

II Всеукраїнська науково-практична конференція
з нагоди 125-річчя КПІ ім. Ігоря Сікорського

«Системні науки та інформатика»

Збірник доповідей

4–8 грудня 2023 року

Київ, Україна

ЗМІСТ

Секція 01

Системний аналіз і управління

Методи заповнення пропусків у даних по електроспоживанню	8
<i>Бондаренко Ю.Г., Губарев В.Ф.</i>	
Лінійні моделі для прогнозування та керування фінансово-економічними показниками на основі когнітивних карт	14
<i>Зуєвський Ю.В., Мілявський Ю.Л.</i>	
СППР для дослідження ринкових ризиків	21
<i>Квашук І.О., Кузнєцова Н.В.</i>	
Стабілізація демографічних процесів за допомогою управління по прогнозуючій моделі	27
<i>Нетудихата А.С., Губарев В.Ф.</i>	
Інформаційна система для кластеризації стану країн за показниками сталого розвитку	33
<i>Самсонюк М.В., Бідюк П.І.</i>	
Інтелектуальна система підтримки прийняття рішень для дослідження актуарних фінансових ризиків	38
<i>Чеманова А.О., Кузнєцова Н.В.</i>	
Моделювання впливу чат-ботів на основі штучного інтелекту на якість вищої освіти методами системного аналізу	45
<i>Чернюк О.Р., Тимошук О.Л.</i>	

Секція 02

Системний аналіз фінансового ринку

Прогнозування гетероскедастичних процесів для оцінювання фінансового ризику	58
<i>Байбара А.Г., Кузнєцова Н.В.</i>	
Підхід щодо визначення факторів впливу на ринок логістики США з використанням LLM	64
<i>Балькін Я.Ю., Савастьянов В.В.</i>	
Прогнозування ціни на золото методами машинного навчання	70
<i>Білоус К.С., Кузнєцова Н.В.</i>	
Порівняльний аналіз та програмна реалізація методів розв'язання задачі про максимальний потік у мережах	76
<i>Боднар М.С., Статкевич В.М.</i>	
Вибір та оцінка якості моделей прогнозування фінансових показників з урахуванням характеристик вхідних даних	84
<i>Бойніцька С.В., Мілявський Ю.Л.</i>	

Аналіз текстових повідомлень за допомогою методів машинного навчання <i>Ведмедєв Д.О., Шаповал Н.В.</i>	90
Система прогнозування енерговитрат будівель різного призначення <i>Данилов В.Я., Дука О.О.</i>	94
Математичні моделі суспільних процесів для аналізу впливу війни на динаміку розвитку економіки та економічних показників <i>Діхтяр А.В., Лопатін О.К.</i>	102
Система підтримки прийняття рішень для оптимізації рекламних кампаній підприємства на основі методу моделювання впливу з залежним представленням даних <i>Заїка Б.Ю., Терентьєв О.М.</i>	108
Модель управління ресурсами гетерогенних баз даних в хмарному середовищі <i>Зайцев О.В., Мухін В.Є.</i>	117
Система прогнозування метеорологічних умов на основі методів аналізу даних та штучного інтелекту <i>Іванійчук А.П., Гуськова В.Г.</i>	121
Кластеризація за допомогою OPTICS: аналіз та оптимізація з використанням графіків та метрик <i>Іванюта О.О., Недашківська Н.І.</i>	127
Моделювання сейсмічних хвиль з використанням машинного навчання <i>Каніовська І.Ю., Кавара А.О.</i>	134
Моделі інтелектуального аналізу даних для оцінювання фінансових моделей <i>Коваленко О.М., Гуськова В.Г.</i>	139
Сучасні моделі оцінювання фінансових ризиків <i>Костенко М.О.І, Кузнєцова Н.В.</i>	146
Система підтримки прийняття рішень для вибору S-моделей економічного зростання <i>Кузьмінчук А.В., Лопатін О.К.</i>	153
Розробка моделей оцінювання ризиків зелених проектів <i>Кузнєцова Н.В., Шевчук О.С.</i>	162
Аналіз відкритих даних про якість повітря в міському середовищі та розробка прогностичних моделей для прогнозу забруднення повітря в місті <i>Луцкер Р.О., Гуськова В.Г.</i>	168
Порівняльний аналіз моделей для методів прогнозування <i>Макухін Є.І., Макаренко О.С., Бідюк П.І.</i>	172
Підходи до прогнозування фінансового стану підприємства та оцінки інвестиційної привабливості сільськогосподарського підприємства на основі статистичних даних <i>Мілявський Ю.Л., Павлуша А.О.</i>	181
Порівняльний аналіз та покращення моделей прогнозування цін акцій на фінансовому ринку <i>Муравльов А.Д., Гуськова В.Г.</i>	187
Моделі оптимального розподілу даних <i>Мухін В.Є., Яковлева А.П., Шмідт А.Є.</i>	192

Прогнозування кредитної спроможності клієнтів банку на основі аналізу фінансових даних	198
<i>Петровський В.Є., Гуськова В.Г.</i>	
Розв'язання задачі заповнення пропусків даних альтернативними методами при створенні прогнозних моделей	201
<i>Попов А.Ю., Макаренко О.С., Бідюк П.І.</i>	
Система аналізу впливу кластеризації на якість рішень в моделях штучного інтелекту	207
<i>Симонов Є.Д., Макаренко О.С.</i>	
Системний підхід до аналізу кредитних ризиків в банківському секторі	213
<i>Сумін О.О., Шубенкова І.А.</i>	
Інтелектуальні засоби підтримки автоматизації управління бізнес процесами	218
<i>Тагільцев Д.І., Мухін В.Є.</i>	
Intellectual decision support system for estimation of financial risks	223
<i>Tymoshchuk O.L., Levenchuk L.B., Vidyuk P.I.</i>	
Чат-бот як середовище розгортання системи підтримки прийняття рішень: приклад телеграм-боту по наданню рекомендацій щодо вибіркових дисциплін	229
<i>Харабара Д.В., Статкевич В.М.</i>	
Автоматична сегментація об'єктів на зображеннях за допомогою напівкерованого навчання з використанням активного навчання та стабільної дифузії	237
<i>Шаповал Н.В., Крутий І.В.</i>	

Секція 03

Інтелектуальні сервіс-орієнтовані розподілені обчислювання

Керування відносинами з клієнтом як сервіс в галузі закладів освіти	243
<i>Бабіч К.О., Петренко А.І.</i>	
Технології WASM, WASI та WAGI для розробки та розгортання мережевих застосунків	247
<i>Булах Б.В., Бондаренко С.Д.</i>	
Інтелектуальні агенти для системи автоматизації управління бізнес-процесами у сфері логістики	252
<i>Вайнер Г.О., Мухін В.Є.</i>	
Інструменти тестування цифрової доступності веб-сайтів	259
<i>Волосожар Д.В., Кисельов Г.Д.</i>	
Гарантування унікальної доставки повідомлень в розподілених системах	263
<i>Гапонюк М.О., Письменний І.О.</i>	
Резервне копіювання даних як послуга	268
<i>Головін Б.О., Булах Б.В.</i>	
Методика редуції обсягу інформації в системах обробки великих даних	273
<i>Дзиговський В.І., Рогоза В.С.</i>	
Дослідження бібліотеки React.js для побудови веб-інтерфейсів для людей з обмеженими можливостями	279
<i>Забельський В.В., Кисельов Г.Д.</i>	

Програмний сервіс для раціонального управління персоналом комерційної компанії	284
<i>Кисельов Г.Д., Гречко Д.М.</i>	
Розробка бібліотеки компонентів для веб інтерфейсів медичних платформ з використанням бібліотеки React.js	291
<i>Кисельов Г.Д., Коваль Д.О.</i>	
Розробка WordPress плагіна для керування файловою системою	297
<i>Кисельов Г.Д., Таран А.І.</i>	
Адаптивні засоби захисту комп'ютерних систем на основі апарата нейронних мереж	303
<i>Коновал В.О., Мухін В.Є.</i>	
Методології розробки мобільних додатків для медичних інформаційних систем на платформі iOS	309
<i>Люлька Р.О., Харченко К.В.</i>	
Розробка і конфігурація інтелектуальної розподіленої системи контролю споживання електроенергії в багатоквартирному будинку	314
<i>Мунтян Д.М., Кисельов Г.Д.</i>	
Сервер для мобільних застосунків як сервіс	320
<i>Науменко Є.О., Булах Б.В.</i>	
Методи аналізу новин	325
<i>Орловський А.В., Кислий Р.В.</i>	
Огляд пріоритезації трафіку в програмно визначених мережах	330
<i>Переяславський С.К., Письменний І.О.</i>	
Хмарний сервіс для дистанційного догляду за пацієнтами	335
<i>Петренко А.І., Редька М.Ю.</i>	
Оркестрування обчислень у мікросервісній архітектурі	341
<i>Полещук В.О., Булах Б.В.</i>	
Аналіз ризиків в задачах інформаційної безпеки	347
<i>Северин М.С., Мухін В.Є.</i>	
Платформи з низькою затримкою для обробки потоків даних у реальному часі	353
<i>Сіркович А.І., Харченко К.В.</i>	
Сервіс ведення фінансової звітності для ФОП	356
<i>Скрипник А.В., Булах Б.В.</i>	
Оптимізація передачі даних у багатокористувацьких відеоіграх	360
<i>Сухарев О.М., Безносик О.Ю.</i>	
Система розумного будинку з керуванням телеграм ботом та Google Assistant	366
<i>Харченко К.В., Кушовий Д.І.</i>	
Дослідження продуктивності мікросервісних архітектур через кешування даних	370
<i>Хоміч Л.І., Яременко В.С.</i>	
Геолокація позицій об'єктів робочої карти за даними зображень БПЛА	375
<i>Хом'як К.В., Петренко А.І.</i>	

Самонавчальна система розпізнавання людської діяльності з використанням навчання з підкріпленням	382
<i>Цибін М.Д., Кислий Р.В.</i>	
Інтернет речей в сфері охорони здоров'я	388
<i>Цимбалюк Р.С.</i>	

Секція 04

Системи і методи штучного інтелекту

Filter for confidential information for chats with LLM and using local texts databases	392
<i>Bezymiannyi O.E., Shapoval N.V.</i>	
Аналіз наукових статей за допомогою штучного інтелекту	396
<i>Овчаренко О.С.</i>	
Задача пошуку шляхів з використанням спектральної теорії графів	400
<i>Сабітова Р.Р., Статкевич В.М.</i>	
Довгострокове прогнозування попиту: використання ансамблю нейронних мереж для підвищення точності	406
<i>Самошин А.О., Синєглазов В.М.</i>	
Пошук відповідності між зображенням і його текстовим описом	412
<i>Шаповал Н.В., Крижанівська О.В.</i>	

МЕТОДИ ЗАПОВНЕННЯ ПРОПУСКІВ У ДАНИХ ПО ЕЛЕКТРОСПОЖИВАННЮ

Бондаренко Ю.Г.¹, Губарев В.Ф.

Національний технічний університет України «Київський політехнічний інститут
імені Ігоря Сікорського», Київ, Україна

¹ bondarenkojulia423@gmail.com

Метою роботи є дослідження методів боротьби з пропущеними значеннями. Експериментально перевірити ефективність існуючих методів для вирішення задачі заповнення пропусків на даних по електроспоживанню. Порівняти отримані результати для кожного методу. Визначити які методи мають потенціал для ефективного практичного застосування в розглянутій галузі.

Ключові слова: заповнення пропусків, електроспоживання, часові ряди, регресійні моделі.

1. ВСТУП

На сьогоднішній день важливим є питання ефективного аналізу, передбачення та прогнозування електроспоживання для подальшого запобігання можливості тимчасових відключень електроенергії чи навіть зatoryжних блекаутів. Тому важливо, щоб вхідні дані були якомога якіснішими та повними, тобто не містили пропусків.

Наявність пропущених значень в часових рядах – це проблема, яку досліджують вже не перше десятиліття науковці по всьому світу. Поява пропусків може бути із низки причин: технічні проблеми, зміни у методології збору інформації, неможливість отримати дані в певний період часу, тощо. Наявність відсутніх значень значно спотворює вхідну інформацію, що може призвести до неможливості коректного аналізу даних та подальшого прогнозування [1]. Вважається, що наявність пропусків менше 5 відсотків є досить несуттєвою та може не впливати на результат дослідження. Проте, якщо відсоток втрати даних становить 20 та більше, то результати такого дослідження потрібно ставити під сумнів [2].

Хоч методів для відновлення відсутніх значень існує досить багато, однак, універсальної методології немає. Тому важливо дослідити наявні способи боротьби з пропусками та оцінити їх ефективність для часового ряду електроспоживання.

2. МЕТОДИ ЗАПОВНЕННЯ ПРОПУСКІВ

Найпростішим методом обробки даних, що містять пропуски, є видалення відсутніх значень. Проте такий спосіб не є ефективним і лише спотворює наявні дані, тому використання його не є доцільним.

Метод медіани полягає у тому, що всі пропущені значення замінюються статичною мірою центральної тенденції – медіаною. Він простий у реалізації та використанні, але може призвести до заниження якості даних.

За схожою аналогією працює метод моди. Його принцип в тому, що замість пропусків записується значення, що найчастіше зустрічається в наборі даних. Цей метод корисний особливо у випадках категоріальних змінних.

Метод інтерполяції. Він допомагає оцінити відсутні значення між двома сусідніми точками даних, що відомі. Цей метод в основному можна вважати чудовим компромісом між

швидкістю роботи та точністю результатів. Проте ефективність цього методу залежить від характеру набору даних, тому в певних випадках він може бути недоцільним.

Регресійний метод. Основна його суть в тому, що заповнення пропусків виконується за допомогою прогнозування їх різними регресійними моделями. Для цього дані спостережуваної змінної, що не містять пропусків, використовуються як залежна змінна, а інші змінні використовуються як незалежні для побудови відповідної регресійної моделі. Наступний кроком є використання отриманої моделі для прогнозування відсутніх значень. Головною перевагою використання регресійного методу є те, що такий підхід враховує зв'язки між змінними та не змінює вихідний розподіл.

Для подальшого дослідження розглядалися два типи регресійних моделей: лінійні (Linear Regression, Bayesian Ridge) та нелінійні (Random Forest, Support Vector Regression, Adaptive Boosting).

Метод K-Nearest Neighbor. Його принцип роботи полягає в тому, що для кожного відсутнього значення вибираються значення k-найближчих сусідів, що не містять пропусків, які і будуть опорними значеннями для подальшого розрахунку. Далі кожен пропуск обраховується як функція від значень цих k-найближчих сусідів. Цей метод є досить ефективним, коли дані містять певні закономірності.

3. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Для проведення досліджень використано дані про щоденне споживання електроенергії українцями за період з 2014 по 2022 рік, отримані з офіційних даних УкрЕнерго. Розмір датасету становить 3223 записи. Графічне представлення даних наведено на Рис. 1.

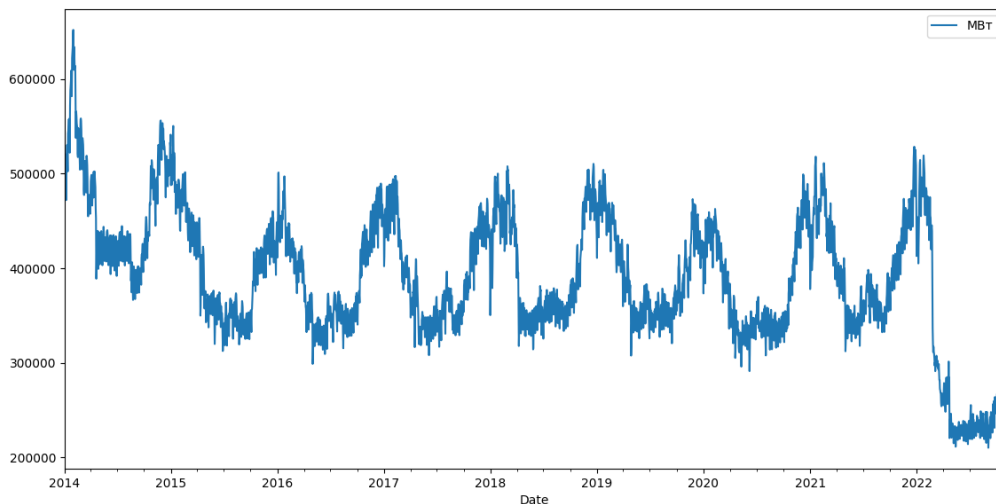


Рисунок 1. Споживання електроенергії 2014–2022 р.

Спочатку виконався попередній аналіз датасету. Для наявних даних було побудовано кореляційні матриці. Результати наведені в Табл. 1 демонструють кореляцію між споживанням електроенергії, днем, місяцем, роком та годиною. Результати наведені в Табл. 3 містять дані кореляції між споживанням, генерацією та температурою навколишнього повітря.

Таблиця 1. Кореляційна матриця 1

	Day	Month	Year	Hour	Consumption
Day	1,000000	0,008851	-0,003293	0,000494	0,020389
Month	0,008851	1,000000	-0,046340	-0,000223	-0,107189
Year	-0,003293	-0,046340	1,000000	0,000066	-0,354161
Hour	0,000494	-0,000223	0,000066	1,000000	0,370651
Consumption	0,020389	-0,107189	-0,354161	0,370651	1,000000

Таблиця 2. Кореляційна матриця 2

	Generation	Temperature	Consumption
Generation	1,000000	-0,673325	0,981036
Temperature	-0,673325	1,000000	-0,695876
Consumption	0,981036	-0,695876	1,000000

З таблиці 1 видно, що дата та час не мають значного впливу на обсяг спожитої електроенергії. Це пояснюється тим, що останні роки в країні не було вираженої сезонності, адже часто траплялося так, що зими були відносно теплі, а весни досить холодні. Однак відомо, що споживання електроенергії часто зростає у жаркі чи навпаки дуже холодні дні, адже люди активно починають використовувати кондиціонери, вентилятори, електрообігрівачі, тощо. Тому, як ми бачимо з таблиці 2, коефіцієнт кореляції між температурою та обсягом електроспоживання становить $-0,695876$, що означає те, що цей параметр може бути вагомим для побудови моделей. Тому доцільно буде перевірити його вплив на роботу методів.

Також з таблиці 2 видно, що обсяг згенерованої електроенергії має високий коефіцієнт кореляції, тому доцільно використовувати його як параметр для моделей.

Для перевірки методів було створено три різних датасети. Перший набір даних включає поля «Дата» та «Спожито, МВт». До нього застосовувалися прості методи заповнення пропусків (мода, медіана та інтерполяція). Другий та третій датасети – «Дата», «Згенеровано, МВт», «Спожито, МВт» та «Дата», «Згенеровано, МВт», «Спожито, МВт» та «Середня денна температура, C°» відповідно.

Для проведення експериментів штучним шляхом в усіх наборах даних було згенеровано пропуски із відсотком втрати інформації в полі «Спожито, МВт» 10%, 20%, 40% та 60%. Для порівняння ефективності методів в кожному випадку було обчислено коефіцієнт детермінації, середня абсолютна похибка та корінь середньоквадратичного відхилення.

Порівняння результатів методів моди, медіани та інтерполяції для трьох датасетів наведено у Табл. 3.

Порівняння результатів шести моделей для другого та третього датасетів наведено в Табл. 4.

Таблиця 3. Порівняння результатів для моди, медіани та інтерполяції

Похибка	Мода	Медіана	Інтерполяція
Перший датасет, 10%			
R2	0,81869714	0,8729142	0,9974808
RMSE	83,96228	77,28661	28,53677
MAE	7049,66449	5973,2203	814,3475067
Другий датасет, 10%			
R2	0,90443874393	0,95085209	0,9990900
RMSE	66,4259656	54,88019	16,4613682
MAE	4412,40890890	3011,83608	406,46496
Третій датасет, 10%			
R2	0,9362924	0,967234	0,999393
RMSE	54,236573	44,8094	16,4613682
MAE	2941,6059	2007,89072	270,97664
Перший датасет, 20%			
R2	0,6805497	0,76390084	0,994414
RMSE	111,8694	104,6918	40,963454
MAE	12514,7768	10960,384	1678,004
Другий датасет, 20%			
R2	0,79201352	0,887542	0,99771
RMSE	95,60291793	77,9939	29,803414
MAE	9139,9179	6083,0630	888,2435
Третій датасет, 20%			
R2	0,86134234	0,925028	0,9984792
RMSE	78,0594	63,681829	24,33438641
MAE	6093,2786	4055,3753	592,16236
Перший датасет, 40%			
R2	0,146221	0,33185800	0,9843825
RMSE	166,1744	149,0019	64,038066
MAE	27613,937	22201,57	4100,873
Другий датасет, 40%			
R2	0,3866518	0,6854566	0,99455428
RMSE	173,46507	111,89123	44,057018
MAE	30090,13	12519,6476	1941,02084
Третій датасет, 40%			
R2	0,5911012	0,7903044	0,9963695
RMSE	141,63363	91,3588078	35,9724
MAE	20060,0874	8346,43176	1294,013
Перший датасет, 60%			
R2	-0,601119	-0,3704258	0,96869
RMSE	195,543	182,021658	84,75633
MAE	38237,2045	33131,884	7183,6361
Другий датасет, 60%			
R2	-0,008068	0,30906	0,98832
RMSE	211,12529903	137,327202	58,896317
MAE	44573,891	18858,7605	3468,77624
Третій датасет, 60%			
R2	0,32795	0,53937485	0,992217
RMSE	172,38308	112,12719	48,0886
MAE	29715,927	12572,5070	2312,517497

Таблиця 4. Порівняння результатів для шести моделей

Похибка	Linear Regression	Random Forest	Bayesian Ridge	SVR	Ada Boost	KNN
Другий датасет, 10%						
R2	0,9935750	0,9852721	0,9935746	0,9852721	0,98527214	0,99815743
RMSE	38,61722	45,498611	1491,32252	80,221261	49,514787	29,0084641
MAE	1491,2899	2070,12365	2224042,868	6435,4508	2451,7141	841,49099099
Третій датасет, 10%						
R2	0,9922817	0,984748	0,992286	0,98474804	0,984748	0,9981645
RMSE	40,11553	44,59644	1608,8032	80,412648	47,57422	28,941684835
MAE	1609,2559	1988,8433	2588247,8	6466,19404	2263,3068	837,621121
Другий датасет, 20%						
R2	0,986310	0,9684657	0,9863098	0,96846574	0,968465	0,99468875
RMSE	55,65723	65,907300	3097,799486	116,921829	71,73040	44,51679
MAE	3097,72832	4343,77227	9596361,65	13670,7143	5145,2504	1981,74474
Третій датасет, 20%						
R2	0,9834775	0,9676315	0,98348760	0,9676315	0,9676315	0,9962415
RMSE	57,50567	64,6782397	3306,04511	117,186482	70,230005	41,053518623
MAE	3306,90220	4183,27469	10929934,27	13732,67161	4932,2536	1685,391391391
Другий датасет, 40%						
R2	0,971916	0,93806022	0,97191507	0,93806022	0,9380602	0,9898630
RMSE	79,6785807	92,4901632	6348,8339	167,048671	103,1290	61,86479817
MAE	6348,6762	8554,43029	40307692,5	27905,2587	10635,60085	3827,2532
Третій датасет, 40%						
R2	0,96749847	0,93494680	0,96751581	0,93494680	0,93541447	0,99337741
RMSE	81,6670297	91,1252390	6668,055854	167,470421	99,67827105	56,2941473958
MAE	6669,50374	8303,80918	44462968,88	28046,3422	9935,75772	3169,0310
Другий датасет, 60%						
R2	0,95884694	0,9052382	0,9588443	0,905238	0,90523821	0,9809668
RMSE	96,98512259	113,1810	9406,3757	206,49897	124,65692	78,6695678194
MAE	9406,114005	12809,9551	88479905,0	42641,8253	15539,34856	6188,9009
Третій датасет, 60%						
R2	0,95153	0,901126	0,951561	0,9011265	0,901126	0,98438803
RMSE	100,4346994	111,625299	100,4214981	206,96689	121,729431	74,164647
MAE	10087,128	12460,2074	10084,47728	42835,2951	14818,0544	5500,39489

4. ВИСНОВКИ

З отриманих результатів видно, що найкращим методом виявився K-Nearest Neighbor. Він дозволяє ефективно боротися з пропусками навіть на високих відсотках втрати інформації. Метод інтерполяції також показує високі результати, проте у випадках, коли датасет містить пропуски на початку, він не завжди може виконатися, що не дозволяє йому бути універсальним, на відміну від KNN. Також, слід виділити метод Linear Regression. Хоч

показники цього методу трохи нижчі за показники попередніх двох, та все ж він веде себе досить стабільно не залежно від кількості пропусків.

Методи Random Forest, Ada Boost, SVR та Bayesian Ridge показують гарні результати для 10 та 20 відсотків пропусків, проте, при збільшенні відсотку до 40 і більше, отримані результати значно погіршуються в порівнянні з трьома попередніми методами.

Найгірше себе показали методи заповнення модою та медіаною. При відсотку втрачених даних більше 20 ці методи можна вважати неточними та неефективними.

Зазначимо, що додавання параметра «температура» покращило роботу більшості моделей, особливо на етапах з високим відсотком пропусків.

Отже, для заповнення пропусків в даних по електроспоживанню рекомендується використовувати метод K-Nearest Neighbor, інтерполяцію чи Linear Regression. Також, для великих наборів даних з високим відсотком пропущених значень доцільно буде враховувати показник температури зовнішнього середовища для отримання більш точних результатів.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Методи заповнення пропусків даних у задачах прогнозного моделювання соціально-економічних процесів / П. І. Бідюк, О. М. Терентьев, Т. І. Просянкіна-Жарова // Інтелектуальні системи прийняття рішень та проблеми обчислювального інтелекту : матеріали міжнародної наукової конференції (ISDMCI-2017). – Херсон, ПП Вишемирський В. – С. 185-187. – Бібліогр.: 2 назви.

2. Dziura, J. D., Post, L. A., Zhao, Q., Fu, Z., & Peduzzi, P. (2013). Strategies for dealing with missing data in clinical trials: from design to analysis. *The Yale journal of biology and medicine*, 86(3), 343–358.

3. Balances - Energy Map. Energy Map. URL: <https://map.ua-energy.org/en/datasets/d1fb93e6-7751-4d70-9ee9-633747ff83d9/resources/> (date of access: 18.09.2023).

ЛІНІЙНІ МОДЕЛІ ДЛЯ ПРОГНОЗУВАННЯ ТА КЕРУВАННЯ ФІНАНСОВО-ЕКОНОМІЧНИМИ ПОКАЗНИКАМИ НА ОСНОВІ КОГНІТИВНИХ КАРТ

Зуєвський Ю.В., Мілявський Ю.Л.

Національний технічний університет України «Київський політехнічний інститут
ім. Ігоря Сікорського», Київ, Україна

Фінансова сфера постійно зазнає впливу багатьох факторів, і точне прогнозування фінансових показників стає ключовим завданням для успішного управління. Одним із підходів до аналізу цих показників є побудова лінійних моделей когнітивних карт, моделювання, аналіз та керування на їх основі. Однією з проблем при побудові когнітивної карти є визначення вагових коефіцієнтів ребер карти на основі історичних даних системи, що моделюється. У даній роботі розглянуто деякі підходи до ідентифікації, а саме, на основі лінійної регресії та на основі кореляційного аналізу. Показано, що ці підходи можуть співіснувати і доповнювати одне одного у випадках, коли можна виділити залежну і незалежні змінні у моделі. Далі, пропонується відповідно два підходи до прогнозування - один на основі лінійної регресії для прогнозування залежної змінної, інший - на основі сценарного аналізу імпульсного процесу когнітивної карти для прогнозування решти змінних. Проведено моделювання побудованої таким чином когнітивної карти конкретної системи на історичних даних.

Ключові слова: ідентифікація, когнітивні карти, лінійні моделі, регресія, кореляційний аналіз, фінансові ризики, сценарний аналіз, прогнозування.

1. ВСТУП

В сучасному світі фінансова діяльність відіграє важливу роль у глобальній економіці, і її результати мають значущий вплив на стабільність та розвиток різних сфер життя. Однак, незважаючи на значення фінансових операцій, вони завжди супроводжуються елементом невизначеності та ризику. Ризик є невід'ємною частиною будь-якої фінансової діяльності, і відсутність можливості передбачити його вплив може призвести до негативних наслідків для фірми, інвесторів та всієї економіки.

Фінансові ризики, як випадкові величини, виникають внаслідок впливу багатьох факторів, включаючи зміни в ринкових умовах, процентних ставках, валютних курсах, інфляції та багатьох інших. Тому, для забезпечення ефективного управління фінансовими ризиками, необхідно розробляти моделі, які допомагають прогнозувати їх вплив та вчасно приймати відповідні рішення.

Отже, **актуальність теми** полягає в проведенні досліджень спрямованих на вивчення можливостей використання лінійних моделей імпульсних процесів когнітивних карт для ефективного прогнозування та керування фінансово-економічними показниками. Наша робота спрямована на розвиток нових методів управління фінансами, які допомагають компаніям та інвесторам краще розуміти та керувати фінансовими ризиками, забезпечуючи стабільність та успішність їхньої діяльності.

Метою роботи є вдосконалення методів прогнозування та керування фінансово-економічними показниками на основі когнітивних карт. Для досягнення поставленої мети в роботі сформульовані наступні задачі:

- аналізу структурних залежностей між змінними та ідентифікація зв'язків між змінними та їх ваги
- математичне моделювання прогнозування електроспоживання населенням в часі з урахуванням зміни напруги в мережі з розробкою когнітивної карти

2. МЕТОДИ ПОБУДОВИ ТА АНАЛІЗУ КОГНІТИВНИХ КАРТ

Припустимо, що у динамічній фінансово-економічній складній системі, яку можна моделювати у формі лінійного імпульсного процесу в когнітивній карті, існує залежна змінна, яка представляє найбільший практичний інтерес, і ряд незалежних змінних, пов'язаних із нею та між собою. Пропонується використати різні підходи до оцінювання ваг ребер когнітивної карти, які з'єднують вершину - залежну змінну з вершинами – незалежними змінними, і ребер, які з'єднують вершини - незалежні змінні між собою.

Перевагою лінійної регресії є простота інтерпретації, здатність визначати кількісний вплив незалежних змінних на залежну. Недоліком є вимога до лінійності зв'язку між змінними, чутливість до викидів і нелінійних взаємозв'язків.

Основна математична модель, яка використовується в лінійній регресії, представлена як

$$y = Xa + \varepsilon, \text{ де}$$

y – вектор залежної змінної,

X – матриця незалежних змінних,

a – вектор невідомих коефіцієнтів,

ε – вектор випадкових помилок.

Далі на основі побудованої лінійної регресії можна здійснювати прогнозування залежної змінної та аналіз її чутливості до змін у незалежних змінних. При цьому будуватимемо регресію між приростами незалежних і залежних змінних, наближаючись таким до підходу Ф. Робертса у когнітивних картах.

Але як відомо, сценарний аналіз імпульсних процесів когнітивних карт також має ряд переваг, зокрема, він дозволяє системно, комплексно аналізувати динаміку складної системи, бачити взаємовпливи між змінами в будь-яких вершинах карти і розуміти, як зміна однієї вершини вплине за всі інші. Для того, щоб знайти ваги ребер когнітивної карти, можна використати коефіцієнти лінійної регресії, побудовану перед цим. але це лише частина ребер карти. Для решти пропонується використати коефіцієнти кореляції між рядами приростів значень у вершинах – незалежних змінних, які можна легко отримати за наявних історичних даних.

Обидва ці підходи, застосовані разом, можуть надати більш повне уявлення про потенційні фінансові результати та ризики, даючи можливість краще управляти фінансово-економічними показниками. Лінійна регресія дозволяє здійснювати точні кількісні прогнози, тоді як сценарний аналіз допомагає адаптуватися до різних можливих умов ринку.

3. МОДЕЛЮВАННЯ ТА РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Для дослідження було обрано датасет, що називається "Household Power Consumption", що містить інформацію про споживання електроенергії в домогосподарствах. Як залежну змінну було обрано змінну `globalactiverpower`, що відображає загальну активну потужність, цей атрибут представляє собою середню потужність, споживану домогосподарством, в кіловаттах (кВт). Вибір лінійної регресії обґрунтований, тим що залежність між змінними лінійна. Цей метод добре підходить для простих, малих та середніх датасетів, де очікується, що зв'язок між

змінними можна апроксимувати лінійною функцією. Для обчислення матриці було використано метод найменших квадратів (МНК) для оцінки коефіцієнтів, що є класичним підходом в лінійній регресії. МНК мінімізує суму квадратів різниць між спостережуваними та прогнозованими значеннями.

Логіка полягає в аналізі та прогнозуванні змін в споживанні електроенергії, виходячи з різних параметрів, що відображаються у датасеті. Використовуючи лінійну регресію, можна зрозуміти, як різні фактори (наприклад, реактивна потужність, напруга тощо) впливають на активну потужність. Візуалізації допомагають інтерпретувати залежності та зв'язки між змінними.

Для оцінки моделі було визначено R^2 – коефіцієнт детермінації, що вимірює, яка частина варіативності залежної змінної пояснюється незалежними змінними. Хороша модель матиме R^2 близько до 1. Також використовуємо MSE – середньоквадратичну похибку, що вимірює середнє квадратичне відхилення між фактичними та прогнозованими значеннями. Чим нижче значення MSE, тим краще модель.

Використання когнітивної карти з вагами, визначеними на коефіцієнтах регресії та кореляціях, є корисним для візуалізації та аналізу структури залежностей між змінними. Когнітивна карта дозволяє легко ідентифікувати важливі зв'язки та потенційні джерела мультиколінеарності.

В результаті реалізації моделі було отримано когнітивну карту, що зображено на рис. 1.

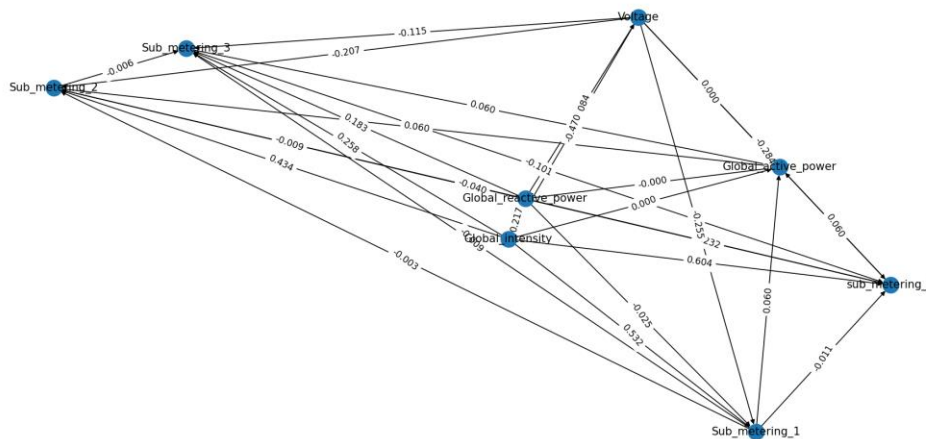


Рисунок 1. Когнітивна карта

Ми можемо спостерігати відображення когнітивної карти, на якій відображаються зв'язки між змінними та їх ваги. З рис. 2 можемо бачити теплокарту кореляційних зв'язків змінних. Теплокарта – це графічне зображення даних, де окремі значення в матриці відображаються за допомогою кольорів.

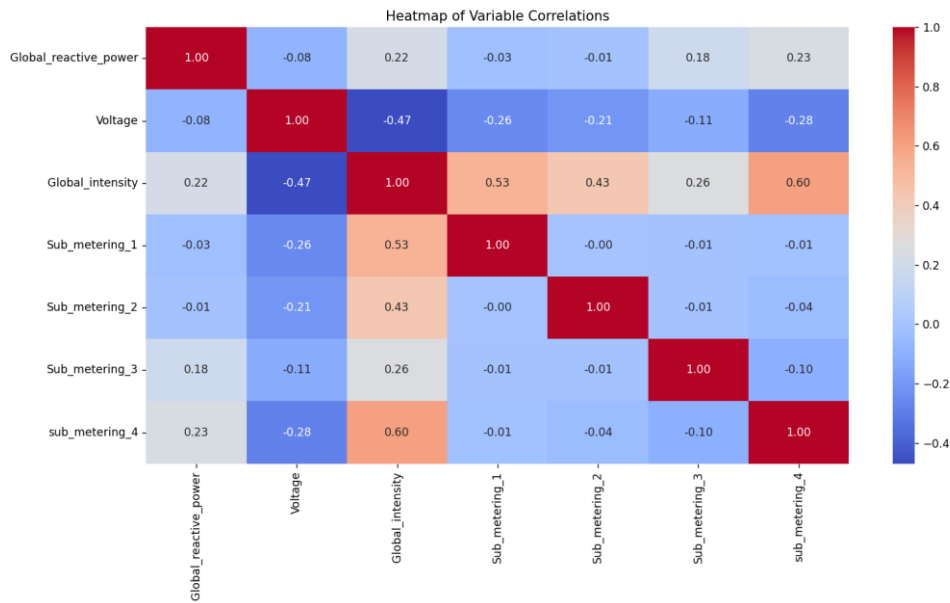


Рисунок 2. Теплокарта значень кореляції змінних

Також було отримано результат передбачення зміни середнього використання електроенергії, результат зображено на рис. 3.

```
C:\Users\Hp\Desktop\linear>python pr.py
R-square (R2): 0.999999999999537
Mean Squared Error (MSE): 3.3046878919759077e-15
Predicted change in Global_active_power: [0.93200002]
```

Рисунок 3. Результати програми

Показник R^2 є майже ідеальним, оскільки він наближається до 1. Такий високий R^2 означає, що модель майже ідеально відповідає даним. MSE є мірою середньої квадратичної помилки між фактичними та прогнозованими значеннями. Дуже низьке значення MSE, як у вашому випадку, вказує на те, що помилка між прогнозованими та фактичними значеннями є мінімальною, що є хорошим показником ефективності моделі.

Прогнозоване змінення у `Global_active_power` становить приблизно 0.932 одиниць. Цей результат означає, що згідно з лінійною регресійною моделлю та вхідними даними для прогнозу, очікується, що `Global_active_power` збільшиться на 0.932 одиниць порівняно з останнім відомим значенням у датасеті.

Також було здійснено сценарний аналіз для відображення впливу зміни показників на прогнозоване значення використання електроенергії. На рис. 4–6 відображено вплив зміни показників.

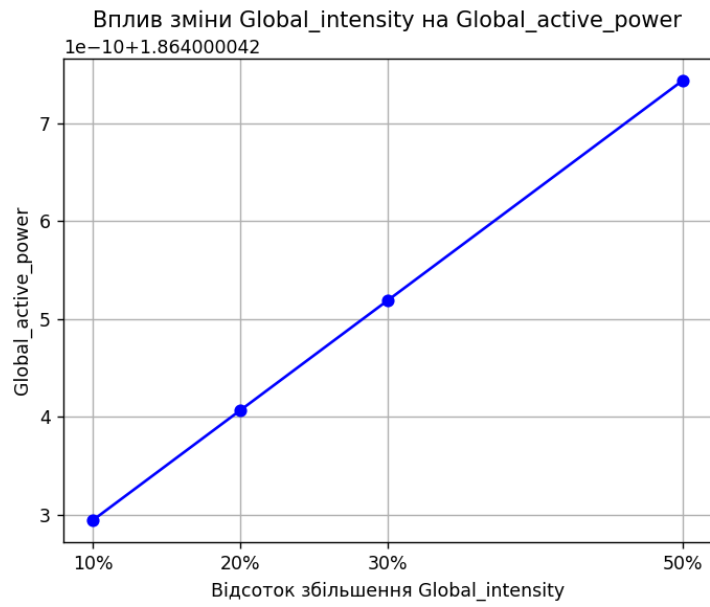


Рисунок 4. Зміна показника Global Intensity

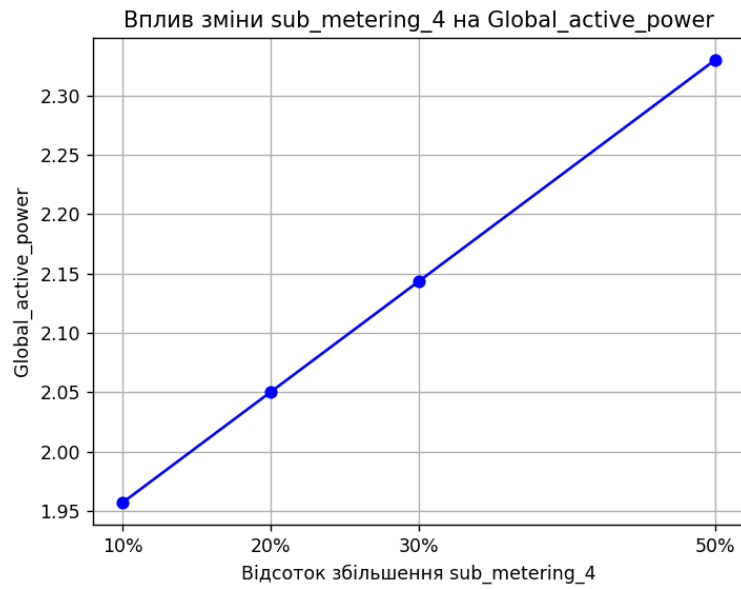


Рисунок 5. Зміна показника Sub Metering 4

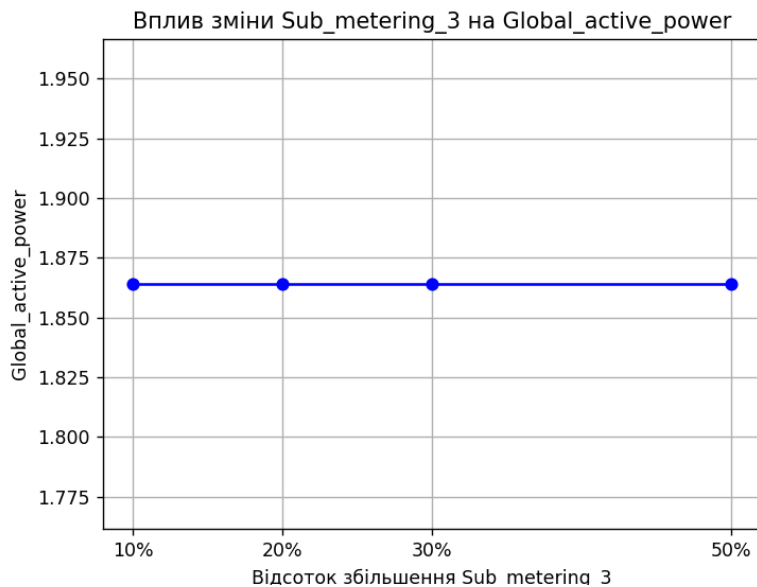


Рисунок 6. Зміна показника Sub Metering 3

Збільшення використання різних видів електроенергії дав наступні результати на зміну загального використання електроенергії (рис. 7).

Прогнозоване значення Global_active_power: 1.864000042182632

Рисунок 7. Прогнозоване значення

Даний прогноз може бути корисним у різних сценаріях, таких як планування навантаження на електромережу, аналіз енергоефективності тощо. В залежності від контексту, таке збільшення може бути важливим. Даний прогноз є індикатором можливих змін у споживанні електроенергії за заданими умовами.

4. ВИСНОВКИ

Прогнозування та побудова когнітивних карт є важливими інструментами для аналізу фінансових даних та прийняття рішень. Вони допомагають розуміти складну інформацію, визначати взаємозв'язки та розробляти стратегії для подальшого розвитку. У ході дослідження було досліджено питання побудови когнітивних карт на основі фінансових даних та прогнозування цих даних на основі побудованих когнітивних карт. Це відіграє важливу роль у сучасному бізнесі та науці, допомагаючи покращити прийняття рішень та аналізувати складну інформацію. Було побудовано когнітивні карти та здійснено прогнозування за допомогою двох методів – лінійної регресії та сценарного аналізу імпульсного процесу когнітивної карти. Сценарний аналіз дає більші можливості для відслідковування зміни та впливу показників на середнє використання електроенергії. Висока точність моделі та здатність виявляти зв'язки між різними параметрами споживання електроенергії може бути корисною для розуміння та оптимізації використання енергоресурсів у домогосподарствах.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Mikhail Z. Zgurowsky, Victor D. Romanenko, Yuriy L. Milyavskiy. Principles and Methods of Impulse Processes Control in Cognitive Maps of Complex Systems. Part I // Journal of Automation and Information Sciences. – 2016. – Vol. 48, No. 3. – P. 36-45.
2. Mikhail Z. Zgurowsky, Victor D. Romanenko, Yuriy L. Milyavskiy. Principles and Methods of Impulse Processes Control in Cognitive Maps of Complex Systems. Part II // Journal of Automation and Information Sciences. – 2016. – Vol. 48, No. 7. – P. 4-16.
3. Mykhailo Zgurovsky, Victor Romanenko, Yuriy Milyavsky. Adaptive Control of Impulse Processes in Complex Systems Cognitive Maps with Multirate Coordinates Sampling // Advances in Dynamical Systems and Control, Studies in Systems, Decision and Control, 69. – Springer International Publishing . - Switzerland, 2016. - P. 363 – 374.
4. V. Gubarev, V. Romanenko, Y. Miliavskiy. Identification and Control Automation of Cognitive Maps in Impulse Process mode / Kuntsevich, Gubarev, Kondratenko, Lebedev, Lysenko (eds.), *Control Systems: Theory and Applications* – River Publishers, 2018. - P. 43 – 64.
5. Miliavskiy Yu.L. Identification in cognitive maps in impulse process mode with incomplete measurement of nodes coordinates // Кибернетика и вычислительная техника. – 2019. – № 1 (195). – С. 49–63.
6. V. F. Gubarev, V. D. Romanenko, Yu. L. Miliavskiy. Methods for Finding a Regularized Solution When Identifying Linear Multivariable Multiconnected Discrete Systems // Cybernetics and Systems Analysis. – 2019. - Volume 55, Issue 6. – P. 881–893.

СППР ДЛЯ ДОСЛІДЖЕННЯ РИНКОВИХ РИЗИКІВ

Квашук І.О.¹, Кузнецова Н.В.²

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

¹ illiakvashuk@gmail.com [0009-0006-5585-3045], ² natalia-kpi@ukr.net [0000-0002-1662-1974]

Створення системи підтримки прийняття рішень є особливо актуальним для ринкових фінансових операцій. Можливість прогнозування акцій, оцінка основних параметрів та розрахунок оптимального портфелю є ключовими задачами, які можуть бути виконані СППР. В роботі було виконано тестування системи на наборі даних Apple, Google, Amazon та Tesla. Було виконано прогнозування акцій методами лінійної регресії, випадкового лісу та SVM. Було виконано моделювання величини VaR методом Монте-Карло та розрахований оптимальний портфель на основі спрогнозованих акцій.

Ключові слова: ринковий ризик, СППР, VaR, балансування портфелю

1. ВСТУП

Фінансові операції на ринку пов'язані з ризиком, що виникає в результаті дій інших гравців та агентів. Разом вони формують тренд або зміну на фінансовому ринку: коливання акцій, інший склад і прибутковість оптимального портфелю. Все це разом призводить до того, що дії на фінансовому ринку пов'язані з невизначеністю в майбутньому та мають тенденцію до змін. Кожний гравець на ринку прагне до максимізації власних прибутків та оптимальності дій. Інформація про стан та умови на ринку зараз та в майбутньому дозволяє оцінити величину ризику та визначити найкращу модель поведінки або навіть конкретну дію: продаж акції, купівля, збільшення інвестицій. Для цього потрібно провести розрахунки, виконати оцінку та підбір параметрів. З метою пришвидшення процесу прийняття рішення і зручної роботи спеціалісту, розроблюються системи підтримки прийняття рішень. Вони містять як модуль введення даних, так і модель виведення – графіки, прогнози та інформацію, що буде надана користувачу.

Системи підтримки прийняття рішень та їх модулі широко представлені та доступні. Разом з тим кожна окрема комбінація модуля та закладеного в ньому функціоналу, варіативність моделей та можливості вирізняються [1, 2]. Кожний підхід до розробки подібної системи підтримки прийняття рішень дозволить надати трейдерам або іншим користувачам більшу варіативність в виборі оптимальної стратегії інвестування.

2. ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

Метою магістерської роботи є розробка системи підтримки прийняття рішень, яка буде надавати інформацію щодо фінансових даних відповідно до введеної користувачем інформації. СППР повинно розв'язувати наступні задачі: прогнозування цін акцій, що були обрані користувачем, моделювання стану ринку з метою отримання величини можливих втрат. Окремою задачею є побудова портфелю та розрахунок його оптимального складу на основі спрогнозованих результатів.

3. ОСНОВНІ МЕТОДИ

Акції компанії залежать від великої кількості внутрішніх змінних – стан компанії, новини, зміна керівників та CEO, рівні продажів – та від значної кількості зовнішніх факторів – стану сектору та ринку, нових правил регулювання. Економічні та політичні кризи також значно впливають на стан акцій. Це робить прогнозування складною задачею, адже через складеність процесів – великої кількості факторів, що лежать в основні змін – неможливо врахувати всі значення та важливі змінні для надання точного та безпомилкового прогнозу. Розрахунок стану акцій – це основа для всіх інших дій. Для розв’язання задачі прогнозування акцій було використано декілька моделей. Це дозволить знайти найбільш оптимальну модель для конкретного проміжку часу. Поведінка акцій під час різних періодів часу може мати різний характер та описуватися різними моделями.

Однією з обраних моделей є модель, що будується за допомогою SVM (Support Vector Machine) і застосовує методи регресії та класифікації. Основна ідея методу базується на знаходженні гіперплощини, що буде застосовуватися до розділення класів. Такий підхід використовується для розв’язання задачі класифікації, проте після модифікацій він може використовуватися і для задачі розв’язання регресії. Особливістю методу є те, що він може не обмежуватися розмірами простору початкових даних, а використовуючи техніку Kernel Trick, може переходити до просторів більшої розмірності. Це дозволяє розв’язати проблему, коли в даному просторі елементи не роздільні гіперплощиною [3, 4].

Наступною моделлю було вирішено обрати модель лінійної регресії. Модель лінійної регресії – це універсальний інструмент, що використовується в багатьох задачах в реальному світі, включаючи і фінансові операції. Не зважаючи на свою простоту, ця модель – це потужний інструмент розв’язання задач, та її поведінка та особливості добре вивчені та дослідженні в багатьох роботах [5, 6]. Суть лінійної регресії полягає в оптимальному виборі параметрів, що мінімізують певний критерій.

Модель випадкового лісу – це модель, що належить до сфери машинного навчання. Особливістю моделі є використання наборів дерев рішень, кожне з яких навчається на окремій складовій даних, та прийнятті рішення на засадах ранжування пропозицій наданих кожним окремим деревом. Пропозиція, за яку голосує найбільша кількість дерев, обирається як результат роботи моделі.

Всі моделі будуть використовуватися для прогнозування акцій. Водночас, учасники ринку можуть надавати перевагу іншому підходу, що базується на розрахунках певної, майже гарантованої, величини втрат. На основі цього можна проводити оцінку стану та рівню ризику для окремих операцій. Розрахована величина – це VaR і для її моделювання можна застосовувати різні підходи. Ми зупинили свій вибір на моделі Монте-Карло.

VaR (Value-at-Risk) – це величина, що знаходиться під ризиком втрати. Це один з основних фінансових показників, яка розраховується для отримання розуміння про величину потенційних (можливих) втрат.

Метод Монте-Карло – це метод, що застосовується для генерацій вибірок. В даному випадку метод використовується для отримання можливих траєкторій розвитку акцій з метою оцінки показників. Метод може використовувати різні розподіли і найкращим [7].

4. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Реалізація системи підтримки прийняття рішень з метою розв’язання поставленої задачі вимагає перевірки роботи моделі. Для цього було вирішено обрати набір певних акцій і для них провести прогнозування, балансування портфелю та моделювання значення VaR. Дані можуть бути отримані в реальному часі, оскільки ринкові показники компаній наявні на біржах, де відбувається їх лістинг. З множини всіх доступних акцій було прийнято рішення

взяти акції великих технологічних компаній, оскільки багато учасників ринку купують їх акції, і перевірка роботи системи дозволить отримати максимально наближені до реальних.

Для перевірки якості моделей було обрано акції компанії Apple та прогнозувалась ціна їх закриття (Рис. 1). Існує декілька характеристик, що описують стан акцій – ціна відкриття, ціна закриття, проте було вирішено зупинитися на ціні закриття по тій причині, що прогноз закриття включає в себе зміни протягом одного дня, а також додаткову величину варіативності, що залишає трейдерам можливість розраховувати стан акцій та планувати власні дії.

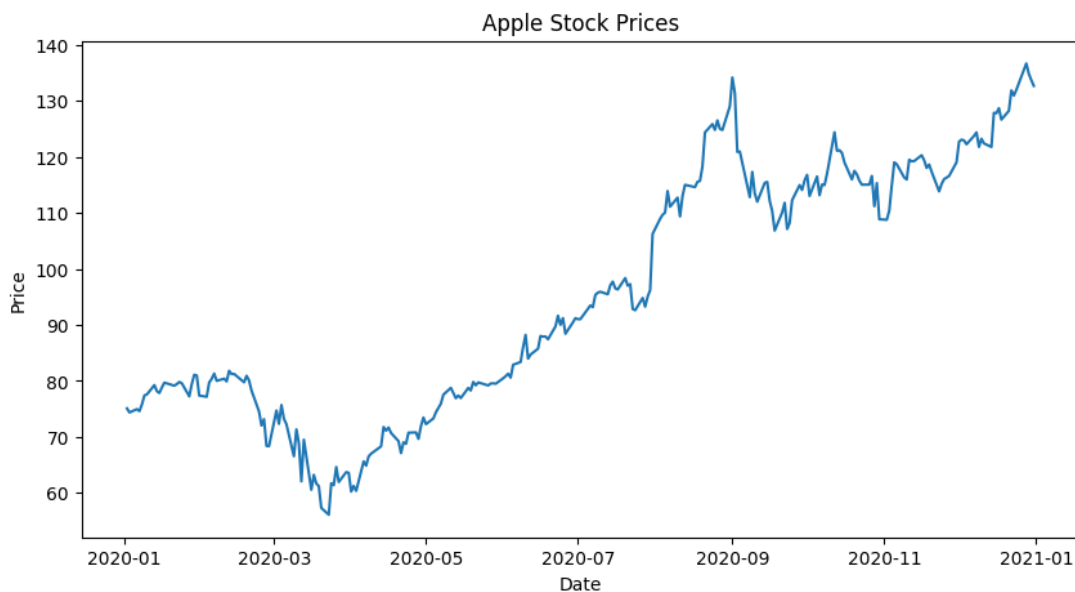


Рисунок 1. Графік акцій Apple

Для моделювання було обрано період в один рік. В цінах на акцій наявні всі історичні данні, тому для навчання моделі можна використовувати будь-який проміжок. Було виконано моделювання трьома моделями – випадковий ліс, лінійна регресія та метод SVM.

На Рис. 2 представлено графічні результати роботи моделей. Як видно, всі побудовані моделі відображають зміни руху в акціях. Разом з тим, очевидно що SVM в цьому випадку був найменш вдалою моделлю.

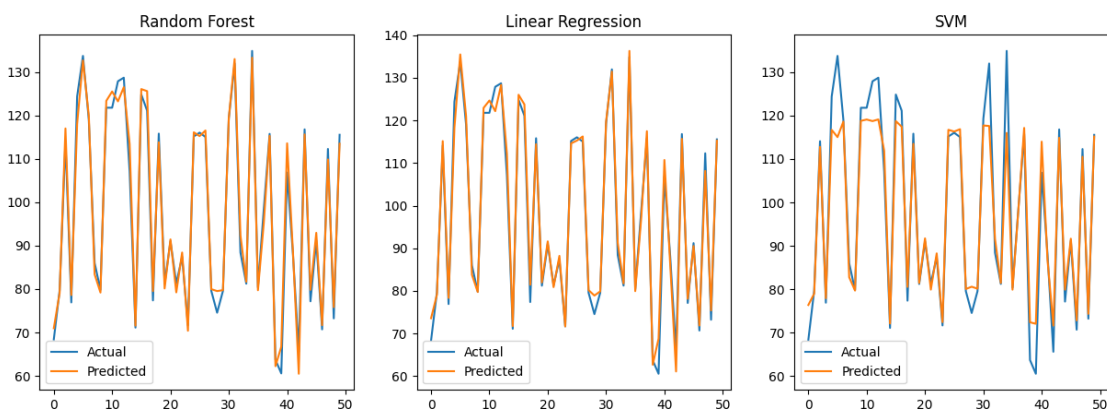


Рисунок 2. Графіки роботи моделей

Для порівняння наведемо таблицю результатів (Табл. 1) і порівняємо тільки дві найкращі моделі. Отримано значення VaR, а також величина помилки відповідно до різних критеріїв – MAE, MSE, RMSE. Найбільш вдалою є модель лінійної регресії, що має найкращі показники для помилки, а, отже, і точність. Загалом, лінійна регресія найкраще показує себе для задачі прогнозування стану акцій.

Таблиця 1. Параметри моделі

Metric	Random Forest	Linear Regression
VaR	3538.68	3518.17
MAE	2.15	1.91
MSE	7.78	6.99
RMSE	2.79	2.64
R ²	0.98	0.99

Отримані результати дозволяють розрахувати значення VaR. Альтернативним підходом до розрахунку VaR може слугувати моделювання методом Монте-Карло. Тут було обрано більший проміжок для дослідження ціни та виконане моделювання (Рис. 3). Метод Монте-Карло може використовувати різні розподіли під час моделювання, при цьому відомо, що розподіли акцій в більшій мірі слідує розподілу Стюдента, аніж нормальному розподілу.

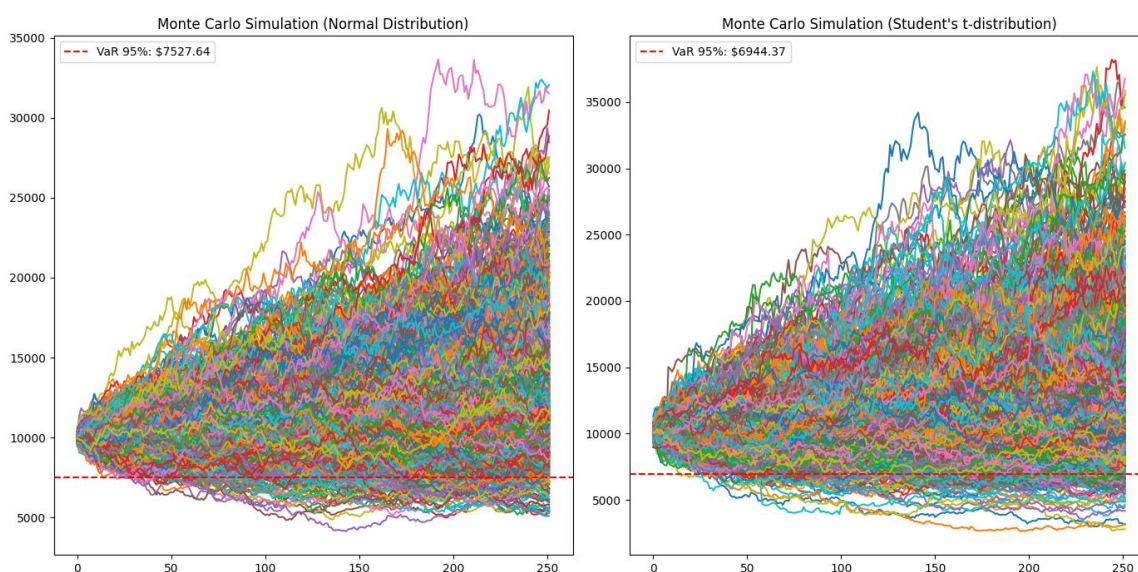


Рисунок 3. Моделювання методом Монте-Карло з різними розподілами

Розуміння поведінки акцій, які отримуються в результаті моделювання та прогнозування, надає можливість створення збалансованого портфолію з декількох акцій. Задача його розрахунку є ключовою для отримання прибутків.

Нами було обрано 4 акції – Apple, Google, Amazon та Tesla. Для них буде виконано оптимальний розрахунок співвідношення в портфелі протягом трьох років. Як видно з Рис. 4, оптимальне співвідношення змінювалося протягом часу разом зі зміною прибутковості акцій.

Під час певного періоду найбільш прибутковими були акції компанії Google, що майже повністю повинні складати портфель при його оптимальній побудові.



Рисунок 4. Графік зміни оптимальних пропорцій портфелю

Починаючи з середини 2020 року його прибутковість падає (Рис. 5). Акції компанії Amazon мають частку 0 для портфелю. Це пояснюється величиною зростання інших акцій, для яких прибуток був більший. Відповідно до складеного портфелю можна отримати значення портфелю за весь час.

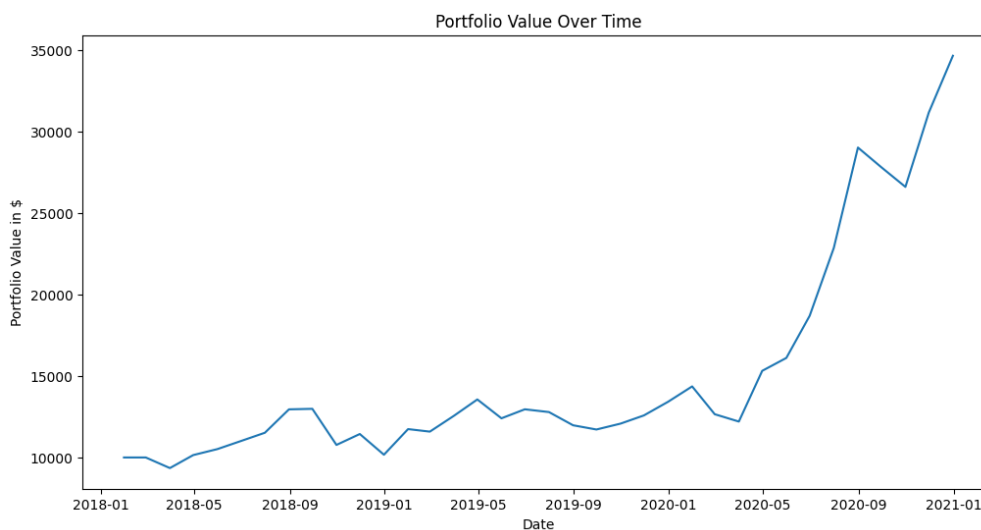


Рисунок 5. Графік прибутковості акцій

5. ВИСНОВКИ

Для проведення безбиткових операцій на фондових ринках потрібно опиратися на достовірний прогноз стану акцій. Для максимізації доходів інвестора необхідно розробити оптимальний портфель інвестора, що включатиме якнайбільше прибуткових акцій та найменше спадаючих акцій під час кризи. Розроблена СППР допомагає виконати отримані необхідні дані. Три розроблених моделі дозволяють отримати варіативність, що разом з модулем моделювання дозволяє отримати значення ключового значення параметру VaR різними способами.

Отримані за результатом моделі задовільно справились з задачею прогнозування, показали достатньо високу точність та низьку величину помилки. Шляхом відбору найкращої моделі можливо отримати найбільш підходящий для даної акції модель та прогноз.

Альтернативний спосіб обчислення VaR методом Монте-Карло дозволяє використовувати різні розподіли та надає додаткову варіативність. При наявності інформації про стан ринку та можливий розподіл – точність розрахунку даним методом зросте.

Метод побудови оптимального портфелю акцій дозволяє інтегрувати результати роботи модулів прогнозування та розрахувати найбільш прибуткове співвідношення для портфелю.

Загалом, побудована СППР виконує поставлені задачі та реалізує необхідний для роботи з фінансовими даними функціонал. Можливими шляхами до вдосконалення є автоматичний підбір розподілу для методу Монте-Карло використовуючи наявну інформацію про акції. Більша кількість моделей для прогнозування акцій також дозволить, при наявності відповідних можливостей для покриття апаратних вимог, покращити якість та точність прогнозування.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. G. H. Van Bruggen, A. Smidts, and B. Wierenga, “Improving decision making by means of a marketing decision support system”, *Management Science*, 44(5), pp. 645-658, 1998. doi: 10.1287/mnsc.44.5.645.

2. J. D. Little, “Decision support systems for marketing managers”, *Journal of Marketing*, 43(3), pp. 9-26, 1979. doi: 10.1177/002224297904300302.

3. R. Yang, L. et al., “Big data analytics for financial Market volatility forecast based on support vector machine”, *International Journal of Information Management*, 50, pp. 452-462, 2020. doi: 10.1016/j.ijinfomgt.2019.05.027.

4. W. Huang, Y. Nakamori, and S. Y. Wang, “Forecasting stock market movement direction with support vector machine”, *Computers & operations research*, 32(10), 2513-2522, 2005. doi: 10.1016/j.cor.2004.03.016.

5. T. L. Lai, and H. Xing, “Linear regression models”, *Statistical Models and Methods for Financial Markets*, pp.3-35, 2008. doi: 10.1007/978-0-387-77827-3_1.

6. R. Seethalakshmi, “Analysis of stock market predictor variables using Linear Regression”, *International Journal of Pure and Applied Mathematics*, 119(15), pp. 369-378. 2018. doi: 10.1109/eSmarTA52612.2021.9515762.

7. M. C. Fu, and J. Q. Hu, “Sensitivity analysis for Monte Carlo simulation of option pricing”, *Probability in the Engineering and Informational Sciences*, 9(3), pp. 417-446, 1995. doi: 10.1017/S0269964800003958.

СТАБІЛІЗАЦІЯ ДЕМОГРАФІЧНИХ ПРОЦЕСІВ ЗА ДОПОМОГОЮ УПРАВЛІННЯ ПО ПРОГНОЗУЮЧІЙ МОДЕЛІ

Нетудихата А.С.¹, Губарев В.Ф.

Національний технічний університет України «Київський політехнічний інститут
ім. Ігоря Сікорського», Київ, Україна

¹ netudykhata.arina@lil.kpi.ua

Мета дослідження полягає в стабілізації демографічних процесів через застосування управління по прогнозуючій моделі. Дослідження демографічних процесів здійснюється на основі моделі когнітивної карти. Отримана модель є нестійкою, тому при відсутності управління параметри демографічних процесів можуть необмежено збільшуватись. Тому необхідна стабілізація таких процесів. Модель було створено аналітичним методом на основі реальних даних. Методами моделювання досліджено поведінку демографічних процесів по отриманій моделі. Наукова новизна полягає в ефективному застосуванні прогнозного управління для стабілізації демографічних процесів. Практична значимість полягає в тому, що за допомогою запропонованого методу вдалося вирішити демографічну проблему в Україні. В роботі проведено багато чисельних експериментів, які демонструють ефективність прогнозного управління.

Ключові слова: демографічні показники, управління по прогнозуючій моделі, когнітивне моделювання, стабілізація, нестійкі процеси.

1. ВСТУП

Згідно з даними, опублікованими на офіційному веб-сайті The World Bank Group [1], показник загального населення України скоротився з 52 мільйонів до 38 мільйонів людей. Цей демографічний зсув створює необхідність у глибокому розумінні основних демографічних показників України для розробки ефективних стратегій управління населенням.

У контексті когнітивного моделювання важливо враховувати, що система може бути нестабільною, що ускладнює аналіз та передбачення демографічних процесів. Для досягнення стабільності та ефективного управління демографічними змінами, використання управління по прогнозуючій моделі [2] може виявитися важливим інструментом. У даному дослідженні розглядається можливість застосування цього підходу для стабілізації системи демографічних процесів в Україні.

Враховуючи нестабільність системи та важливість демографічних показників, дослідження спрямовано на розуміння та вирішення проблем, пов'язаних зі змінами у населенні. Підкреслюється актуальність вивчення демографічних тенденцій та їх впливу на суспільство для розробки ефективних стратегій управління населенням.

2. КОГНІТИВНЕ МОДЕЛЮВАННЯ ТА СТАБІЛІЗАЦІЯ

2.1. Стабілізація за допомогою управління по прогнозуючій моделі

Керовані демографічні процеси описуються рівняннями

$$\Delta\bar{Y}(k+1) = A\Delta\bar{Y}(k) + B\Delta\bar{U}(k), \quad (1)$$

де A і B – відомі матриці розмірностей 11×11 та 11×6 ; $\Delta\bar{Y}$ і $\Delta\bar{U}$ – вектори стану та керування розмірностей 11 та 6 відповідно.

Вважається, що $\Delta\bar{Y}$ або вимірюється в кожний момент часу, або оцінюється за допомогою оцінювача стану. Параметри матриці A такі, що система (1) є нестійкою (існують власні числа, що лежать за межами одиничного кола на комплексній площині).

Ставиться задача: синтезувати керування $\Delta\bar{U}$ таке, щоб де б система не знаходилася в момент часу k , існує керування, яке за скінчений час приводить систему до нульового стану $\Delta Y(k+s) = 0$, де s – скінчений час або горизонт керування. Іноді розглядають задачу стабілізації, яку формулюють як асимптотична стабілізація, коли s прямує до нескінченності.

Таку задачу будемо вирішувати не апіорі для всіх довільних станів, а безпосередньо у реальному часі, де система за результатами вимірювань опинилася. Таким чином реалізується зворотний зв'язок, що забезпечує стабілізацію. Якщо в точці k ми його знайшли, то в точці $k+1$ за результатами прогнозного розв'язку система повинна мати певне значення стану. За результатами вимірювань перевіряємо, чи співпадає прогнозне значення з вимірюваним чи ні. Якщо співпадає, то обчислене керування залишається незмінним. В протилежному випадку вирішується задача прогнозу для часу $k+1$ так, щоб система на горизонті s мала нульовий стан. Так робиться на кожному кроці функціонування системи у реальному часі. Це означає, що синтез керування робиться не апіорі, а у кожен наступний час або через декілька кроків у реальному процесі. В цьому полягає зміст управління по прогнозуючій моделі зі зворотним зв'язком.

Для реалізації переходимо до траєкторного опису моделі (1):

Нехай k – початкова точка, що змінюється у часі в процесі функціонування системи або є ковзною точкою. Траєкторію системи на ковзному інтервалі $[k, k+s]$ згідно з (1) запишемо як

$$\begin{aligned} \Delta\bar{Y}(k+1) &= A\Delta\bar{Y}(k) + B\Delta\bar{U}(k), \\ \Delta\bar{Y}(k+2) &= A^2\Delta\bar{Y}(k) + AB\Delta\bar{U}(k) + B\Delta\bar{U}(k+1), \\ &\vdots \\ \Delta\bar{Y}(k+s) &= A^s\Delta\bar{Y}(k) + A^{s-1}B\Delta\bar{U}(k) + \dots + B\Delta\bar{U}(k+s-1). \end{aligned} \quad (2)$$

Вирішується задача термінального керування: щоб за s кроків система мала нульовий стан. Перепишемо останнє рівняння в (2) у наступному вигляді

$$[B \ AB \ A^2B \ \dots \ A^{s-1}B] \cdot U(k, s) = \Delta\bar{Y}(k+s) - A^s\Delta\bar{Y}(k), \quad (3)$$

де $U(k, s)$ – це каскадний вектор, що має вигляд $U(k, s) = [\Delta\bar{U}^T(k+s-1) \ \Delta\bar{U}^T(k+s-2) \ \dots \ \Delta\bar{U}^T(k)]^T$, де «Т» – операція транспонування.

Вирішується задача стабілізації, коли термінальний стан в кінці горизонту повинен бути нульовим, тоді $\Delta\bar{Y}(k+s) = 0$ і рівняння (3) можна записати як

$$\Omega \cdot U(k, s) = -A^s\Delta\bar{Y}(k),$$

де Ω – матриця керуваності системи, тобто

$$\Omega = [B \ AB \ A^2B \ \dots \ A^{s-1}B]. \quad (4)$$

При синтезі керування на основі управління по прогнозуючій моделі зі зворотнім зв'язком спочатку обчислюється невизначеність у вигляді похибки кожної із компонент поточного вектора стану. В результаті замість точного значення $\{\Delta\bar{Y}(k)\}$ у кожній точці процесу керування маємо

$$\Delta\tilde{Y}(k) = \Delta\bar{Y}(k) + \delta\bar{Y}(k)$$

де $\Delta\bar{Y}(k)$ – точне прогнозне значення вектора стану у точці k , а $\delta\bar{Y}(k)$ – сумарний вектор похибки від різних факторів (похибка вимірювань, зовнішні збурення та інші). Опишемо, як

будемо визначати $\delta\bar{Y}(k)$. Для цього в розглядуваній точці знаходимо середнє по компонентах значення вектора $\Delta\bar{Y}(k)$, тобто

$$\Delta\bar{Y}_{\text{сеп}}(k) = \frac{\sum_{i=1}^n \Delta\bar{Y}_i(k)}{n},$$

де $\Delta\bar{Y}_i(k)$ – i -а компонента вектора $\Delta\bar{Y}(k)$, а n його розмірність ($n=11$).

Формуємо скалярну випадкову послідовність $\{\delta_i\}$, елементи якої рівномірно розподілені на інтервалі

$$-\varepsilon \leq \delta_i \leq \varepsilon,$$

де $\varepsilon = 0,1 \cdot \Delta\bar{Y}_{\text{сеп}}(k)$.

Розмірність випадкової послідовності $\{\delta_i\}$ повинна бути більша або дорівнювати u_n ($n = 11$), щоб визначити всі компоненти вектора $\delta\Delta\bar{Y}(k)$. Вони обираються з $\{\delta_i\}$ довільно.

Після цього синтезується управління по прогнозуючій моделі із зворотним зв'язком.

2.2. Визначення показників вершин для побудови когнітивної карти

Проаналізувавши показники з офіційного сайту Державної статистики України [3] на 2021 рік, було обрано наступні вершини для побудови когнітивної карти: 1 – середньомісячна заробітна плата одного працівника, 2 – індекс споживчих цін, 3 – експорт товарів, 4 – імпорт товарів, 5 – чисельність населення, 6 – народжуваність населення, 7 – смертність населення, 8 – міграційний рух населення, 9 – ВВП на душу населення, 10 – загальний рівень безробіття, 11 – інфляція.

Експертним шляхом було визначено причинно-наслідкові зв'язки, виділені у таблиці 1:

Таблиця 1. Причинно-наслідкові зв'язки між показниками для побудови когнітивної карти

№ вершини відповідно назві	1	2	3	4	5	6	7	8	9	10	11
1	0	0	0	0	0,02	0,1	0	0	0,7	-0,2	0,3
2	0,4	0	0	0	0	0,2	-0,3	0,3	0,4	0,2	0,5
3	0,3	-0,3	0	0	0	0	0	-0,2	0,4	0	0
4	-0,3	0,8	-0,2	0	0	0	0	0,3	-0,4	0	0,5
5	0	0	0	0	0	0	0	0,4	0	0,85	0
6	0	0	0	0	1	0	0	0	0,3	0	0
7	0	0	0	0	-1	0	0	-0,7	-0,55	0	0
8	0	0,3	0,2	0,2	0,5	0,1	0	0	0	0	0
9	0,7	0,5	0,1	-0,1	0	0,2	-0,2	-0,3	0	-0,45	0,3
10	-0,8	-0,3	0	0	0	0,2	0,4	0	-0,4	0	0
11	0,8	0,9	0,3	-0,9	0	-0,15	0	0,7	0,3	0,7	0

Примітка: напівжирним шрифтом в таблиці 1 виділено сильні абсолютні зв'язки з пороговим значенням 0,5.

Значення таблиці 1 сформували матрицю A . З даної матриці отримано наступні власні числа: $\lambda_1 \approx -0,286$, $\lambda_2 \approx 0,142$, $\lambda_3 \approx 1,389$, $\lambda_{4,5} \approx -0,359 \pm i \cdot (0,205)$, $\lambda_{6,7} \approx 0,689 \pm i \cdot (0,331)$, $\lambda_{8,9} \approx -0,883 \pm i \cdot (0,361)$, $\lambda_{10,11} \approx -0,068 \pm i \cdot (0,511)$. Серед власних чисел матриці A є числа більші за одиницю (наприклад λ_3). Таким чином можна зробити висновок, що система є нестійкою.

Було проведено дослідження обумовленості матриці керованості Ω (4). Було обчислено число обумовленості для різних значень горизонту s , починаючи від $s = 2$. Визначено, що найкраще число обумовленості отримано при $s = 2$, при наступних значеннях горизонту число обумовленості зростає.

3. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

3.1. Когнітивне моделювання без застосування управління по прогнозуючій моделі

Візьмемо початковий стан системи $[0.0 \ 0.0 \ 0.0 \ 0.0 \ 0.0 \ 0.0 \ 0.0 \ 0.0 \ 0.0 \ 0.0 \ 0.0 \ 0.0]$ і подамо імпульс -0.5 на вершини 10 і 11 та імпульс $+1$ на вершину 6. Зробимо 10 ітерацій.

На рисунку 1 зображено результати. Як бачимо, всі показники мають не дуже стабільну поведінку, при цьому тренд смертності має плавний зріст, а тренд народжуваності навпаки вийшов в нуль.

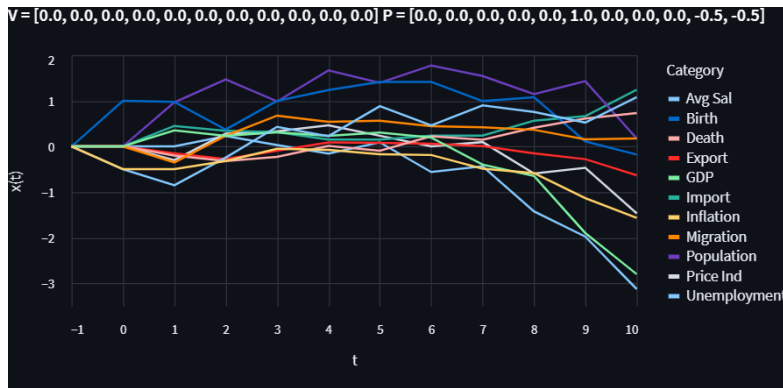


Рисунок 1. Імпульс $+1$ на народжуваність населення та -0.5 на безробіття та інфляцію

Візьмемо аналогічний початковий стан системи та подамо імпульс на вершини 1 та 6. Зробимо 8 ітерацій. З рисунку 2 можемо побачити, що всі параметри відповідають очікуванням і є більш-менш стабільними. Також варто зазначити, що показники народжуваності та чисельності населення, експорту товарів та ВВП зростають, показники імпорту товарів та смертності спадають. При цьому показники експорту товарів, міграційного руху та безробіття мають незначний зріст. Мінусом є те зростає інфляція, проте це очікувано відповідно до сильного зросту ВВП.

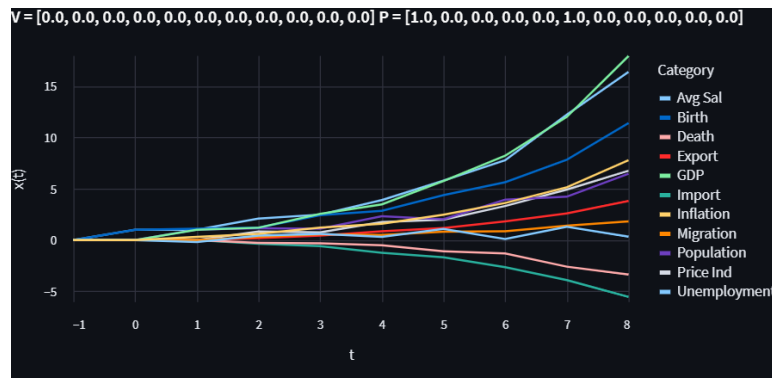


Рисунок 2. Імпульс $+1$ на народжуваність населення та середньомісячну заробітну плату одного працівника

3.2. Когнітивне моделювання із застосуванням управління по прогнозуючій моделі без зворотного зв'язку

Візьмемо компоненти початкового стану всі рівні 10, подамо імпульс на вершини 1 та 6.

На рисунку 3 можемо спостерігати, що система проявляє нестабільність, ускладнюючи можливість знаходження ефективного розв'язку. Графіки, які відображають поведінку системи, демонструють синусоїдальний характер, що свідчить про циклічні коливання та непередбачувані зміни у демографічних процесах, проте система приходиться в нуль. Очевидно,

що значення компонент початкового вектора завеликі та могли мати вплив на отриманий результат.

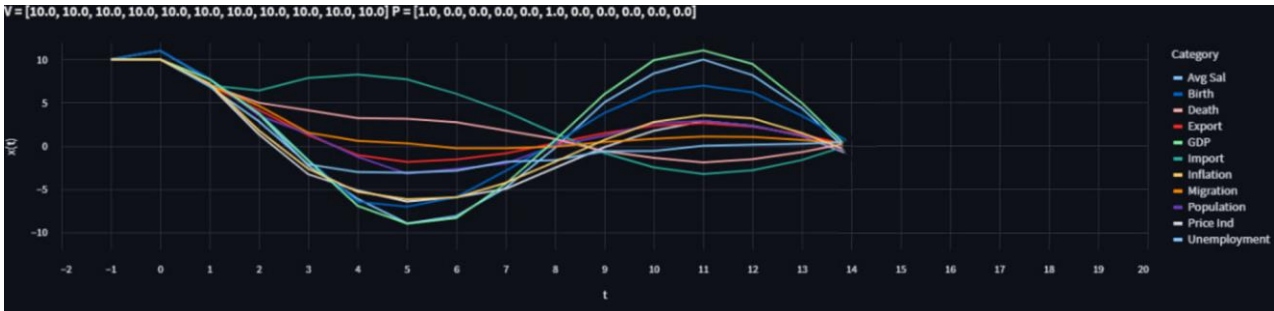


Рисунок 3. Нестабільне керування на основі управління по прогнозуючій моделі

Візьмемо початковий стан системи, де всі компоненти рівні 0. Застосуємо той самий імпульс на показники 1 та 6. Зменшимо довжину горизонту до оптимального, що виконуватиме умову найкращої обумовленості матриці керованості.

На рисунку 4 можемо спостерігати стабілізовану поведінку демографічних процесів. Графіки, що відображають поведінку системи, показують плавні та прогнозовані коливання, що свідчить про успішну стабілізацію демографічних процесів. Застосування управління по прогнозуючій моделі дозволило зменшити синусоїдальний характер графіків та забезпечити більш плавну та передбачувану динаміку. Система досягає нульового стану.

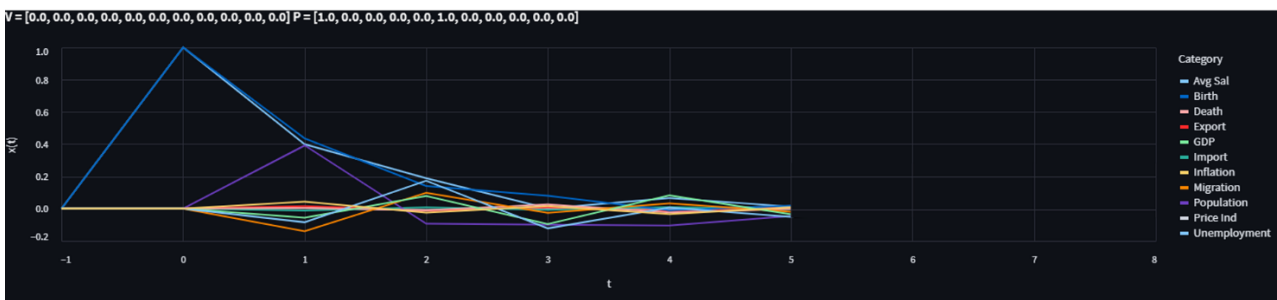


Рисунок 4. Стабільне керування на основі управління по прогнозуючій моделі

3.3. Когнітивне моделювання із застосуванням управління по прогнозуючій моделі зі зворотнім зв'язком

Візьмемо компоненти початкового стану всі рівні 10, подамо імпульс на вершини 1 та 6. На рисунку 5 можемо спостерігати графіки реального та прогнозованого значень для компоненти середньої заробітної плати. Система стабілізується, навіть враховуючи великі значення компонент початкового вектора, що свідчить про кращі результати за рахунок використання зворотного зв'язку.

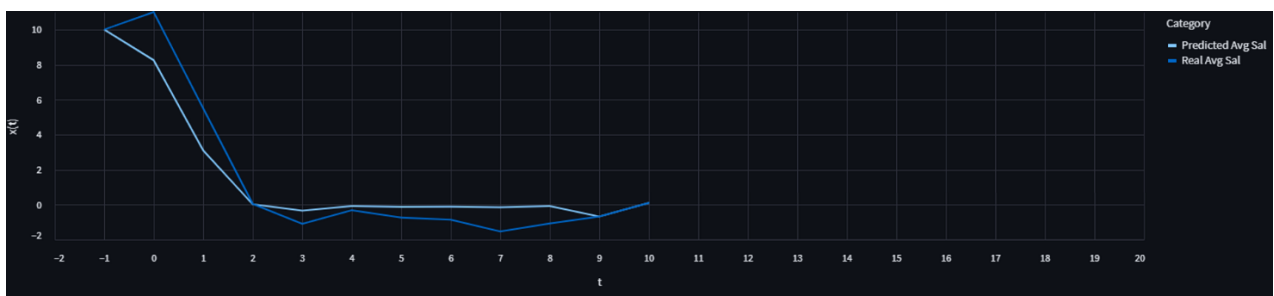


Рисунок 5. Керування на основі управління по прогнозуючій моделі

Візьмемо початковий стан системи, де всі компоненти рівні 0. Застосуємо той самий імпульс на показники 1 та 6. Зменшимо довжину горизонту до оптимального, що виконуватиме умову найкращої обумовленості матриці керованості.

На рисунку 6 бачимо графіки реального та прогнозованого значень для компоненти середньої заробітної плати – графіки відтворюють оптимальний результат вирішення проблеми. Система є достатньо стабільною, показники кращі ніж при моделюванні без зворотнього зв'язку, система плавно досягає нульового стану за скінченний час.



Рисунок 6. Керування на основі управління по прогнозуючій моделі

4. ВИСНОВКИ

В роботі було досліджено когнітивне моделювання демографічних показників України, за допомогою методу модального керування з декількома керуючими сигналами, управління по прогнозуючій моделі без зворотного зв'язку та зі зворотнім зв'язком.

Було визначено, що використання управління по прогнозуючій моделі, особливо зі зворотнім зв'язком, дозволяє досягти стабілізації демографічних процесів. Оптимальна ініціалізація та врахування умов найкращої обумовленості матриці керованості, горизонту, початкових компонент вектора $\Delta \bar{Y}(0)$ грають важливу роль у досягненні ефективних результатів. Управління по прогнозуючій моделі з врахуванням зворотного зв'язку є перспективним методом для стабілізації та оптимізації демографічних систем.

Результати дослідження підтверджують, що управління по прогнозуючій моделі є перспективним інструментом для стабілізації та оптимізації демографічних процесів. Порівняно з альтернативними методами, цей підхід виявляється більш ефективним та пристосованим до змін в системі.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. The World Bank Group. World Bank Open Data Ukraine. World Bank Open Data. URL: <https://data.worldbank.org/country/ukraine> (date of access: 05.10.2023).
2. Camacho E. F. Model predictive control. Berlin : Springer, 1999. 280 p.
3. Держстат України. Основні показники соціально-економічного розвитку України. Державна служба статистики України. URL: https://ukrstat.gov.ua/operativ/operativ2021/mp/arh_op2021.html (дата звернення: 12.09.2023).
4. Романенко В. Д., Мілявський Ю. Л. КОГНІТИВНЕ МОДЕЛЮВАННЯ ДИНАМІКИ ПРИЙНЯТТЯ РІШЕНЬ ДЛЯ СТАБІЛІЗАЦІЇ НЕСТІЙКИХ РЕЖИМІВ У СОЦІАЛЬНО-НАВЧАЛЬНОМУ ПРОЦЕСІ СТУДЕНТА. Наукові вісті НТУУ "КПІ". 2016. № 5. С. 48–53. URL: <https://doi.org/10.20535/1810-0546.2016.5.67264> (дата звернення: 15.08.2023).
5. V. Romanenko and Y. Milyavskiy, "Methods of impulse processes control for cognitive maps with delays", Naukovi visti NTUU "KPI", no. 5, pp. 57 – 63, 2015 (in Ukrainian).
6. В.Д. Романенко, Ю.Л. Мілявський. Методи керування імпульсними процесами когнітивних карт із запізненнями // Наукові вісті НТУУ «КПІ». 2015. – № 5. – С. 57 – 63.

ІНФОРМАЦІЙНА СИСТЕМА ДЛЯ КЛАСТЕРИЗАЦІЇ СТАНУ КРАЇН ЗА ПОКАЗНИКАМИ СТАЛОГО РОЗВИТКУ

Самсонюк М.В.¹, Бідюк П.І.²

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

¹ ms-0807@ukr.net [0000-0001-6593-3504], ² pbidyuke_00@ukr.net [0000-0002-7421-3565]

Метою дослідження є побудова інформаційної системи підтримки прийняття рішень (ІСППР), що ґрунтується на методах інтелектуального аналізу даних (ІАД). Вона дозволить провести кластеризацію стану країн за індикаторами сталого розвитку, а також порівняти існуючі моделі кластеризації за допомогою спеціальних критеріїв. У роботі використані експериментальні дослідження та методи математичного моделювання.

Ключові слова: кластеризація, сталий розвиток, критерії якості, математичні моделі, інтелектуальний аналіз даних, машинне навчання

1. ВСТУП

Сталий розвиток є дуже важливим для стабільного та щасливого життя людей на нашій планеті. У випадку, якщо такі проблеми людства як військові конфлікти, розрив між бідними і багатими, руйнування навколишнього середовища не будуть вирішені, людству загрожує повна загибель. Отже, дуже важливою задачею є побудова математичних моделей на основі даних сталого розвитку. Це допоможе оцінити те, у яких саме сферах є найбільші проблеми, які цілі сталого розвитку не досягаються, який вплив на систему керування державою ми можемо здійснити, щоб досягти цілей сталого розвитку та як саме прийти до мінімізації ймовірності настання військових конфліктів. Для України дана тема є особливо актуальною, оскільки вплив пандемії COVID-19 та повномасштабного вторгнення на життя людей є дуже значним. У рамках даного дослідження вирішується задача кластеризації стану країн за індексами сталого розвитку шляхом створення інформаційної системи, що базується на основі методів та моделей інтелектуального аналізу даних. Для дослідження були обрані наступні типи моделей: – k -means, agglomerative, fuzzy c -means, dbscan. Ці моделі були об'єднані в єдиний програмний продукт – інформаційну систему, яка за допомогою критеріїв оцінювання кластеризації (критерій компактності кластерів та Silhouette criterion) дала можливість визначити найкращі моделі.

2. МОДЕЛІ КЛАСТЕРИЗАЦІЇ І КРИТЕРІЇ ЯКОСТІ

За останні десятиліття було створено велику кількість моделей та методів для кластеризації, які застосовуються у багатьох сферах (медицина, бізнес, екологія, економіка тощо). Опишемо основні методи кластеризації, які були застосовані у даному дослідженні, а також методи для порівняння адекватності таких моделей.

2.1. Моделі інтелектуального аналізу для кластеризації даних

K -means або метод k -середніх є одним із найбільш популярних моделей. Він працює наступним чином. Обирається початкова кількість кластерів k . Для цього застосовується ліктьовий (elbow) метод. Він полягає у тому, що будується певна кількість моделей (для різної

кількості кластерів) та обчислюється метрика WCSS (within cluster sum of squares) – сума квадратів відстаней від точок до центрів кластерів:

$$WCSS = \sum_{P_i \text{ in Cluster } 1} distance(P_i, C_1)^2 + \dots + \sum_{P_i \text{ in Cluster } N} distance(P_i, C_N)^2,$$

де C_i – відповідний центроїд і-того кластеру.

Зі збільшенням кількості кластерів дана метрика прямуватиме до 0, але починаючи з певної кількості кластерів зміна буде незначною. Відповідно, така кількість кластерів і буде оптимальною.

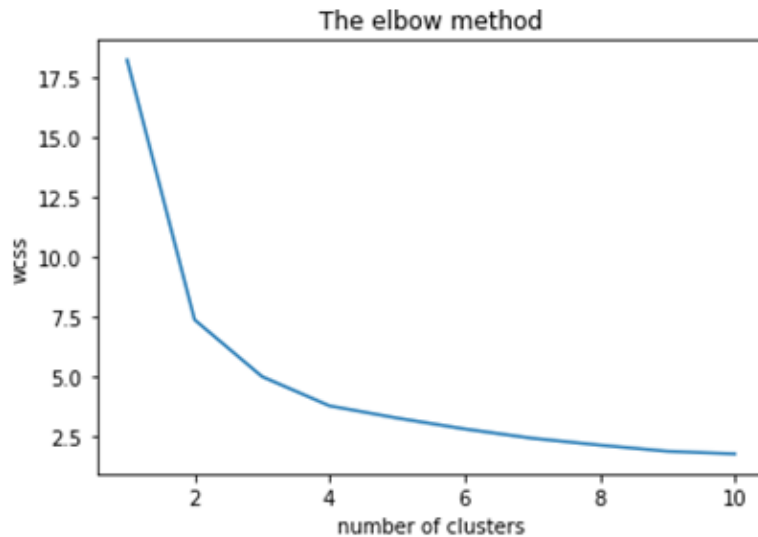


Рисунок 1. Приклад визначення кількості кластерів

Загалом, існують й інші методи визначення оптимальної кількості кластерів (наприклад, метод дендрограм), але в даному дослідженні застосовується саме ліктювий метод. Наступним етапом алгоритму є обчислення для кожної точки даних його евклідова відстань до кожного центру кластера. У якості відстані береться середньоквадратична норма l_2 , тобто цільовою функцією вважається:

$$S = \sum_{j=1}^k \sum \{|x_i - \mu_j|^2 | x_i \in c_j\},$$

де x_i – і-тий об'єкт, c_j – j-тий кластер з центром μ_j [1].

Алгоритм обчислює центроїди (centroids) - центри мас кластерів. Кожен центроїд - це вектор, елементи якого являють собою середні значення характеристик, обчислені по всіх точках кластера. Центр кластера зміщується в його центр ваги. Точки заново призначаються найближчого центру кластера. Етапи зміни центрів кластерів і перепризначення точок ітераційно повторюються до тих пір, поки кордони кластерів і розташування центроїдів не перестануть змінюватися, тобто, на кожній ітерації в кожен кластер будуть потрапляти одні і ті ж точки даних.

Agglomerative clustering відноситься до сімейства алгоритмів кластеризації, в основі яких лежать однакові принципи: алгоритм починає свою роботу з того, що кожен пункт даних заносить в свій власний кластер і по мірі виконання об'єднує два найбільш схожих між собою кластера до тих пір, поки не буде задоволено певний критерій зупинки. В основі цих критеріїв лежить відстань між двома існуючими кластерами.

$$\tilde{d}(c_i, c_j) = \frac{1}{n_i n_j} \sum_{x \in c_i} \sum_{y \in c_j} \|x - y\|,$$

де n_i, n_j – кількість об'єктів відповідно у і-тому та j-тому кластері.

Нечіткі методи (fuzzy c-means) відрізняються тим, що в результаті роботи алгоритму ми отримуємо розподілення точок на кластери з деяким значенням рівня належності: $w_{k,j} \in [0,1]$ k-го вектору ознак до j-го кластеру. При цьому виконується розрахунок $N \times m$ матриці $W = \{w_{k,j}\}$, яка називається матрицею нечіткого розбиття. Алгоритми фаззи-кластеризації, що засновані на цільових функціях, призначені для вирішення задач шляхом оптимізації деякого наперед заданого критерію якості кластеризації і являються найбільш строгими з математичної точки зору. Цільова функція, що мінімізується, має наступний вигляд:

$$E(w_{k,j}, c_j) = \sum_{k=1}^N \sum_{j=1}^m w_{k,j}^{\beta} d^2(x_k, c_j)$$

при обмеженнях:

$$\sum_{j=1}^m w_{k,j} = 1, \quad k = 1, \dots, N$$

$$0 < \sum_{k=1}^N w_{k,j} < N, \quad j = 1, \dots, m,$$

де c_j – прототип (центр) j-го кластера, β – невід’ємний параметр «фаззифікатор» (зазвичай, $\beta = 2$), $d^2(x_k, c_j)$ – відстань між x_k і c_j [3].

2.2. Критерії оцінки якості кластеризації

Так як ми достеменно не знаємо, яке розбиття даних є правильним, досить важко оцінити адекватність побудованих моделей кластеризації. На сьогоднішній день таких методик існує не так багато, проте є деякі метрики, що допомагають вирішити це питання. В рамках дослідження було застосовано два підходи – компактність кластерів та критерій силуету.

Ідея компактності кластерів полягає в тому, що чим ближче один до одного знаходяться об’єкти всередині кластерів, тим краще поділ. Таким чином, необхідно мінімізувати суму квадратів відхилень:

$$WSS = \sum_{j=1}^M \sum_{i=1}^{|c_j|} (x_{ij} - \bar{c}_j)^2,$$

де M – кількість кластерів [4].

Критерій силуету в свою чергу вказує на те, наскільки об’єкти схожі на свій кластер в порівнянні з іншими кластерами. Критерій силуету визначається наступним чином:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}.$$

Інакше можна записати:

$$s(i) = \begin{cases} 1 - \frac{a(i)}{b(i)}, & \text{if } a(i) < b(i) \\ 0, & \text{if } a(i) = b(i) \\ \frac{b(i)}{a(i)} - 1, & \text{if } a(i) > b(i) \end{cases},$$

де $a(i)$ – середня відстань між точкою і та іншими точками у тому ж кластері, $b(i)$ – середня відстань і до всіх точок в будь-якому іншому кластері (не в тому, де знаходиться точка і).

3. РЕЗУЛЬТАТИ МОДЕЛЮВАННЯ

Масив даних для дослідження складається з таких ознак: індикатор GINI, індекс стану здоров’я населення, показник рівня життя, індекс сталого розвитку. Використовуючи описані вище алгоритми, була розроблена інформаційна система для визначення оптимальної кількості кластерів, побудови моделей, обчислення критеріїв адекватності. Далі розглянемо отримані результати моделювання.

Таблиця 1. Порівняння реалізованих алгоритмів кластеризації

Алгоритм	WSS	Silhouette criterion
KMeans (K=3)	4.969	0.419
KMeans (K=4)	3.742	0.431
Agglomerative clustering (K=3)	5.032	0.422
Agglomerative clustering (K=4)	4.101	0.402
Нечіткий метод (K=3)	5.013	0.419
Нечіткий метод (K=4)	3.867	0.425

Отже, можемо бачити, що всі алгоритми з приблизно однаковою якістю розбили вхідні дані на кластери. Серед найкращих по метрикам можемо виділити метод K-середніх та нечіткий метод Бездека при кількості кластерів K=4. Далі опишемо отримані кластери та визначимо, які країни потрапили до відповідних кластерів.

Таблиця 2. Результат кластеризації методом k-середніх (K=4)

Номер кластеру	Країни
1	Australia, Austria, Belgium, Canada, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Israel, Italy, Japan, Korea Republic, Latvia, Lithuania, Luxembourg, Niderlandi, New Zeland, Norway, Poland, Portugal, Rumuniya, Slovachchina, Slovenia, Spain, Sweden, Switzerland, Great Britain, USA, Uruguay
2	Albania, Algeria, Armenia, Azerbaijan, Bosnia and Herzegovina, China, Egypt, Georgia, Indonesia, Yordaniya, Kazakhstan, Kyrgyzstan, Moldova..Republic, Mongolia, Morocco, Filippini, Russian Federation, Sri Lanka, Trinidad, Tunisia, Turkey, Ukraine, United by Arabskiye Emirati, Venezuela
3	Bangladesh, Benin, Cambodia, Cameroon, Ethiopia, Gambia, India, Kenya, Madagascar, Malawi, Mozambique, Nepal, Nicaragua, Niger, Pakistan, Senegal, Tajikistan, Tanzania., Uganda, Uzbekistan, Vjetnam, Zambia, Zimbabwe
4	Argentina, Bolivia, Botswana, Brazil, Bulgaria, Chile, Colombia, Costa Rica, Dominican Republic, Ecuador, El Salvador, Guatemala, Honduras, Jamaica, Malaysia, Mexico, Namibia, Panama, Paraguay, Peru, South Africa, Thailand

При розбитті на 4 кластери отримано такі результати. До першого кластеру потрапили найбільш розвинені країни з найвищими показниками (Західна Європа, Америка, Північна Європа). До другого кластеру потрапили країни з середніми значеннями усіх показників. Переважно сюди потрапили країни, які й раніше були у другому кластері (Albania, Algeria, Armenia, Turkey, Ukraine,..) – Східна Європа, країни Балканського півострову тощо. До третього кластеру віднесли країни з найнижчими показниками та середнім показником Джині. Це такі країни як Bangladesh, Benin, Cambodia, Cameroon, Ethiopia, Gambia, Honduras, India. До четвертого кластеру належать країни з найнижчим показником Gini та середніми іншими індексами. В основному, це країни Латинської Америки (Argentina, Bolivia, Botswana, Brazil, Bulgaria, Chile, Colombia, Costa Rica, Dominican Republic, ...).

4. ВИСНОВКИ

Отже, у ході виконання дослідження розроблена інформаційна система підтримки прийняття рішень, яка базується на основі методів інтелектуального аналізу даних та дає можливість виконувати кластеризацію країн світу за показниками сталого розвитку. Запропонована система виконує попередню обробку даних, побудову математичних моделей, перевірку якості (адекватності) моделей за спеціальними статистичними критеріями. Система розроблена мовою програмування Python, використовуючи бібліотеку scikit-learn. В ході роботи виконано експериментальне дослідження на основі даних за показниками сталого розвитку вибраних країн світу. В результаті було визначено, що серед представлених та апробованих моделей найкращими для застосування до заданого набору даних є модель k -середніх з кількістю кластерів $k = 4$. Для подальшого покращення результатів планується використання більш складних методів нечіткої кластеризації (Густафона-Кесселя, Гета-Леві), а також розширення масиву вхідних даних (додавання більшої кількості економічних та соціальних показників сталого розвитку).

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Шумейко А.А., Сотник С.Л. Інтелектуальний аналіз даних. Введення в data mining. Дніпро: Біла Е.А., 2012. 212 с.
2. Sustainable Development Report 2021. URL: <https://dashboards.sdgindex.org/rankings> (дата звернення: 20.11.2023).
3. Зайченко Ю.П., Гончар М.А. Нечіткі методи кластерного аналізу в задачах автоматичної класифікації в економіці. – Київ: Вісник НТУУ «КПІ», 2007. – с.197-204.
4. Алгоритми кластеризації на службі Data Mining. URL: <https://loginom.ru/blog/data-mining-clustering> (дата звернення: 25.11.2023).

ІНТЕЛЕКТУАЛЬНА СИСТЕМА ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ДЛЯ ДОСЛІДЖЕННЯ АКТУАРНИХ ФІНАНСОВИХ РИЗИКІВ

Чеманова А.О.¹, Кузнецова Н.В.²

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

¹ ankachemanova@gmail.com [0009-0008-6647-9364], ² natalia-kpi@ukr.net [0000-0002-1662-1974]

Дослідження актуарних (страхових) ризиків є однією з ключових задач у галузі страхування. Дослідження актуарних процесів є доволі складним процесом, оскільки потребує врахування різних показників і параметрів, використання різних підходів та методик оцінювання, а також достатньої статистичної бази. Метою нашого дослідження є розробка інтелектуальної системи підтримки прийняття рішень (ІСППР), яка буде надавати необхідну користувачу інформацію щодо конкретного страхового випадку з точки зору доцільності надання полісу. Для реалізації даної системи було обрано модульну структуру, а також три різні за принципом та структурою – інтелектуальної системи підтримки прийняття рішень. При цьому основною особливістю розробленої ІСППР є надання користувачеві конкретних рішень та порад щодо видачі полісу страхування.

Ключові слова: актуарні фінансові ризики, інтелектуальна система підтримки прийняття рішень, лінійна регресія, випадковий ліс, метод екстремального градієнтного підсилення, скорингова карта.

1. ВСТУП

У сучасному світі страхування широко представлене в багатьох сферах людського життя. Процес страхування невід’ємно пов’язаний з ризиком та включає в себе два протилежних за моделями поведінки учасники: страхувальника та застрахованої особи. При чому страхувальник прагне до страхування виключно тих випадків, де не очікуються збитки, а людина, що має поліс, не може спрогнозувати величину збитку або ж кількість таких випадків і час їх настання [1]. Це робить дослідження актуарних процесів складним – необхідно врахувати фактори та виконувати оцінку збиткам, імовірностям їх настання [2]. Окремою проблемою є проблема інтерпретації результатів. Особа, що приймає рішення (ОПР), використовує результати та оцінки, однак разом з тим повинна враховувати лише їх наближену точність, при цьому спосіб отримання та фактори, що впливають, залишаються ключовими, не зважаючи на лише наближену величину їх оцінки. Популярним способом розв’язання поставлених задач є використання інтелектуальної системи підтримки прийняття рішень [3]. Вона дозволяє отримати можливості для швидшої обробки даних, варіативності та легкої взаємодії з користувачем.

2. ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

Задачею даної роботи є побудова інтелектуальної системи підтримки прийняття рішень з набором необхідних модулів для виконання обробки, аналізу та перевірки даних. Побудована

система повинна надавати необхідну користувачу інформацію та мати можливість бути розширеною при необхідності додатковими модулями, тобто мати модульну структуру.

Метою роботи системи є надання користувачеві рішень та рекомендацій на основі прогнозів полісів. Це дозволить вирішити проблему автоматизованої видачі полісів та надасть ОПР готові результати.

В системі мають бути передбачені наступні модулі – модуль введення (завантаження) інформації, модуль виведення, модуль попередньої обробки даних, модуль побудови моделі та модуль побудови скорингової карти.

3. ОСНОВНІ МЕТОДИ

Для врахування можливих варіантів зв'язків між змінними, в тому числі і нелінійних, було вирішено побудувати декілька класів обрати моделей, що за своїм принципом і структурою повинні відрізнятися один від одного та належати до різних сімейств моделей. Було обрано наступні моделі: лінійна регресія, випадковий ліс та метод екстремального градієнтного підсилення.

Лінійна регресія – це метод, що базується на ідентифікації незалежних змінних, від яких залежить шукана цільова змінна, яка в свою чергу вважається залежною. Ключовим аспектом моделі є дві складові: визначення незалежних змінних, що будуть використанні для моделювання, та способу оптимізації параметрів моделі з метою найбільш оптимального, з точки зору критерію, підбору параметрів. Для розв'язання першої проблеми можливі два підходи, що будуть використовуватися в системі. Перший підхід – це пошук найбільш корельованих із залежною змінною незалежних змінних шляхом побудови матриці кореляцій. Слід також зазначити, що деякі змінні з набору даних можуть мати великий коефіцієнт кореляції з цільовою змінною, а також один з одним. Такі змінні було прийнято рішення видаляти. Другий підхід – це підхід експертний з метою надання особі, що відповідає за прийняття рішення, можливості вибрати деякі змінні, що точно будуть використовуватися для побудови моделі [4].

Метод випадкового лісу – це метод машинного навчання, що представляє собою комбінацію дерев рішень, що були побудовані на окремих частинах набору даних, та за допомогою яких приймається єдине рішення шляхом врахування та подальшого зваження рішення з кожного дерева. Цей підхід набирає популярність та застосовується у фінансовій сфері та не тільки, оскільки дозволяє моделювати нелінійні зв'язки між змінними [5].

Метод екстремального градієнтного підсилення – це ще один метод машинного навчання, що також використовує дерева, проте на відміну від методу випадкового лісу, він базується на поступовому додаванні нової складової до моделі – слабкої моделі. На кожному кроці навчання основна модель отримує додаткове дерево, що позитивно впливає на точність моделі. Існує цілий клас методів підсилення, зокрема метод екстремального градієнтного підсилення є одним з найбільш популярних [6, 7].

Окремим модулем є побудова скорингової карти. Скорингова карта є набором значень з кожної змінної-ознаки (для категорійних змінних – це окрема категорія, для неперервних – це інтервал) з певним числовим значенням. Кожне спостереження відповідно до карти отримує загальний бал або оцінку. Цей підхід є основним під час видання кредитів та є ключовим інструментом, яким користуються під час оцінки та класифікації клієнта. Це ефективний та легко зрозумілий метод, результати використовуються прямо на місці під час взаємодії з клієнтом [8].

Термінологія скорингових карт пов'язана з двома поняттями: WoE та IV.

WoE (1Weight of Evidence або вага (величина) спостереження (доказу)) – це коефіцієнт наявності залежності між певною групою в середині даних та цільовою змінною. Він розраховується за формулою (1).

$$WoE = \ln\left(\frac{Bad\ Results}{Good\ Results}\right), \quad (1)$$

де Bad Results – це кількість подій (позовів, скарг) у групі, тобто записів, в яких наявний позов або певна його критична величина, а Good Results – це кількість записів, по яким не було події, тобто поліси без скарг або такі, по яким не було збитків.

IV (Information Value або Величина інформації) – це коефіцієнт величини зв’язку між незалежною та цільовою змінною, що розраховується за формулою (2).

$$IV = \sum(\% \text{ of Good Results} - \% \text{ of Bad Results}) * WoE. \quad (2)$$

Ці величини будуть використанні для визначення змінних та категорій, що не несуть корисної інформації для побудови карти та можуть бути відкинуті.

4. РЕЗУЛЬТАТИ ДСЛІДЖЕННЯ

З метою перевірки роботи побудованої системи підтримки прийняття рішення було обрано тестовий набір даних, запропонований на Kaggle. [9] Це набір даних звернень від осіб, що мали страховий поліс та страховий випадок, пов’язаний з травмуванням. Слід зазначити, що подібні поліси представляють собою окрему категорію, оскільки для них не виконується припущення про незалежність кількості випадків та їх величини збитку.

Набір даних включає в себе 54 000 записів для побудови моделі та 36 000 записів для перевірки. Цільовою змінною є кінцева величина затрат по полісу.

На Рис. 1 побудований розподіл категоріальних змінних, змінної заробітку та змінних витрат по полісам, а на Рис. 2 – числовий розподіл цільової та фінансових змінних. Попередній огляд цих даних вказує на певну величину дисбалансу класів категорійних змінних та центрованості значень.

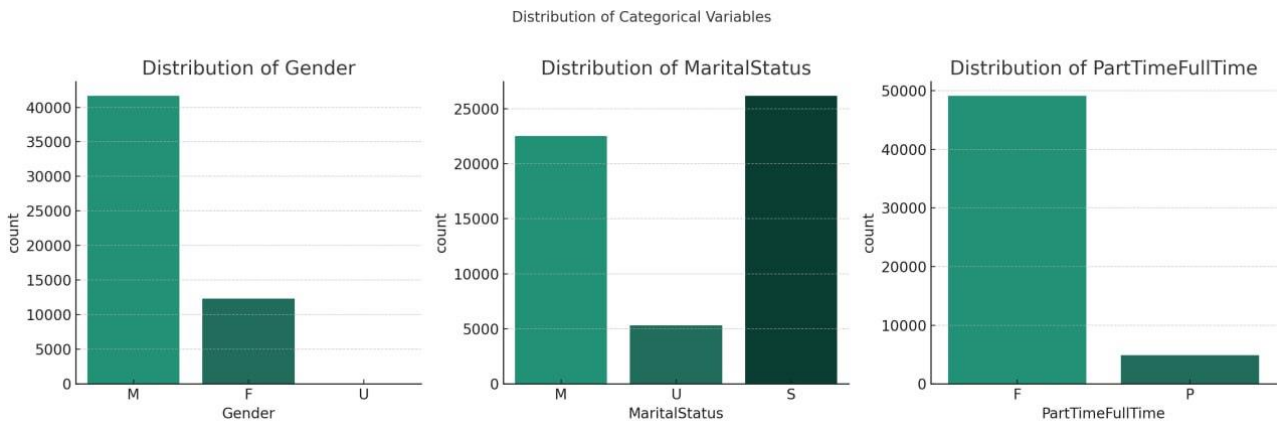


Рисунок 1. Розподіл категоріальних змінних, змінної заробітку та змінних витрат за полісами

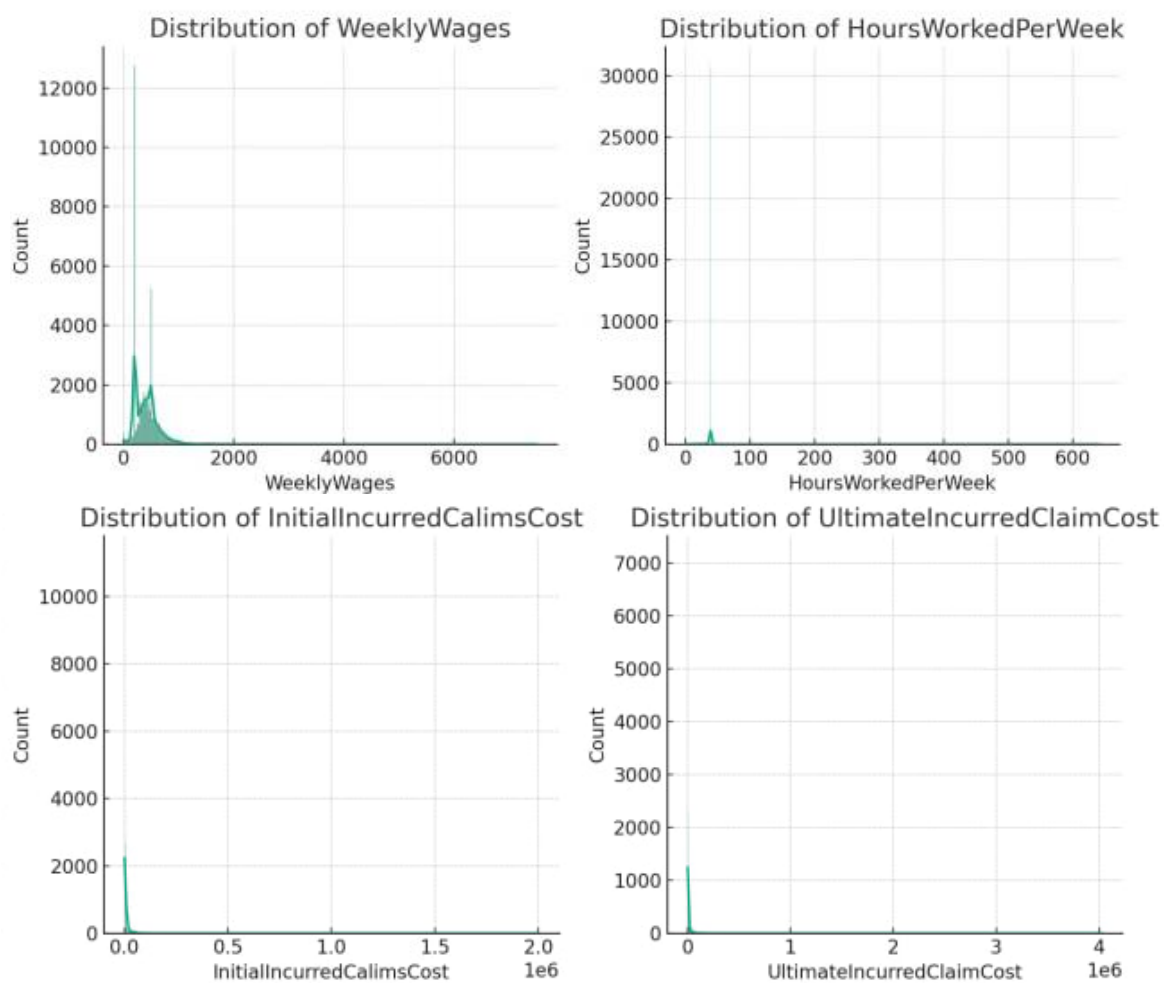


Рисунок 2. Числовий розподіл цільової та фінансових змінних

Проблема наявності незбалансованих класів є актуальною для проведення аналізу будь-яких вибірок. Значна диспропорція у класах, як, наприклад, видно для змінної `PartTimeFullTime`, може призвести до невірної оцінки та інтерпретації параметрів. Важливо використовувати моделі, що нечутливі до диспропорцій та будуть враховувати малозначущі класи.

Матриця кореляцій (Рис. 3) вказує про наявність набору ознак, що має відносно великий коефіцієнт кореляції.

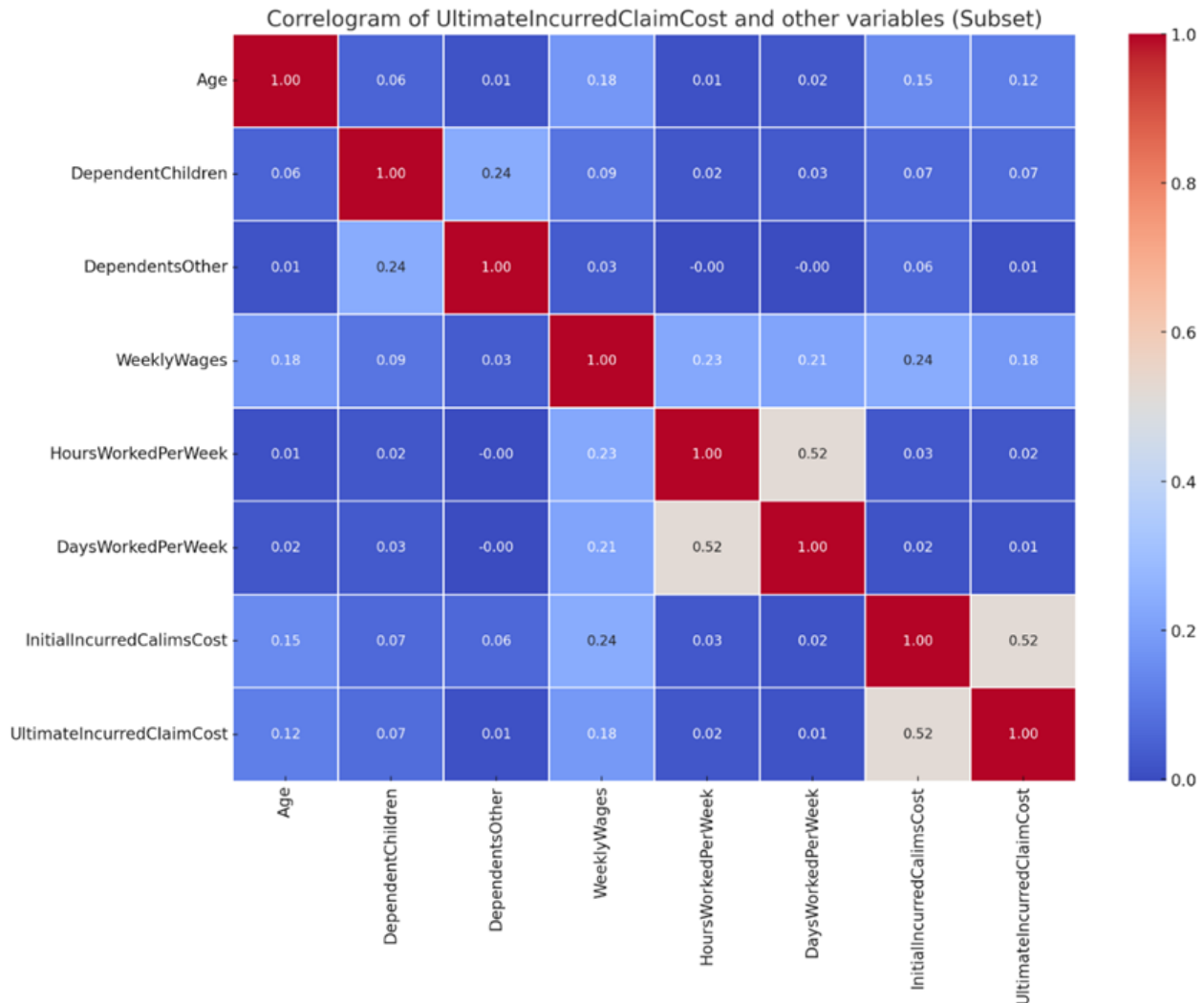


Рисунок 3. Матриця кореляцій

Для подальшої роботи системи було виконано синтез додаткової ознаки з метою покращення результатів моделі: кількість днів між подією та зверненням за полісом, а також було розраховано зміну відхилення початкової оцінки звернення та кінцевої величини збитків. Це дозволить визначити та розрахувати величину ризику, яка буде величиною помилки.

Результати роботи моделей представлені у Табл. 1.

Таблиця 1. Оцінки прогнозів побудованих моделей

Модель	MSE	RMSE	MAE	R2
Linear Regression	712949507,6221465	26701,11435169226	8668,957348620825	0,2146899381959123
Random Forest	32846600,81837761	5731,195409194979	361,3268974754968	0,9638196452301848
XGBoost	132286446,40038218	11501,584516942967	1006,4390728725784	0,8542871882400046

Найкращою моделлю виявилася модель випадкового лісу.

Результати роботи модулю з розрахунку скорингових оцінок представлені на Рис. 4–5.

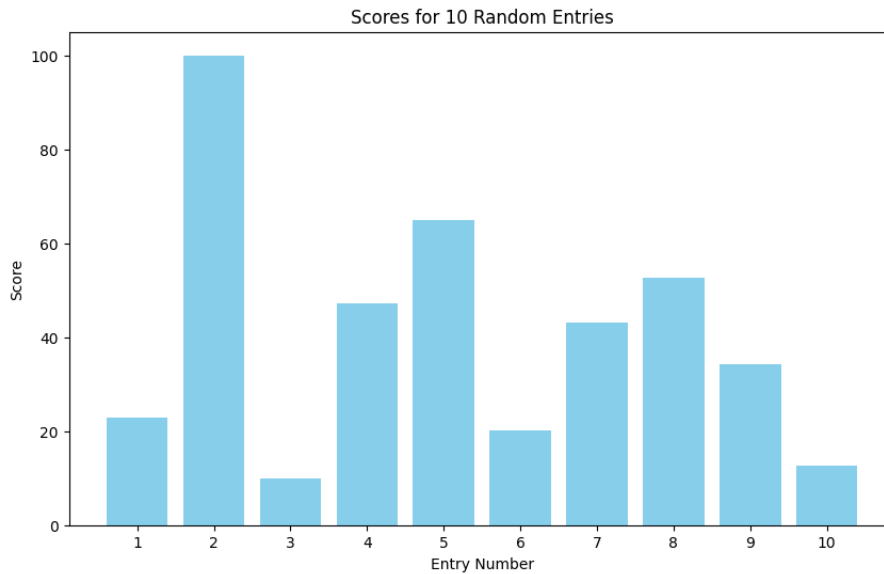


Рисунок 4. Розрахунок значення оцінки для 10 полісів

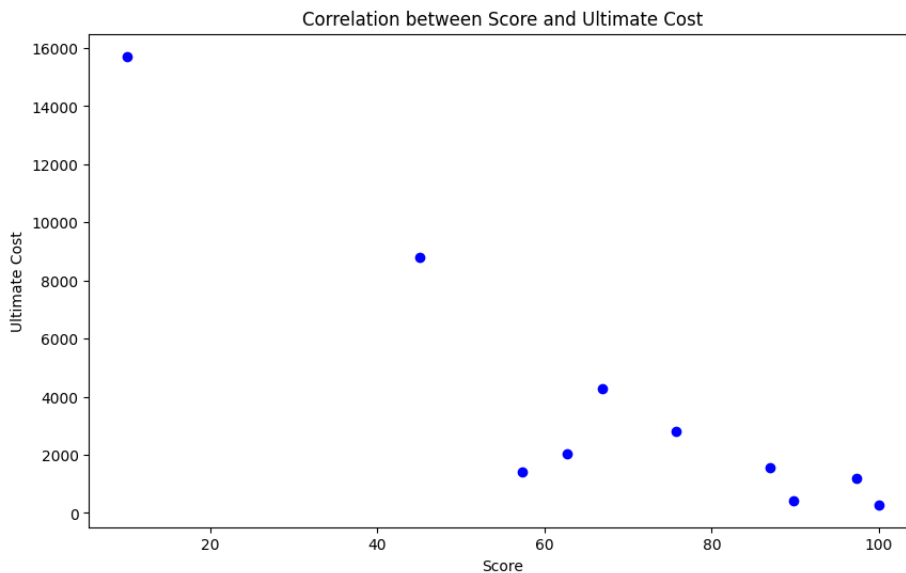


Рисунок 5. Зв'язок між величиною витрат та оцінкою (скором)

Результати показують ефективність скорингової моделі, проте разом з тим наявні і випадки розходження результатів. Позитивним результатом застосування моделі є те, що найменшу кількість балів отримують поліси з найбільшою величиною потенційного збитку. Це дозволяє відразу отримати уявлення про потенційну величину витрат, орієнтуючись на отриманий результат скорингової оцінки.

5. ВИСНОВКИ

Актуарні операції становлять значну частку всіх фінансових операцій. Для надання можливості користувачам проводити швидкий аналіз даних та функціоналу для розрахунку ключових показників, було побудовано інтелектуальну систему підтримки прийняття рішень.

Запропонована ІСППР надає можливість проводити розрахунки найбільш ключових параметрів та дозволяє експертну оцінку з метою врахування параметрів, які хоч і мають малу величину кореляції, проте мають відомий, потенційно, нелінійний зв'язок з цільовою змінною.

Використані та застосовані моделі дозволяють підходити до задачі прогнозування більш комплексно, враховувати можливі нелінійні зв'язки та шукати найкращу модель для прогнозування. Отримані результати для моделі випадкового лісу говорять про можливість застосування розробленого модуля на реальних даних.

Окремим модулем є модуль побудови скорингової карти. Сам концепт розроблявся з метою покращення розуміння результатів моделі та надання працівникам на місцях швидкого способу інтерпретації стану клієнта та потенційної величини збитків.

Все разом, реалізує прикладну систему, що виконує аналіз даних та надає всю необхідну для прийняття рішень інформацію.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Н. М. Внукова, В. І. Успенко, та Л. В. Временко, *Страховання: теорія та практика : навч. посіб. / за заг. ред. проф. Н. М. Внукової*. Харків, Україна: Бурун Книга, 2004.
2. Н.В Кузнецова, *Теорія і практика аналізу фінансових ризиків: системний підхід: монографія / Н. В. Кузнецова, П. І. Бідюк*, Київ, Україна: Ліра-К, 2020.
3. О. В. Нестеренко, О. І. Савенков, та О. О. Фаловський, *Інтелектуальні системи прийняття рішень: Навч. посібник*. Київ, Україна: Національна академія управління, 2016.
4. В. В. Головачко, та В. М. Головачко, "Поняття лінійної регресії", на *III Міжнародній науково-практичній конференції Сучасні тенденції розвитку науки й освіти в умовах поглиблення євроінтеграційних процесів*, Мукачево, 2023, с. 348-350.
5. J. Ali, R. Khan, N. Ahmad, and I. Maqsood, "Random Forests and Decision Trees", *IJCSI International Journal of Computer Science Issues*, Vol. 9, Issue 5, No 3, 2012.
6. T. Chen, and C. Guestrin, "XGBoost: A Scalable Tree Boosting System", in proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16, ACM Press, 2016, pp. 785-794. doi: 10.1145/2939672.2939785.pdf.
7. S. F. N. Islam, A. Sholahuddin, and A. S. Abdullah, "Extreme gradient boosting (XGBoost) method in making forecasting application and analysis of USD exchange rates against rupiah", *Journal of Physics: Conference Series* 1722 (2021) 012016, 2021, pp. 1-11. doi:10.1088/1742-6596/1722/1/012016.
8. N. Siddiqi, *Credit Risk Scorecards: Developing and Implementing Intelligent Credit Scoring*, New Jersey, Canada: John Wiley & Sons, Inc., Hoboken, 2006.
9. Actuarial loss prediction. [Електронний ресурс]. Доступно: <https://www.kaggle.com/competitions/actuarial-loss-estimation/overview>. Дата звернення: листопад 2023.

МОДЕЛЮВАННЯ ВПЛИВУ ЧАТ-БОТІВ НА ОСНОВІ ШТУЧНОГО ІНТЕЛЕКТУ НА ЯКІСТЬ ВИЩОЇ ОСВІТИ МЕТОДАМИ СИСТЕМНОГО АНАЛІЗУ

Чернюк О.Р.¹, Тимошук О.Л.²

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

¹ cherniuk.oleksii@iit.kpi.ua, ² o.tymoshchuk@kpi.ua [0000-0003-1863-3095]

У дослідженні запропоновано ефективні стратегії для покращення якості вищої освіти шляхом мінімізації негативних наслідків, пов'язаних із зловживанням чат-ботами студентами, і максимізації навчальних, практичних і наукових переваг, які можна отримати від взаємодії учасників навчального процесу зі штучним інтелектом чат-ботів. У дослідженні використано двоетапний метод модифікованого морфологічного аналізу та метод когнітивного моделювання. В результаті було розроблено дві взаємопов'язані моделі, які виявили чисельну ієрархію ефективності освітніх втручань для покращення якості вищої освіти. Ця ієрархія дій може служити цінним інструментом для педагогів і бути рекомендованою для впровадження в систему вищої освіти.

Ключові слова: якість вищої освіти, морфологічний аналіз, когнітивна карта, когнітивне моделювання, прийняття рішень, системний аналіз.

1. ВСТУП

Вища освіта є критично важливим компонентом суспільного розвитку, що надає людям знання, навички та кваліфікацію, необхідні для особистого та професійного зростання. Одним із визначних аспектів галузі вищої освіти, яка розвивається, є інтеграція технологій в освітні процеси.

Чат-боти, що базуються на технологіях штучного інтелекту (ШІ) і обробки природної мови (NLP), стали цінними інструментами в різних сферах, включаючи вищу освіту. Чат-боти – це комп'ютерні програми, призначені для імітації людської розмови та надання автоматичних відповідей на запити користувачів. Значний технологічний прорив здійснили генеративні чат-боти на основі штучного інтелекту, такі як Chat GPT (OpenAI), Bard (Google), Bing Chat (Microsoft), Perplexity AI, YouChat, Chatsonic, Aria та багато інших. Хоча основною функцією генеративних чат-ботів є імітація співрозмовника-людини, вони можуть виконувати дуже багато завдань. Наприклад, писати і налагоджувати код програм; відповідати на тестові питання з поясненнями; генерувати бізнес-ідеї; писати вірші, твори, статті, тексти пісень; перекладати, переписувати та резюмувати текст; емулювати систему Linux; моделювати цілі чати; грати в такі ігри, як «хрестики-нулики»; змоделювати банкомат; надавати психологічні консультації; розпізнавати зображення; вирішувати математичні задачі та багато іншого [1–5].

Зв'язок чат-ботів на основі штучного інтелекту з вищою освітою є неминучим, тому існує потреба критично вивчити їхній вплив на якість освіти. Це дослідження прагне надати інформацію та рекомендації для навчальних закладів, які, хочуть цього чи ні, змушені

взаємодіяти з новою реальністю, коли студент стає одним цілим з чат-ботом, і стає незрозуміло, де справжні знання студента, і в чому вони полягають.

2. ШКАЛА ВИМІРЮВАННЯ ВЗАЄМОВПЛИВУ АЛЬТЕРНАТИВ

Усі впливи вершин одна на одну відбуваються на інтервалі $[-1; +1]$. Вагу кожного впливу будемо розуміти так (Табл. 1):

Таблиця 1. Міри впливу

Дуже сильно негативний вплив	$[-1; -0,8)$
Сильно негативний вплив	$[-0,8; -0,6)$
Досить негативний вплив	$[-0,6; -0,4)$
Помірно негативний вплив	$[-0,4; -0,2)$
Легко негативний вплив	$[-0,2; -0)$
Прямого впливу немає	$\{0\}$
Легко позитивний вплив	$(0; +0,2]$
Помірно позитивний вплив	$(+0,2; +0,4]$
Досить позитивний вплив	$(+0,4; +0,6]$
Сильно позитивний вплив	$(+0,6; +0,8]$
Дуже сильно позитивний вплив	$(+0,8; +1]$

3. МОДЕЛЬ НА ОСНОВІ МЕТОДУ ДВОЕТАПНОГО МОДИФІКОВАНОГО МОРФОЛОГІЧНОГО АНАЛІЗУ

3.1. Характеристика вхідних даних

Для дослідження нашої тематики застосуємо метод двоетапного модифікованого морфологічного аналізу (МММА) [6–9, 17]. Експериментальна модель об'єкта буде заснована на морфологічних таблицях з двох характеристичних параметрів, кожен з яких має свої альтернативні сценарії.

Усі параметри морфологічної таблиці є якісними за своєю природою, тобто альтернативи таких параметрів принципово (якісно) відрізняються між собою, і між такими альтернативами неможливо встановити відношення переваги, як для кількісних параметрів.

Усі параметри є релевантними, тобто параметр повинен залежати або впливати на хоча б один інший параметр. В рамках деталізації, яка обрана для задачі. з параметрів, на основі яких визначається вплив на якість вищої освіти, було вилучено усі нерелевантні альтернативи.

У державних та університетських документах, підручниках та інших джерелах [10–12] можна знайти декілька десятків параметрів для оцінки якості вищої освіти, але в контексті нашої задачі було підібрано тринадцять релевантних показників ефективності студентської навчальної діяльності (Табл. 2).

Таблиця 2. Морфологічна таблиця першого етапу МММА (морфологічна таблиця сценаріїв)

Вплив чат-ботів на основі штучного інтелекту на якість вищої освіти	
Мета використання чат-ботів студентами	Студентські показники ефективності навчальної діяльності
1.1. Списування і халатність у навчанні	2.1. Оцінки студентів
1.2. Саморозвиток, навчальна комунікація, удосконалення навичок	2.2. Розуміння навчального матеріалу
	2.3. Базові практичні навички, здобуті під час навчання
	2.4. Навички критичного, творчого та незалежного мислення
	2.5. Запам'ятовування інформації
	2.6. Швидкість виконання навчальних завдань
	2.7. Дослідницька майстерність
	2.8. Відвідуваність занять та активність на них
	2.9. Здатність ефективно передавати думки усно і письмово
	2.10. Співпраця та командна робота
	2.11. Адаптивність та стійкість до нових ситуацій та викликів
	2.12. Організація часу (тайм-менеджмент)
	2.13. Технологічна компетентність

На основі джерел про генеративні чат-боти на основі ШІ [3–5] та експертного актуального навчального студентського досвіду було відібрано дев'ять способів контролю якості вищої освіти (Табл. 3) в контексті використання студентами чат-ботів. Альтернативи параметрів є взаємовиключними, що було враховано при проведенні процедури експертного оцінювання початкових наближень параметрів при експертному оцінюванні попарними порівняннями. Множина альтернатив є повною.

Таблиця 3. Морфологічна таблиця другого етапу МММА (морфологічна таблиця стратегій)

Способи контролю якості вищої освіти в контексті використання студентами чат-ботів
3.1 Використання чат-ботів в навчальних програмах предметів
3.2 Заборона використання гаджетів під час очних контрольних робіт і суворий контроль
3.3 Усні опитування віч на віч
3.4 Оцінювання за творчими індивідуальними унікальними роботами
3.5 Використання спеціального контролюючого програмного забезпечення в умовах дистанційного навчання під час контрольних робіт
3.6 Усунення гуманітарних або тестових завдань в якості способів оцінювання студентів
3.7 Вимога дотримання спеціального стилю оформлення робіт, який чат-боти не зможуть повторити
3.8 Написання контрольних робіт від руки перед камерою, яка фіксує руки та обличчя
3.9 Суворі часові обмеження для написання контрольних робіт

3.2. Перший етап МММА

Для початку оцінимо початкові ймовірності альтернатив. Головна мета першого етапу морфологічного аналізу - отримати початкові наближення $p_j^{(i)}$ для ймовірностей кожної з альтернатив $a_j^{(i)}$ характеристичних параметрів. Для альтернатив $a_j^{(i)}$, $j \in \overline{1, n}$ параметра «Мета використання чат-ботів студентами» експертами надається оцінка $p_j^{(i)}$. Для альтернативи «1.1» обрали значення 0,8, знаючи що в основному студенти використовують чат-боти для списування і халтури, для «2» відповідно значення 0,2. Для альтернатив параметра «Студентські показники ефективності навчальної діяльності» використали рівномірний розподіл, тобто однакову ймовірність для кожної його альтернативи. Оскільки ми не можемо апріорно отримати адекватні оцінки ймовірностей, використання експертної процедури для цього не є раціональним через значну невизначеність оцінок і через їх близькість. У такому випадку результат роботи МММА над параметром «Студентські показники ефективності навчальної діяльності» буде базуватись виключно на використанні матриці взаємозв'язків альтернатив параметрів і стане одним із важливих результатів даного дослідження.

Наведемо оцінену уже нормовану морфологічну таблицю (табл. 4).

Таблиця 4. Початкові ймовірності альтернатив

Мета використання чат-ботів студентами		Студентські показники ефективності навчальної діяльності	
Номер альтернативи	Ймовірність альтернативи	Номер альтернативи	Ймовірність альтернативи
1.1	0,8	2.1	0,076
1.2	0,2	2.2	0,077
		2.3	0,077
		2.4	0,077
		2.5	0,077
		2.6	0,077
		2.7	0,077
		2.8	0,077
		2.9	0,077
		2.10	0,077
		2.11	0,077
		2.12	0,077
		2.13	0,077

Далі оцінимо матрицю взаємозв'язків параметрів першого етапу.

Для врахування зв'язків між параметрами морфологічної таблиці (МТ) на основі таблиці 1 побудуємо числову матрицю взаємозв'язків альтернатив параметрів (Табл. 5).

Таблиця 5. Матриця взаємозв'язків параметрів першого етапу

	1.1	1.2
2.1	0,4	0,5
2.2	-0,8	0,3
2.3	-0,5	0,15
2.4	-0,3	0,15
2.5	-0,7	-0,2
2.6	0,8	0,2
2.7	-0,7	-0,2
2.8	-0,4	-0,2
2.9	-0,25	0,05
2.10	-0,35	-0,1
2.11	0,4	0,4
2.12	0,15	0,3
2.13	0,5	0,8

Нарешті, проведемо процедури з розрахунку ймовірностей альтернатив і конфігурацій. Щоб отримати остаточні значення ймовірності, необхідно розв'язати задачу розрахунку ймовірностей альтернатив параметрів. Подібні розрахунки повторюємо для всіх 26 конфігурацій, отримуємо результат першого етапу морфологічного аналізу – оцінки ймовірностей альтернатив з урахуванням зв'язків між ними (Табл. 6).

Таблиця 6. Результат першого етапу МММА

Мета використання чат-ботів студентами		Студентські показники ефективності навчальної діяльності	
Номер альтернативи	Ймовірність альтернативи	Номер альтернативи	Ймовірність альтернативи
1.1	0,74806734	2.1	0,11668415
1.2	0,25193266	2.2	0,03496632
		2.3	0,05244948
		2.4	0,06576998
		2.5	0,03330126
		2.6	0,13986528
		2.7	0,03330126
		2.8	0,05328201
		2.9	0,06743505
		2.10	0,0582772
		2.11	0,1165544
		2.12	0,09823871
		2.13	0,1298749

3.3. Другий етап МММА

Специфіка другого етапу МММА полягає в тому, що вибір альтернатив параметрів морфологічної таблиці стратегій залежить не від випадкових зовнішніх факторів, а від особи,

що приймає рішення. Тому на другому етапі для оцінки альтернатив і конфігурацій використовується величина очікуваної результативності, тобто вірогідності того, що вибір цієї альтернативи або конфігурації призведе до бажаних результатів.

Матриця зв'язків, яка співставляє кожен пару альтернатив першого і другого етапів наведена в таблиці 7.

Таблиця 7. Матриця зв'язків альтернатив першого та другого етапів

	3.1	3.2	3.3	3.4	3.5	3.6	3.7	3.8	3.9
1.1	0,5	1	1	0,65	0,9	0,6	0,15	1	0,4
1.2	0,5	0	0	0	0	0	0,1	0	0
2.1	0	-0,5	-0,4	0	-0,4	0	-0,1	-0,5	-0,5
2.2	0,3	0,5	0,4	0,6	0,3	-0,1	0	0,5	0,3
2.3	-0,25	0,7	0,3	0,5	0,3	-0,1	0	0,5	0,3
2.4	-0,15	0,2	0,3	0,6	0,1	0,2	0	0,2	-0,45
2.5	-0,35	0,5	0,3	0,15	0,3	0,2	0	0,4	0,2
2.6	0,7	-0,5	0	-0,25	0	-0,2	-0,2	-0,4	0,5
2.7	-0,1	-0,2	0	0,4	0	0,3	0	0	0
2.8	0,2	0,5	0,8	-0,2	0	0	0,1	0	0
2.9	-0,2	0	0,45	0,4	0	-0,25	0	0	0
2.10	0,2	0,3	0,15	0	-0,1	0	0,1	0	-0,4
2.11	0,6	0,35	0,4	0,1	0,1	0	0,1	0,25	0,45
2.12	0,55	0,3	0	0,25	0,3	0	0	0,2	0,7
2.13	0,85	-0,1	0	0,4	0,15	0	0,15	0,05	0

У результаті процедури з розрахунку оцінок альтернатив другого етапу отримуємо таблицю 8.

Таблиця 8. Розраховані на другому етапі МММА оцінки альтернатив

3.1. Використання чат-ботів в навчальних програмах предметів	0,13713944
3.3. Усні опитування віч на віч	0,129650906
3.4. Оцінювання за творчими індивідуальними унікальними роботами	0,116804045
3.2. Заборона використання гаджетів під час очних контрольних робіт і суворий контроль	0,116340931
3.5. Використання спеціального контролюючого програмного забезпечення в умовах дистанційного навчання під час контрольних робіт	0,114872605
3.8. Написання контрольних робіт від руки перед камерою, яка фіксує руки та обличчя	0,112691262
3.6. Усунення гуманітарних або тестових завдань в якості способів оцінювання студентів	0,096571773
3.9. Суворі часові обмеження для написання контрольних робіт	0,096144714
3.7. Вимога дотримання спеціального стилю оформлення робіт, який чат-боти не зможуть повторити	0,079784324

3.4. Результати моделювання МММА

У результаті першого етапу МММА отримали початкові наближення для ймовірностей кожної з альтернатив «1.1-1.2», «2.1-2.13». З отриманих даних можна зробити висновки, що приблизно три використання студентами генеративного чат-бота із чотирьох відбувається для списування і халтури. Було отримано різноманітну ієрархію показників студентської ефективності навчальної діяльності саме в контексті використання чат-ботів студентами (Рис. 1). Пам'ятаємо що вибіркове середнє рівномірного розподілу появи будь-якого параметра альтернативи «Студентські показники ефективності навчальної діяльності» дорівнює 0,077. Згідно з таблицею 8 можна зробити висновок, що використання чат-ботів студентами позитивно вплинуло на швидкість виконання навчальних завдань, технологічну компетентність, оцінки студентів, адаптивність та стійкість до нових ситуацій, організацію часу. І негативно - на запам'ятовування інформації, дослідницьку майстерність, розуміння навчального матеріалу, базові практичні навички, відвідуваність занять та активність на них, співпрацю та командну роботу, навички критичного, творчого та незалежного мислення, здатність ефективно передавати думки усно і письмово. Порівнявши кожне числове значення ефективності з початковим вибірковим середнім рівномірного розподілу, можна поррахувати приблизно, у скільки разів покращилася або погіршилася ситуація з відповідним показником студентської ефективності.

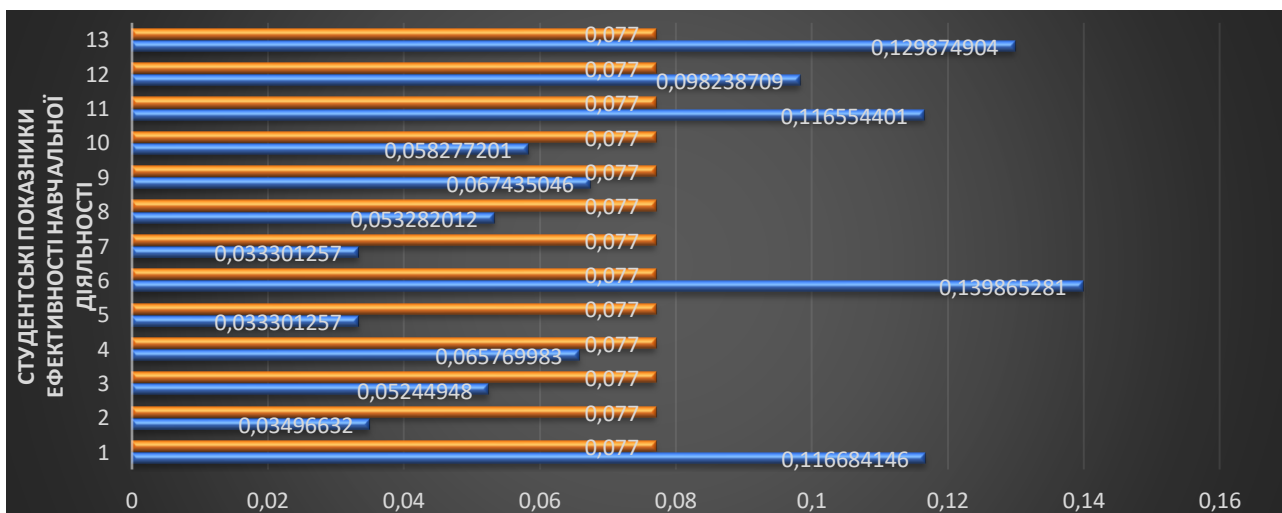


Рисунок 1. Порівняння ймовірностей альтернатив за рівнем позитивно-негативного впливу генеративних чат-ботів на студентські показники ефективності навчальної діяльності

У результаті другого етапу МММА отримали рейтинг способів контролю якості вищої освіти (Рис. 2).

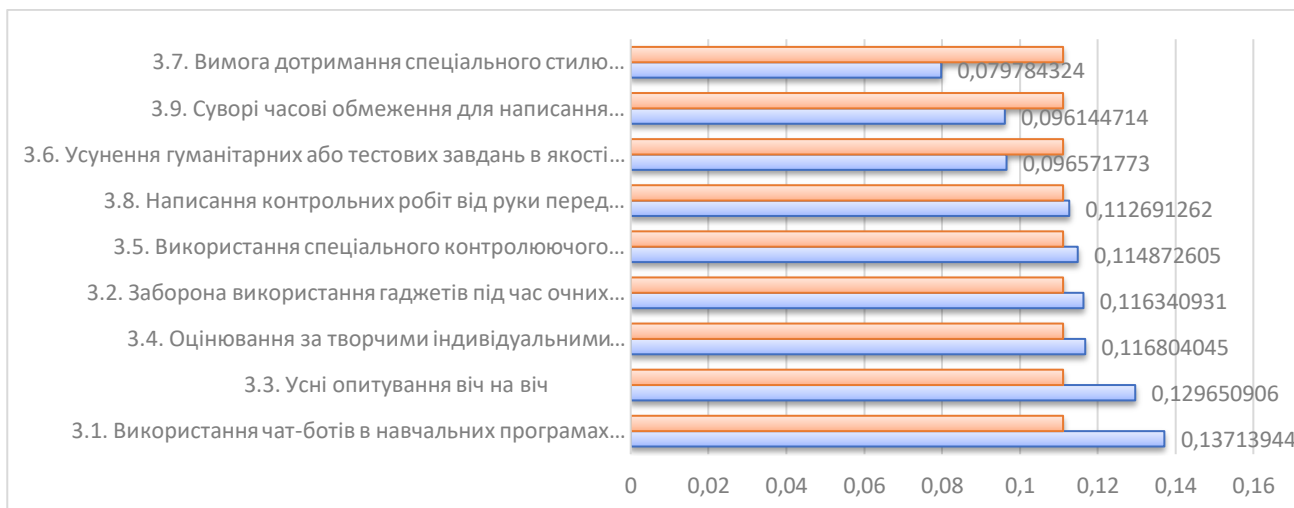


Рисунок 2. Порівняння ймовірностей результативності альтернатив для способів контролю якості вищої освіти

З діаграми можемо бачити, що майже усі величини очікуваної результативності близькі до вибіркового середнього 0,(11), тобто усі способи мають деякий позитивний вплив на якість вищої освіти в умовах використання чат-ботів студентами. Проте можемо виділити 6 найефективніших способів контролю якості вищої освіти в контексті використання студентами чат-ботів, а саме: використання чат-ботів в навчальних програмах предметів, усні опитування віч на віч, оцінювання за творчими індивідуальними унікальними роботами, заборона використання гаджетів під час очних контрольних робіт і суворий контроль, використання спеціального контролюючого програмного забезпечення в умовах дистанційного навчання під час контрольних робіт, написання контрольних робіт від руки перед камерою, яка фіксує руки та обличчя. Тобто 4 із 6 найефективніших способів є способами-обмежувачами, які спрямовані на посиленій контроль студента. Спосіб використання чат-ботів в навчальних програмах предметів набрав високий рівень ефективності через те, що на відміну від інших способів він має значний вплив на альтернативу «1.2»: студенти активно використовують чат-боти на основі штучного інтелекту для саморозвитку, навчальної комунікації і удосконалення своїх навичок, тобто таке рішення не лише допомагає зменшити шкідливий вплив чат-ботів, а також надає можливість отримувати від них значну користь.

4. КОГНІТИВНЕ МОДЕЛЮВАННЯ

В роботі також було застосоване когнітивне моделювання [13–17]. Вершини когнітивної карти (КК) повністю співпадають з альтернативами параметрів методу МММА з додаванням однієї цільової вершини «4. Якість вищої освіти», нумерація вершин також співпадає (Рис. 3).

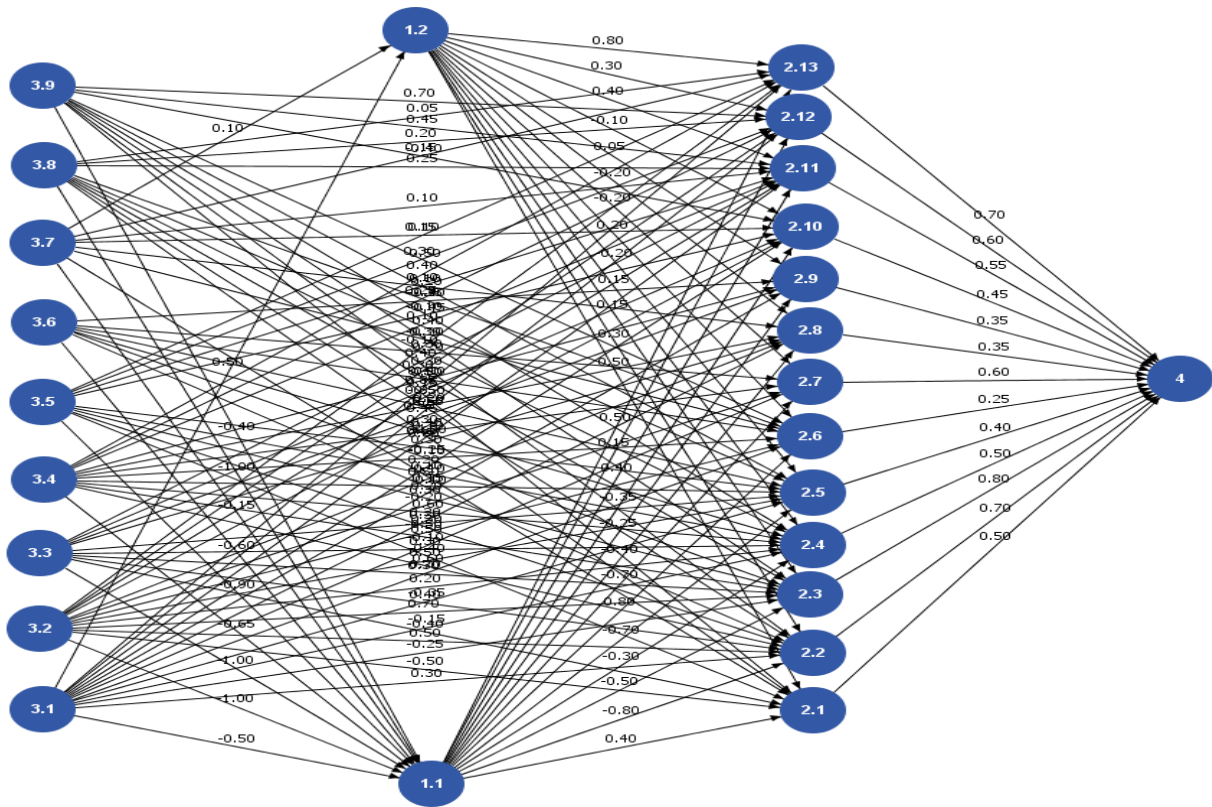


Рисунок 3. Когнітивна карта впливу способів контролю якості вищої освіти на студентські показники ефективності навчальної діяльності, які в свою чергу впливають на цільовий цільовий параметр «Якість вищої освіти»

Взаємовпливи між вершинами КК засновані на даних з таблиць 5, 8. До матриці суміжності КК (Рис. 4) були додані взаємодії параметрів 2.1 - 2.13 з параметром 4.

	1.1	1.2	2.1	2.2	2.3	2.4	2.5	2.6	2.7	2.8	2.9	2.1	2.11	2.12	2.13	3.1	3.2	3.3	3.4	3.5	3.6	3.7	3.8	3.9	4		
1.1	0	0	0.4	-0.8	-0.5	-0.3	-0.7	0.8	-0.7	-0.4	-0.25	-0.35	0.4	0.15	0.5	0	0	0	0	0	0	0	0	0	0	0	
1.2	0	0	0.5	0.3	0.15	0.15	-0.2	0.2	-0.2	-0.2	0.05	-0.1	0.4	0.3	0.8	0	0	0	0	0	0	0	0	0	0	0	
2.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.5	
2.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.7
2.3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.8
2.4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.5
2.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.4
2.6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.25
2.7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.6
2.8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.35
2.9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.35
2.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.45
2.11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.55
2.12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.6
2.13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.7
3.1	-0.5	0.5	0	0.3	-0.25	-0.15	-0.35	0.7	-0.1	0.2	-0.2	0.2	0.6	0.55	0.85	0	0	0	0	0	0	0	0	0	0	0	0
3.2	-1	0	-0.5	0.5	0.7	0.2	0.5	-0.5	-0.2	0.5	0	0.3	0.35	0.3	-0.1	0	0	0	0	0	0	0	0	0	0	0	0
3.3	-1	0	-0.4	0.4	0.3	0.3	0.3	0	0	0.8	0.45	0.15	0.4	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3.4	-0.65	0	0	0.6	0.5	0.6	0.15	-0.25	0.4	-0.2	0.4	0	0.1	0.25	0.4	0	0	0	0	0	0	0	0	0	0	0	0
3.5	-0.9	0	-0.4	0.3	0.3	0.1	0.3	0	0	0	0	-0.1	0.1	0.3	0.15	0	0	0	0	0	0	0	0	0	0	0	0
3.6	-0.6	0	0	-0.1	-0.1	0.2	0.2	-0.2	0.3	0	-0.25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3.7	-0.15	0.1	-0.1	0	0	0	0	-0.2	0	0.1	0	0.1	0.1	0	0.15	0	0	0	0	0	0	0	0	0	0	0	0
3.8	-1	0	-0.5	0.5	0.5	0.2	0.4	-0.4	0	0	0	0	0.25	0.2	0.05	0	0	0	0	0	0	0	0	0	0	0	0
3.9	-0.4	0	-0.5	0.3	0.3	-0.45	0.2	0.5	0	0	0	-0.4	0.45	0.7	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Рисунок 4. Матриця суміжності КК

Усі власні числа отриманої матриці суміжності рівні 0, тому дана КК є імпульсно стійкою і стійкою за значеннями (абсолютно). Над отриманою когнітивною моделлю було

проведено експериментальне тестування шляхом направлення на кожну керуючу вершину 3.1 – 3.9 одиничного додатного імпульсу (Рис. 5).



Рисунок 5. Ієрархія впливу способів контролю якості вищої освіти в контексті використання студентами чат-ботів на основі ШІ

Отже, створена когнітивна модель реагує на зовнішні зміни адекватно, адже при моделюванні вона показала цілком реалістичні закономірності. За результатами моделювання можемо виділити 5 найефективніших способів контролю якості вищої освіти, а саме: використання чат-ботів в навчальних програмах предметів, усні опитування віч-на-віч, заборона використання гаджетів під час очних контрольних робіт і суворий контроль, використання спеціального контролюючого програмного забезпечення в умовах дистанційного навчання під час контрольних робіт, написання контрольних робіт від руки перед камерою, яка фіксує руки та обличчя.

5. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Результати роботи двох методів для порівняння і наочності були зведені в єдину таблицю (Табл. 9), в якій можемо бачити ієрархію способів максимізації позитивного впливу чат-ботів на основі ШІ на якість вищої освіти.

Таблиця 9. Порівняння результатів моделювання методами морфологічного аналізу та когнітивного моделювання

Ієрархія пріоритетності	Метод двоетапного модифікованого морфологічного аналізу	Метод когнітивного моделювання
1	Використання чат-ботів в навчальних програмах предметів	Використання чат-ботів в навчальних програмах предметів
2	Усні опитування віч на віч	Усні опитування віч на віч
3	Оцінювання за творчими індивідуальними унікальними роботами	Написання контрольних робіт від руки перед камерою, яка фіксує руки та обличчя
4	Заборона використання гаджетів під час очних контрольних робіт і суровий контроль	Заборона використання гаджетів під час очних контрольних робіт і суровий контроль
5	Використання спеціального контролюючого програмного забезпечення в умовах дистанційного навчання під час контрольних робіт	Використання спеціального контролюючого програмного забезпечення в умовах дистанційного навчання під час контрольних робіт
6	Написання контрольних робіт від руки перед камерою, яка фіксує руки та обличчя	Оцінювання за творчими індивідуальними унікальними роботами
7	Усунення гуманітарних або тестових завдань в якості способів оцінювання студентів	Усунення гуманітарних або тестових завдань в якості способів оцінювання студентів
8	Суворі часові обмеження для написання контрольних робіт	Суворі часові обмеження для написання контрольних робіт
9	Вимога дотримання спеціального стилю оформлення робіт, який чат-боти не зможуть повторити	Вимога дотримання спеціального стилю оформлення робіт, який чат-боти не зможуть повторити

6. ВИСНОВКИ

Отримані результати є досить реалістичними, хоч і засновані на суб'єктивній оцінці. Дане дослідження є досить наочним і актуальним, адже побудоване на оцінці студента, який сам знаходиться у розглянутій ситуації і знає, як це працює зсередини. Була розроблена модель, яка може допомогти проаналізувати способи контролю якості вищої освіти в контексті використання студентами генеративних чат-ботів на основі ШІ.

Новизна роботи полягає в тому, що було досліджено конкретну складову впливу чат-ботів на якість вищої освіти і застосовано конкретні рішення, які дозволять підвищити якість вищої освіти. Особливістю даної роботи є те, що в ній для двох різних методів системного аналізу було використано майже однакові базові дані. Результати виявилися вражаюче подібними не дивлячись на різні способи їх отримання. Обидва методи показали майже однакові пріоритетні напрями для покращення якості вищої освіти. Такого результату вдалося досягти через експериментальне комбінування властивостей когнітивної і морфологічної моделей. Завдяки комплексному підходу визначення і оцінки параметрів результуючі моделі стали більш структурованими, репрезентативними, повноцінними, логічними, зрозумілими і адекватними.

Результати даного дослідження дозволять викладачам поставити пріоритети в тому, які методи взаємодії зі студентами будуть найбільш ефективними для покращення якості вищої освіти.

Ці моделі можна покращити шляхом уточнення початкових оцінок за допомогою великої кількості компетентних у даній області експертів, або за допомогою використання певних статистичних даних. У методі морфологічного аналізу можна додати нові контекстні параметри. У методі когнітивного моделювання можливо застосувати методи керування імпульсними процесами.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Abdullahi A. 10 Best AI Chatbots 2023. eWEEK. URL: <https://www.eweek.com/artificial-intelligence/best-ai-chatbots/#comparison-chart> (дата звернення: 10.11.2023).
2. Contributors to Wikimedia projects. ChatGPT. Wikipedia, the free encyclopedia. URL: <https://en.wikipedia.org/wiki/ChatGPT> (дата звернення: 15.11.2023).
3. Fengchun M., Wayne H. Guidance for generative AI in education and research. 7, place de Fontenoy, 75352 Paris 07 SP, France : UNESCO, 2023. 44 с. URL: <https://unesdoc.unesco.org/ark:/48223/pf0000386693> (дата звернення: 13.11.2023).
4. Hulick K. How ChatGPT and similar AI will disrupt education [Електронний ресурс] / Kathryn Hulick // ScienceNews. – 2023. – Режим доступу до ресурсу: <https://www.sciencenews.org/article/chatgpt-ai-artificial-intelligence-education-cheating-accuracy>.
5. Kamalov F., Santandreu Calonge D., Gurrib I. New Era of Artificial Intelligence in Education: Towards a Sustainable Multifaceted Revolution. *Sustainability*. 2023. Т. 15, № 16. С. 12451. URL: <https://doi.org/10.3390/su151612451> (дата звернення: 11.11.2023).
6. Морфологічний аналіз. Теорія, проблеми, застосування : навчальний посібник / Н.Д. Панкратова, І.О. Савченко ; М-во освіти і науки України, НТУУ "КПІ", Ін-т прикладного системного аналізу. Київ : Наукова думка, 2015. - 244 с.
7. Методологічне і математичне забезпечення розв'язання задач передбачення на основі модифікованого методу морфологічного аналізу / І.О. Савченко // Систем. дослідж. та інформ. технології. — 2011. — № 3. — С. 18-28. — Бібліогр.: 14 назв. — укр. URL: <http://dspace.nbuv.gov.ua/handle/123456789/50108>
8. Pankratova, N. & Haiko, Hennadii & Savchenko, Illia. (2021). Morphological model for underground crossings of water objects. *System research and information technologies*. 53-67. 10.20535/SRIT.2308-8893.2021.4.04. URL: https://www.researchgate.net/publication/358359378_Morphological_model_for_underground_crossings_of_water_objects
9. Панкратова, Н. Д. Стратегія застосування методу морфологічного аналізу в процесі технологічного передбачення / Н. Д. Панкратова, І. О. Савченко // Наукові вісті НТУУ «КПІ» : науково-технічний журнал. – 2009. – № 2(64). – С. 35–44. – Бібліогр.: 16 назв.
10. Педагогіка вищої школи [Електронний ресурс] : підручник / В. П. Головенкін ; КПІ ім. Ігоря Сікорського. – 2-ге вид., переробл. і доповн. – Електронні текстові дані (1 файл: 3,6 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2019. – 290 с.
11. Анненкова І. П. Критерії і показники якості освіти у ВНЗ. *Наука і освіта*. 2011. URL: https://scienceandeducation.pdpu.edu.ua/doc/2011/8_2011/1.pdf (дата звернення: 01.11.2023).
12. Гапон Л. О. Показники ефективності освітньої діяльності педагога. Методист ТКМЦНОІМ Гапон Л. О. Блог учителів української мови і літератури міста Тернополя. URL: <https://gapon.te.ua/rubryka-metodysta/dorobok-metodysta/metodychni->

rekomendatsii/item/1360-mekhanizm-pobudovy-u-zakladi-osvity-vnutrishnoyi-systemy-otsinyuvannya-yakosti-osvity (дата звернення: 21.10.2023).

13. Мілявський, Ю. Л. Ідентифікація та керування складними системами на основі моделей імпульсних процесів когнітивних карт : дис. ... д-ра техн. наук. : 01.05.04 Системний аналіз і теорія оптимальних рішень / Мілявський Юрій Леонідович. – Київ, 2019. – 297 с. URL: <https://ela.kpi.ua/handle/123456789/43829>

14. Roberts F. Discrete Mathematical Models with Applications to Social, Biological, and Environmental Problems. – Englewood Cliffs, Prentice-Hall, 1976. – 559 p.

15. Романенко В. Д. Когнітивне моделювання динаміки прийняття рішень для стабілізації нестійких режимів у соціально-навчальному процесі студента / В. Д. Романенко, Ю. Л. Мілявський // Наукові вісті НТУУ «КПІ» : науково-технічний журнал. – 2016. – № 5(109). – С. 48–53. – Бібліогр.: 10 назв. URL: <https://ela.kpi.ua/handle/123456789/18873>

16. Метод проектування когнітивної карти для оптимізації профорієнтаційної діяльності ЗВО. ВБ Мокін, ОВ Бурдейна, КО Коваль, АР Ящолт. ВНТУ, 2018. URL: https://www.researchgate.net/publication/330089909_METOD_PROEKTUVANNA_KOGNITIVNOI_KARTI_DLA_OPTIMIZACII_PROFORIENTACIINOI_DIALNOSTI_ZVO

17. Системный анализ : проблемы, методология, приложения: монография / М.З. Згуровский, Н.Д. Панкратова ; Министерство образования и науки, молодежи и спорта Украины, НАН Украины, Институт прикладного системного анализа. Киев : Наукова думка, 2011 - 726 с.

ПРОГНОЗУВАННЯ ГЕТЕРОСКЕДАСТИЧНИХ ПРОЦЕСІВ ДЛЯ ОЦІНЮВАННЯ ФІНАНСОВОГО РИЗИКУ

Байбара А.Г.¹, Кузнєцова Н.В.

Національний технічний університет України «Київський політехнічний інститут ім. Ігоря Сікорського»

¹ angelinabaibara@gmail.com

Ризики є елементами ведення фінансової діяльності та є наслідком невизначеності, тому для зменшення і компенсації їх негативних ефектів використовують усталені методи оцінки фінансового ризику. Метою роботи є оцінювання ризиків різними методами та їх порівняння для виявлення найбільш ефективних. В роботі представлені моделі для оцінки ризиків VaR та CVaR, а також гетероскедастичні моделі для опису динаміки волатильності фінансових процесів. Такий аналіз фінансово-економічних процесів та ризиків дає можливість для обґрунтованого прийняття рішень та управління ризиками в реальному світі.

Ключові слова: фінансові ризики, системний підхід, VAR, CVAR, аналіз ризиків, ринковий ризик.

1. ВСТУП

В сучасному фінансовому середовищі, де ринкові коливання можуть мати значущий вплив на інвестиційні портфелі та фінансові установи, точна оцінка ризику є надзвичайно важливим завданням для інвесторів та гравців на фінансових ринках при прийнятті рішень та побудови стратегій інвестування. Тож для всіх учасників ринку постає задача точної оцінки ризиків в умовах, коли більшість фінансово-економічних процесів характеризуються нестаціонарністю, нелінійністю та сильною волатильністю, а також сильною залежністю від багатьох зовнішніх факторів. Це, в свою чергу, ще більше обумовлює необхідність і важливість мати точне розуміння реального становище своїх позицій на ринку та можливих втрат. Дане дослідження спрямоване на вдосконалення методів прогнозування ризиків (VaR та CVaR), а також на їхнє застосування в контексті ефективного управління ризиком та його контролю на фінансових ринках.

2. МЕТОДИ ОЦІНЮВАННЯ ФІНАНСОВИХ РИЗИКІВ

Фінансові ризики зазвичай поділяють на чотири основні категорії: операційний ризик, ризик ліквідності, кредитний ризик і ринковий ризик. Під час формування вартості портфеля та розрахунку ризикового капіталу враховують всі ці ризики, зазвичай вони розглядаються окремо і працюють з ними незалежно. Ефективне управління ризиками виявляється критичним для визначення можливих втрат та захисту інвесторів від потенційних збитків у негативних сценаріях, або навіть від настання банкрутства фінансових установ та фондів.

Оскільки фінансовий ризик обумовлений невизначеністю щодо результатів фінансових операцій у майбутньому, отримання прийнятних оцінок не завжди є тривіальним завданням. Тому розвиток ефективних та науково обґрунтованих методів аналізу та оцінювання фінансових ризиків визначається як важливий напрямок науки.

Фінансово-економічні процеси характеризуються нестационарністю, нелінійністю та волатильністю. Традиційно волатильність відіграє важливу роль у вимірюванні ризику: щоб оцінити волатильність, необхідно розробити модель, що враховує зміни волатильності в часовому ряді на основі гетероскедастичних процесів [1]. Методи оцінювання фінансового ризику, такі як методологія VaR (Value at Risk) та моделі прогнозування дисперсії втрат, включаючи ARCH (Autoregressive Conditional Heteroskedasticity) та GARCH (Generalized Autoregressive Conditional Heteroskedasticity), стають об'єктом дослідження для розробки точних прогнозів та стратегій мінімізації ризику.

Вартість під ризиком (VaR) — метод, який використовується для оцінки фінансового ризику шляхом оцінки потенційно можливої несприятливої зміни в ринковій вартості портфеля із заданим рівнем довірчої ймовірності за певний період часу [2]. Перевагами даного підходу були простота і легкість в інтерпретації, а також наявність всього одного агрегованого показника. VaR показує величину збитків, яку компанія не перевищить з певною ймовірністю за певний період часу.

Прогнозування VaR та CVaR визначається необхідністю адекватної кількісної оцінки ризиків для прийняття обґрунтованих фінансових рішень [3]. У цьому контексті, прогнозування VaR надає можливість визначити верхню межу можливого збитку з певним рівнем ймовірності, що стає ключовим фактором для управління портфелем та прийняття рішень щодо розподілу активів, проте VaR може не дати повної картини ризиків, оскільки ця міра байдужа до всього, що виходить за межі її власного порогу. Спільно з цим, прогнозування CVaR допомагає врахувати не тільки екстремальні величини збитків, але й їхню середню інтенсивність, що дозволяє ліпше розуміти загальний ризик у разі великих втрат.

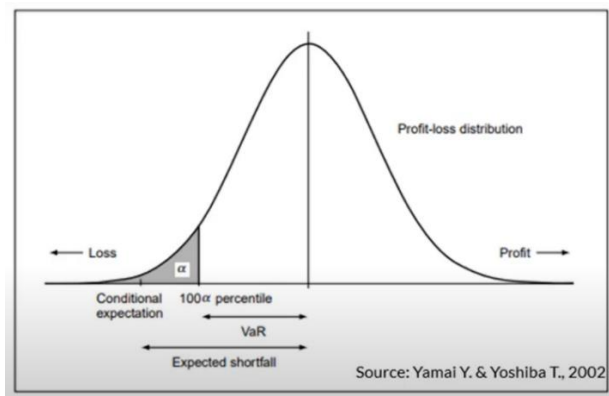


Рисунок 1. Візуалізація VaR та CVaR на графіку розподілу портфелю

Методи оцінювання VaR поділяються на дві групи: параметричні та непараметричні. До непараметричних відносяться історичний метод та метод моделювання Монте Карло, який відрізняється від попередніх більшою складністю обчислень, однак показує більш точні результати.

Основною проблемою під час обчислення VaR параметричними методами є моделювання та прогнозування волатильності доходності інструментів. Використання дельта-нормального методу для VaR передбачає оцінку волатильності доходності фінансових інструментів, для чого важливо врахувати змінність дисперсії в часі.

Динаміка дисперсії фінансових процесів може бути адекватно описана моделлю авторегресії з умовною гетероскедастичністю (ARCH) [4]. Гетероскедастичність вказує на змінність дисперсії у часі, і використання ARCH моделі дозволяє побудувати математичну структуру, яка враховує цю змінність і на її основі можна робити прогнози на наступні кроки. Також для моделювання моделювання волатильності широко використовується узагальнена

модель авторегресії з умовною гетероскедастичністю (GARCH модель) [5], та їх модифікації, такі як EGARCH, FIGARCH моделі [6].

Створивши адекватну модель, ми зможемо суттєво покращити якість управління ризиком, оскільки це дозволить нам краще розуміти та передбачати зміни в динаміці ризику у фінансових процесах

Для впевненості в адекватності використовуваної моделі оцінки ризику, необхідно провести її верифікацію, що включає процес "бек-тестування" [7]. Верифікація дозволяє визначити, наскільки точно модель VaR відображає реальні умови ринку.

3. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Для застосування методологій оцінювання ринкового ризику було сформовано інвестиційний портфель з акцій компаній Apple (AAPL), META, The Walt Disney Company (DIS), Citigroup Inc (C). Будемо оцінювати ризик на основі щоденних цін закриття акцій за 2 роки. Таким чином, часовий період для значень акцій компаній — 4 роки (з 1 січня 2019 по 1 січня 2023), а кількість спостережень — 1008.

Найпростішим методом розрахунку VaR є історичний метод, згідно з яким модель історичної прибутковості вказує на модель майбутніх прибутків. Припущення щодо розподілу доходів не робиться, а моделюється на основі історичних емпіричних даних. Візуалізуємо для цього методу на гістограмі розподілу доходностей портфелю підраховані значення VaR та CVaR для рівня довіри 95%, який використовується в системі RiskMetrics (Рис. 2).

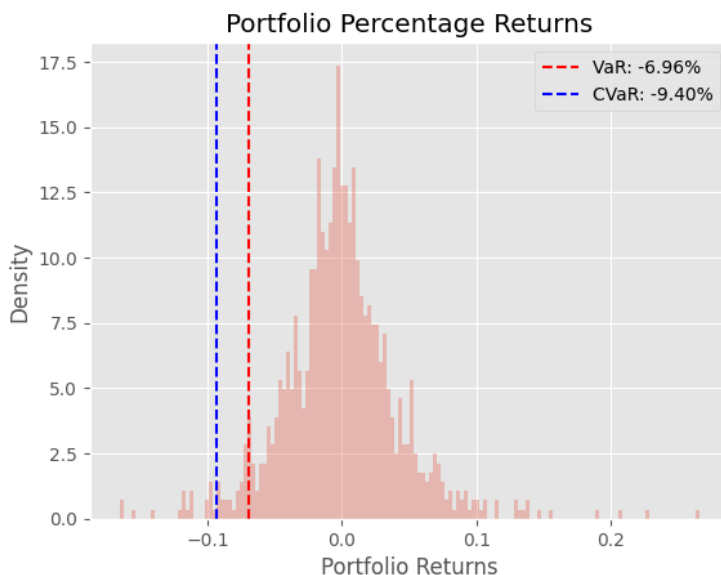


Рисунок 2. Розподіл прибутковості портфелю з VaR та Cvar за історичним методом

Для розрахунку VaR варіаційно-коваріаційна методом припущення полягає в тому, що доходи розподілені нормально і використовуються історичні доходи портфелю та стандартні відхилення (оцінки) для визначення параметрів моделі. Визначаються параметри нормального розподілу, що найкращим чином апроксимує фактичний розподіл розглянутого ринкового показника. Далі необхідно визначити значення зворотного нормального розподілу – для довгої позиції, а отже, негативної зміни, на основі параметрів, отриманих раніше, і зворотного рівня довіри.

Наступним методом оцінки VaR є параметричний метод на основі прогнозованого значення волатильності ряду. Для моделювання дисперсії, що є гетероскедастичним процесом, було застосовано такі моделі – ARCH, GARCH, EGARCH, FIGARCH.

Результати порівняння якості побудованих моделей зведено до Таблиці 1.

Таблиця 1. Характеристики різних моделей прогнозування волатильності портфелю акцій

Тип моделі	AIC	BIC
ARCH(1)	-6213,287764	-6203,013207
GARCH(1,1)	-6372,040331	-6356,628496
EARCH(1,1)	-6194,814760	-6184,540203
FIGARCH(1,1)	-6362,575627	-6342,026513

За результатами моделювання всі моделі виявились задовільними та можуть використовуватись для подальших оцінок рівня ризику, проте за інформаційними критеріями Акайке і Баеса-Шварца, найкращою моделлю для волатильності і найбільшим точним описом серії є GARCH(1,1).

Останнім методом оцінки ризику є метод Монте-Карло. Метод Монте-Карло є найбільш складним методом розрахунку VaR, проте його точність може бути значно вищою, ніж у інших методів. Метод Монте-Карло передбачає здійснення великої кількості випробувань – разових моделювань розвитку ситуації на ринку з розрахунком одержаного результату за портфелем. За підсумком випробувань формується розподіл можливих результатів. Відповідно до обраного рівня вірогідності відсікаються найгірші варіанти і одержується VaR-оцінка. Для найбільшої точності, характеристики стохастичного процесу, покладеного в основу симуляції, мають збігатись з аналогічними показниками процесу, який ми досліджуємо, але це не обов'язково.

Оскільки оцінка VaR методом Монте-Карло майже завжди виконується за допомогою програмних засобів, які вимагають великої кількості симуляцій для формування прогнозу. Іншими словами, метод Монте-Карло дозволяє використовувати моделі ризиків практично будь-якої складності при розрахунках. При цьому, збільшення кількості симуляцій підвищує точність обчислень. Застосувавши 400 симуляцій, було визначено VaR для довірчого інтервалу – 95%, а також визначено показники очікуваних втрат CVaR при початковій інвестиції у розмірі 100000\$ (Рис. 3).

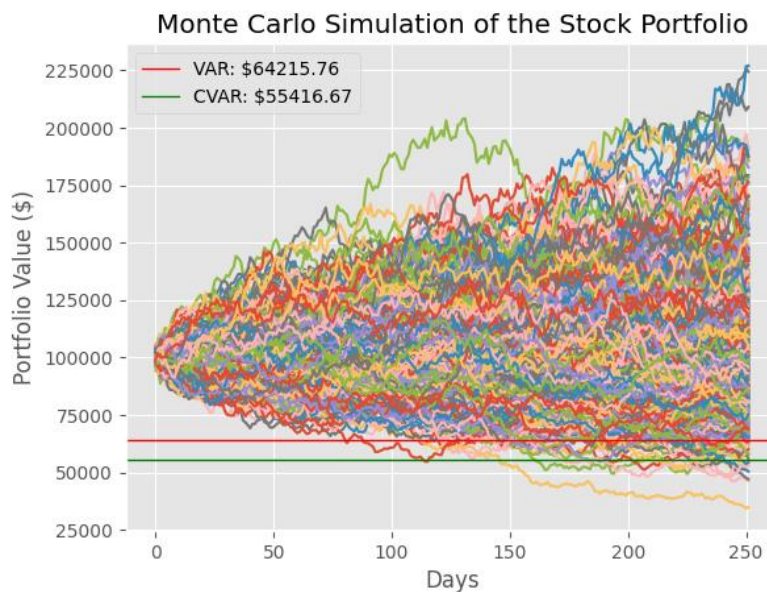


Рисунок 3. Значення VaR та CVaR для методу Монте Карло alpha = 5%

Результати обчислень VaR та CVaR різними методами для нашого портфелю наведені у таблиці нижче (Табл. 2).

Таблиця 2. Значення VaR та CVaR для різних рівнів довіри та різних методів обчислення

Довірчий інтервал	Метод оцінки ризику	Історичний	Параметричний N(0,1)	Параметричний GARCH(1,1)	Монте Карло
95%	VaR	-4,1%	-5,12%	-5,01%	-4,83%
	CVaR	-6,85%	-6,32%	-6,26%	-5,56%
99%	VaR	-5,55%	-5,63%	-6,32%	-5,06%
	CVaR	-9,4%	-6,42%	-7,15	-5,97%

Результати верифікації моделей для оцінювання ризику за допомогою процедури бек-тестування подано у Таблиці 3.

Таблиця 3. Результати процедури бек-тестування для оцінювання VaR

Довірчий інтервал	Метод оцінки ризику	Історичний	Параметричний N(0,1)	Параметричний GARCH(1,1)	Монте Карло
95%	VaR	94,9%	96,4%	96,54%	95,3%
	CVaR	97,5%	97,3%	97,1%	96,7%
99%	VaR	98,4%	98,45%	98,67%	98%
	CVaR	99,1%	98,64%	99,21%	98,56%

Загалом, можна зробити висновок, що усі моделі є у прийнятній мірі адекватними. Історичний метод дає гіршу оцінку ризику, що цілком закономірно, враховуючи, що цей метод базується тільки на історичних даних, таким чином історичний метод може не зреагувати на раптові зміни на ринку. Найкращими виявились параметричний метод на основі GARCH(1,1) та метод Монте Карло. Вони не завищують ризику і не занижують ризику, і, як показує бектестування, є гнучкими до змін на ринку.

Методи оцінки VaR з довірчим інтервалом 99% можуть дещо недооцінювати ризик при настанні екстремальних ситуацій. Саме в таких випадках треба звертати увагу на значення CVaR, яке є більш прийнятним і забезпечує точнішу оцінку ризику при 99% довірчому інтервалі. Таким чином, ми надаємо інвестору більш адекватну картину щодо його інвестицій, щоб він міг, по-перше, прийняти рішення щодо інвестування, або навпаки, про виведення інвестиції вчасно з урахуванням всіх ризиків, а по друге, мати достатній капітал для покриття збитків.

Враховуючи складність в обчисленні ризиків за методом Монте Карло, метод оцінки ризиків на основі прогнозування волатильності GARCH(1,1) моделлю можна вважати хорошою альтернативою. На основі оцінки VaR та CVaR даним методом з різними довірчими інтервалами, можна приймати рішення щодо інвестувань та стратегію, яка вам найбільше підходить на ринку.

4. ВИСНОВКИ

В даній роботі було розглянуто фінансові ризики, які є важливою складовою на фінансовому ринку, та дозволяють комплексно оцінити можливі майбутні втрати. Складання оптимального портфеля цінних паперів є важливою практичною задачею на фондовому ринку. Велике значення мають наукові дослідження в галузі математичного моделювання процесів оцінювання фінансових ризиків та управління ними. Тож використання ймовірнісних функціоналів VaR та CVaR і розрахунок їх за допомогою математичних моделей буде дуже корисним для даної практичної задачі.

У ході дослідження було реалізовано методології VaR та CVaR оцінки ринкового ризику на прикладі інвестиційного портфелю, сформованого з акцій, представлених на біржі NASDAQ. Було виконано порівняння отриманих результатів для низки методів, а саме історичного, параметричного, методу Монте-Карло та методу на основі прогнозування волатильності за допомогою моделювання гетероскадестичних процесів, зокрема GARCH моделлю. Беручи за основу GARCH-модель волатильності для опису змінної в часі дисперсії була досягнута висока точність прогнозів, придатна для прийняття рішень під час виконання фінансових операцій. Також була проведена процедура бек-тестування для верифікації використаних моделей оцінки ризику, що дозволяє зрозуміти наскільки точно модель VaR та CVaR відображає реальні умови ринку. Це дозволило комплексно оцінити ризики, а також порівняти точність різних методів оцінки VaR та CVaR, що дозволить надалі інвесторам приймати більш обґрунтовані рішення під час гри на фінансовому ринку.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Кузнєцова Н.В., Бідюк П.І. Теорія і практика аналізу фінансових ризиків: системний підхід, монографія. Київ 2020, 400 с.
2. Башкіров О.В. Порівняльний аналіз VAR-методів оцінки ризику активів банку О.В. Башкіров, Проблеми і перспективи розвитку банківської системи України : зб. наук. праць ДВНЗ «УАБС НБУ». Вип. 14. С. 302–309.
3. Moraux F. How valuable is your VaR? Large sample confidence intervals for normal VaR. F. Moraux. Journal of risk management in financial institutions. 2011. № 4.2. P. 189–200. – [Електронний ресурс]. – Режим доступу : <https://perso.univ-rennes1.fr/franck.morau/research/JRMFI.pdf>
4. Bollerslev T., Chow R., Kroner K. ARCH modeling in finance: a review of the theory and empirical evidence. Journal of Econometrics. 1992. Vol. 52. P. 5–59.
5. Bollerslev T. General autoregressive conditional heteroscedasticity. Journal of Econometrics. 1986. Vol. 31. P. 518–537.
6. Kilic, R. Conditional Volatility and Distribution of Exchange Rates: GARCH and FIGARCH Models with NIG Distribution. Studies in Nonlinear and Econometrics. Vol. 11. P. 1-31
7. Longerstae J. Risk Metrics TM – Technical Document . J. Longerstae, M. Spencer. Morgan Guaranty Trust Company of New York: New York, 1996. [Електронний ресурс]. – Режим доступу : <http://yats.free.fr/papers/td4e.pdf>

ПІДХІД ЩОДО ВИЗНАЧЕННЯ ФАКТОРІВ ВПЛИВУ НА РИНОК ЛОГІСТИКИ США З ВИКОРИСТАННЯМ LLM

Балькін Я.Ю.¹, Савастьянов В.В.

Національний технічний університет України «Київський політехнічний інститут ім. Ігоря Сікорського»

¹ yarik.balkin@gmail.com

Ринок логістики одним з найвразливіших ринків в економіці США, тому для ефективної оцінки та якісного прогнозування потрібно знаходити фактори впливу, та працювати з цими факторами. Метою роботи є розробка підходу задля ефективного визначення факторів впливу з застосуванням потужностей штучного інтелекту. Результатом дослідження є основні фактори впливу інших галузей на галузь логістики. У роботі було використано теоретичні та емпіричні методи дослідження, а також Large Language Model.

Ключові слова: логістика, вантажні перевезення, LLM, TSI, текстовий аналіз, фактори впливу.

1. ВСТУП

Сектор вантажних перевезень в США займає значне місце в формуванні економіки. На перевезення вантажів витрачається сотні мільярдів доларів на рік, а вплив інших сфер на цей сектор непомірно великий. Через це вантажні перевезення і викликають непомірний інтерес для аналітиків з усього світу. Ціноутворення на логістичні послуги має дуже складну систему через вплив сотень факторів, і дослідження цих факторів є необхідним задля моделювання ризиків та прогнозування.

На сьогоднішній день були проведені дослідження щодо ціноутворення в окремих галузях вантажних перевезень, а також дослідження які вивчали фактори впливу між вузькоспрямованими вантажними перевезеннями та іншими галузями. Ми ж спробуємо використовуючи більш новітні методи аналізу знайти та оцінити фактори більш широкого спектру впливу.

2. МЕТОДИ ПОШУШУ ФАКТОРІВ ВПЛИВУ

Задля більш точного та кваліфіковано пошуку факторів впливу було вирішено пропрацювати методи описані в роботах інших дослідників [1–5]. Так наприклад в роботах [4, 5] аналітика вантажних перевезень була проведена завдяки створенню та пропрацюванню часових рядів різними методами. Також були взяті роботи Українських дослідників [1–3] для більш точного розуміння сфери вантажних перевезень.

Стаття [1] досліджує темпи розвитку логістики в Україні. Акцент цієї роботи припадає на оцінку географічне положення України і розглядання динаміки росту та падіння об'єму вантажних перевезення за 2010–2018 роки. Також в роботі було розглянуто сфери які мають вплив на логістику та описана сила впливу цих сфер.

В наступній статті [2] були розглянуті тенденції розвитку Українського ринку логістики. Задля спостереження динаміки було введено індекс ефективності логістики. Були розглянуті основні країни експортери та імпортери, була проведена оцінка сили впливу експорту та імпорту на логістичну сферу. Також було розглянуто зв'язок ВВП країни та логістичної сфери.

У результаті цього дослідження було виділено фактори зростання попиту на логістичні послуги, проведено сегментацію ринку логістики в Україні та класифікацію цих послуг.

Стаття [3] розглядала логістику як галузь прикладних наук. В ній було проведено паралель між «минулим» та «теперішнім» логістики. Розглядалась еволюція ринку логістики, які питання були актуальні на той час, які фактори впливу були у 1960х роках та їх еволюціонування у сучасні проблеми цієї сфери. Тенденції та темпи розвитку того часу і їх розвиток з часом. Також було розглянуто нові ідеї для визначення сил факторів впливу на основі минулих досліджень та спостережень ринку перевезень.

Статті [4, 5] використовують більш новітні методи прогнозування та дослідження ринку вантажних та пасажирських перевезень. В них запроваджені моделі ARIMA, ARMA, SARIMA та інші для прогнозування попиту та ціноутворення логістичних послуг в США. А для аналізу були взяті індекси TSI для вантажних та пасажирських перевезень, який характеризує об'єми цих перевезень. Також у статті [4] результатами досліджень були модельні прогнози ціни на транспортні перевезення вантажівками Ці дані були отримані на основі аналізу TSI та сьогodenних цін на транспортні перевезення вантажними машинами (рис. 1). Але ціноутворення вантажних перевезення сильно вразливе до попиту, тому треба враховувати що пристосування перевізників до зміни цін займає час до бти місяців.

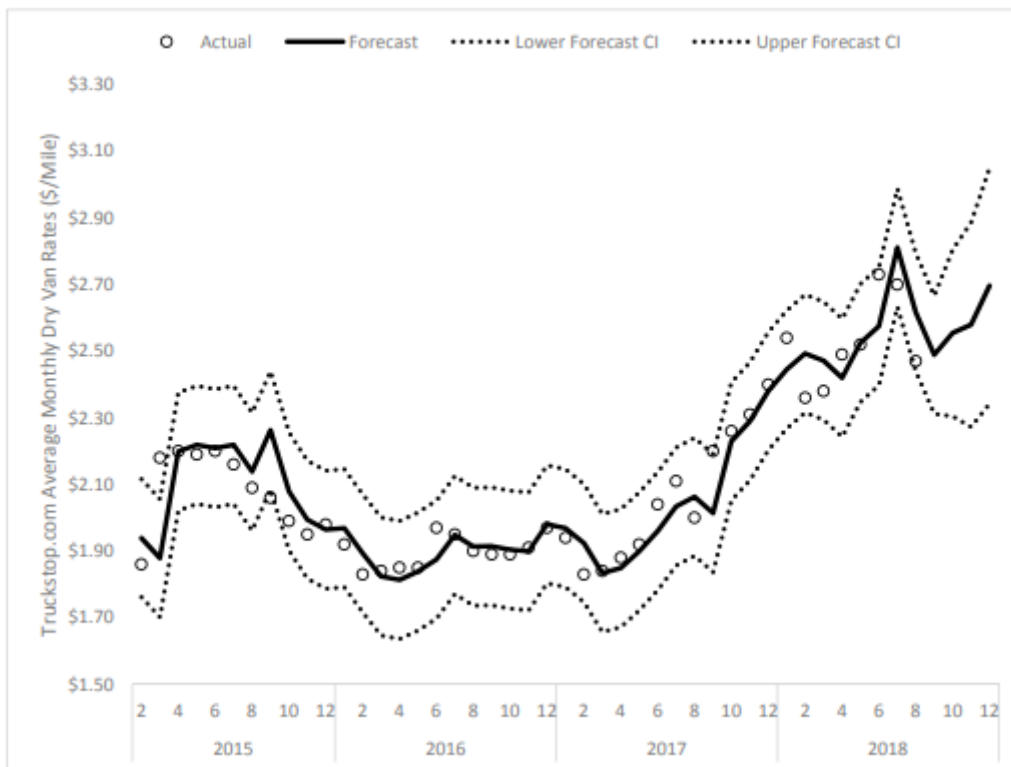


Рисунок 1. Модельні прогнози на вантажні перевезення

На основі аналізу робіт інших авторів було вирішено використовувати більш новітню методіку аналітики, а саме LLM. Завдяки LLM ми можемо урахувати більше факторів впливу, а також сезонність та вразливість до інших нестандартних факторів впливу. LLM дозволить нам обробити великі об'єми даних, а також допоможе їх правильно структурувати та класифікувати. Ще однією з причин щодо вибору LLM було врахування всіх недоліків всіх попередніх методів, та можливість їх модифікування. Так як LLM дуже гарно працює з

текстами, було вирішено шукати фактори впливу в новинних джерелах, і так як LLM дозволяє гарно сортувати та відсіювати непотрібне, ми можемо дозволити собі працювати з несумісними, на перший погляд, текстовими даними. Також для пошуку факторів впливу будемо розглядати числовий регресор TSI для вантажних перевезень. Цей індекс допоможе нам якісніше оцінити фактори впливу та кореляцію між галузями. Також цей індекс являє собою готові проаналізовані дані і завдяки ньому ми можемо бачити зріст та падіння логістичної сфери в різних проміжках часу (Рис. 2).



Рисунок 2. Графік змін TSI

3. ВХІДНІ ДАНІ ТА ОЧІКУВАНІ РЕЗУЛЬТАТИ

Для дослідження факторів впливу було сформовано базу даних текстової інформації, яка являє заголовки новин (Табл. 2), та часовий ряд числової інформації, яка являє собою TSI – Freight індекс (Табл. 1). Інформацію було взято за останні 12 років задля підвищення точності. Для пошуку залежностей було проведено накладання подій, на наш графік змін TSI(рис). Це дозволить нам визначати час, коли відбувались значущі події для сфери логістики і тим самим визначити ці події.

Таблиця 1. TSI – Freight індекс за 2011 рік

Date	Transportation Services Index - Freight
01/01/2011 12:00:00 AM	110,3
02/01/2011 12:00:00 AM	109,7
03/01/2011 12:00:00 AM	111,1
04/01/2011 12:00:00 AM	110,6
05/01/2011 12:00:00 AM	108,7
06/01/2011 12:00:00 AM	110,6
07/01/2011 12:00:00 AM	110,4
08/01/2011 12:00:00 AM	110,9
09/01/2011 12:00:00 AM	111,4
10/01/2011 12:00:00 AM	111,7
11/01/2011 12:00:00 AM	112
12/01/2011 12:00:00 AM	115,1

Таблиця 2. Приклад текстових даних

June 2, 2023	Why Americans Want Part-Time Jobs Again
June 1, 2023	IEA Head Wants Fossil Fuel Industry To Set Climate Targets
June 1, 2023	Airlines Are Weighing Passengers
June 1, 2023	How Grab Became a Super-App
May 31, 2023	Why Janet Yellen Doesn't Lose Sleep Over U.S. Borrowing
May 31, 2023	Twitter Now Worth One-Third of What Musk Paid for It
May 30, 2023	AI Is as Risky as Nuclear War, Top CEOs Say
May 27, 2023	Twitter Withdraws From EU Disinformation Code, Commissioner Says
May 26, 2023	Why Gas Prices Are Cheaper Right Now
May 25, 2023	Why Europe's New Climate Rules Matter to American Companies
May 25, 2023	Germany Endures First Recession Since COVID on Consumers
May 24, 2023	Yellen: Treasury 'Not Involved' in Planning With Investors for Default

Текстові дані потребують попередньої обробки, тому в першу чергу треба зробити відсів даних які зовсім не стосуються логістики, наприклад стаття з заголовком «Philippines' Duterte Subpoenaed Over Alleged Death Threat» ніяк не стосується логістики навіть опосередковано. В той час як стаття «China Invests \$5.4 Billion in Chipmaker» цілком може вплинути на коливання сфери логістики. Початковий дата сет має в собі більш ніж 200 тисяч рядків, коли після обробки цей список зменшується до 175 тис. (Рис. 3).

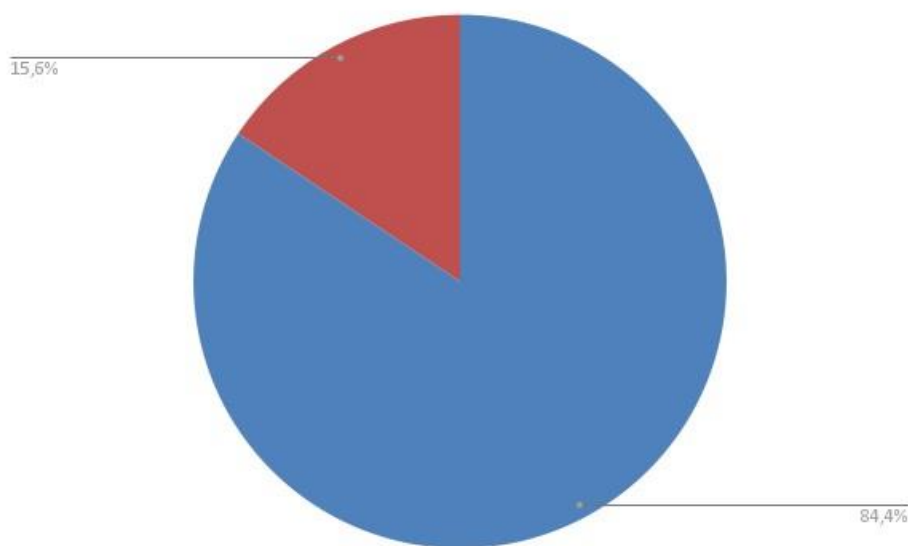


Рисунок 3. Відношення ефективних та неефективних текстових даних

Наступним етапом нашої роботи йде класифікація. Для початкової класифікації нам підійде модель STEEP(Social, Technological, Economical, Environmental and Political), для початкової класифікації. Для уточнення будемо користуватись підходом «few-shot prompting». На вхід будуть подаватись заголовки – після цього кожний заголовок буде оброблюватись моделлю GPT-3.5, використовуючи заздалегідь прописані правила класифікації(на цьому етапі застосовується підхід «few-shot prompting», який окрім правил використовує ще й підказки) – на виході отримуємо класифікований заголовок.

Основна задача цієї роботи – це знайти фактори впливу на сферу логістики, та визначити їх силу. Щоб цього досягти кожна класифікована новина буде подаватися до моделі GPT-3.5, оброблюватися завдяки правилам, прописаним заздалегідь, та буде отримувати свою силу впливу. Після всієї обробки планується обрахувати які сектори мають більший вплив на сферу логістики, а також які саме події викликали різкі збільшення/зменшення об'єму вантажоперевезень.

4. ВИСНОВКИ

Об'єми вантажних перевезень збільшуються із року в рік, тому для працівників цієї галузі дуже важливим є розуміння ринку, його оцінка та аналітика економічної ситуації. Також вважаючи те що ринок логістики є одним найбільшим у США, його кардинальні зміни приведуть до змін майже у всіх економічних сферах.

Завдяки новітнім технологіям, а саме LLM ми можемо розглядати та оцінювати такі великі структури. Займатися прогнозуванням та плануванням. А завдяки тому що LLM має велику бібліотеку даних та вміє в ній орієнтуватись, це може стати заміною експертної думки одразу у багатьох галузях. Завдяки цьому LLM може пов'язувати факти, опосередковано, або й зовсім не зв'язаних з логістикою. Також через гарне розуміння LLM текстової інформації наш підхід буде не тільки інноваційним, але й досить точним порівняно з іншими методами аналізу(використання часових рядів, тощо)

У ході дослідження планується розробити універсальний продукт виявляючий фактори впливу на логістичний ринок. Продукт який буде підлаштований до нештатних ситуацій, та буде використовувати не тільки статистичну модель, а також LLM.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Статистичний аналіз структури та тенденцій розвитку логістичного ринку України / Р. В. Ціщик, Н. В. Котис // Проблеми системного підходу в економіці. - 2018. - Вип. 3(1). - С. 54-59. - [Електронний ресурс] - Режим доступу: http://nbuv.gov.ua/UJRN/PSPE_print_2018_3%281%29__11
2. АНАЛІЗ РИНКУ ЛОГІСТИЧНИХ ПОСЛУГ В УКРАЇНІ ЗА 2010-2018 РОКИ / Наталія Ю. Кирлик - 2019. – С. 83-95. - [Електронний ресурс]. – Режим доступу <https://dspace.uzhnu.edu.ua/jspui/bitstream/lib/31105/1/83--95.pdf>
3. Logistics research: A 50 years' march of ideas/ Peter Klaus – 2008. – С. 53-65. - [Електронний ресурс]. – Режим доступу https://www.researchgate.net/publication/220232983_Logistics_research_A_50_years%27_march_of_ideas
4. ARIMA Time Series Models for Full Truckload Transportation Prices/Jason W. Miller – 2019. – [Електронний ресурс]. – Режим доступу <https://www.mdpi.com/2571-9394/1/1/9>
5. Forecasting Daily and Weekly Passenger Demand for Urban Rail Transit Stations Based on a Time Series Model Approach/ Dung David Chuwang, Weiya Chen – 2022. - [Електронний ресурс]. – Режим доступу <https://www.mdpi.com/2571-9394/4/4/49>
6. ChatGPT takes the stand in its defense/DAN LUCARINI – 2023. - [Електронний ресурс]. – Режим доступу <https://www.deep-analysis.net/chatgpt-takes-the-stand-in-its-defense/>
7. What is a STEEP Analysis? - [Електронний ресурс]. – Режим доступу <https://www.utsdesignindex.com/researchmethod/steep-analysis/>

ПРОГНОЗУВАННЯ ЦІНИ НА ЗОЛОТО МЕТОДАМИ МАШИННОГО НАВЧАННЯ

Білоус К.С.¹, Кузнєцова Н.В.

Національний технічний університет України «Київський політехнічний інститут
ім. Ігоря Сікорського»

¹ katyabelousal@gmail.com

Золото - є ключовим дорогоцінним металом для економіки світу, хоча його ціна постійно коливається. Саме тому своєчасна актуальна інформація про його ціну є вкрай важливою. Методами машинного навчання можна виконати прогнозування ціни на золото з високою точністю, що дозволяє забезпечити краще управління ризиками та приймати обґрунтовані рішення щодо продажу чи придбання дорогоцінного металу. Метою роботи є розробка системи для прогнозування цін на золото. Результатом дослідження є розроблена система прогнозування цін на золото.

Ключові слова: золото, машинне навчання, прогнозування, дорогоцінні метали, прогнозування цін.

1. ВСТУП

Золото – це дорогоцінний метал, що відомий своєю стійкістю до корозії, високою пластичністю та хорошою електропровідністю. Саме через його фізичні властивості його високо цінують і використовують у ювелірній сфері, при виробництві електроніки, медичного обладнання, а також золото має статус "безризикового активу" і використовують як спосіб зберігання грошей [1].

Золото як спосіб зберігання вартості можуть використовувати як приватні особи, так і країни. Такими прикладами є золотовалютні резерви – запас золота та іноземної валюти, що утримується центральним банком країни. Золотовалютний резерв впливає на загальну економічну динаміку країни, а його правильне управління є важливим елементом для забезпечення стабільності та розвитку [2].

Зважаючи на це золото є важливим елементом економіки країн і бізнесу. Відповідно, є важливим вміти прогнозувати ціни на золото.

2. ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

Мета даного дослідження – розробка системи прогнозування ціни на золото методами машинного навчання. В роботі розглядаються різні існуючі методи роботи з прогнозуванням цін дорогоцінних металів. Виконується порівняння конкретних методів для роботи з даною задачею на основі методів машинного навчання. Об'єкт дослідження – ціни на золото. Предмет дослідження – методи та моделі машинного навчання для прогнозування цін на золото.

3. ОГЛЯД КЛАСИЧНИХ МЕТОДІВ ПРОГНОЗУВАННЯ ЦІН НА ЗОЛОТО

Існують різні методики прогнозування цін на дорогоцінні метали, такі як золото, срібло, платина та інші. Основними напрямками в даній задачі є наступні методи:

- Аналіз фундаментальних факторів.

Цей підхід враховує фактори, які можуть впливати на попит і пропозицію на ринку дорогоцінних металів. Це може включати політичні події, економічний стан країн, геополітичні конфлікти, інфляцію, торгові відносини між країнами, зміни відсоткових ставок та інші фактори. Аналізуючи ці дані, можна робити припущення щодо майбутньої динаміки цін [3].

Зміни у політиці, такі як введення нових законодавчих актів щодо фінансової політики, можуть вплинути на ціни дорогоцінних металів. Економічний стан країн, зокрема країн-виробників і споживачів, також має велике значення. Нестабільність у виробничих регіонах, можливі воєнні конфлікти або санкції можуть призвести до змін в постачанні металів і вплинути на їхні ціни. Зміни у торгових угодах між країнами можуть мати великий вплив на експорт та імпорт дорогоцінних металів. Зміни в рівні інфляції або відсоткових ставках також можуть впливати на ціни металів.

- Технічний аналіз.

Цей метод полягає у вивченні та аналізі графіків цін, обсягів торгівлі, патернів поведінки цін та інших технічних показників. Технічний аналіз може допомогти виявити тренди, підтримку та опір на ринку, що може бути використано для прогнозування майбутніх цін на метали.

Аналіз графіків допомагає виявити тренди цін, патерни поведінки цін, рівні підтримки та опору на ринку. Використання індикаторів, таких як ковзне середнє, стохастичний осцилятор, MACD тощо, допомагає прогнозувати можливі рухи цін на метали.

- Модель ARIMA (Autoregressive Integrated Moving Average).

Це статистична модель, яка використовується для аналізу часових рядів. ARIMA дозволяє прогнозувати майбутні значення на основі попередніх даних. Застосовується до історичних даних цін на дорогоцінні метали для прогнозування їх майбутнього руху.

ARIMA використовує попередні значення часових рядів для прогнозування майбутніх значень. Він може враховувати тренди, сезонність та інші закономірності в даних для прогнозування цін.

- Моделювання з використанням машинного навчання.

Методи машинного навчання, такі як нейронні мережі, випадкові ліси, градієнтний бустінг і інші, можуть бути використані для аналізу великої кількості даних і прогнозування цінових рухів [4].

Ці методи машинного навчання можуть аналізувати великі обсяги даних та виявляти складні зв'язки між різними факторами для прогнозування цін на метали.

- Аналіз геополітичних та макроекономічних подій.

Події на світовій арені, такі як конфлікти, зміни режимів, торгові угоди, а також макроекономічні показники, такі як зміни у ВВП, безробіття, можуть мати великий вплив на ціни дорогоцінних металів.

Події на світовій арені, такі як військові конфлікти або зміни у владних режимах, можуть призвести до нестабільності на ринку дорогоцінних металів. Зміни у макроекономічних показниках, таких як ВВП, безробіття, інфляція, можуть впливати на попит і пропозицію на ринку металів [5].

Ці методики можуть використовуватися окремо або комбінуватися для отримання більш точного прогнозу цін на дорогоцінні метали. Жоден метод не може гарантувати точний прогноз цін на дорогоцінні метали. Зазвичай, найефективніше використовувати комбінацію кількох підходів для отримання більш точного прогнозу.

3. ОГЛЯД МЕТОДІВ МАШИННОГО НАВЧАННЯ ДЛЯ ПРОГНОЗУВАННЯ ЦІН НА ЗОЛОТО

Методи машинного навчання можуть бути потужним інструментом для прогнозування цін на золото, використовуючи різні моделі та алгоритми. Вони можуть описувати складну структуру взаємозв'язків між факторами, що описують закон формування цін на золото. Ось деякі з них:

- Нейронні мережі.

Це можуть бути різні типи нейронних мереж, такі як рекурентні нейронні мережі (RNN), згорткові нейронні мережі (CNN) або глибокі нейронні мережі (DNN) [6]. Вони можуть аналізувати складність та неоднорідність даних для прогнозування цін на золото.

- Випадкові ліси.

Випадкові ліси – це тип алгоритму, що базується на деревах рішень. Вони використовуються для аналізу великих обсягів даних та можуть добре прогнозувати цінові рухи на основі різних факторів [7].

- Градієнтний бустінг.

Градієнтний бустінг – це метод, що поєднує декілька слабких моделей для створення потужної ансамблевої моделі. Він може виявляти складні зв'язки між різними факторами та прогнозувати ціни на золото з високою точністю.

- Метод опорних векторів (SVM).

Цей метод може бути застосований до прогнозування цін на золото, шукаючи оптимальну гіперплощину для розділення даних на категорії та прогнозування майбутніх значень.

- Лінійна регресія.

Лінійна регресія – це один з найпростіших методів, який моделює залежність між залежною та незалежними змінними [8]. Він може бути використаний для простого прогнозування цін на золото на основі історичних даних.

Ці методи машинного навчання використовують різні підходи та алгоритми для аналізу великих обсягів даних про ціни на золото, враховуючи різноманітні фактори, які можуть впливати на ціни. Точність прогнозів може залежати від якості даних, використаної моделі та правильно підібраних параметрів.

4. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Для даного дослідження було використано дані про ціни на золото за період з 18 листопада 2011 року до 1 січня 2019 року. Дані було зібрано з різних джерел. В датасеті містяться 1718 рядків даних та 80 колонок характеристик. До даних входить інформація про ціни на нафту та інші дорогоцінні метали, ставки облігацій США, обмінні курси євро і долара США та індекси Standart and Poor's і Доу-Джонса.

Спершу дані оброблялись, перевірялись і аналізувались з допомогою статистичних показників і методів. Було розраховано додаткові технічні індикатори для покращення прогнозування цін на золото: MACD, RSI, SMA, UpperBand, LowerBand, DIFF, Open-Close, High-Low. Також дані було нормалізовано для роботи з методами машинного навчання.

В роботі використовувались наступні методи машинного навчання: метод Бенчмарк, SVR, Random Forest, Lasso, Ridge, Bayesian Ridge, Gradient Boosting, Stochastic Gradient Descent, LGBM Regressor with Repeated stratified K fold, XGBRegressor, GRU NN, LSTM NN. Впродовж дослідження проводився відбір показників для покращення методів машинного навчання. На наступних Рисунках 1–2 представлено графіки прогнозів на рік вказаними вище методами.

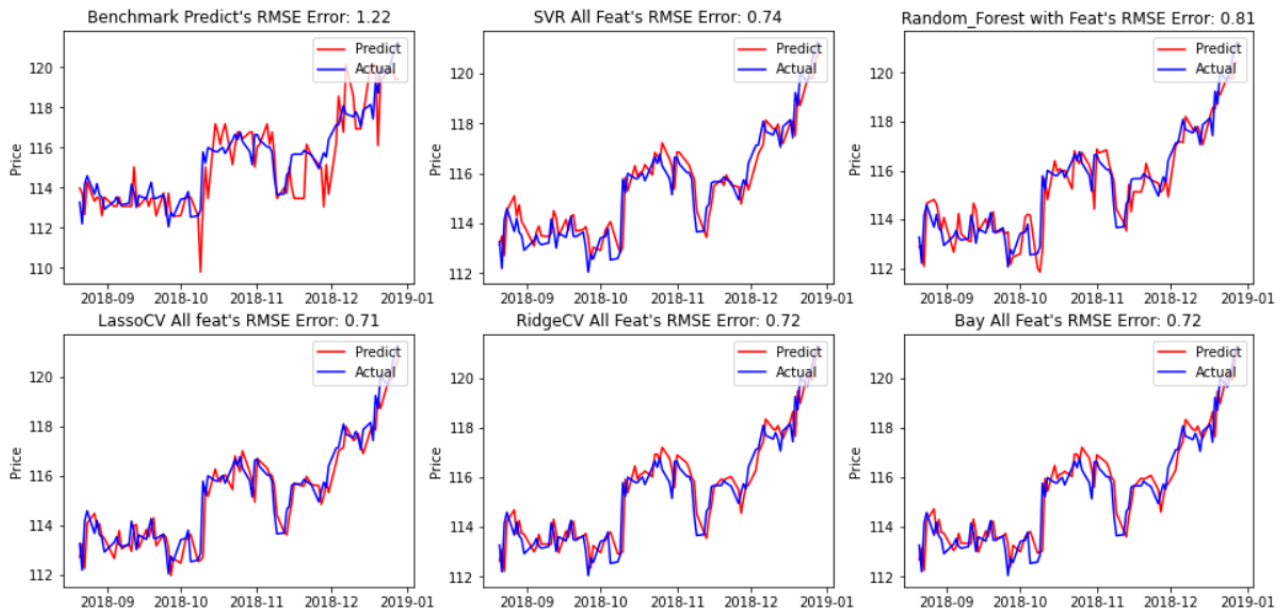


Рисунок 1. Порівняльні графіки прогнозування моделей в порівнянні з реальними даними

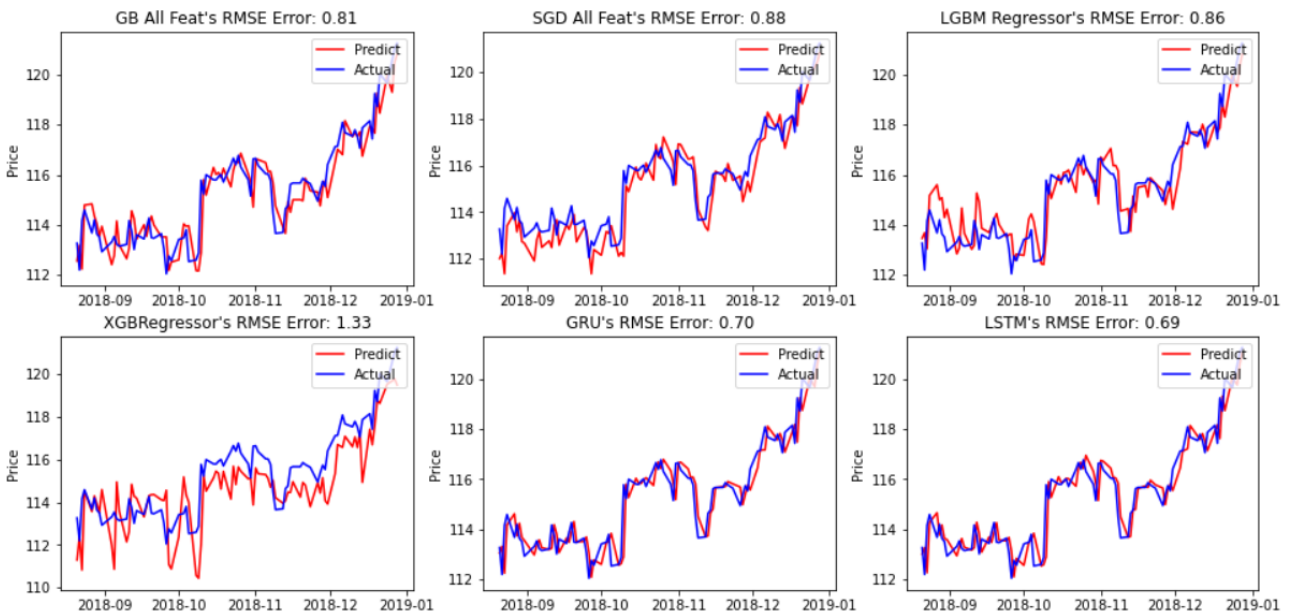


Рисунок 2. Порівняльні графіки прогнозування моделей в порівнянні з реальними даними

На Рисунку 3 представлено графік RMSE для оцінки найкращої моделі для методів навчених на всіх показниках і з відібраними параметрами.

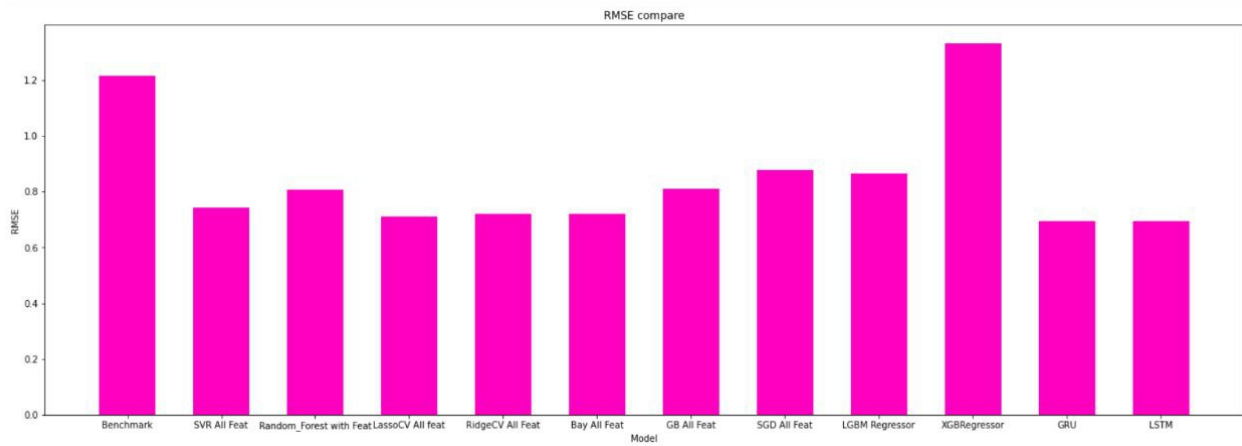


Рисунок 3. Гістограма RMSE для всіх методів

Відповідно до Рисунку 3, найкраще показали себе моделі навчені на відібраних параметрах. Для уточнення роботи моделей оцінимо усереднені похибки кожного методу прогнавши їх 50 разів. В наступній Таблиці 1 представлено результати дослідження.

Таблиця 1. Порівняльна таблиця результатів кращих моделей

Модель	RMSE	R2
Benchmark Model	1,2166	0,6590
LSVR	0,8136	0,8474
Random Forest	0,8078	0,8496
Lasso	0,7117	0,8833
Ridge	0,7186	0,8810
Bayesian Ridge	0,7195	0,8807
Gradient Boosting	0,8094	0,8490
Stochastic Gradient Descent	0,8763	0,8231
LGBM Regressor with Repeated stratified K fold	0,8643	0,8279
XGBRegressor	1,3319	0,5913
LSTM NN	0,6942	0,8889
GRU NN	0,6955	0,8885

Отже, відповідно до результатів представлених в Таблиці 1 найкращими виявились моделі LSTM NN. Але варто зазначити, що модель GRU NN є співставною до LSTM NN, і різниця RMSE між ними лежить у межах статистичної похибки.

5. ВИСНОВКИ

В даному дослідженні було розглянуто математичні методи і моделі машинного навчання для прогнозування цін на золото. В процесі дослідження було розглянуто багато напрямків роботи над поставленою задачею і відібрано основний, на основі якого і проводились експерименти. Проведено аналіз даних і обробка даних перед використанням методів машинного навчання для максимізації точності методів.

В результаті дослідження було обрано модель, яка найкраще описує закон ціноутворення на золото. Модель створена на базі методу LSTM NN, яка дала найкращі результати.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Диба М.І., Бахтарі Е.А. Еволюція золота: історія і сучасність : Вісник НБУ Жовтень 2008. С. 20–28.
2. Морозов А.С. Еволюція економічної ролі золота. Економіка та держава. – №12. – 2010. С. 53 – 57.
3. Johnston J., DiNardo J. Econometric methods. New York: McGraw-Hill, Inc., 1997. 530 p.
4. Бідюк П.І. Методи прогнозування. Луганськ: Альма Матер, 2008. С. 605.
5. Ксенжук, О. С. Тенденції функціонування світового ринку дорогоцінних металів в умовах глобальної фінансової нестабільності. Економічний аналіз: зб. наук. праць. Тернопільський національний економічний університет. 2017. Т. 27. № 4. – С. 289-298
6. П. Бідюк, Є. Гуць, В. Гавриленко, Н. Рудоман. Прогнозування цін акцій з використанням рекурентної нейронної мережі lstm / Системи управління, навігації та зв'язку. Збірник наукових праць. – Полтава: ПНТУ, 2021. – Т. 3 (65). – С. 64-68. – doi:<https://doi.org/10.26906/SUNZ.2021.3.064>.
7. Leo Breiman. Random Forests. Berkeley: Statistics Department, University of California, 2001. 33 p.
8. Altman, N. and Krzywinski, M. (2015). Simple linear regression. Nature Methods, 2(11), 999–1000 ст. [Електронний ресурс]. – Режим доступу : <https://www.nature.com/articles/nmeth.3627>

ПОРІВНЯЛЬНИЙ АНАЛІЗ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ МЕТОДІВ РОЗВ'ЯЗАННЯ ЗАДАЧІ ПРО МАКСИМАЛЬНИЙ ПОТІК У МЕРЕЖАХ

Боднар М.С.¹, Статкевич В.М.²

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Навчально-науковий інститут прикладного системного аналізу, кафедра математичних методів системного аналізу, Київ, Україна

¹ poit25@ukr.net, ² mstatkevich@yahoo.com [0000-0001-5210-9890]

Розглянуто задачу пошуку максимального потоку в мережах. Для алгоритмів Форда-Фалкерсона, Едмондса-Карпа та просування передпотуку наведено асимптотичну складність, виконано порівняльний аналіз за часом залежно від розміру та характеристик мережі. У результаті, алгоритм Форда-Фалкерсона зазвичай є найповільнішим, а просування передпотуку – найшвидшим, проте для розрідженого графа з великою кількістю вершин алгоритм Едмондса-Карпа може бути швидшим. Розроблено програмне забезпечення, яке реалізує вказані алгоритми та графічно відображає процес розв'язку, проведено експерименти на різних множинах тестових мереж.

Ключові слова: максимальний потік, задача про максимальний потік, алгоритм Форда-Фалкерсона, алгоритм Едмондса-Карпа, алгоритм просування передпотуку.

1. ВСТУП

Необхідність оптимізації поточкових мереж стає все більш актуальною в різних сферах, включаючи транспортні системи, телекомунікації та комп'ютерні мережі [1]. Ефективне управління потоками ресурсів, інформації або товарів є життєво важливим для підвищення загальної продуктивності системи, мінімізації вузьких місць і забезпечення оптимального використання ресурсів [2].

Це дослідження має на меті зробити внесок в існуючу базу знань шляхом аналізу, порівняння та реалізації трьох відомих алгоритмів для вирішення задач про максимальний потік: алгоритмів Форда-Фалкерсона, Едмондса-Карпа та алгоритму "просування передпотуку" (англ. *Push-Relabel*).

Задача про максимальний потік (англ. *Maximum Flow Problem*) є однією з фундаментальних проблем в теорії оптимізації, яка має багато практичних застосувань у різних галузях, таких як проектування мереж, транспорт, логістика, планування та розподіл ресурсів. Наприклад, ця задача може використовуватися для моделювання потоків транспорту в дорожніх мережах, потоків даних в комунікаційних мережах, мережевій маршрутизації, потоків товарів у ланцюгах постачання, потоків крові в кровоносній системі, потоків електрики в електричних мережах та інших [3].

2. ЗАДАЧА ПРО МАКСИМАЛЬНИЙ ПОТІК

Задача про максимальний потік може бути сформульована так: дано орієнтований граф (також називається мережею) з джерелом та стоком, де кожне ребро має додатну пропускну здатність, яка визначає максимальну кількість потоку, що може проходити через нього, знайти

спосіб надсилання якомога більшого потоку від джерела до стоку з урахуванням обмежень на пропускну здатність. Потік на кожному ребрі повинен бути невід'ємним і задовольняти умови збереження потоку: загальний потік, що входить у вузол (крім джерела та стоку), повинен бути рівним загальному потоку, що виходить з цього вузла (див. рис. 1). Більш формально, **потік** (англ. *flow*) у G – це двовимірна дійснозначна функція на декартовому квадраті множини вершин $f: V \times V \rightarrow \mathbb{R}$, яка визначає кількість потоку, що проходить по кожному ребру. Така функція задовольняє наступним умовам:

- 1) **обмеження пропускну здатності** (*capacity constraint*):

$$\forall u, v \in V: f(u, v) \leq c(u, v),$$

- 2) **антисиметричність** (*skew symmetry*):

$$\forall u, v \in V: f(u, v) = -f(v, u),$$

- 3) **збереження потоку** (*flow conservation*):

$$\forall u \in V \setminus \{s, t\}: \sum_{v \in V} f(u, v) = 0.$$

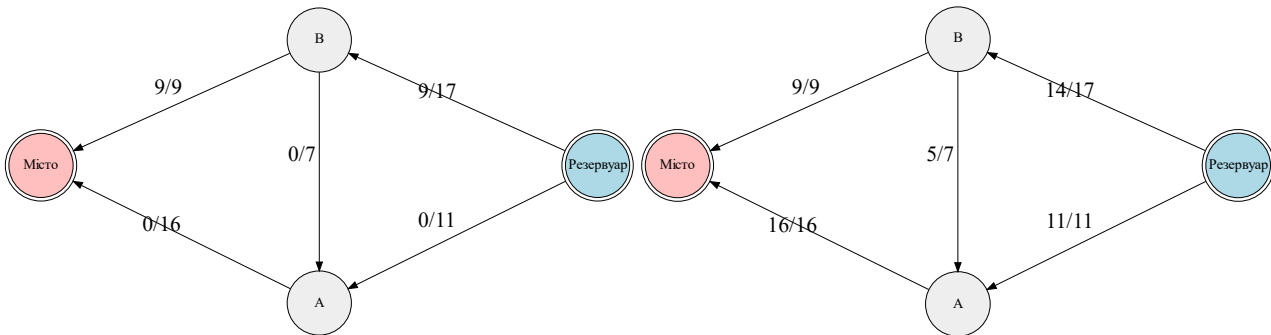


Рисунок 1. Зліва – можливий потік по мережі, справа – максимальний

Можемо збільшити потік, знайшовши збільшувачий шлях – це шлях від джерела до стоку, що має деяку невикористану пропускну здатність на своїх ребрах. Збільшувачий шлях також називають доповнюючим шляхом або шляхом, що збільшує (англ. *augmenting path*). Теорема про максимальний потік та мінімальний розріз стверджує, що максимальна величина потоку від s до t дорівнює мінімальній пропускну здатності розрізу від s до t у мережі.

Задача про максимальний потік була вперше сформульована у 1954 році Т. Е. Гаррісом та Ф. С. Россом як спрощена модель радянського залізничного транспортного потоку [4]. У 1956 році Лестер Рендольф Форд-молодший та Делберт Реймонд Фалкерсон створили перший відомий алгоритм – алгоритм Форда-Фалкерсона [5], який базується на пошуку збільшувачих шляхів.

З тих пір було розроблено різні покращені алгоритми для вирішення задачі максимального потоку, такі як алгоритм Едмондса-Карпа, алгоритм "просування передпотіку" (Push-Relabel), алгоритм Голдберга-Рао та інші [3]. Ці алгоритми мають різну складність виконання та характеристики продуктивності залежно від структури та розміру мережі. Ці алгоритми використовують різні техніки для пошуку збільшувачих шляхів, різні структури даних та методи для відстеження залишкової мережі.

Залишкова мережа G_f – це граф, який показує можливий додатковий потік у мережі G . Якщо є шлях із джерела в стік у залишковій мережі, то можна збільшити потік у вихідній мережі. Кожне ребро залишкової мережі має значення, зване залишковою пропускну здатністю, яка дорівнює різниці вихідної пропускну здатності ребра та поточного потоку по ньому. Залишкова пропускну здатність – це, по суті, поточна невикористана пропускну здатність ребра.

2.1. Алгоритм Форда-Фалкерсона

Це один із найкласичніших і найпростіших алгоритмів розв'язання задачі про максимальний потік. Основна ідея алгоритму полягає в наступному. На кожній ітерації алгоритму знаходимо збільшуючий шлях із джерела до стоку у залишковій мережі, знаходимо ребро з найменшим значенням залишкової пропускної здатності та за допомогою цього значення збільшуємо потік уздовж цього шляху. Потім шукаємо інший збільшуючий шлях, і так далі (ітеративно), доки в отриманій залишковій мережі не буде доступного збільшувального шляху. Алгоритм завершується, коли в залишковій мережі більше не існує збільшувальних шляхів. Складність за часом (або асимптотична складність) даного алгоритму залежить від способу вибору збільшувальних шляхів, і може бути як поліноміальною, так і експоненціальною щодо розміру вхідних даних.

Приклад роботи. Розглянемо наступний граф G з вершинами $V = \{s, a, b, c, d, t\}$, де s – джерело, t – стік. Ребра та їх пропускні здатності зображені на рис. 2.

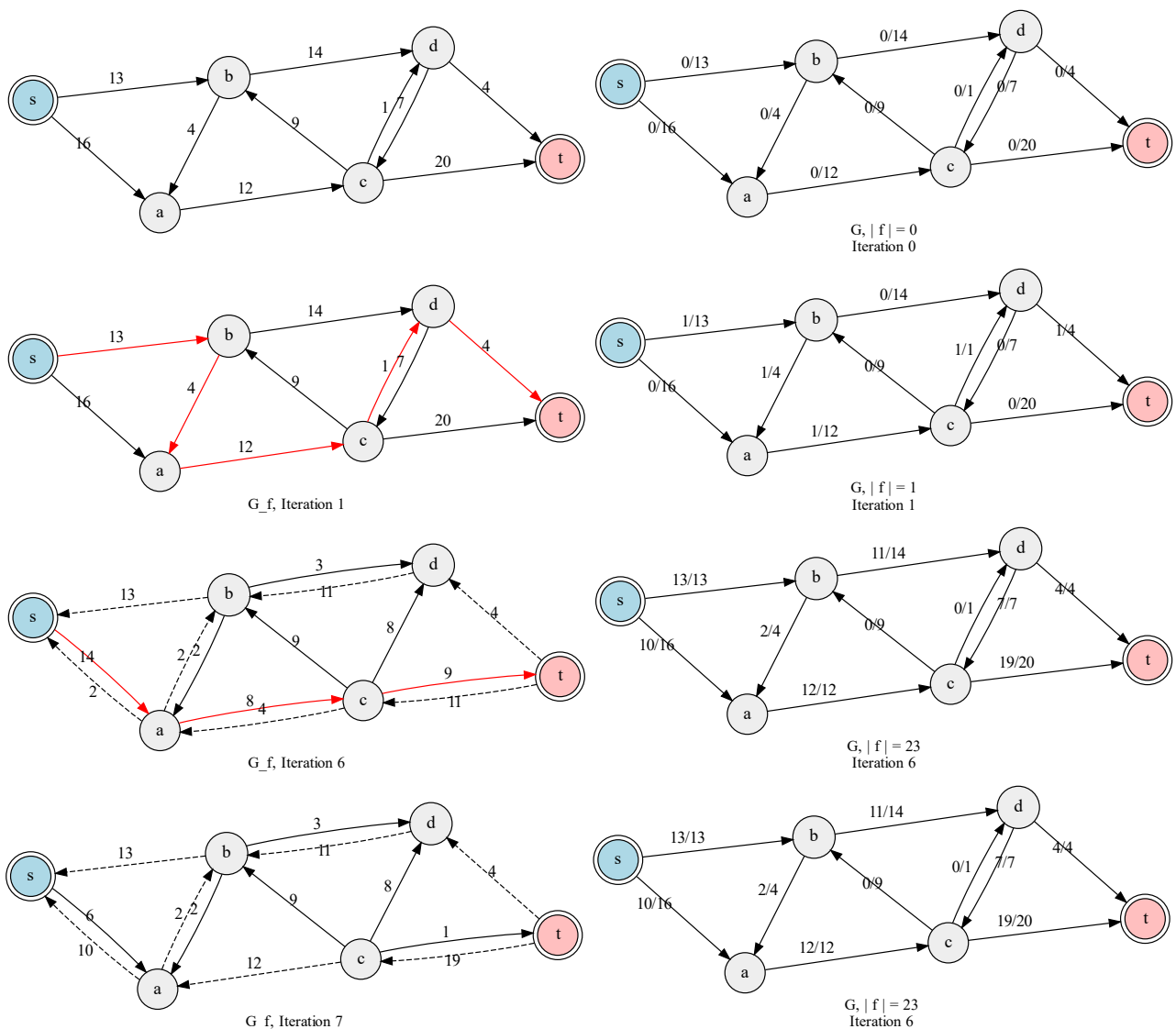


Рисунок 2. Мережа G , ітерації 0, 1, 6 та 7 (ітерації 2–5 випущено)

2.2. Алгоритм Едмондса-Карпа

Це версія алгоритма Форда-Фалкерсона, яку модифікували Джек Едмондс та Річард Меннінг Карп у своїй роботі 1972 року [6], вона гарантує поліноміальну складність за часом алгоритма. Відмінність полягає в тому, що для вибору збільшуючих шляхів використовується алгоритм пошуку в ширину (англ. *Breadth First Search* або скорочено *BFS*), який знаходить найкоротший за кількістю ребер збільшуючий шлях у залишковій мережі (вважаючи, що кожне ребро має одиничну довжину). Нагадаємо, що пошук у ширину – це один із найдавніших і найвідоміших алгоритмів обходу графа, який починається із заданої вершини та відвідує всі вершини, які можна досягти з неї, у порядку зростання відстані до неї. Складність алгоритму Едмондса-Карпа за часом становить $O(|V||E|^2)$, де $|V|$ – кількість вершин, а $|E|$ – кількість ребер у вихідному графі.

Приклад роботи. Розглянемо приклад роботи алгоритму Едмондса-Карпа в транспортній мережі, яка була розглянута на рис. 2. Наступні кроки аналогічні крокам, які наведені на рис. 2, лише за відмінністю, що треба шукати **найкоротший** збільшуючий шлях на кожній ітерації (див. рис. 3).

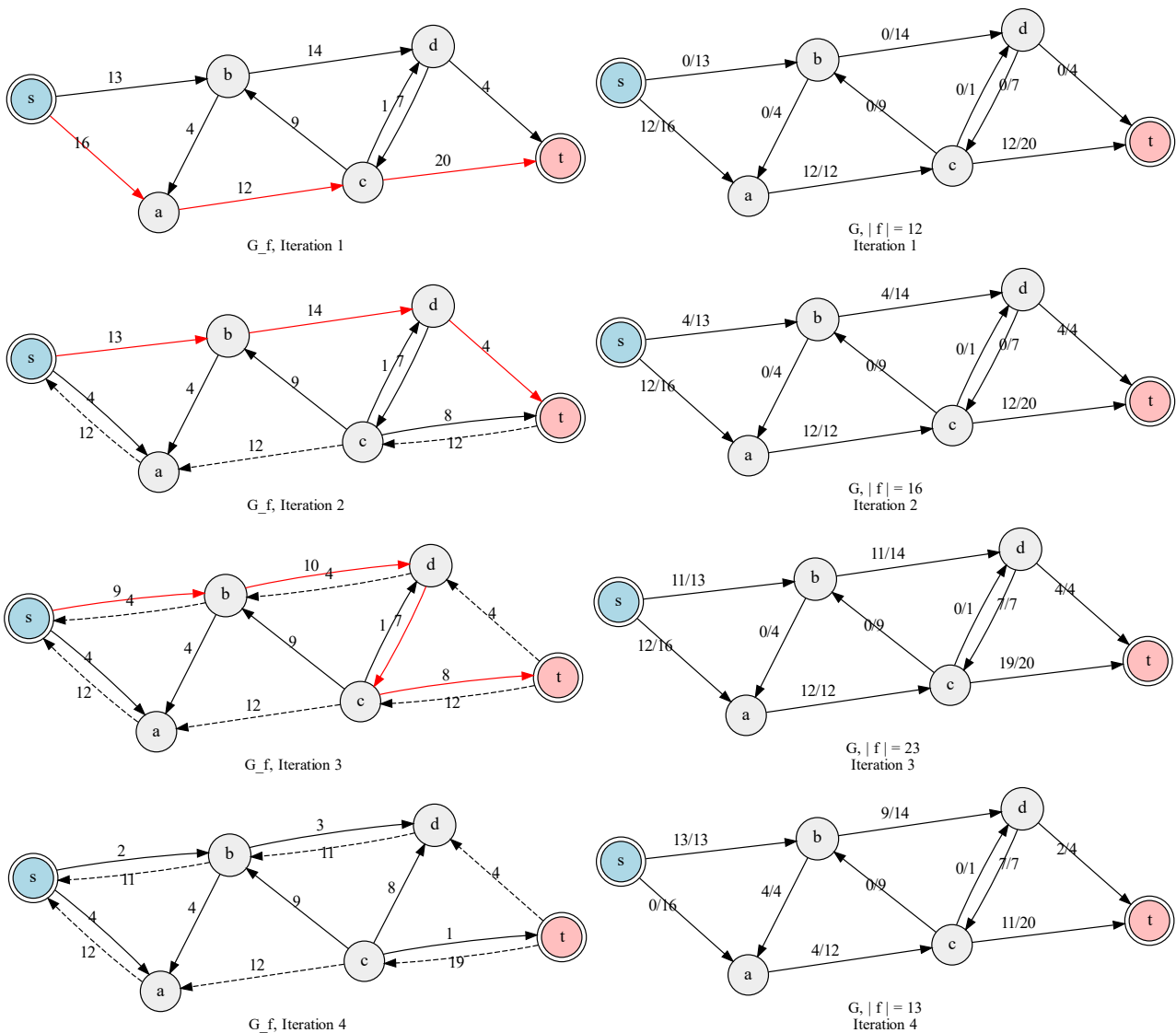


Рисунок 3. Мережа G, ітерації 0–4

Як можна побачити, алгоритмом Едмондса-Карпа максимальний потік знайдений швидше, ніж алгоритмом Форда-Фалкерсона, якому знадобилось 7 ітерацій. Тобто краща ефективність алгоритма Едмондса-Карпа у порівнянні з алгоритмом Фордом-Фалкерсоном помітна неозброєним оком. Це все за рахунок зміни вибору збільшуючого шляху.

2.3. Алгоритм просування передпоток

Алгоритм просування передпоток (або «алгоритм підняття передпоток», англ. *Push-Relabel*), розроблений Ендрю Голдбергом і Робертом Тар'яном у 1986–1988 роках [7], представляє собою важливий прорив у теорії потоків у графах. Автори ввели поняття передпоток і запропонували дві основні операції: просування і підняття, які дають змогу поступово перетворювати передпотік на максимальний потік. Вони також довели, що цей алгоритм має сильно поліноміальну складність $O(|V|^2|E|)$, що є кращою за складність алгоритму Форда-Фалкерсона та його модифікацій (Едмондс-Карп, Дініц). Вони використовували ідею міток висоти у кожній вершині для управління потоком. Це один з найефективніших алгоритмів розв'язання задачі про максимальний потік.

На відміну від класичного методу Форда-Фалкерсона, де аналізують усю залишкову мережу для пошуку шляху, що збільшує потік, алгоритми просування передпоток діють більш локально. Тобто опрацювання вершин відбувається по одній, розглядаючи тільки їх сусідів у залишковій мережі. Це дає змогу скоротити обсяг обчислень і прискорити процес знаходження максимального потоку. У кожній ітерації алгоритму вибирають вершину з надлишковим потоком і просовують зайвий потік у її сусідні вершини. Таким чином, алгоритми просування передпоток працюють поетапно, збільшуючи потік від джерела до стоку, доки не досягнуто межі пропускної здатності або не буде досягнуто максимального потоку.

Надлишковий потік (англ. *excess flow*) представляє собою кількість потоку, яка перевищує пропускну здатність вихідної вершини в мережі, тобто це означає, що алгоритм просування передпоток не забезпечує збереження потоку під час свого виконання, на відміну від методу Форда-Фалкерсона. Натомість він підтримує **передпотік** (англ. *preflow*), який являє собою функцію $f: V \times V \rightarrow \mathbb{R}$, яка схожа на потік, за виключенням того, що замість звичайної умови збереження потоку вона задовольняє **послаблену умову збереження потоку**: $f(V, u) \geq 0$ для всіх вершин $u \in V \setminus \{s\}$. На кожному кроці алгоритма, коли відбувається просування надлишкового потоку, функція передпоток оновлюється відповідно до перерозподілу потоку.

Також вводиться поняття **висоти вершини**, що представляє собою функцію $h: V \rightarrow \mathbb{N} \cup \{0\}$, причому $h(s) = |V|$, $h(t) = 0$ та $h(u) \leq h(v) + 1$ для $\forall (u, v) \in E_f$. Тут варто зазначити, що f є саме передпоток у G , а не просто потоком, як у попередніх алгоритмах.

Алгоритм заснований на двох основних операціях: просуванні та піднятті.

Просування (англ. *push*) – це операція, під час якої вершина u пересилає частину свого надлишкового потоку (тобто різниці між вхідним і вихідним потоком) по одному з ребер до сусідньої вершини v . Просування можливе лише за наступних умов:

- 1) вершина u має бути переповненою, тобто $e(u) > 0$;
- 2) ребро (u, v) має належати залишковій мережі, $(u, v) \in G_f$;
- 3) $h(u) = h(v) + 1$.

Підняття (англ. *relabel*) – це операція, за якої для вершини u збільшуємо висоту до мінімально можливої, щоб зробити хоча б одне ребро допустимим для просування надлишкового потоку в бік стоку t . Підняття можливе лише за наступних умов:

- 1) $e(u) > 0$;
- 2) $h(u) \leq h(v)$ для всіх ребер (u, v) залишкової мережі.

Ці операції повторюються в циклах до досягнення максимального потоку, алгоритм завершується, коли більше немає переповнених вершин або можливих просувань.

Таким чином, алгоритм просування передпотіку працює поступово збільшуючи потік і оптимізуючи його розподіл у мережі. У результаті, він забезпечує ефективне розв'язання задачі максимального потоку, особливо у великих і складних мережах.

Алгоритм працює наступним чином.

1. Ініціалізація передпотіку: встановлюємо висоту витіку, що дорівнює кількості вершин у мережі, а висоту решти вершин рівною нулю. Потім просуваємо максимально можливий потік із витіку по всіх вихідних ребрах. Таким чином, витік стає порожнім, а всі його сусіди – переповненими.

2. Поки існує переповнена вершина (крім стоку), обираємо її і виконуємо одну з операцій: просування або підняття. Якщо у вершини є допустиме ребро для просування, то просуваємо по ньому частину або весь надлишковий потік. Якщо у вершини немає допустимих ребер для просування, то піднімаємо її на мінімально можливу висоту.

3. Коли всі вершини (крім стоку) стають порожніми, алгоритм завершується. Поточний передпотік є максимальним потоком у мережі.

Приклад роботи зображено на рис. 4.

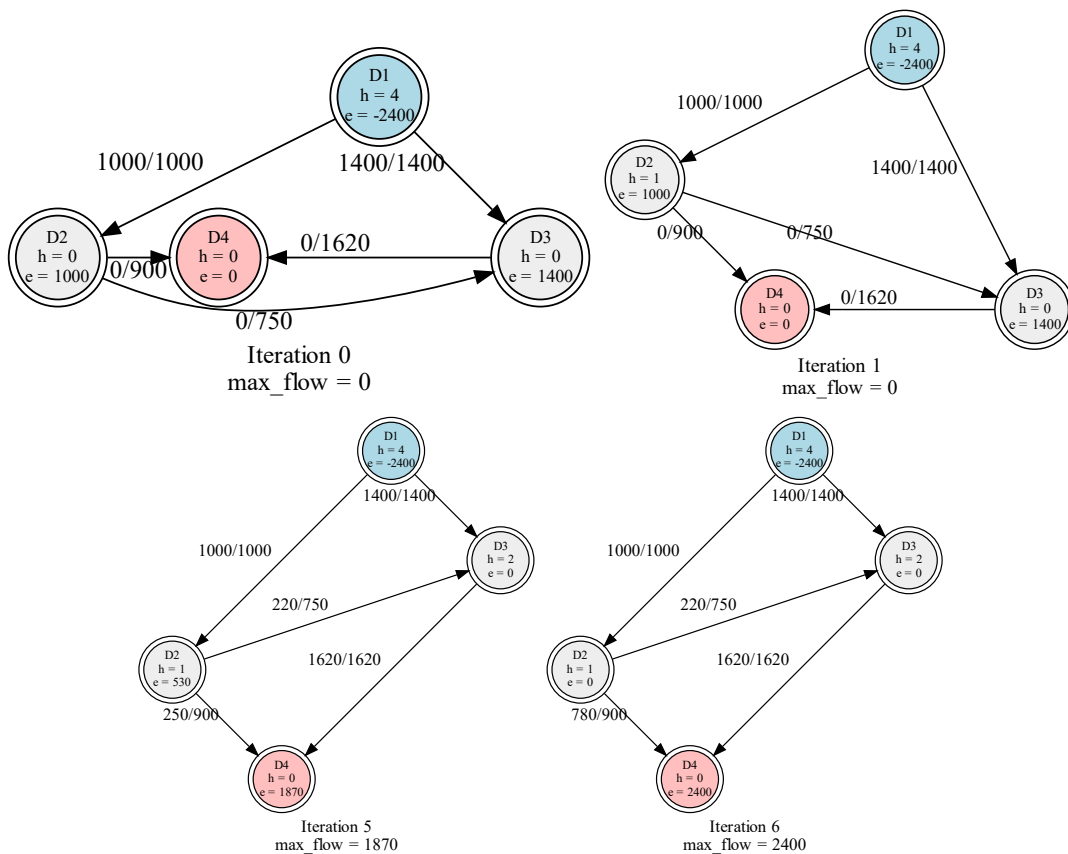


Рисунок 4. Алгоритм просування передпотіку

Алгоритм просування передпотіку може бути застосовано для розв'язання різних завдань, пов'язаних з оптимізацією потоків у мережах, таких як задача про мінімальний вартісний потік, задача про розбиття графа, задача про максимальне паросполучення.

2.4. Порівняння алгоритмів

Загалом, алгоритм Форда-Фалкерсона є досить простим, але дієвим алгоритмом для розв'язання задач про максимальний потік, але він має певні обмеження, зокрема, при роботі з великими графами та необхідності ретельної реалізації для забезпечення завершення та

ефективності. Також вагомим недоліком є дуже висока залежність від значень максимального потоку та пропускних здатностей мережі, через свою асимптотичну складність.

Алгоритм Едмондса-Карпа має переваги, такі як краща складність за часом, простота реалізації та гарантоване завершення роботи. Його асимптотична складність у найгіршому випадку є значно кращою за часом, ніж у алгоритму Форда-Фалкерсона, що покращує швидкість виконання. Проте він має деякі недоліки, зокрема, повільну роботу у щільних графах, можливість виконувати неефективну роботу і чутливість до топології графа. При виборі алгоритму слід враховувати специфіку задачі та особливості графа.

Алгоритм просування передпотуку має кілька значних переваг. По-перше, він має кращу асимптотичну складність порівняно з іншими алгоритмами, що дозволяє ефективно вирішувати задачі потоку в графах. Крім того, його локальний підхід і простота розуміння роблять його придатним для паралельних та розподілених реалізацій. Більш того, алгоритм може бути легко модифікований для вирішення різних проблем, пов'язаних з потоком, забезпечуючи гнучкість в його застосуванні.

Однак, варто враховувати деякі недоліки алгоритму. Перш за все, його реалізація може бути складною і вимагати глибокого розуміння теорії графів і структур даних. Зокрема, ефективність алгоритму значно залежить від евристики, що потребує приділяти значну увагу до правила вибору вершини. Нестабільність алгоритму і можливість виконання зайвої роботи є ще двома недоліками, які можуть ускладнити оцінку його продуктивності на реальних даних. Проте, незважаючи на його недоліки, алгоритм просування передпотуку є потужним і ефективним інструментом для вирішення задач потоку в графах.

3. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Усі три алгоритми було програмно реалізовано з візуалізацією їх роботи. Для їх реалізації, була обрана мова програмування Python та середовище програмування Jupyter Lab. Для простішого розуміння роботи алгоритмів було реалізоване графічне представлення. Для цього була використана бібліотека Graphviz. Також Graphviz, на відміну від більш популярного NetworkX, дозволяє дуже просто графічно представляти графи з мультиребрами, у той час коли NetworkX з'єднує декілька таких ребер в одне, що унеможлиблює коректне відображення роботи алгоритмів. Також, у візуалізації роботи алгоритма просування передпотуку висота кожної вершини відповідає висоті розміщення цієї вершини на рисунку, що покращує сприйняття роботи алгоритма.

Для порівняння роботи алгоритмів будемо вимірювати час роботи кожного з них, в залежності від графа. Спочатку порівняємо час роботи в залежності від кількості ребер графу, при фіксованому значенні вершин, у кількості 100 вершин. Пропускна здатність кожного ребра буде випадкова, від 1 до 40.

Так як кількість вершин у нас 100 і граф зв'язний та орієнтований, то максимальна кількість ребер буде становити $N(N - 1) - 2N = 100 \cdot 99 - 200 = 9700$. Ми тут віднімаємо $2N$ через те, що з джерела ребра можуть тільки виходити N ребер, а у стік тільки надходити.

На рис. 5 зліва показана залежність часу виконання у секундах від кількості ребер у графі, а справа – залежність від кількості вершин. Залежно від щільності графа, алгоритми показують різні результати.

На рис. 5 зліва показано, що в щільному графі найшвидшим є алгоритм просування передпотуку, а найповільнішим – Форда-Фалкерсона. На противагу справа видно, що в розрідженому графі ситуація змінюється: найшвидшим стає алгоритм Едмондса-Карпа, а найповільнішим – просування передпотуку. Це пояснюється квадратичною залежністю асимптотичної складності алгоритма просування передпотуку, яка дорівнює $O(|V|^2|E|)$.

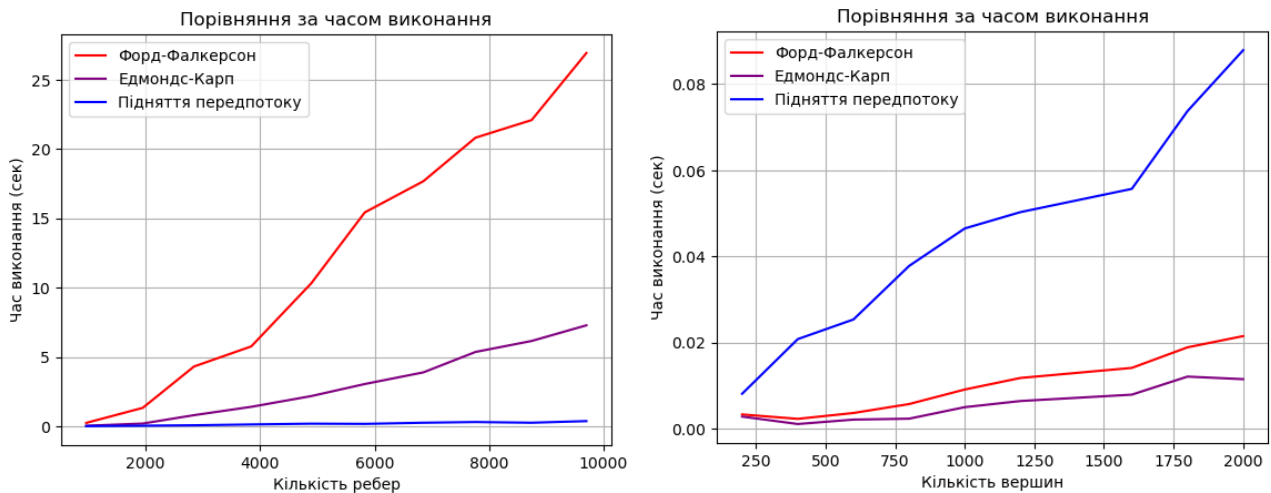


Рисунок 5. Залежність часу роботи від кількості ребер та вершин

Спираючись на результати, можемо зробити висновок, що алгоритм Форда-Фалкерсона завжди програє у швидкості алгоритму Едмондса-Карпа. В свою чергу, алгоритм просування передпотoku у загальному випадку виявився набагато краще за Едмондса-Карпа, за винятком ситуації з розрідженим графом, у якого багато вершин, але такі випадки не дуже розповсюджені та грають велику роль лише в окремих випадках.

4. ВИСНОВКИ

У роботі проведено детальне аналітичне порівняння та продемонстрований результат програмної реалізації задачі про максимальний потік. Досліджені основні теоретичні поняття, визначення та важливі алгоритми, такі як алгоритми Форда-Фалкерсона, Едмондса-Карпа та просування передпотoku. Результати порівняння алгоритмів показали, що найшвидшим серед них є алгоритм просування передпотoku, проте кожен з них має свої переваги та недоліки, і вибір конкретного алгоритму залежить від вхідних умов та вимог ефективності.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Abd-Alsabour N. The maximum flow problem. *International journal of engineering research and technology*. 2020. Vol. 13, no. 7. P. 1534–1545.
2. Tayyebi J., Deaconu A. Inverse generalized maximum flow problems. *Mathematics*. 2019. Vol. 7, no. 10. P. 899.
3. Introduction to algorithms / T. H. Cormen et al. 3rd ed. Cambridge, MA, USA : The MIT Press, 2009. 1312 p.
4. Harris T. E., Ross F. S. Fundamentals of a method for evaluating rail net capacities. Santa Monica, CA : RAND Corporation, 1955. 63 p.
5. Ford L. R., Fulkerson D. R. Maximal flow through a network. *Canadian journal of mathematics*. 1956. Vol. 8. P. 399–404.
6. Edmonds J., Karp R. M. Theoretical improvements in algorithmic efficiency for network flow problems. *Journal of the ACM*. 1972. Vol. 19, issue 2. P. 248–264.
7. Goldberg A. V., Tarjan R. E. A new approach to the maximum-flow problem. *Journal of the ACM*. 1988. Vol. 35, issue 4. P. 921–940.

ВИБІР ТА ОЦІНКА ЯКОСТІ МОДЕЛЕЙ ПРОГНОЗУВАННЯ ФІНАНСОВИХ ПОКАЗНИКІВ З УРАХУВАННЯМ ХАРАКТЕРИСТИК ВХІДНИХ ДАНИХ

Бойніцька С.В.¹, Мілявський Ю.Л.²

Національний технічний університет України «Київський політехнічний інститут
імені Ігоря Сікорського», Київ, Україна

¹ sonia.boinitska@gmail.com, ² yuriy.milyavsky@gmail.com

Складність взаємодії різних факторів, які визначають фінансові показники, вимагає розробки методів прогнозування, здатних враховувати різноманітність вхідних даних. Головний внесок дослідження полягає в створенні універсального програмного продукту, який автоматично аналізує характеристики даних і на їхній основі рекомендує найкращу модель прогнозування. Використання такого підходу значно підвищує якість та швидкість прогнозів, що сприяє ефективному фінансовому плануванню.

Ключові слова: прогнозування, фінансові показники, часові ряди, вибір моделей, характеристики, стратегічне управління.

1. ВСТУП

У сучасному світі, де обсяги фінансової інформації неспинно зростають, вирішення завдань прогнозування фінансових показників стає важливим етапом стратегічного управління. З метою вдосконалення цього процесу та надання ефективних інструментів для прийняття рішень, в рамках даного дослідження була створена програма, яка використовує інноваційний підхід до вибору моделей прогнозування фінансових показників.

Метою дослідження є розробка методології для оцінки якості моделей прогнозування фінансових показників, в залежності від характеристик, які притаманні вхідним даним, а також створення програмного продукту, який дозволить автоматизувати цей процес.

Завдання роботи полягають у аналізі теоретичної інформації щодо методів оцінювання моделей прогнозування та їх залежності від вхідних даних, а також у розробці методології для порівняння моделей.

Актуальність дослідження зумовлена швидкими темпами змін у фінансовому середовищі та необхідністю оперативної та точної інформації для ефективного управління ресурсами. Завдяки автоматизованому підходу до вибору та оцінки моделей прогнозування, створений продукт відповідає потребам ринку, дозволяючи зробити процес прогнозування більш ефективним, швидким і надійним.

Наукова новизна дослідження полягає в тому що у ньому вперше була розроблена методологія яка співвідносить характеристики вхідних часових рядів до найбільш доцільних моделей для прогнозування для кожного випадку, а також у створенні програмного забезпечення, універсального для різноманітних вхідних фінансових даних.

Розроблена програма, спираючись на результати теоретичного аналізу, систематично визначає ключові параметри вхідних часових рядів та на основі виділених характерних ознак формує рекомендації щодо вибору найбільш придатної моделі прогнозування.

Однією з ключових переваг розробленого продукту є універсальність підходу до моделювання різноманітних фінансових часових рядів. Програма враховує індивідуальні

особливості кожного датасету, автоматично конфігуруючи параметри моделей, беручи до уваги велику кількість факторів, що можуть впливати на його поведінку.

Це відкриває нові можливості для прогнозування, що робить дане дослідження інноваційним у сфері автоматизації аналізу та прогнозування часових рядів.

Розроблений підхід може знайти застосування у різних галузях, де прогнозування фінансових показників є ключовим елементом стратегічного управління – наприклад, в сферах економіки, інвестицій, торгівлі чи виробництва, де швидкі і точні прогнози є ключовим елементом для досягнення успішних стратегій та оптимізації ресурсів.

2. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

В результаті дослідження було створено універсальний алгоритм для вибору найкращої моделі для вхідного набору. Алгоритм наведений на рисунку 1:

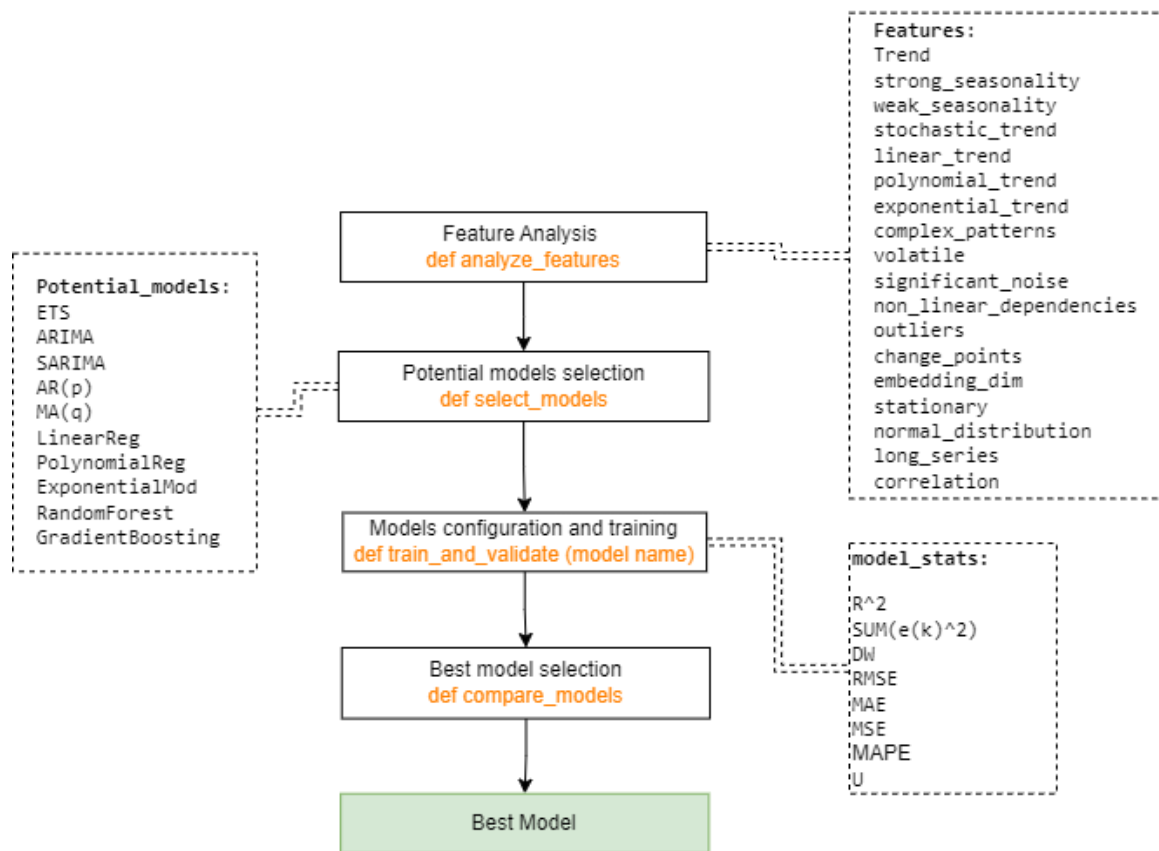


Рисунок 1. Алгоритм створеного програмного продукту

Етап 1. Аналіз ознак вхідних даних (Feature Analysis)

Можна виділити основний перелік ознак, які впливають на вибір майбутньої моделі та які перевіряються для кожного вхідного датасету: стаціонарність, кореляція, наявність тренду, тип тренду: лінійний, поліноміальний, експоненційний чи стохастичний, сезонність, волатильність, шум та викиди, нелінійні залежності в даних, наявність точок зміни, нормальний розподіл залишків, довжина вибірки

Етап 2. Вибір потенційних моделей (Potential models selection)

Тепер, коли вже відомі характеристики, які притаманні вхідному ряду, оберемо, які моделі будуть доцільними для прогнозування. Отже, на цьому етапі в залежності від ознак, що були обраховані на етапі 1, обирається набір потенційних моделей, які підходять до даного часового ряду. Кожна модель обирається в залежності від певних умов.

Загальний алгоритм вибору моделей зображено на рис. 2. Тут зеленим позначені моделі, які додаються до масиву «потенційних» при умові наявності відповідної характеристики у вхідних даних, а червоним навпаки позначені моделі, які видаляються.

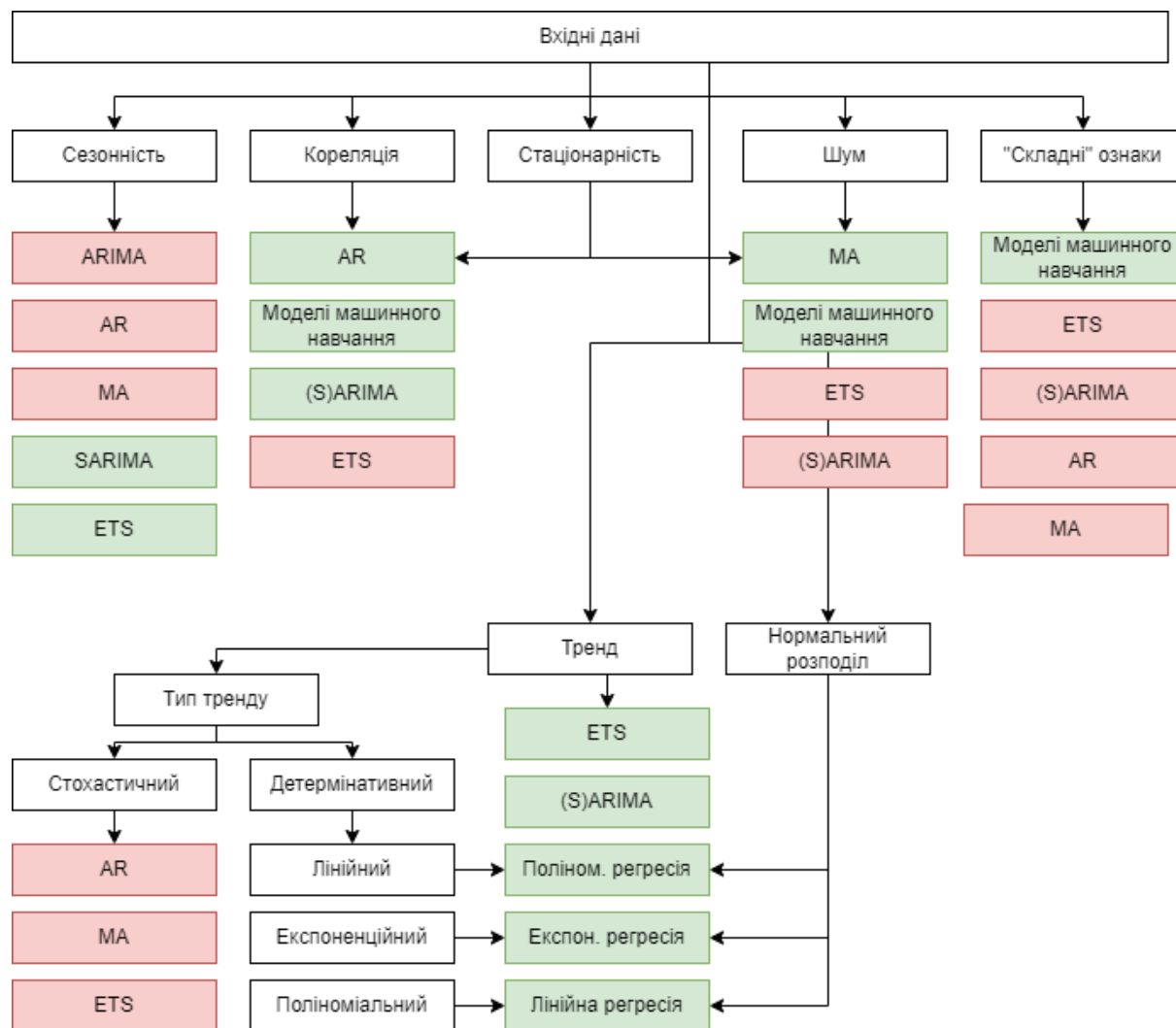


Рисунок 2. Залежність вибору потенційної моделі від наявних ознак

Розглянемо детальніше зв'язки, зображені на рисунку вище.

Умови на моделі Лінійної і Поліноміальної регресії доволі прості, їх пробуємо при наявності відповідного тренду – лінійного чи поліноміального.

Аналогічно Експоненційна модель будується для експоненційного тренду. Сама модель являє собою застосування логарифмічної трансформації для вхідних даних і потім лінійної регресії.

Також для моделей регресії дуже важливим є щоб залишки були розподілені за нормальним розподілом, тож цей параметр також додано до «визначальних» характеристик, необхідних для моделі.

Додавання цих моделей до переліку потенційних необхідно для того, щоб мати певну базу для порівняння роботи більш складних моделей. Також у випадку простих вхідних даних використання такої моделі може бути більш доцільним, ніж налаштування і використання більш складних систем.

Модель машинного навчання Gradient Boosting застосовується у випадках, коли наявні «складні» характеристики вхідних даних і звичайні моделі, як Експоненційне згладжування і ARIMA, можуть бути не доцільними. Ці моделі обираються, коли присутні такі характеристики як волатильність, шум, кореляція, викиди, нелінійні залежності, точки змін.

В той же час при виборі ETS, (S)ARIMA, AR і MA стоїть умова на відсутність цих ознак.

Основним рушієм в виборі моделі AR(p) є присутність кореляції. При виборі MA(q) важливим є присутність шуму. Відсутність сезонності та стохастичного тренду є критерієм для обох моделей. Також обидві моделі вимагають, щоб дані були стаціонарні.

ARIMA поєднує в собі ознаки AR(p) і MA(q), проте завдяки параметру d ще здатна працювати з трендом. А SARIMA здатна працювати з сезонністю. Також розглядається сценарій, за якого ARIMA обирається для нестационарних даних, адже вона може привести їх до стаціонарності за допомогою I компоненти.

ETS добре працює з трендом і сезонністю, проте, на відміну від ARIMA, в ETS стоїть умова на відсутність кореляції. Також ETS погано працює зі стохастичним трендом, при його наявності обирається ARIMA яка має I компоненту для роботи з трендом.

Етап 3. Вибір параметрів моделей, їх тренування і валідація (Models configuration and training)

Після того, як на етапі 2 ми отримали масив потенційних моделей, починається робота з кожною з них. Моделі конфігуруються, навчаються і валідуються. Також на цьому етапі обчислюються метрики якості прогнозів, такі як R^2 , $\text{SUM}(e(k)^2)$, 'DW', 'RMSE', 'MAE', 'MSE', 'MAPE', 'U'.

Етап 4. Вибір найкращої з моделей (Best model selection)

На цьому етапі для кожної моделі, на основі її метрик, обчислюється її score. Score враховує кожен метрику обчислену для моделі зважаючи на її вагу. Після обчислення кількості балів у моделі обирається модель з найнижчим score. Ця модель і вважається найкращою для вхідного набору даних.

3. РОБОТА ПРОГРАМИ

Наведемо приклад роботи програмного продукту. В ролі вхідних даних використаємо акції компанії Google з 1 жовтня 2010 по 1 січня 2017.

Дані зображені на рис. 3.



Рисунок 3. Графік часового ряду значення акцій Google

Застосуємо розроблений алгоритм до цих даних. Схема зображена на рисунку 4.

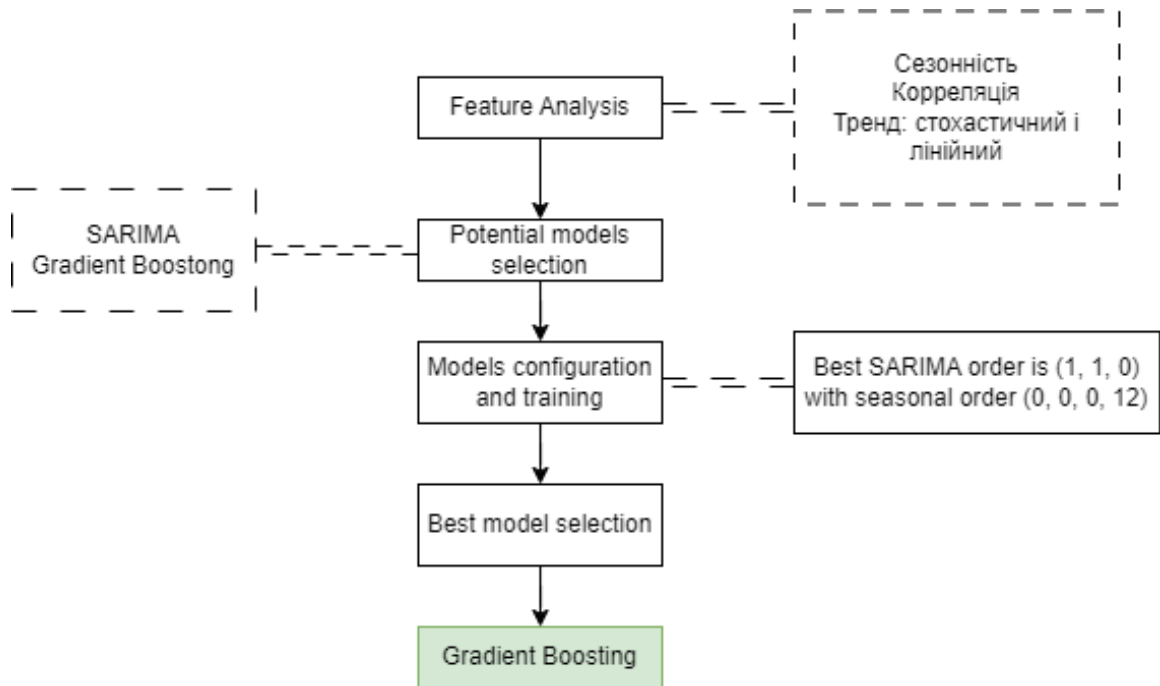


Рисунок 4. Робота алгоритму для даних з прикладу

Етап 1. Під час аналізу даних виділено такі характеристики, як наявність сезонності і тренду, а також кореляції в ряді.

Етап 2. На основі цих характеристик обрано моделі SARIMA (для роботи з трендом, сезонністю і кореляцією) та Gradient Boosting (для роботи з кореляцією)

Етап 3. Під час конфігурації моделей визначено, що найкращим варіантом параметрів SARIMA є порядок $p = 1$, $d = 1$, $q = 0$, а також сезонний період $s = 12$. Результати тренування моделей зображені на рис. 5 та 6 відповідно.

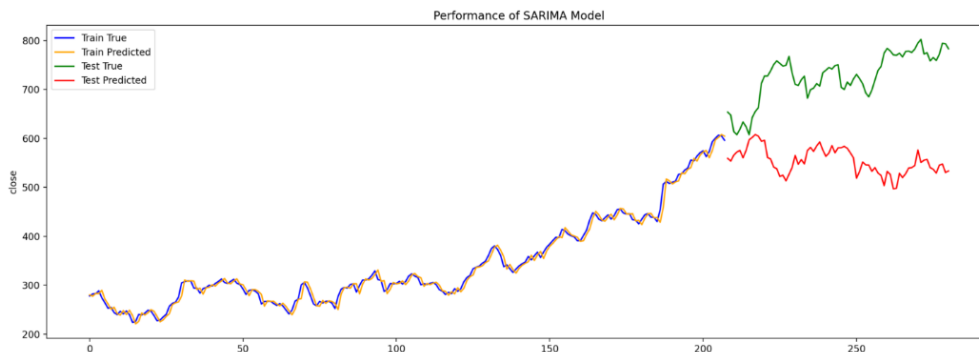


Рисунок 5. Робота моделі SARIMA на даних з прикладу

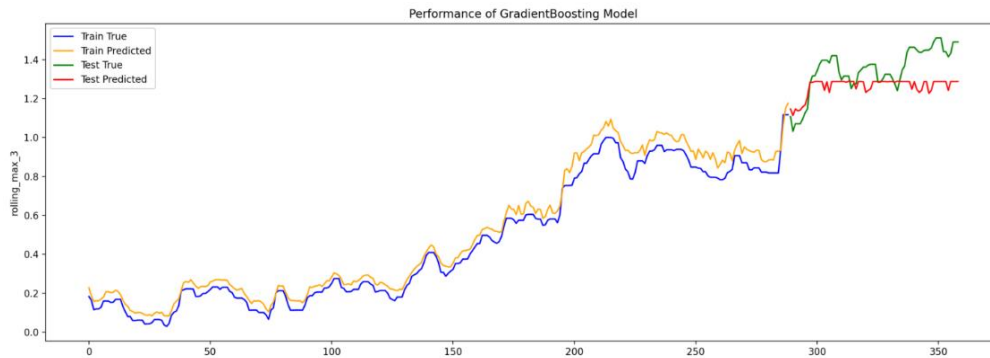


Рисунок 6. Робота моделі Gradient Boosting на даних з прикладу

Обчислені значення метрик наведені у таблиці 1.

Таблиця 1. Значення метрик для даних з прикладу

model_name	R ²	SUM(e(k) ²)	DW	MSE	RMSE	MAE	U
SARIMA(1, 1, 0)x(0, 0, 0, 12)	0,99	18735,55	1,846	34531,106	185,825	173,161	1,576
GradientBoosting	0,97	0,245	0,468	0,016	0,125	0,102	0,00035

Етап 4. Як результат, на етапі вибору найкращої моделі обрано модель Gradient Boosting.

4. ВИСНОВКИ

В даній роботі досліджено автоматизацію вибору оптимальних моделей прогнозування фінансових показників на основі характеристик вхідних даних. Розроблено програму та логіку вибору моделей, базовану на теоретичному дослідженні. Отримані результати підтверджують ефективність алгоритму та його універсальність для різних фінансових часових рядів. Застосування цього підходу має великий потенціал для автоматизації прогнозування та управління фінансовими ресурсами.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Бідюк П.І., Романенко В.Д., Тимошук О.Л. Аналіз часових рядів (навчальний посібник). Київ: Політехніка, 2010. 317 с.
2. Hyndman, R.J., & Athanasopoulos, G. Forecasting: principles and practice, 3rd edition, 2021 [Електронний ресурс] – Режим доступу до ресурсу: OTexts: Melbourne, Australia. URL: <https://otexts.com/fpp3/>
3. Box, G.E.P., Jenkins, G.M. (1970). Time Series Analysis: Forecasting and Control. Holden-Day, San Francisco.
4. Hyndman, R. J., Koehler, A. B., Ord, J. K., & Snyder, R. D. (2008). Forecasting with exponential smoothing: The state space approach. Springer-Verlag. [Електронний ресурс] – Режим доступу до ресурсу: <https://robjhyndman.com/expsmooth/http://www.exponentialsMOOTHING.net>

АНАЛІЗ ТЕКСТОВИХ ПОВІДОМЛЕНЬ ЗА ДОПОМОГОЮ МЕТОДІВ МАШИННОГО НАВЧАННЯ

Ведмедєв Д.О.¹, Шаповал Н.В.²

Національний технічний університет України «Київський політехнічний інститут ім. Ігоря Сікорського», ПСА, Київ, Україна

¹ vedmedevdanil@gmail.com, ² shovgun@gmail.com [0000-0002-8509-6886]

Ріст кількості текстових даних, зокрема СМС, вимагає розвитку аналізу та кластеризації для виявлення закономірностей і групування повідомлень. Дане дослідження об'єднує в собі використання вбудованих векторних представлень слів (Embedded Word2Vec), метод Mini Batch K-means та метод найбільшої спільної підпоследовності. Мета даного дослідження - розробити ефективний метод кластеризації СМС-повідомлень, яка може бути використана для автоматичного виявлення текстових шаблонів.

Ключові слова: аналіз текстових повідомлень, Embedded Word2Vec, Mini Batch K-means, метод найбільшої спільної підпоследовності, кластеризація, СМС-повідомлення.

1. ВСТУП

З кожним днем у сучасному світі із розвитком соціальних мереж, розвиваються і методи комунікації. Одним із таких методів є короткі текстові повідомлення (СМС). Хочу наголосити, що саме короткі, тому що останнім часом у людей розвивається активно “кліпова увага”, і одним із головних атрибутів даного явища, це розсіяний фокус та відсутність довготривалої концентрації на одній події довгий час. Саме це явище безпосередньо впливає на актуальність саме коротких СМС.

Метою даного дослідження виявлення ефективних методів комунікації, а також виявлення спам розсилок серед “живого трафіку”. Тобто результатом роботи буде скрипт, що виконує аналіз та кластеризацію великого обсягу коротких текстових повідомлень за допомогою методів машинного навчання, та оцінка результатів кластеризації.

Ключовими аспектами дослідження є розробка вбудованих векторних представлень слів (Embedded Word2Vec), використання Mini Batch K-means для ефективною кластеризації великого обсягу даних, а також використання методу найбільшої спільної підпоследовності для аналізу схожості текстів.

2. АНАЛІЗ ТА КЛАСТЕРИЗАЦІЯ ТЕКСТІВ

Для аналізу вирішення поставленої задачі в першу чергу потрібно звернутись до постановки задачі. В якій сказано: “Із вхідного великого масиву коротких СМС-повідомлень виділити шаблони СМС-ної розсилки, отримавши на виході назву шаблону та кількість екземплярів вхідної вибірки, що до цього шаблону належать”. Отже, отримавши якісне формулювання задачі, можна загальну задачу розбити на декілька підзадач:

1. *Перетворення текстів у числові вектори*
2. *Кластеризація числових векторів*
3. *Оцінка кластеризації*

2.1. Перетворення текстів у числові вектори

Ця задача була сформульована таким чином, щоб можна було працювати із текстами, як із векторами ознак. Далі вектори ознак можна подати на вхід до підзадачі Кластеризації числових векторів.

Отже, для перетворення текстів у числові вектори був використаний метод вбудованих представлень слів **embedded Word2Vec**. Embedded Word2Vec – це техніка векторного представлення слів у вигляді числових векторів у просторі з низькою розмірністю. Основна ідея полягає в тому, щоб кожному слову призначити унікальний вектор, таким чином, що подібні слова будуть мати схожі вектори. Використання Embedded Word2Vec має декілька переваг:

- **Семантична схожість**: Слова зі схожим значенням розташовані близько одне до одного у векторному просторі, відображаючи їхню семантичну схожість.
- **Збереження контексту**: Враховує синтаксичні та семантичні відносини між словами, зберігаючи контекст використання слова у реченні.
- **Зменшення розмірності**: Зменшує розмірність векторів, зберігаючи при цьому важливу інформацію, що полегшує обробку та аналіз текстових даних.

Ця техніка є ефективним інструментом для роботи з текстовою інформацією, особливо в завданнях машинного навчання, де слова потрібно представити у вигляді числових векторів для подальшого використання у моделях.

2.2. Кластеризація числових векторів

Для кластеризації числових векторів був використаний давно всім відомий K-means, а точніше його модифікація для обробки великих обсягів даних Mini Batch K-means.

Mini Batch K-means – це варіант алгоритму K-means для кластеризації даних. Основна відмінність полягає в тому, що **Mini Batch K-means** використовує випадковий підмасив даних (міні-партію) для оновлення центрів кластерів, що робить його ефективнішим у великих наборах даних.

Переваги Mini Batch K-means:

- **Швидкість**: Mini Batch K-means часто працює швидше за класичний K-means, особливо на великих обсягах даних, оскільки використовує лише частину набору даних для кожного оновлення.
- **Ефективність великих даних**: Підходить для роботи з великими обсягами даних, оскільки не вимагає обробки всього набору даних на кожній ітерації.
- **Можливість онлайн-навчання**: Цей метод може використовуватися для онлайн-навчання, де нові дані можуть динамічно додаватися для постійного покращення моделі.
- **Зменшення вимог до пам'яті**: Завдяки використанню міні-партій, вимоги до обсягу пам'яті значно менше, що робить його більш ефективним для обробки великих наборів даних у вигляді потоків.
- **Тенденція до локальних мінімумів**: Може уникнути застрягання в локальних мінімумах через випадковий вибір міні-партій, що може призвести до кращого розв'язку.

2.3. Оцінка кластеризації

Після етапу кластеризації, її потрібно оцінити. На жаль емпіричним шляхом було виявлено досить неефективне оцінювання кластеризації за допомогою вже існуючих методів оцінки кластеризації:

Silhouette_score вимірює, наскільки кожен об'єкт в кластері схожий на інші об'єкти свого кластера порівняно з кластером, до якого він не належить,

Silhouette_samples – це функція, яка визначає коефіцієнти силуету для кожного окремого елемента в масиві даних. Коефіцієнт силуету вимірює, наскільки об'єкт добре вписується в свій власний кластер в порівнянні з сусідніми кластерами,

Inertia – це сума квадратів відстаней між кожною точкою в кластері та його центром. У контексті методу k-means, inertia визначає, наскільки кластери компактні,

Calinski-Harabasz – також відомий як критерій варіативності внутрішньокластерної суми квадратів, Суть індексу полягає в порівнянні "внутрішньокластерної" дисперсії з "міжкластерною" дисперсією. Ідеальний результат досягається, коли кластери компактні всередині і вони відокремлені один від одного.

Davies-Bouldin index – вимірює, наскільки кожен кластер відокремлений від інших та наскільки компактний у порівнянні з найближчими кластерами.

Отже, використавши підхід прикладного аналізу потрібно було задуматись над евристикою розуміння, що таке є задовільний результат. Задовільним результатом був шаблон тексту, та напроти нього число(кількість екземплярів СМС із таким самим текстом, або схожим). Провівши дослідження даного питання, знайшли метод найбільшої спільної підпоследовності((Longest Common Subsequence, LCS).

LCS – це метод інтуїтивно доволі легкий в розумінні(нагадує добре всім відоме поняття зі школи Найбільшого спільного дільника, НСД), а саме виконує пошук найбільшої спільної підпоследовності символів між двома текстами. Відрізняється від розглянутих метрик саме тим, що працює не з векторами, а саме безпосередньо з текстами що отримали в результаті кластеризації.

Тут слід зазначити, що розуміння "последовності" доволі гнучке, тому що можна текст сприймати як последовність слів, так і последовність символів. Але для нашої задачі більш доречно буде інтерпретувати СМС, як последовність символів, оскільки при написанні в методах комунікації існує людський фактор, через який, одне слово може мати купу різних варіацій написання.

LCS можна використати для порівняння всіх текстів в межах кожного кластеру, але попарне перевірка має складність алгоритму $O(n^2)$. На цьому етапі, можна полегшити обчислювальну складність алгоритму, знайшовши центр кожного кластеру. Центром кожного кластеру будемо вважати найближчу точку до геометричного центру кожного кластеру. І далі можна порівнювати кожен кластер із його центром, таким чином зменшили обчислювальну складність алгоритму до $O(n)$.

Далі за допомогою методу LCS, можна отримати відносну метрику схожості, яка вираховується наступним чином. Спершу розглядається кластер та його центр, а потім визначається найбільша спільна підпоследовність між ними. Ця довжина вимірюється в кількості символів, що є спільними для обох текстів. Далі, ця довжина спільної підпоследовності нормалізується, поділивши на загальну довжину екземпляру тексту (або відомостей у кластері). Це значення дозволяє отримати відсоток схожості між кластером та його центром. Тому було запропоновано метод LCS в якості метрики для аналізу схожості двох текстів. За допомогою цієї метрики "чистоту" кластерів можна знаходити за середнім кожної вибірки, нижнім квантилем вибірки кожного кластеру. Або ж регулювати також межами ступеня довіри до кожного елемента кластера. Тобто виставити, наприклад умову: "якщо екземпляр даного кластеру схожий на його центр на $p\%$, то цей екземпляр належить до цього кластеру", де $p = \{60, 85, 90\}$.

Для прикладу, якщо маємо тексти "кластер" і "центр кластера", то найбільша спільна підпоследовність буде "кластер", і після нормалізації відношення довжин, ми отримаємо

високий показник схожості. Тепер результат кластеризації можна оцінити з точністю до того, що саме мається на увазі під словосполученням “схожість двох текстів”.

3. ВИСНОВКИ

В результаті дослідження було отримано доволі ефективний метод кластеризації текстових повідомлень для виявлення шаблонів та групування їх за схожістю. При цьому він є доволі гнучким та модульним, а саме кожному з наведених підзадач, можна вирішити різними методами машинного навчання або інтелектуального аналізу даних, що в свою чергу не закриває питання актуальності вирішення даної задачі, а навпаки спонукає до пошуку більш ефективних рішень, використання не тільки суцільного скрипту, а й його частин по окремо, або в різних конфігураціях і послідовностях, для вирішення інших задач у сфері аналізу текстових повідомлень.

4. СУМІЖНІ АКТУАЛЬНІ ЗАДАЧІ

Насамперед, дане дослідження може бути корисним для вирішення таких задач як:

- Пошук *spin-words* тобто взаємозамінними словами та словосполученнями, що є різними по формі, та однаковими по суті. Наприклад: “Привіт”, “Вітаю”, “Доброго часу доби” – ці всі вирази мають однакову суть, але різну форму.
- Розпізнавання іменованих об’єктів (Named Entity Recognition, NER). Тобто в цій задачі на вхід отримуємо текстове повідомлення, а на виході повинні отримати Ім’я та прізвище, якщо такі є. Наприклад: “Доброго дня! Джоне Джонсонюк, пишу вам щодо нашої зустрічі.”-> “Джоне Джонсонюк”.

СИСТЕМА ПРОГНОЗУВАННЯ ЕНЕРГОВИТРАТ БУДІВЕЛЬ РІЗНОГО ПРИЗНАЧЕННЯ

Данилов В.Я., Дука О.О.¹

Національний технічний університет України «Київський політехнічний інститут ім. Ігоря Сікорського»

¹ olga.duka575@gmail.com

Електроенергія є невід’ємною частиною сучасного життя та важливою для економіки усього світу, у тому числі України. Очікується, що в найближчому майбутньому електроенергія замінить інші джерела енергії як основне джерело для використання в будинках, на підприємствах і у транспорті. Для ефективного і правильного прогнозування доцільно створити Систему прогнозування енерговитрат будівель різного призначення. Метою роботи є аналіз існуючих методів прогнозування для покращення його точності. Результатом дослідження є система, що виконує прогнозування енерговитрат із застосуванням методів машинного навчання. У роботі було використано теоретичні та емпіричні методи дослідження.

Ключові слова: машинне навчання, енергоспоживання, прогноз, RMSLE, аналіз енерговитрат.

1. ВСТУП

Електроенергія є невід’ємною частиною сучасного життя та важливою для економіки усього світу, у тому числі України. Очікується, що в найближчому майбутньому електроенергія замінить інші джерела енергії як основне джерело для використання в будинках, на підприємствах і у транспорті [1]. Це підкреслює, наскільки важливо правильно прогнозувати споживання електроенергії, оскільки воно має великий вплив на багато операційних і бізнес-операцій. Електроенергія стає головним аспектом нашого повсякденного життя.

Надзвичайний попит на електроенергію прискорюється останнім часом потужним економічним розвитком і швидкою урбанізацією. Прогнозування попиту на електроенергію стає критичним в електричному секторі, оскільки воно служить основою для прийняття важливих рішень у сфері експлуатації та управління енергосистемою. Через спад економічного розвитку, а також відносно помірну температуру в багатьох великих країнах світовий попит на енергію зріс меншими темпами в 2019 році порівняно з останніми роками (+0,7% порівняно із середнім показником 3% на рік у рік період 2000–2018 рр.) [1].

Незважаючи на це, глобальний попит на енергію впав до 2,5% у першому кварталі 2020 року, незважаючи на те, що карантинні заходи в більшості країн тривали лише близько місяця. Зміни в тому, де і як електроенергія використовувалася під час карантину, ще більше змінили структуру попиту на енергію протягом дня в певних районах, причому шаблони будніх днів тепер збігаються з моделями неділі [2]. Це ілюструє відсотковий зсув у попиті та споживанні енергії за різних умов.

За останні десятиліття попит на енергію в будівельному секторі значно зріс через збільшення кількості населення, швидку урбанізацію та соціальний попит. Будівлі зробили значний внесок у світове споживання енергії та викиди парникових газів. Таким чином, будівлі

повинні бути енергоефективними та сталими. Розуміння моделей енергоспоживання в будівлях є корисним для комунальних підприємств, користувачів та керівників об'єктів, оскільки це може допомогти підвищити енергоефективність.

2. ПОСТАНОВКА ЗАДАЧІ

Метою роботи є розробка програмного забезпечення системи прогнозування енерговитрат будівель різного призначення. Буде виконано аналіз і обробку набору даних та порівняння обраних методів машинного навчання для прогнозування енергоспоживання будівель. Методи машинного навчання будуть застосовані на наборі реальних даних. За вхідними параметрами необхідно буде визначити обсяг енерговитрат. Об'єктом дослідження є набір реальних даних, що описує параметри будівель, показники лічильників та погодні дані з найближчих метеорологічних станцій.

3. ОПИС НАБОРУ ДАНИХ

У задачах машинного навчання дуже важливо виконати правильну попередню обробку даних, для того, щоб результати прогнозування були максимально точними. У даній роботі використаний набір даних, зібраних Американським товариством інженерів з опалення, охолодження та кондиціювання повітря (American Society of Heating, Refrigerating & Air Conditioning Engineers; ASHRAE). Параметрами є характеристики будівель (призначення, площа, рік побудови, поверховість); показники лічильників (тип лічильника і час зняття виміру); погодні дані (температура, хмарність, тиск, напрямок і швидкість вітру).

Під час аналізу і обробки даних створена матриця кореляції для визначення зв'язків між змінними. Матрицю кореляції можна побачити на рис. 1:

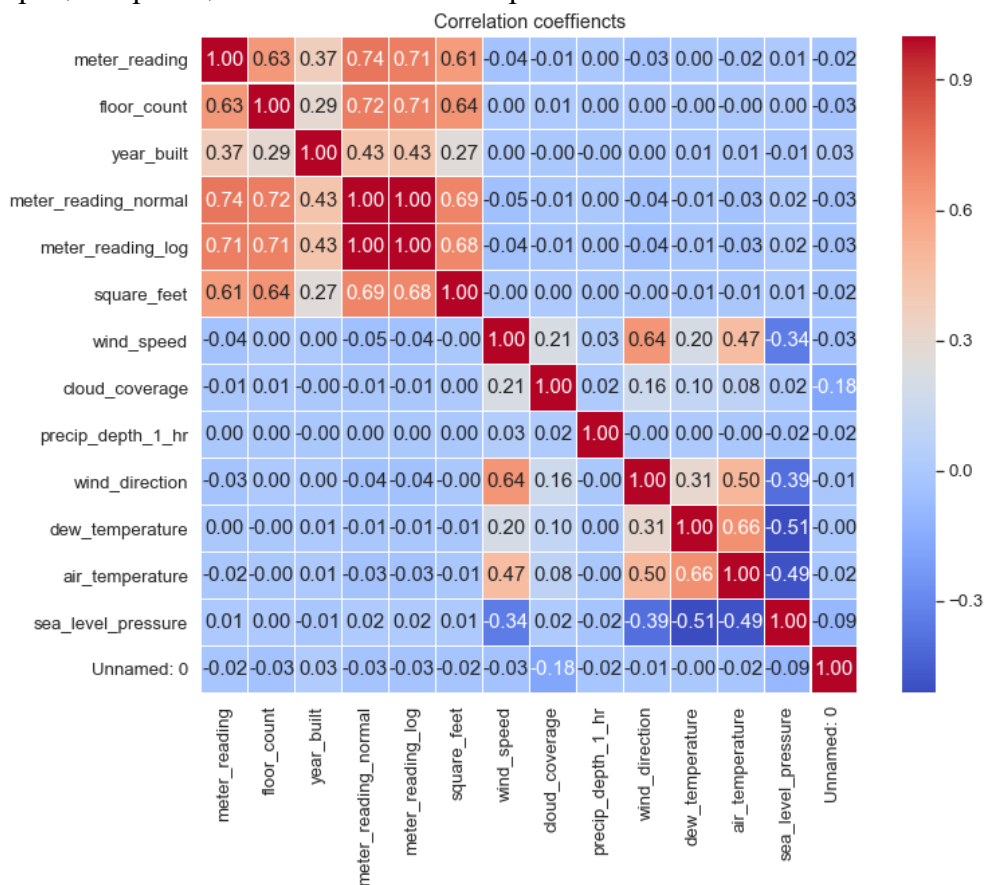


Рисунок 1. Матриця кореляції

Існує висока кореляція між наступними ознаками:

— **dew_temperature & air_temperature** – 0,75;

— **square_feet & floor_count** – 0,56;

— **site_id & building_id** – 0,98.

Решта ознак мають кореляцію менше 0,50.

Розмір будівлі визначає споживання електроенергії, що ми бачимо з наведеного графіка, де кореляція між квадратними футами та кількістю поверхів становить 0,56. Можна також помітити, що температура теж впливає на енергоспоживання.

4. ОПИС АЛГОРИТМУ

У цю нову еру сучасного глибокого навчання дуже спокусливо використовувати сучасну модель глибокого навчання для вирішення будь-якої проблеми. Але завжди треба враховувати вищезазначені фактори, перш ніж відразу переходити до вибору моделі на основі DL.

Розглянемо приклад. Очевидно, що це регресія, оскільки ми прогнозуємо показання лічильників, які є реальними. Крім того, маємо приблизно 40–45 функцій, тому він потрапляє в категорію з низькою кількістю функцій. Ми маємо лише 2380 пар будівельних лічильників з даними за 1 рік для навчання, тому вибірка не є великою.

Як правило, моделі GBDT добре працюють з даними низької розмірності, хоча вони можуть зайняти багато часу через адитивну природу алгоритму.

Крім того, у нас є лише пара будівельних лічильників 2380, тому ми не повинні починати з моделі на основі DL, як LSTM.

Спочатку ми створюємо `u_predict`, використовуючи `train_data`. Потім розділяємо наші `train_data` на `x_train` і `x_cv` як набір для навчання та перехресної перевірки (*cross-validation*) за допомогою бібліотеки Scikit-Learn `train_test_split`. З цього ми створюємо `u_train` і `u_cv`. Тепер, використовуючи ці набори, запустимо різні моделі машинного навчання та зробимо прогнози для `test_data`.

Кросс-валідація (*cross-validation*) — це метод оцінки точності моделі машинного навчання, який використовується для зменшення перенавчання (*overfitting*) та недонавчання (*underfitting*). При кросс-валідації модель будується на частині даних, а потім оцінюється на іншій частині даних. Це дозволяє оцінити, як модель буде працювати на нових даних, які не були використані для її навчання.

Так як цільова змінна має широкий діапазон значень, то є сенс користуватися метрикою RMSLE (Root Mean Squared Logarithmic Error) (1):

$$RMSLE = \sqrt{\frac{1}{n} \sum_{i=1}^n (\log(p_i + 1) - \log(a_i + 1))^2} \quad (1)$$

де n – кількість прикладів у наборі даних; p_i – прогнозоване значення (прогноз); a_i – фактичне (спостережуване) значення.

Це дозволяє приділяти менше уваги великим аномальним значенням і фокусуватися на точності прогнозу в нормальному діапазоні.

Оскільки вже було взято логарифм цільової змінної, то визначається середня квадратична помилка або RMSE між прогнозованими та фактичними значеннями цільової змінної (**meter_reading**) (2):

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (p_i - a_i)^2} \quad (2)$$

де n – кількість прикладів у наборі даних; p_i – прогнозоване значення (прогноз); a_i – фактичне (спостережуване) значення.

Decision Tree Regression Model

Застосовується регресійна модель Decision Tree до наборів x_{train} , x_{test} , y_{train} і y_{test} . Помилки перехресної перевірки для цієї моделі наведено нижче на рис. 2:

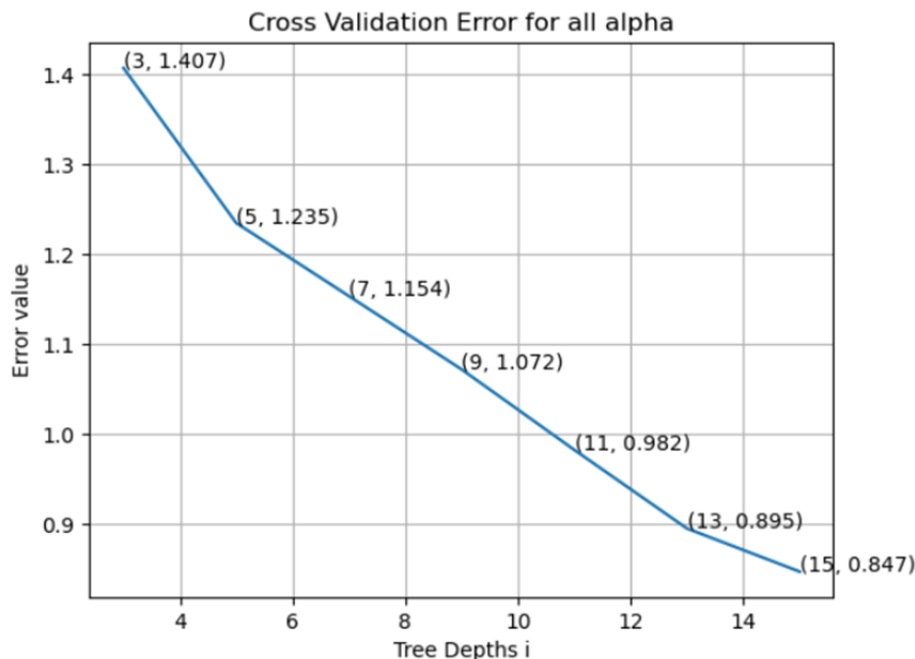


Рисунок 2. RMSLE для Decision Tree

Спостерігається, що найменша помилка перехресної перевірки припадає на глибину дерева 15 із середньоквадратичною помилкою 0,847.

Цю помилку можна ще більше зменшити в інших майбутніх моделях.

Light Gradient Boosting Machine (LGBM) Decision Tree Regression Model

Тепер застосовуємо регресійну модель дерева рішень Light GBM до наборів x_{train} , x_{test} , y_{train} і y_{test} . Помилки перехресної перевірки для цієї моделі наведено нижче на рис. 3:

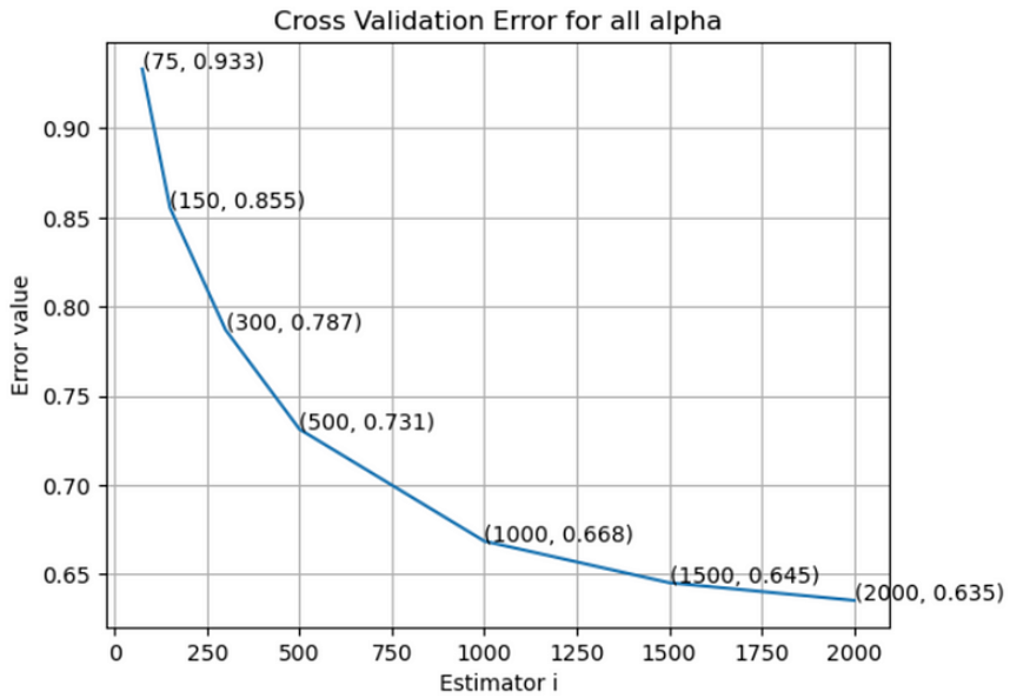


Рисунок 3. RMSLE для LGBM

Random Forest Regression Model

Тепер ми застосуємо Random Forest Regression Model до наборів x_{train} , x_{test} , y_{train} і y_{test} (рис. 4):

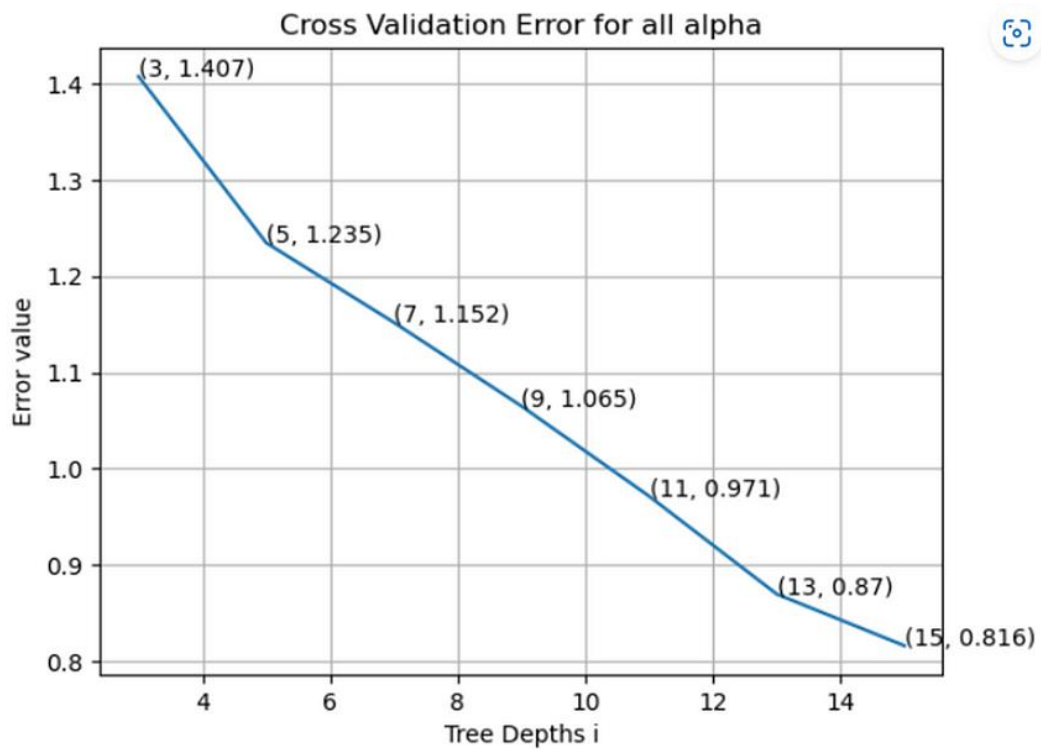


Рисунок 4. RMSLE для Decision Tree

Light Gradient Boost Machine (LGBM) Random Forest Regression Model

Тепер ми застосовуємо регресійну модель випадкового лісу Light GBM до наборів x_{train} , x_{test} , y_{train} і y_{test} (рис. 5):

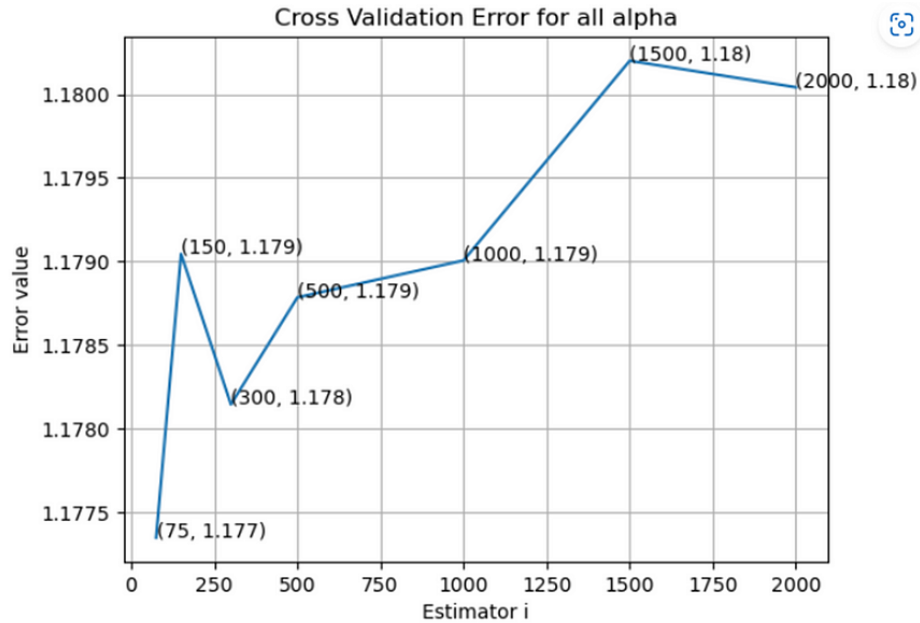


Рисунок 5. RMSLE для LGBM Random Forest

CatBoost Regression Model

Тепер ми застосовуємо модель Categorical Boost або регресійну модель CatBoost до наборів x_{train} , x_{test} , y_{train} і y_{test} (рис. 6):

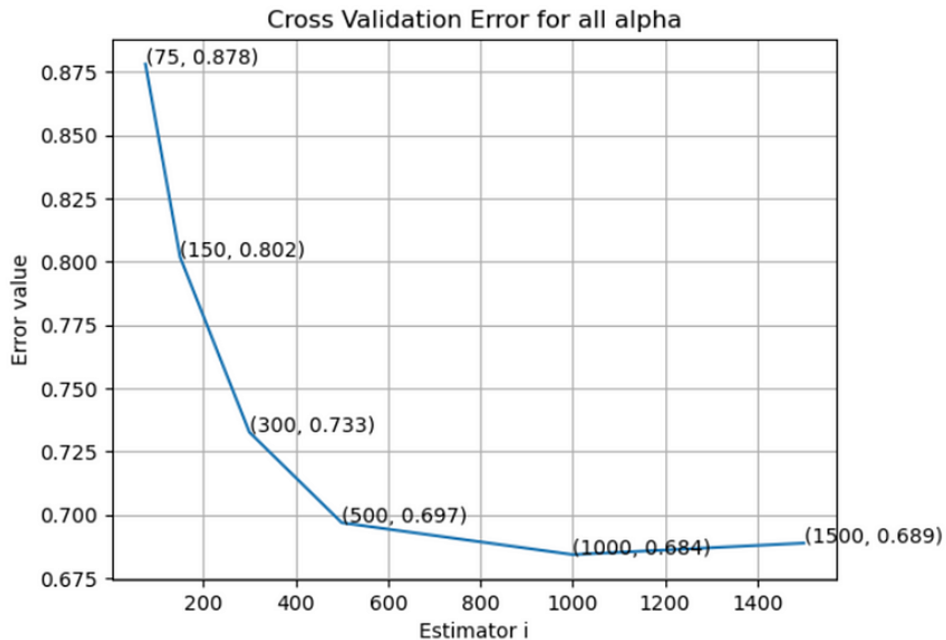


Рисунок 6. RMSLE для CatBoost

Extreme Gradient Boost (XGBoost) Regression Model

Тепер застосовуємо Extreme Gradient Boost або регресійну модель XGBoost до наборів x_{train} , x_{test} , y_{train} і y_{test} .

Помилки перехресної перевірки для цієї моделі наведено нижче на рис. 7:

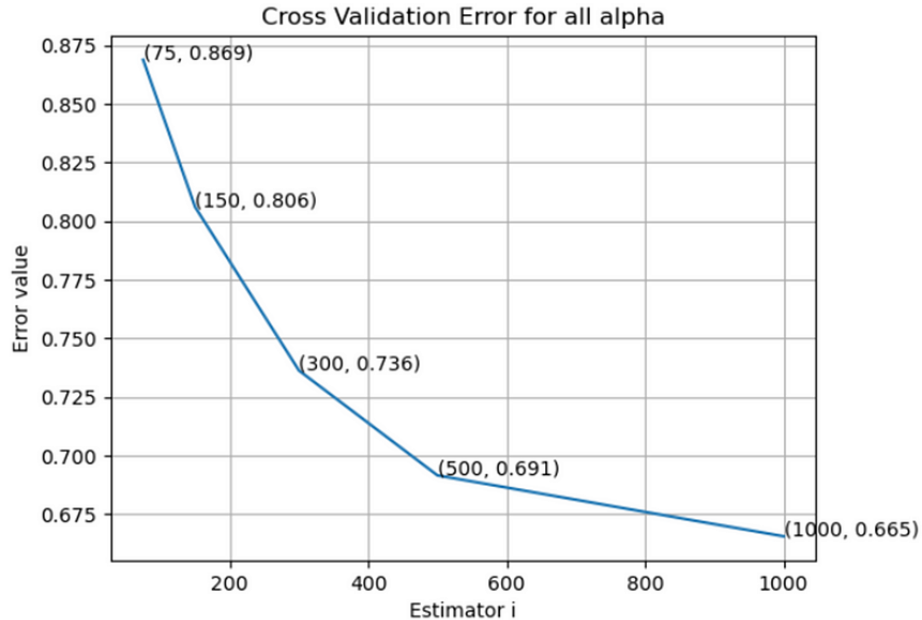


Рисунок 7. RMSLE для XGBoost

Ми зводимо в таблицю всі згенеровані помилки train і перехресної перевірки використаних моделей (табл. 1):

5. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Результат виглядає наступним чином. Ми зводимо в таблицю всі згенеровані помилки train і перехресної перевірки використаних моделей (табл. 1):

Таблиця 1. Результати

Модель	Cross-Validation RMSLE	Train RMSLE
Decision Tree	1,502	1,533
LGBM GBDT	0,578	0,846
Random Forest	0,561	0,816
LGBM Random Forest	1,166	1,177
Catboost	0,575	0,684
XGBoost	0,497	0,665

Найкращою моделлю для прогнозування RMSLE є XGBoost. Вона має найнижчий показник RMSLE на обох наборах даних, тренувальному і крос-валідаційному. Інші моделі також мають порівняно низькі показники RMSLE. Random Forest і Catboost також мають хороші результати, але вони не такі хороші, як XGBoost. Decision Tree і LGBM Random Forest мають вищі показники RMSLE. Це може бути пов'язано з тим, що ці моделі не так добре

справляються з нелінійною залежністю між змінними. Decision Tree має найвищий показник RMSLE. Це може бути пов'язано з тим, що це простий алгоритм, який не може враховувати складні взаємозв'язки між змінними. LGBM GBDT має найнижчий показник RMSLE. Цей алгоритм є одним з найпотужніших алгоритмів машинного навчання для регресії. Він може ефективно враховувати складні взаємозв'язки між змінними. Random Forest також має хороший результат. Цей алгоритм є ансамблем декількох дерев рішень. Він може поліпшити точність прогнозів порівняно з одним деревом рішень. LGBM Random Forest має вищий показник RMSLE, ніж LGBM GBDT. Це може бути пов'язано з тим, що цей алгоритм є більш складним, ніж LGBM GBDT. Він може бути більш сприйнятливим до перенавчання. Catboost має хороший результат. Цей алгоритм є ансамблем декількох моделей CatBoost. Він може ефективно враховувати складні взаємозв'язки між змінними.

6. ВИСНОВКИ

Прогнозування енергоспоживання будівель за допомогою моделей машинного навчання є важливим для оцінки ефективності в контексті модернізації, вимірювання та верифікації, інтеграції відновлюваних джерел енергії, управління системами, виявлення несправностей, енергоспоживання в житловому секторі та моделювання енергетики в міському масштабі.

Висока частка енергії, що споживається в будівлях, спричинила появу багатьох екологічних проблем, які негативно впливають на існування людства. Прогнозування енергоспоживання будівель, по суті, проголошується методом енергозбереження та покращення прийняття рішень щодо зменшення споживання енергії. Крім того, будівництво енергоефективних будівель сприятиме зменшенню загального споживання енергії в новозбудованих будівлях.

Дослідники припускають, що наявність енергетичної системи будівлі з точним прогнозуванням може заощадити від 10 до 30% загального енергоспоживання в будівлях. Без виявлення алгоритму, який може точно прогнозувати енергоспоживання будівлі, це призведе до збільшення викидів парникових газів, будівництва більш неефективних будівель, попиту на енергію та зменшення фінансових заощаджень. У даній роботі було розглянуті методи машинного навчання для прогнозування енерговитрат. Методи машинного навчання було застосовано на наборі реальних даних. Результати прогнозу показали, що найкращою моделлю для прогнозування RMSLE є XGBoost. Вона має найнижчий показник RMSLE на обох наборах даних, тренувальному (0,497) і крос-валідаційному (0,665).

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. World power consumption | Electricity consumption | enerdata. World Energy Statistics Enerdata. Accessed 7 Dec 2020. <https://yearbook.enerdata.net/electricity/electricity-domestic-consumption-data.html>

2. Global energy review 2020. IEA. Accessed 7 Dec 2020. Available from: <https://www.iea.org/reports/global-energy-review-2020/electricity>

МАТЕМАТИЧНІ МОДЕЛІ СУСПІЛЬНИХ ПРОЦЕСІВ ДЛЯ АНАЛІЗУ ВПЛИВУ ВІЙНИ НА ДИНАМІКУ РОЗВИТКУ ЕКОНОМІКИ ТА ЕКОНОМІЧНИХ ПОКАЗНИКІВ

Діхтяр А.В.¹, Лопатін О.К.²

Національний технічний університет України «Київський політехнічний інститут
ім. Ігоря Сікорського»

¹ wdich12@gmail.com, ² lopatinalexey142@gmail.com

На жаль, у сучасному світі військові конфлікти стають невід'ємною складовою геополітичного ландшафту, породжуючи виклики та наслідки, що відчутні у багатьох аспектах глобальної економіки та життя суспільства. Війни призводять до жахливих людських жертв, трагедій, економічного занепаду та руйнування. Одним із суттєвих завдань, яке стоїть перед науковим співтовариством, є розуміння впливу війни на розвиток економіки та ключових економічних показників. Важкі випробування, що виникли внаслідок світових війн минулого сторіччя, неабияк прискорили розвиток математичних методів та моделей для аналізу та прогнозування динаміки цих процесів, включаючи лінійні, нелінійні та перехідні явища, в тому числі економічні процеси під впливом зовнішніх факторів. Метою роботи є розробка інструментарію для прогнозування та проведення порівняльного аналізу макроекономічних показників під впливом війни. Основним результатом дослідження є побудова системи підтримки прийняття рішень (СППР) для прогнозування та подальшого аналізу економічних процесів.

Ключові слова: макроекономічні процеси, регресійні моделі, рекурентні нейронні мережі, прогнозування, війна, внутрішній валовий продукт.

1. ВСТУП

Війна – це жахливе явище, що приносить смерть, руйнування та страждання мільйонам. Люди втрачають свої життя, країни зазнають великих економічних збитків. Війна порушує мирне співіснування та залишає разючий слід у психології людей, особливо в дітей. Вона спричиняє непоправної шкоди природному середовищу і сприяє зростанню глобальної напруги. Глобальні військові конфлікти мають надзвичайно вагомий вплив на світову економіку, створюючи серію складних викликів та наслідків для різних країн та галузей. Цей вплив стосується різних аспектів економіки та суспільства. Війни спричиняють економічний спад через руйнування інфраструктури, збитки виробництву та зниження споживчого попиту. Нестабільність та невизначеність під час військового конфлікту є важливими факторами, які гальмують економічний розвиток.

Населення найбільше страждає від гуманітарних катастроф і соціальних втрат, які виникають внаслідок війни. Руйнування виробництва та інфраструктури призводять до безробіття та погіршення рівня життя, що може мати довгострокові соціальні наслідки. Війна в Україні значно вплинула на економічну активність в країні, призводячи до прогнозованого спаду ВВП та інших економічних показників через воєнні обставини. Різні сектори економіки,

такі як сільське господарство, промисловість та торгівля, відчули серйозні труднощі через перерви в постачанні та зниження попиту. Регіони, які стали ареною активних бойових дій, виявилися особливо вразливими, зазнавши руйнувань і великих втрат. Національна бюджетна система також відчула складність ситуації через значний зріст витрат на потреби армії, соціальні програми та відновлення пошкоджених регіонів. Збільшення державного боргу та використання нових інструментів фінансування, таких як введення військових облігацій, свідчать про необхідність ефективного фінансового управління в умовах воєнного конфлікту.

Соціальний аспект війни неминуче впливає на зайнятість, рівень життя та еміграцію населення, ставлячи під загрозу соціальну стабільність країни. Незважаючи на це, важливо визначити стратегії для подальшого відновлення та розвитку. Заходи, такі як надання податкових пільг, приваблення інвестицій, розвиток важливих секторів економіки, а також соціальна підтримка населення, стають ключовими для подолання викликів воєнного періоду. Українське суспільство, проявляючи солідарність та патріотизм, потребує ефективного управління, стратегічного планування та співпраці з міжнародними партнерами для подальшого економічного та соціального відновлення після нашої перемоги.

2. ВИБІР ТА ОПИС МЕТОДІВ ДЛЯ АНАЛІЗУ ТА ПРОГНОЗУВАННЯ

Регресійні моделі відіграють ключову роль для аналізу взаємозв'язків між залежними та незалежними змінними. Ці моделі визначаються як фундаментальний інструмент для ретельного аналізу, що відбувається у сфері дослідження зазначених залежностей, розширюючи наше розуміння обсягу даних і надаючи можливість ефективного здійснення прогнозів на підставі цих виявлених залежностей. Важливо пам'ятати, що вибір конкретної моделі повинен бути чітко налаштованим під характеристики використовуваних даних у конкретному контексті роботи [6].

Оцінка якості прогнозів, безперечно, є ключовим етапом, і вона виконується за допомогою різноманітних критеріїв, таких як середньоквадратична похибка (MSE), середня абсолютна похибка (MAE), коефіцієнт детермінації (R-squared) та інші. Вибір конкретного критерію визначається специфікою завдання та конкретними аспектами прогнозу, які привертають увагу дослідника. Наприклад, MSE ефективно реагує на великі похибки, в той час як MAE краще враховує середні значення похибок [3].

Нейронні мережі, зокрема рекурентні та згорткові, мають здатність виявляти складні залежності в наборах даних та застосовувати їх для точного прогнозування. Ці мережі показують вражаючу ефективність в різних областях, таких як обробка тексту, зображень, аудіо та часові послідовності, забезпечуючи результативність у розв'язанні складних завдань, що вимагають аналізу великих обсягів даних.

Оцінка якості моделі включає використання критеріїв адекватності, таких як Критерій Акайке, Байєсівський інформаційний критерій (BIC) та Критерій Дарбіна-Ватсона (DW). Важливо розуміти, що вибір конкретної моделі повинен точно відповідати завданням та специфікаціям використовуваних даних. Деякі критерії можуть виявитися кращими, для обробки обширних наборів даних, інші можуть проявити високу ефективність для аналізу складних моделей [4].

Вищезазначені аспекти є ключовими для успішного аналізу та прогнозування даних, і правильне їх використання має потенціал істотно покращити якість отриманих результатів. Обґрунтованість та уважність у проведенні аналізу моделей є вирішальними для досягнення максимально ефективних результатів. Застосування різноманітних методів та критеріїв для оцінювання та порівняння моделей є важливим завданням, і їх вибір обумовлений конкретною ситуацією. Критерії якості, а також мережі та інструменти для аналізу та прогнозування

постійно еволюціонують, тому важливо слідкувати за новими тенденціями та використовувати їх для постійного удосконалення результатів. Загалом важливо підкреслити вагомість ретельного аналізу, вибору відповідних підходів та моделей для проведення аналізу та прогнозування даних [7].

3. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Основними даними для моделювання та подальшого прогнозування є щомісячні дані внутрішнього валового продукту України із січня 2002 по грудень 2021 року [1]. Допоміжними даними для побудови прогнозів виступали також й інші макроекономічні показники, такі як індекс споживчих цін, індекс промислового виробництва, ціни на газ, податкові надходження, неподаткові надходження та офіційний курс долара.

Розглянемо графік вхідних даних ВВП (рис. 1).

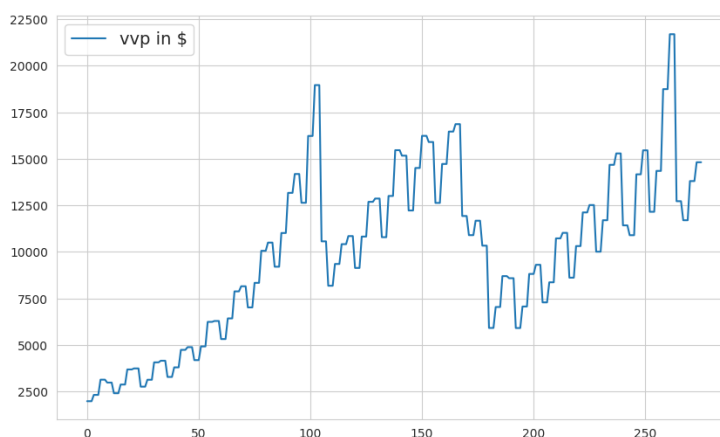


Рисунок 1. Графічне зображення даних

За допомогою системи EViews для вхідних даних були побудовано ряд авторегресійних моделей, в тому числі й вищих порядків [3]. По ходу покращення моделей, вони ускладнювалися та набували складових ендогенних регресорів і нелінійних складових, таких як тренди та періодичні функції. По ходу вдосконалення моделей, їхні параметри та статистики ставали все кращими й кращими. Найкраща модель по всім статистикам це авторегресія 25 порядку із додаванням усіх ендогенних регресорів також із лагом 25 порядку, із додаванням тренду, періодичної функції та квадратичних складових ендогенних регресорів. Графік оцінених найкращою моделлю даних (рис. 2).

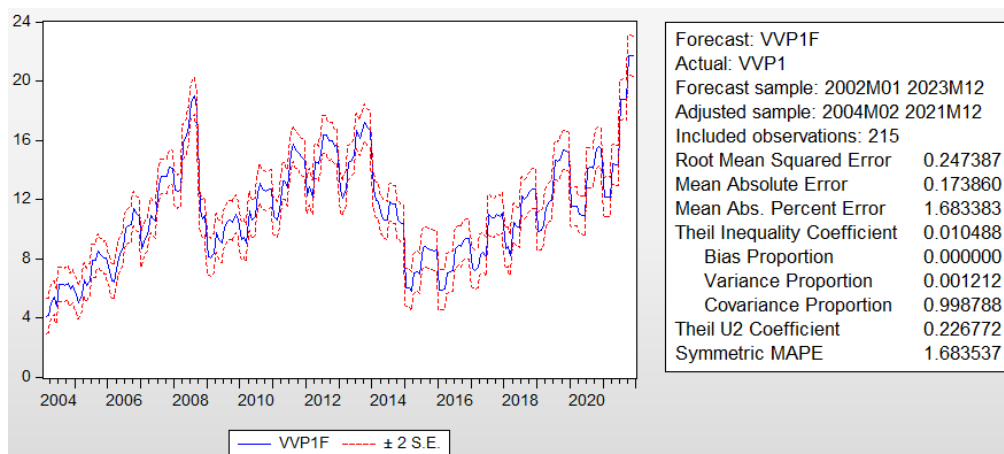


Рисунок 2. Графік оцінених найкращою моделлю даних

Як можемо бачити, характеристики однокрокового прогнозу дуже обнадіюючі.

Також, за допомогою штучних рекурентних нейронних мереж (використовуючи інструмент Google Colab), було побудовано ряд експериментальних одношарових та багатошарових нейронних мереж, використовуючи декілька видів рекурентних нейронів, таких як LSTM та GRU [5]. Не зважаючи на відносно невелику кількість прикладів для навчання, згорткові нейронні мережі показали себе не досить ефективно. Що можна легко пояснити, адже згорткові нейронні мережі мають ефективне прикладне застосування в розпізнаванні зображень та відео, рекомендаційних системах тощо. Процес навчання та статистики якості прогнозів найкращої нейронної мережі подано нижче (рис. 3).



Рисунок 3. Процес навчання та статистики якості прогнозів найкращої нейронної мережі

Порівняємо статистичні характеристики найкращих моделей авторесії та рекурентних нейронних мереж. Будемо оцінювати за наступними критеріями: коефіцієнт детермінації, критерій Дарбіна-Уотсона, сума квадратів похибок, MSE, MAE, MAPE та коефіцієнт Тейла [4] (Табл. 1).

Таблиця 1. Порівняння статистичних параметрів найкращих моделей

Тип моделі	Характеристики моделі			Характеристики прогнозу			
	R^2	$\sum e^2(k)$	DW	MSE	MAE	MAPE	U
AR(25)+REG	0,995	13	2	0,061	0,174	1,683	0,228
RNN	0,972	278		0,401	0,462	4,561	

Як можемо бачити, авторегресійна модель по всіх розглянутих показниках є кращою за модель штучної рекурентної нейронної мережі. Це легко пояснюється даними для навчання: авторегресійна модель приймає на вхід не лише власні часові данні, а ще й ендогенні данні, в

той час як нейронна мережа навчається на своїх попередніх даних, яких, до того ж, недостатньо для гарного навчання останньої. Проте, не зважаючи на недоліки, характеристики адекватності та однокрокових прогнозів обох моделей є дуже хорошими, і це дає нам підставу вважати, що прогнози цих моделей на майбутнє будуть близькими до реальності [2].

За результатами прогнозування авторегресійною моделлю можна припускати, яким було б ВВП України, якби не повномасштабне вторгнення. За 2022 рік ВВП України склав 159,124 мільярдів доларів. За спрогнозованими моделлю даними, отримуємо, що ВВП України за 2022 рік міг би бути 257,85 мільярдів доларів. Падіння ВВП за цей рік в складає 99 мільярдів доларів, або 38.3% (рис. 4).

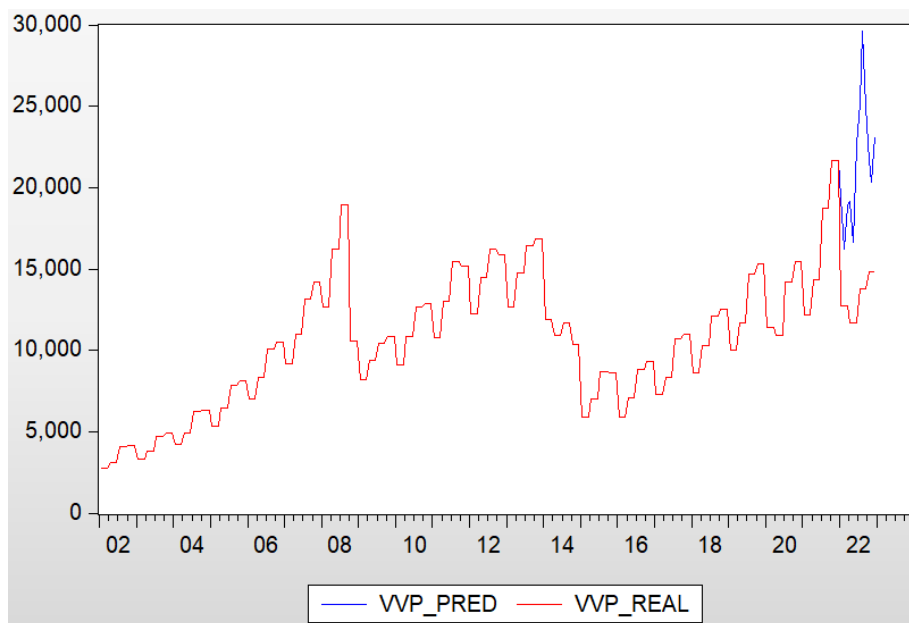


Рисунок 4. Графік розриву прогнозованого регресійною моделлю та реального ВВП

За результатами прогнозування нейронною мережею отримуємо наступне: за 2022 рік ВВП України склав 159,124 мільярдів доларів; по спрогнозованим моделлю даним, маємо, що ВВП України за 2022 рік міг скласти 225,169 мільярдів доларів. Падіння ВВП за останній квартал в складає 20,5 мільярдів доларів, або 31,7% (рис. 5).

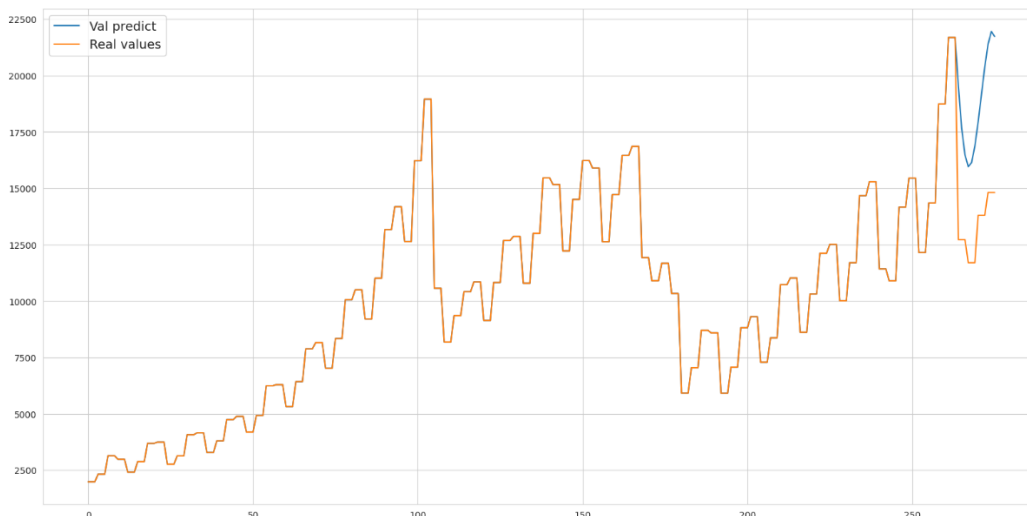


Рисунок 5. Графік розриву прогнозованого RNN та реального ВВП за 2022 рік

Аналізу окремих показників, таких як бюджетні надходження України, виявив цікаві висновки: до початку повномасштабного вторгнення, приблизно 80% податкових надходжень до державного бюджету надходило від внутрішніх джерел, в той час як зовнішні інвестиції та трансї становили лише 1,5%. Однак після першого року повномасштабного вторгнення ця динаміка змінилася: частка внутрішніх податкових надходжень знизилася до 53%, а надходження від іноземних інвестицій та трансїв зростали до 27,5%. За даними на 2023 рік, частка внутрішніх податкових надходжень в бюджет складає всього 45%, в той час як надходження від іноземних джерел становлять практично 21%.

4. ВИСНОВКИ

Це дослідження спрямоване на ретельний аналіз та вивчення впливу війни на динаміку розвитку економіки та макроекономічних показників України, у межах якого було проведено обширне вивчення різних аспектів, включаючи огляд існуючих методів моделювання та прогнозування, розгляд критеріїв якості для оцінки моделей, методи оцінювання параметрів моделі, та обробку важливих макроекономічних показників.

Були проаналізовані окремі макроекономічні показники, що дозволило виявити руйнівний вплив військового конфлікту в Україні на економічні процеси, характер наповнення бюджету та рівень міжнародної підтримки.

Дослідження також включало ряд експериментів з розробки прогнозів для різних макроекономічних показників у випадку, якби вторгнення не відбулося. Отримані результати вражаючі: розрив між реальним і прогнозованим внутрішнім валовим продуктом України становить від 66 до 99 мільярдів доларів, що відповідає 41,5% до 62% від реального ВВП за 2022 рік. У 2014 році ситуація була подібною: виявлено розрив внутрішнього валового продукту України між фактичними та прогнозованими показниками на рівні 77,5 мільярдів доларів. Це становило близько 57,6% у порівнянні з реальним ВВП за 2014 рік. Ці цифри є надзвичайно високими!

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Валовий внутрішній продукт (ВВП) України [Електронний ресурс]: Режим доступу до ресурсу: <https://index.minfin.com.ua/ua/economy/gdp/>
2. Бідюк П.І., Половцев О.В. Аналіз та моделювання економічних процесів перехідного періоду. К.: НТУ КПП, 1999. 230 с.
3. Бідюк П. І., Романенко В. Д., Тимошук О. Л. Аналіз часових рядів: навч. посіб. / ННК «Інститут прикладного системного аналізу» Національний технічний університет України «Київський політехнічний інститут», 2010. 317 с.
4. Бідюк П. І. Економетричний аналіз часових рядів. Київ: Політехніка, 2007. 250 с.
5. Бідюк П. І. Проектування комп'ютерних інформаційних систем підтримки прийняття рішень: навч. посіб. / ННК «Інститут прикладного системного аналізу» Національний технічний університет України «Київський політехнічний інститут», 2010. 340 с.
6. Ставицький А. В. Навчально-методичний комплекс з курсів «Прогнозування» та «Фінансове прогнозування». Київ: Центр учб. літ., 2006. 107 с.
7. Половцев О. В. Системний підхід до моделювання, прогнозування та управління фінансово-економічними процесами. Донецьк: Східний видавничий дім, 2009. 286 с.

СИСТЕМА ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ДЛЯ ОПТИМІЗАЦІЇ РЕКЛАМНИХ КАМПАНІЙ ПІДПРИЄМСТВА НА ОСНОВІ МЕТОДУ МОДЕЛЮВАННЯ ВПЛИВУ З ЗАЛЕЖНИМ ПРЕДСТАВЛЕННЯМ ДАНИХ

Заїка Б.Ю.¹, Терент'єв О.М.²

Національний технічний університет України «Київський політехнічний інститут
імені Ігоря Сікорського», Київ, Україна

¹ zaikabohdan3@gmail.com, ² o.terentiev@gmail.com

У сучасному світі динамічного розвитку технологій та конкурентної боротьби у сфері бізнесу, прогресивні методи аналізу даних стають крайньою необхідністю. Рекламні кампанії є невід'ємною складовою будь якого підприємства та потребують правильного підходу для раціонального використання ресурсів та уникнення негативного впливу на клієнтів. Проблемою традиційних підходів моделювання в даному контексті є фокусування на прогнозі ймовірності виконання цільової дії користувачем після комунікації з ним. Моделювання впливу, на відміну від традиційних моделей, прогнозує вплив взаємодії з користувачем на ймовірність виконання ним цільової дії. Використання такого підходу дозволяє раціональніше використовувати ресурси компанії, фокусуючи комунікацію на користувачів, взаємодія з якими матиме найбільший позитивний вплив на виконання ними цільової дії.

Ключові слова: моделювання впливу, система підтримки прийняття рішень, рекламні кампанії, кероване навчання.

1. ВСТУП

У сучасному світі великої конкуренції та постійних змін, використання даних для розуміння поведінки споживачів та ефективного впливу на них стали критичними завданнями для бізнесу. Рекламні кампанії мають величезне значення для підприємства, оскільки вони забезпечують залучення уваги клієнтів, збільшення продажів, підтримку конкурентоспроможності та комунікацію з аудиторією. Реклама дозволяє підтримувати зв'язок зі споживачами, повідомляти їх про новинки, акції, зміни та відповідати на їхні запитання. Проте нераціональне використання цього інструменту бізнесом може нести втрати у вигляді ресурсів та клієнтів, через зайві комунікації або невдалу персоналізацію реклами.

Традиційні методи класифікації прогнозують ймовірність належності користувача до класу користувачів, які зробили цільову дію після взаємодії з ними. На основі цих прогнозів часто приймаються рішення щодо взаємодії з класифікованими особами. Однак справжньою метою рекламних кампаній має бути визначення різниці у виконання цільової дії користувачем з та без взаємодії з ним. Традиційні методи класифікації не використовують інформацію стосовно контрольних груп і тому мають обмежене застосування в цьому контексті.

На відміну від них, моделювання впливу дозволяє включати контрольну групу та спрямовується на явне моделювання різниці в ймовірності результату між двома групами, тому воно набагато краще підходить для аналізу потенційних отримувачів реклами. Крім того,

моделі впливу дозволяють безпосередньо ідентифікувати користувачів, взаємодія з якими є найефективнішою [1]. Такий підхід до комунікації з клієнтами дозволяє раціонально використовувати ресурси та уникати контакту з клієнтами, яких реклама може відштовхнути від виконання цільової дії. Саме в цьому і полягає актуальність використання моделювання впливу для прийняття рішень стосовно рекламних кампаній підприємства: можливість спрогнозувати як взаємодія вплине на користувача, щоб уникнути небажаної комунікації та розуміти з якими користувачами вигідніше взаємодіяти для досягнення цілі.

Метою роботи є розробка системи підтримки прийняття рішень (СППР) [2] на основі методу моделювання впливу з залежним представленням даних (DDR – Dependent Data Representation) [3], що стане невід’ємним інструментом аналітиків та бізнес користувачів підприємства для швидкого аналізу та створення моделей впливу для проведення оперативних та персоналізованих рекламних кампаній.

2. МЕТОДИКА ПОРІВНЯЛЬНОГО АНАЛІЗУ

Для порівняння методу залежного представлення даних з іншими в першу чергу треба визначити методику порівняння, що дозволить об’єктивно оцінити кожен модель та визначити їх слабкі та сильні сторони. Методика включає наступні кроки:

1. Вибір методів для порівняння
2. Визначення метрик ефективності
3. Збір та підготовка даних
4. Навчання, тестування та порівняння моделей
5. Висновки

Для порівняльного аналізу використано методи моделювання впливу одного учня (S-Learner) [4, 5] та трансформації змінної класів (Class Variable Transformation approach або Z-Learner) [1, 5]. Оскільки всі методи, що порівнюються, є методами мета-навчання (Meta-learners), які використовують звичайні моделі класифікації з певним принципом подачі на вхід даних, то для справедливості порівняння всі методи будуть порівнюватись на одному виді моделей класифікації - XGBoost класифікаторі.

В рамках аналізу зосереджено увагу на метриках оцінки ефективності моделі таких як стовпчикова діаграма впливу за перцентильними рангами (uplift by percentile barchart), вплив на топ k% (uplift at top k%, для поточного аналізу розраховано на топ 30%), середньозважений вплив (weighted average uplift), крива Квіні (Qini curve) та коефіцієнт Квіні (Qini Coefficient) [5].

Стовпчикова діаграма впливу за перцентильними рангами будується за наступним алгоритмом:

1. Користувачі сортуються за спаданням спрогнозованого значення впливу
2. Відсортовані дані діляться на перцентилі
3. В кожному перцентилі окремо оцінюється вплив як різниця між середнім значенням цільової змінної в цільовій та контрольній групах.

Для побудови кривої Квіні необхідно відсортувати дані за спаданням спрогнозованого значення впливу та побудувати графік за наступною формулою [6]:

$$Qini\ curve(t) = Y_t^T - \frac{Y_t^C N_t^T}{N_t^C}, \text{ де}$$

- t – кількість включених в комунікацію користувачів;
- Y_t^T, Y_t^C – кількість виконаних цільових дій в цільовій (T – Target) та контрольній (C – Control) групі відповідно;
- N_t^T, N_t^C – кількість користувачів в цільовій та контрольній групі відповідно.

Для розрахунку впливу на топ k% необхідно скористатись наступною формулою на топ k% користувачах за спрогнозованим значенням впливу [7]:

$$Uplift\ at\ top\ k\% = \frac{Y_{top\ k\%}^T}{N_{top\ k\%}^T} - \frac{Y_{top\ k\%}^C}{N_{top\ k\%}^C}, \text{ де}$$

- $Y_{top\ k\%}^T, Y_{top\ k\%}^C$ - кількість виконаних цільових дій в цільовій та контрольній групі відповідно серед топ k% користувачів за спрогнозованим значенням впливу;
- $N_{top\ k\%}^T, N_{top\ k\%}^C$ - кількість користувачів в цільовій та контрольній групі відповідно серед топ k% користувачів за спрогнозованим значенням впливу;

Середньозважений вплив є числовим відображенням інформації зі стовпчикової діаграми впливу за перцентильними рангами та розраховується за наступною формулою:

$$Weighted\ average\ uplift = \frac{1}{\sum_{i=1}^{10} N_i^T} \sum_{i=1}^{10} N_i^T uplift_i, \text{ де}$$

- N_i^T – розмір цільової групи в i-ому перцентилі;
- $uplift_i$ – вплив в i-ому перцентилі.

Коефіцієнт Квіні є числовим відображенням інформації з кривої Квіні розраховується за наступною формулою:

$$Qini\ coefficient = \frac{S_{model}}{S_{ideal}}, \text{ де}$$

- S_{model} – площа між кривими Квіні побудованої та випадкової моделей;
- S_{ideal} – площа між кривими Квіні ідеальної та випадкової моделей;

Для тренування та порівняння моделей взято анонімізований набір даних телекомунікаційної компанії [8], який містить в собі наступні ознаки:

- id – ідентифікатор користувача;
- X_1, \dots, X_{50} – 50 анонімізованих ознак, які описують користувача;
- $treatment_group$ – до якої групи відноситься користувач (цільова чи контрольна);
- $conversion$ – чи зробив користувач цільову дію.

Дані розбито випадковим чином на тренувальну (розміром 480 тис. рядків) та тестову (розміром 60 тис. рядків) вибірки. Розподіл на контрольну та цільову групи складає 50% та 20% користувачів виконали цільову дію. При цьому в цільовій групі 22.9% користувачів виконали цільову дію, а в контрольній 17.94%.

Для аналізу даних використано коефіцієнт рангової кореляції Спірмена, який на відміну від коефіцієнта кореляції Пірсона шукає монотонну залежність між даними, а не тільки лінійну [9]. В тренувальних даних було помічено 16 пар змінних, які мали високу кореляції між собою (більше 0.7). Для кожної пари було визначено змінну, що має меншу за модулем кореляцію з цільовою змінною, та видалено її.

Після внесення аналогічних змін у тестові дані можна переходити до тренування та порівняння методів.

3. РЕЗУЛЬТАТИ ПОРІВНЯЛЬНОГО АНАЛІЗУ

З результатами порівняльного аналізу методів моделювання впливу можна ознайомитись в таблиці 1 та на рисунках 1, 2.

Таблиця 1. Табличні результати порівняльного аналізу

Назва методу	Вплив на топ 30%	Середньозважений вплив	Коефіцієнт Квіні
S-Learner	0,18825	0,04996	0,21522
Z-Learner	0,18951	0,04989	0,20253
DDR(feature='control')	0,20224	0,04926	0,22559
DDR(feature='treatment')	0,20367	0,05023	0,22279

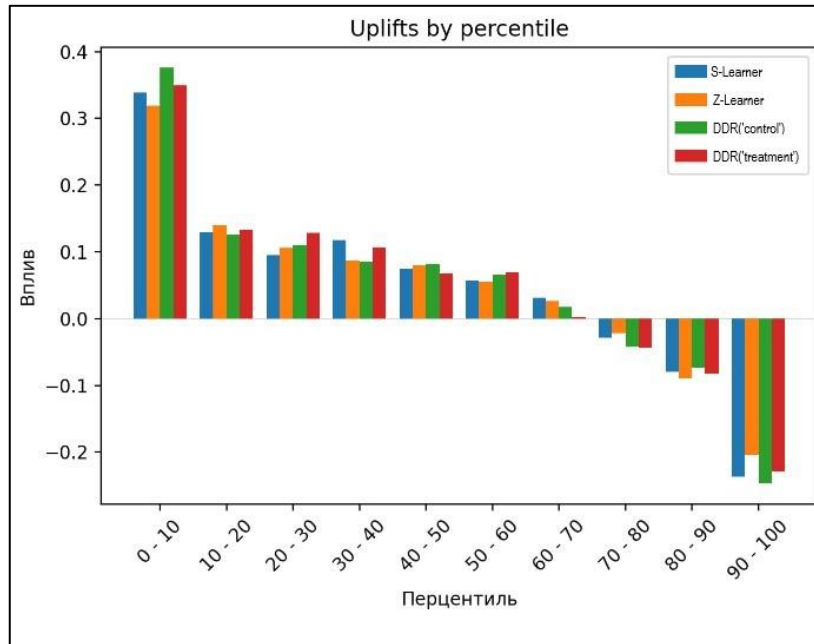


Рисунок 1. Графік впливів побудованих моделей за перцентильними рангами

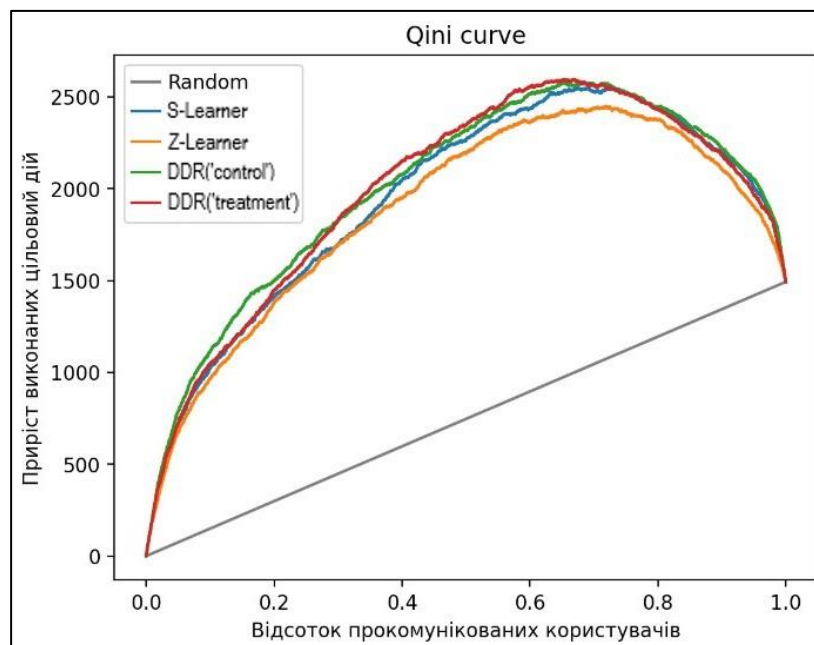


Рисунок 2. Графік кривої Квіні для побудованих моделей

Отримано непогані результати для методу S-Learner, проте при аналізі стовпчикової діаграми впливу за перцентильними рангами можна помітити, що значення впливу 30-го перцентилля нижче 40го. Тобто модель має проблеми з пріоритезацією користувачів за впливом в рамках 30-го та 40-го перцентилів.

Z-Learner трохи краще визначає топ 30% найкращих користувачів для комунікації і немає проблеми з перцентильями, які були у S-Learner. Проте гірші значення середньозваженого впливу та коефіцієнту Квіні свідчать про те, що на всьому обсязі тестових даних Z-Learner гірше справляється з оцінкою впливу комунікації на користувачів.

Метод DDR з використанням результатів моделі контрольної групи в моделі цільової групи має найкращі значення коефіцієнту Квіні та впливу на топ 30% користувачів серед розглянутих методів. Також на стовпчиковій діаграмі можна помітити що впливи 10-го та 30-го перцентилів збільшилися, а 20-го трохи зменшився в порівнянні з минулими методами. Можна прийти до висновку, що загалом модель показала кращі результати, особливо на топ 30% користувачів.

Метод DDR з використанням результатів моделі цільової групи в моделі контрольної групи отримав найкращі значення впливу на топ 30% користувачів та середньозваженого впливу, а також друге найвище значення коефіцієнту Квіні. Це робить його найкращим методом серед усіх розглянутих для поточного набору даних.

Порівняльний аналіз показав, що обидва варіанти обраного методу показали кращі результати за інші розглянуті методи. Опираючись на результати, можна прийти до висновку, що якщо бюджет рекламної кампанії розрахований до 30% користувачів, то для розглянутого набору даних краще використовувати метод DDR з використанням результатів моделі контрольної групи в моделі цільової групи. У всіх інших випадках, краще використовувати метод DDR з використанням результатів моделі цільової групи в моделі контрольної групи для розглянутого набору даних.

4. ОГЛЯД РЕАЛІЗОВАНОЇ СППР

При створенні до СППР було висунуто наступні умови: інтерактивність - система повинна мати інтуїтивно зрозумілий графічний інтерфейс користувача (GUI), який дозволяє користувачам взаємодіяти з системою за допомогою кнопок, меню, графічних елементів тощо; доступність – система повинна бути легкодоступною для користувачів, чим менше користувачу потрібно зробити для початку користування продуктом, тим більше ймовірності привернути його увагу; повнота – в системі має бути можливість провести повноцінний аналіз впливу: починаючи від завантаження даних для тренування моделі, закінчуючи використанням нових даних на створеній моделі та вивантаженням отриманих результатів.

Блок-схема алгоритму роботи з СППР зображена на рисунку 3.

Реалізацію вищезгаданих етапів роботи з СППР зображено на рисунках 4, 5, 6 та 7. На рисунку 4 зображено завантаження тренувальної та тестової вибірок. На рисунку 5 зображено таблиці для аналізу кореляцій Спірмена на тренувальних даних. Рисунок 6 відображає інтерфейс для тренування та порівняння моделей впливу на тренувальних та тестових даних. На рисунку 7 зображено використання обраної натренованої моделі на нових даних з можливістю вивантаження отриманих результатів для подальшого планування рекламної кампанії на їх основі. За наведеними рисунками можна побачити, що інтерфейс є інтуїтивно зрозумілим і СППР дозволяє провести повноцінний аналіз впливу, тобто вона відповідає висунутим до неї вимогам.

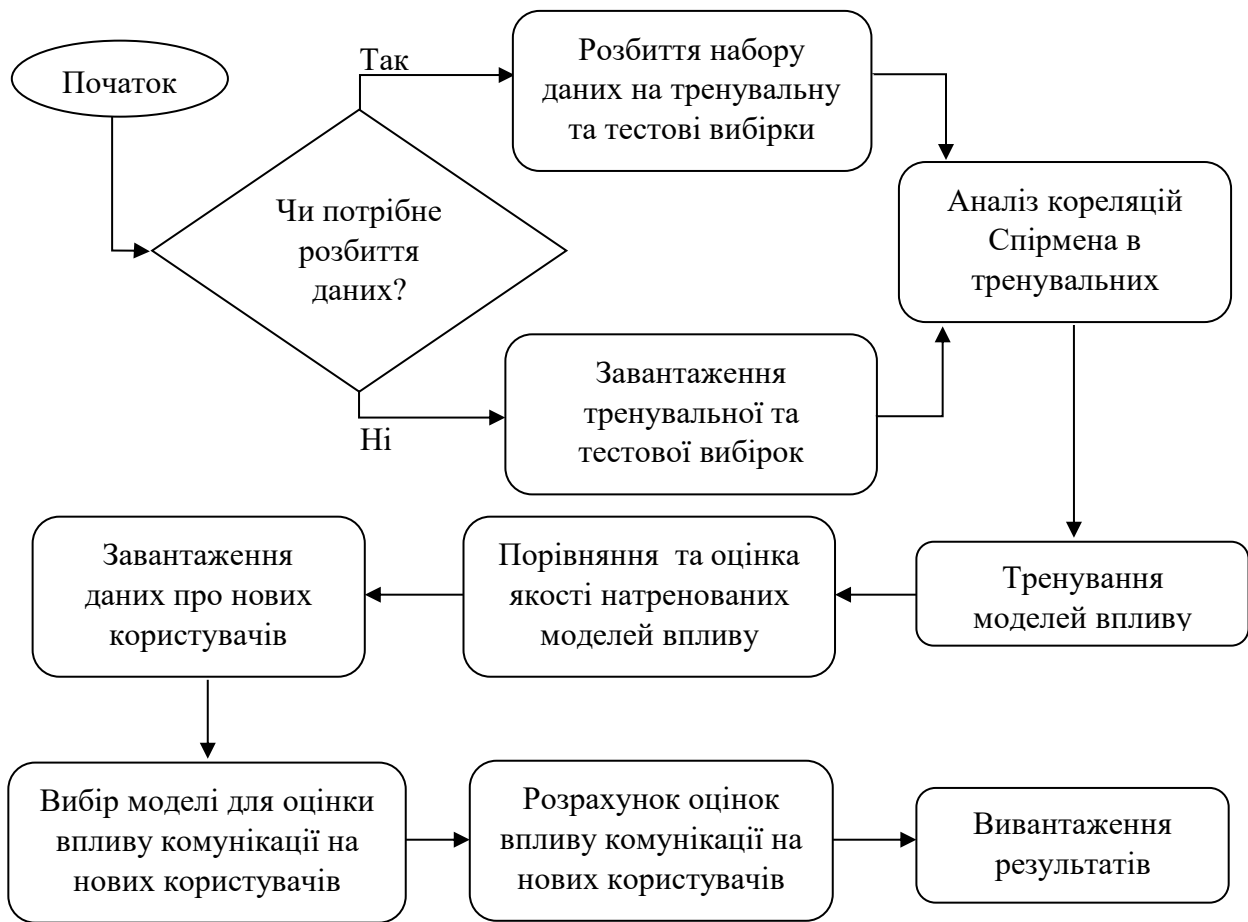


Рисунок 3. Блок-схема алгоритму роботи з СППР

1.1. Вибір способу розбиття даних на тренувальні та тестові

Спосіб розбиття даних на тренувальні та тестові

- Завантажити один файл з даними та розбити їх на тренувальні та тестові в програмі
- Завантажити два окремих файли з тренувальними та тестувальними даними відповідно

1.2. Вибір файли з даними

Вибір файли з тренувальними даними

Drag and drop file here
Limit 200MB per file • XLSX, CSV, PARQUET

upload_mega_train.parquet (181.9MB)

Вибір файли з тестовими даними

Drag and drop file here
Limit 200MB per file • XLSX, CSV, PARQUET

upload_mega_valid.parquet (26.6MB)

Decision Support System for Uplift Modeling with Dependent Data Representation method

1. Огляд завантажених даних

Тренувальні дані
Розмір даних: (488889, 52)
Даніс: (488889, 52)

id	treatment_group	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	
0	504,629	1	0.4566	0.4619	0.3195	0.4143	0.3852	0.4355	0.5527	0.4042	0
1	201,079	1	0.4802	0.5598	0.465	0.4637	0.4222	0.4167	0.4202	0.3985	0.4
2	171,235	1	0.7382	0.5687	0.3886	0.3409	0.3056	0.5361	0.3836	0.4462	0.3
3	85	1	0.5225	0.4588	0.5109	0.5154	0.4555	0.4403	0.5145	0.4026	0
4	308,084	0	0.4324	0.4342	0.4711	0.4683	0.4216	0.6336	0.5109	0.4218	0.4
5	364,223	0	0.4866	0.5345	0.483	0.6356	0.5504	0.3008	0.5855	0.5226	0.3
6	49,834	0	0.5956	0.4302	0.5643	0.4336	0.456	0.4442	0.3366	0.3337	0.3
7	237,522	1	0.4816	0.5666	0.414	0.4564	0.6255	0.6174	0.5856	0.5078	0.5
8	432,298	1	0.4038	0.4486	0.496	0.2696	0.5638	0.6241	0.5834	0.4396	0.5
9	469,301	1	0.3953	0.4221	0.3783	0.5709	0.3923	0.5229	0.6038	0.4374	0.4

Статистичні характеристики даних:

id	treatment_group	X_1	X_2	X_3	X_4	X_5	X_6	X_7			
0	509,880	0	0.5366	0.3736	0.4394	0.4928	0.4956	0.3111	0.4019	0.4278	0.4
1	43,675	0	0.6508	0.2906	0.4621	0.4097	0.5066	0.3582	0.4586	0.51	0.5
2	2,656	0	0.3898	0.4907	0.4305	0.4755	0.4869	0.7077	0.6631	0.3998	0.6
3	422,126	0	0.7276	0.4561	0.6537	0.357	0.3452	0.4777	0.3892	0.4856	0.4
4	4,235	1	0.6109	0.658	0.3661	0.4246	0.4891	0.5182	0.5085	0.4659	0.5
5	475,010	0	0.6076	0.4235	0.5084	0.6343	0.5375	0.5363	0.3954	0.502	0.5
6	211,345	1	0.4098	0.4786	0.3653	0.4012	0.6553	0.5533	0.5291	0.5081	0.5
7	132,555	1	0.4874	0.4411	0.5887	0.5795	0.4664	0.5284	0.5564	0.4742	0.5
8	553,104	1	0.3863	0.4208	0.3933	0.4503	0.6227	0.3705	0.6169	0.5587	0.4
9	563,114	1	0.5436	0.554	0.3708	0.5934	0.5309	0.3357	0.5317	0.5406	0

Статистичні характеристики даних:

id	treatment_group	X_1	X_2	X_3	X_4	X_5	X_6	X_7			
0	509,880	0	0.5366	0.3736	0.4394	0.4928	0.4956	0.3111	0.4019	0.4278	0.4
1	43,675	0	0.6508	0.2906	0.4621	0.4097	0.5066	0.3582	0.4586	0.51	0.5
2	2,656	0	0.3898	0.4907	0.4305	0.4755	0.4869	0.7077	0.6631	0.3998	0.6
3	422,126	0	0.7276	0.4561	0.6537	0.357	0.3452	0.4777	0.3892	0.4856	0.4
4	4,235	1	0.6109	0.658	0.3661	0.4246	0.4891	0.5182	0.5085	0.4659	0.5
5	475,010	0	0.6076	0.4235	0.5084	0.6343	0.5375	0.5363	0.3954	0.502	0.5
6	211,345	1	0.4098	0.4786	0.3653	0.4012	0.6553	0.5533	0.5291	0.5081	0.5
7	132,555	1	0.4874	0.4411	0.5887	0.5795	0.4664	0.5284	0.5564	0.4742	0.5
8	553,104	1	0.3863	0.4208	0.3933	0.4503	0.6227	0.3705	0.6169	0.5587	0.4
9	563,114	1	0.5436	0.554	0.3708	0.5934	0.5309	0.3357	0.5317	0.5406	0

Рисунок 4. Завантаження тренувальної та тестових вибірок в СППР

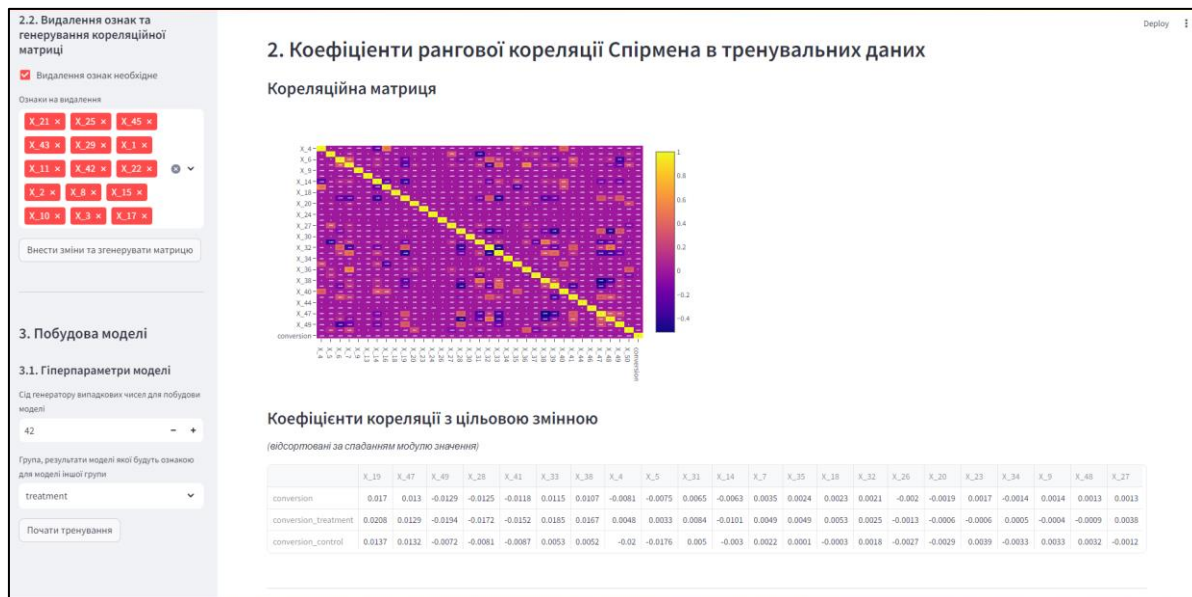


Рисунок 5. Відображення кореляційної матриці Спірмена та можливість видалення ознак

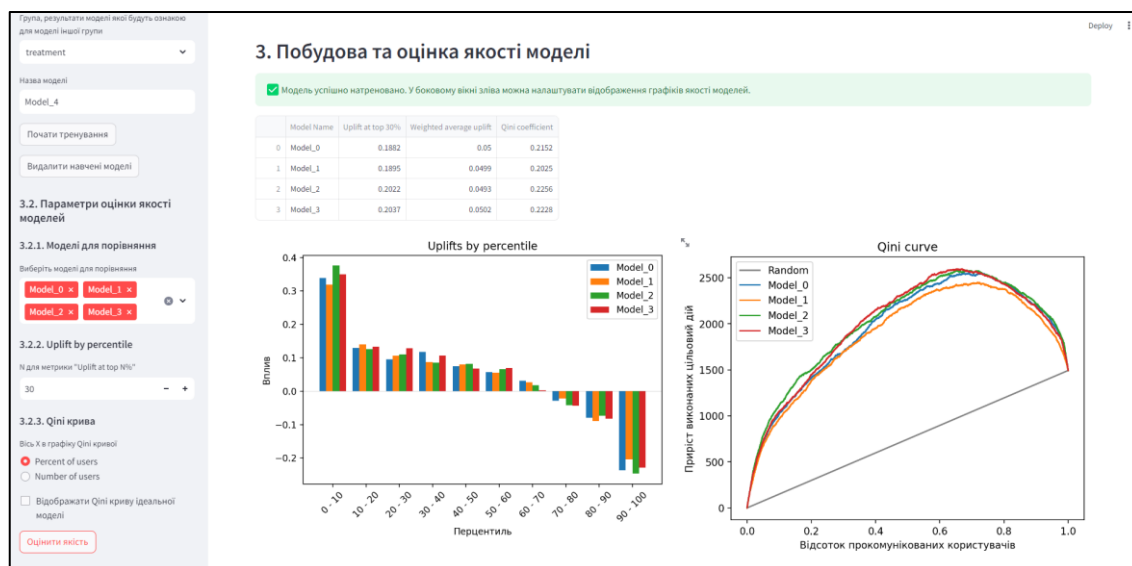


Рисунок 6. Побудова та порівняння моделей впливу

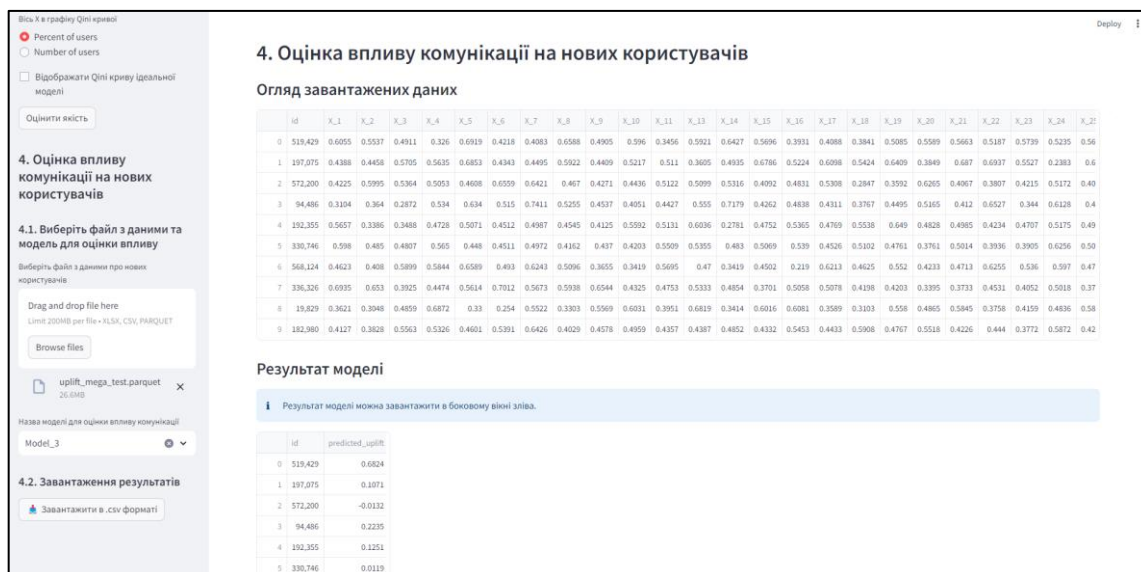


Рисунок 7. Використання побудованої моделі на новому наборі даних та можливість вивантаження отриманих результатів

5. ВИСНОВКИ

В рамках роботи було проведено порівняльний аналіз методу моделювання впливу з залежним представленням даних з іншими та розроблено СППР на його основі.

Порівняльний аналіз показав, що для розглянутого набору даних обидва варіанти методу з залежним представленням даних виявились найкращими, в залежності від потреб підприємства.

Під час розробки СППР було висунуто вимоги до неї та обрано інструменти для створення. Приклад роботи з СППР та огляд інтерфейсу показують, що СППР відповідає висунутим до неї вимогам.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Jaskowski, Maciej; Jaroszewicz, Szymon. Uplift modeling for clinical trial data. In: ICML Workshop on Clinical Data Analysis. 2012. p. 79-95. URL: https://people.cs.pitt.edu/~milos/icml_clinicaldata_2012/Papers/Oral_Jaroszewicz_ICML_Clinical_2012.pdf

2. Бідюк П.І. Проектування комп'ютерних інформаційних систем підтримки прийняття рішень: навч. посіб. / Бідюк П.І., Коршевнюк Л.О. – К.: ННК "ПСА" НТУУ "КПІ", 2010. – 340 с.

3. Betlei, Artem; Diemert, Eustache; Amini, Massih-Reza. Uplift prediction with dependent feature representation in imbalanced treatment and control conditions. In: Neural Information Processing: 25th International Conference, ICONIP 2018, Siem Reap, Cambodia, December 13–16, 2018, Proceedings, Part V 25. Springer International Publishing, 2018. p. 47-57. URL: https://bitlater.github.io/files/iconip_paper.pdf

4. Lo, Victor SY. The true lift model: a novel data mining approach to response modeling in database marketing. ACM SIGKDD Explorations Newsletter, 2002, 4.2: 78-86. URL: https://www.kdd.org/exploration_files/lo.pdf

5. Bon, Michaël; Feutry, Clément; Meftah, Sara. An in-depth benchmark study of the CATE estimation problem: experimental framework, metrics and models Version. URL: https://www.sjscience.org/files/papers/809/CoScience_809.pdf

6. Radcliffe, Nicholas. Using control groups to target on predicted lift: Building and assessing uplift model. *Direct Marketing Analytics Journal*, 2007, 14-21.
7. Radcliffe, Nicholas J.; Surry, Patrick D. Real-world uplift modelling with significance-based uplift trees. *White Paper TR-2011-1*, Stochastic Solutions, 2011, 1-33.
8. Відкритий датасет Megafon: веб-сайт. URL: https://www.uplift-modeling.com/en/latest/api/datasets/fetch_megafon.html
9. De Winter, Joost CF; Gosling, Samuel D.; Potter, Jeff. Comparing the Pearson and Spearman correlation coefficients across distributions and sample sizes: A tutorial using simulations and empirical data. *Psychological methods*, 2016, 21.3: 273. URL:https://www.researchgate.net/profile/Joost-De-Winter/publication/307902372_Comparing_the_pearson_and_spearman_correlation_coefficients_a_cross_distributions_and_sample_sizes_A_tutorial_using_simulations_and_empirical_data/links/63b-aa5d5c3c99660ebdc3f60/Comparing-the-pearson-and-spearman-correlation-coefficients-across-distributions-and-sample-sizes-A-tutorial-using-simulations-and-empirical-data.pdf

МОДЕЛЬ УПРАВЛІННЯ РЕСУРСАМИ ГЕТЕРОГЕННИХ БАЗ ДАНИХ В ХМАРНОМУ СЕРЕДОВИЩІ

Зайцев О.В.¹, Мухін В.Є.

Національний технічний університет України «Київський політехнічний інститут
ім. Ігоря Сікорського»

¹ zaytsev961st@gmail.com

Гетерогенні бази даних представляють складну систему з великою кількістю параметрів. Для ефективного управління процесами обробки інформації важливо оцінювати та передбачати витрати ресурсів на ці процеси, що вимагає урахування багатьох зазначених параметрів. Була розроблена модифікація моделі для оцінювання витрат часу запиту у розподілених гетерогенних базах даних, що дозволяє враховувати поточну кількість запитів у системі, а також метод пошуку оптимальної конфігурації системи для мінімізації цих витрат.

Ключові слова: гетерогенні бази даних, оцінювання часу запиту, конфігурація вузлів, хмарне середовище.

1. ВСТУП

Обсяг інформації, що вимагає постійного та швидкого доступу зростає. Це призвело до використання розподілених баз даних для забезпечення ефективного доступу. Однак для ефективної роботи таких баз даних необхідно постійно оцінювати використані ресурси, зокрема час, необхідний для обробки запитів і даних [1]. Таким чином, виникає потреба у розробці різних моделей для оцінки витрат цих ресурсів, зокрема часу, який необхідний для обробки запитів та даних. Прогнозування часу обробки запитів є ключовою проблемою, оскільки від надійних прогнозів залежить ефективне планування робочих процесів та розподіл ресурсів системи [1].

У даній роботі розглядається модель, яка чисельно оцінює витрати часу на обробку запитів в розподілених гетерогенних базах даних у хмарних середовищах, забезпечуючи ефективне управління цими системами. Для покращення результатів дослідження розглядається модифікована модель, що враховує додаткові параметри системи – навантаження системи та ефективність кожного вузла розподіленої бази даних.

2. ОСНОВНИЙ ЗМІСТ

Архітектура системи керування гетерогенними розподіленими базами даних містить 2 основні компоненти, які є необхідними для роботи системи.

Модуль керування відповідає за надання доступу до системи запитами верифікує запити, створює план виконання, надсилає запити до баз даних та обробляє відповіді.

Модуль сховища містить бази даних, файлові системи та інші джерела.

Користувач створює запити, які авторизуються модулем керування, який далі розподіляє запити до джерел даних, обробляє відповіді, та вертає результат користувачу.

Пропонується підключати інші модулі у якості сторонніх сервісів з різних областей: федеративні системи, які дозволяють користувачам подавати запити через єдиний інтерфейс, не потребуючи детального розуміння реалізації та розташування компонентів; централізований інтерфейс управління для моніторингу, управління та оптимізації ресурсів

гетерогенних баз даних; оптимізатор запитів, що аналізує запити до баз даних та визначає оптимальний спосіб їх виконання, враховуючи розташування даних і поточне навантаження на систему; інструменти моніторингу і аналітики для відстеження стану ресурсів, продуктивності системи та виявлення можливих проблем; а також механізми автоматичного масштабування, які автоматично регулюють ресурси залежно від поточного навантаження.

У [2, 3] розглядається система обробки запитів в розподіленій системі з використанням гетерогенних розподілених баз даних, що задається такою функцією:

$$QPS(t, n) = f(SN, SC, SP, SPDB, SPCN), \text{ де}$$

SN – множина вузлів розподіленої системи обробки даних;

SC – множина зв'язків між вузлами розподіленої системи обробки даних;

SP – множина параметрів розподіленої системи обробки даних;

SPDB – множина параметрів бази даних;

SPCN – множина параметрів вузлів керування розподіленою системою обробки даних;

t – одиниця часу;

n – кількість вузлів системи в даний момент часу.

Множина вузлів розподіленої системи обробки даних визначається як функція кількох параметрів [2, 3]:

$$SN(t, n) = fN(SCN, SDBN, SEIN, SRN), \text{ де}$$

SCN – множина вузлів, в яких встановлено програмне забезпечення для керування розподіленою системою обробки даних;

SDBN – множина вузлів, в яких встановлено програмне забезпечення для гетерогенних розподілених баз даних;

SEIN – множина проміжних вузлів, які виконують функцію маршрутизації пакетів;

SRN – множина вузлів, з яких надходять запити до розподіленої системи обробки даних.

Модель, в якій множина параметрів розподіленої системи обробки даних описана функцією [2, 3]:

$$SP(t, n) = fp(P, V, R, DB, T_{db}, T_c), \text{ де}$$

P – продуктивність вузла обробки даних в розподіленій системі обробки даних;

V – швидкість передачі даних по каналу зв'язку в розподіленій системі обробки даних;

R – надійність вузла розподіленої системи обробки даних;

DB – індикатор присутності в вузлі гетерогенної розподіленої бази даних; T_{db} — час обробки пакету – запиту в вузлі розподіленій системі обробки даних в якому знаходиться система управління базою даних;

T_c – час обробки запиту вузлом керування розподіленої системи обробки даних.

Запропоновано модифікацію моделі:

Вводимо додатковий параметр L – поточну кількість запитів користувача в системі за одиницю часу.

$$SP(t, n) = fp(P, V, R, DB, T_{db}, T_c, L)$$

Множина доступних машин, що можуть бути використані при побудові у системі складається з:

$$U_{db} = (T_{req_cn}, T_{req_dbn}, R_{max}, C),$$

де T_{req_cn} – час обробки запиту машиною, якщо виконує роль вузла керування.

T_{req_dbn} – час обробки запиту машиною, якщо виконує роль вузла сховища даних

R_{max} – пропускна здатність вузла (кількість запитів, що вузол може обробити за одиницю часу)

C – ціна оренди вузла в хмарного провайдера.

Якщо вузол є перевантаженим, тобто $L_{node} > L_{jmax}$, то запит чекає на обробку в черзі. Вважаємо що запити розподіляються між вузлами одного типу циклічно.

Час обробки запиту користувача у системі:

$$T(L) = T_c(L) + T_{db}(L) + T_{transport}, \text{ де}$$

$$T_c(L_{node}, L_{max}, T_{req}) = T_{c_i} * \max\{1, L_{node} - L_{imax}\}$$

$$T_{db}(L_{node}, L_{max}, T_{req}) = T_{db_j} * \max\{1, L_{node} - L_{jmax}\}$$

$$i = \min\left(k \mid \sum_{l=1}^k R_l \geq (i \bmod L_{total})\right), R_{total} = \sum_c n_i L_i$$

$$j = \min\left(k \mid \sum_{l=1}^k R_l \geq (j \bmod L_{total})\right), R_{total} = \sum_{db} m_j L_j$$

$$T_{transport} = \frac{S * N}{V},$$

де S – середній розмір пакету, V – швидкість передачі даних в мережі, N – кількість пересилань даних.

Ціна оренди за період $C_{total} = \sum_c n_i C_i + \sum_{db} m_j C_j$

Задача полягає у знаходженні кількості вузлів кожного типу для мінімізації середніх витрат часу на запит T_r , за обмежень бюджету $C_{total} \leq C_{max}$ та заданої навантаженості системи L запитами користувачів.

3. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Було проведено низку експериментів. Доступні для вибору вузли наведено в таблиці 1.

Таблиця 1. Вузли для формування конфігурацій

Доступні машини	Час обробки запиту керуючим вузлом, с	Час обробки запиту вузлом бази даних, с	Максимальна кількість зв'язків	Вартість, \$
Тип 1	0,005	0,02	50	200
Тип 2	0,003	0,015	40	250
Тип 3	0,009	0,025	70	180

Системні параметри:

- Кількість проміжних вузлів системи: 10;
- Середня швидкість мережі, що з'єднує вузли: 10 мб/с;
- Середній розмір пакету даних: 500 КВ;

Знайдені конфігурації та часові витрати на запит можна побачити у таблиці 2.

Таблиця 2. Результати експериментів

Кількість запитів користувачів	Бюджет, \$	Знайдена конфігурація системи – кількість машин типу 1, 2, 3				Очікуваний час виконання, мс	Грошові витрати, \$
		CN	5	1	2		
2000	1000	CN	5	1	2	32,87	860
		DBN	4	1	2		
10000	1000	CN	1	3	4	35,51	990
		DBN	0	3	5		
10000	1500	CN	1	2	3	35,39	1100
		DBN	2	2	5		
500	1500	CN	1	1	0	32,12	180
		DBN	1	1	0		

Бачимо, що при збільшенні навантаження на систему з 2000 до 10000 запитів у вузлах утворюються черги, витрати часу на запит зростають, також підійшли до межі бюджету та більше не можемо додавати нові вузли. Конфігурація змінилась, за попередньої конфігурації середній час на запит склав 37,11 мс, отже зміна конфігурації пішла на користь.

За рахунок збільшення бюджету до 1500\$ змогли додати нові вузли, за рахунок цього витрати часу трохи зменшились.

Після зменшення навантаження на систему, нехай в системі 500 користувачів надіслали запити, мережа сильно спростилась, оскільки навантаження впало, більше нема необхідності в великій кількості серверів.

4. ВИСНОВКИ

Гетерогенні розподілені бази даних в хмарному середовищі забезпечують ряд переваг, що роблять їх привабливими для великої кількості застосувань. По-перше, такий підхід дозволяє об'єднувати різноманітні джерела даних, що може бути надзвичайно корисним для підтримки різноманітних бізнес-процесів та додатків. Він сприяє ефективному використанню різноманітних технологій та систем у хмарному середовищі, таких як обчислення, зберігання та обробка даних. Крім того, гетерогенність дозволяє оптимізувати розподілені ресурси для різних видів завдань та додатків, забезпечуючи гнучкість та масштабованість.

Було запропоновано архітектуру системи управління ресурсами гетерогенних баз даних в хмарному середовищі, а також модифікацію моделі для оцінювання витрат часу на запит для цієї системи та пошуку оптимальної конфігурації системи для оптимізації цих витрат, проведено експерименти та знайдено оптимальну конфігурацію за заданих параметрів.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Özsu, M. T. and Valduriez, P. (2020). Principles of distributed database systems. Springer, 4th edition. DOI: 10.1007/978-3-030-26253-2.

2. Корнага Я. І. Моделі та методи організації та управління гетерогенними розподіленими базами даних з динамічною структурою на основі мережецентричного підходу : дис. докт. техн. наук : 05.13.06 / Корнага Ярослав Ігорович – Київ, 2020. – 328 с. 11.

3. V. Mukhin, Y. Kornaga, V. Bondarenko, V. Zavgorodnii, O. Herasymenko and O. Sholokhov, "Mathematical Model for Heterogeneous Databases Parameters Estimation in Distributed Systems with Dynamic Structure," 2020 IEEE 2nd International Conference on Advanced Trends in Information Theory (ATIT), 2020, pp. 158-161, doi: 10.1109/ATIT50783.2020.9349331.

СИСТЕМА ПРОГНОЗУВАННЯ МЕТЕОРОЛОГІЧНИХ УМОВ НА ОСНОВІ МЕТОДІВ АНАЛІЗУ ДАНИХ ТА ШТУЧНОГО ІНТЕЛЕКТУ

Іванійчук А.П.¹, Гуськова В.Г.

Національний технічний університет України «Київський політехнічний інститут
ім. Ігоря Сікорського»

¹ arsen.ivaniichuk@gmail.com

Штучний інтелект відкриває перед нами широкі можливості у покращенні прогнозування погодних умов, роблячи цей процес більш точним і надійним. Метою роботи є аналіз існуючих підходів до прогнозування метеорологічних умов за допомогою методів аналізу даних та штучного інтелекту. У результаті дослідження була розроблена система прогнозування метеорологічних умов, яка ґрунтується на використанні моделі штучного інтелекту і може бути використана для забезпечення точних та актуальних даних про погоду, що важливо для різних галузей, включаючи сільське господарство, транспорт та безпеку громадян. У роботі було використано теоретичні та емпіричні методи дослідження.

Ключові слова: прогнозування погоди, аналіз метеорологічних даних, рекурентні нейронні мережі, LSTM.

1. ВСТУП

З використанням штучного інтелекту для прогнозування метеорологічних умов відкриваються перспективи покращення існуючих підходів до розв'язання даної проблеми. Ця нова парадигма дозволяє підвищити точність і надійність погодних прогнозів, що має вагомий науковий та практичний цінність у численних галузях, включаючи аграрну сферу, транспорт та інші.

Використання штучного інтелекту дозволяє здійснювати обробку великого обсягу метеорологічних даних та проводити складний аналіз, урахувавши різноманітні фактори, що впливають на погодні явища. Цей підхід дозволяє отримувати результати, які є точнішими та передбачуванішими.

Для досягнення максимальної ефективності використання можливостей штучного інтелекту в погодному прогнозуванні, є доцільним створення системи, заснованої на ефективній моделі. У рамках наших досліджень було розроблено систему прогнозування метеорологічних умов з використанням моделі Long Short-Term Memory або LSTM, що дозволяє враховувати динаміку погодних явищ з високою точністю та прогнозувати їх на майбутнє з великою достовірністю.

Крім того, наша система використовує технологію REST API, що забезпечує її зручність у використанні та інтеграції з іншими додатками і системами.

2. ОГЛЯД ІСНУЮЧИХ МЕТОДІВ ПРОГНОЗУВАННЯ МЕТЕОРОЛОГІЧНИХ ПОКАЗНИКІВ

Прогнозування погоди – це завдання, яке вимагає великого обсягу даних та аналізу для отримання точних результатів. У цьому контексті існують різні методи аналізу, які допомагають прогнозувати метеорологічні умови.

Один із ключових підходів до прогнозування погоди – це використання математичних моделей, які моделюють атмосферні процеси. Ці моделі базуються на рівняннях, що описують фізичні закони руху повітря, теплообміну та інші атмосферні явища. Вони розділяють атмосферу на велику кількість областей та обчислюють зміни параметрів атмосфери в кожній з них з плином часу. Моделі дозволяють прогнозувати погоду на основі початкових умов, які визначаються спостереженнями та даними з метеорологічних станцій та супутників.

Інший підхід до прогнозування погоди полягає в аналізі статистичних зв'язків між різними погодними явищами на основі історичних даних. Наприклад, метод регресії може бути використаний для встановлення зв'язків між температурою, атмосферним тиском та іншими параметрами. За допомогою цих зв'язків можна прогнозувати майбутні значення погодних параметрів.

Сучасні технології штучного інтелекту, зокрема нейронні мережі, відіграють важливу роль у прогнозуванні погоди. Рекурентні нейронні мережі, такі як LSTM, можуть аналізувати складні закономірності в часових рядах метеорологічних даних. Вони здатні враховувати динаміку погодних явищ і прогнозувати їх на короткий та середній терміни. Цей підхід дозволяє отримувати точні та передбачувані результати в погодному прогнозуванні.

3. АРХІТЕКТУРА МОДЕЛІ LSTM

Архітектура моделі LSTM є різновидом рекурентних нейронних мереж (RNN) і була запропонована з метою вирішення проблем з затуханням та вибуховим зростанням градієнтів у RNN.

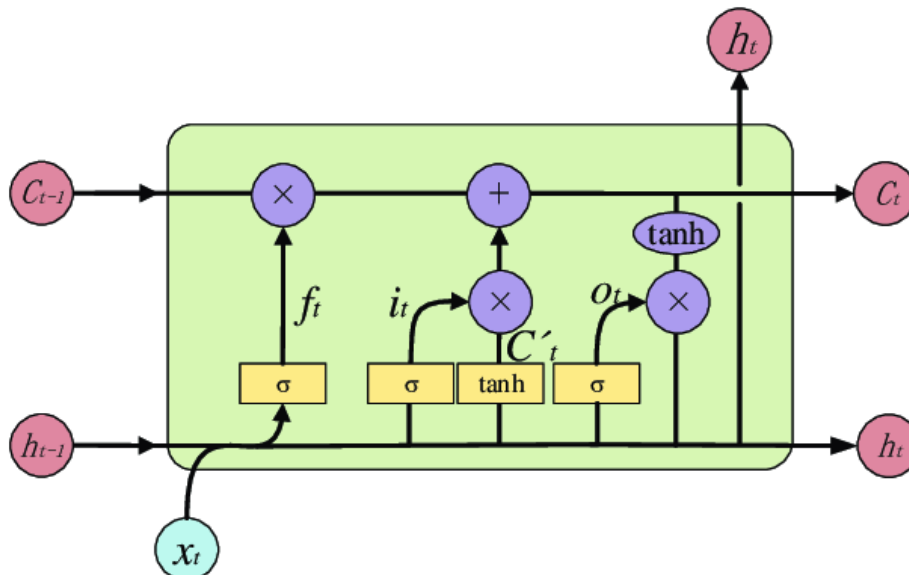


Рисунок 1. Архітектура блоку моделі LSTM

Відмінною особливістю LSTM є використання так званих "комірок пам'яті", які контролюють інформаційний потік за допомогою трьох воріт: вхідного, вихідного та забування. Розглянемо детальніше кожен з воріт:

- Вхідні ворота визначають, яка частина нової інформації буде збережена в комірці пам'яті.
- Ворота забування вирішують, яка частина існуючої інформації у комірці буде втрачена.
- Вихідні ворота визначають, яка частина інформації в комірці пам'яті буде передана до наступного рівня.

Кожні ворота складаються з сігмоїдальної активаційної функції, яка виводить значення між 0 та 1, та покомпонентного множення, що дозволяє контролювати потік інформації.

Ці механізми дозволяють LSTM зберігати, модифікувати або втрачати інформацію з комірки пам'яті в залежності від завдань, і це робить їх ефективними при роботі з послідовностями даних довгого терміну. Тому LSTM широко використовуються в задачах обробки природної мови, прогнозування часових рядів та інших завданнях, де потрібно моделювати завдання з великими відстанями між пов'язаними подіями.

4. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

У результаті проведеної роботи було розроблено систему прогнозування погодних умов, що має архітектуру, зображену на рисунку 2.

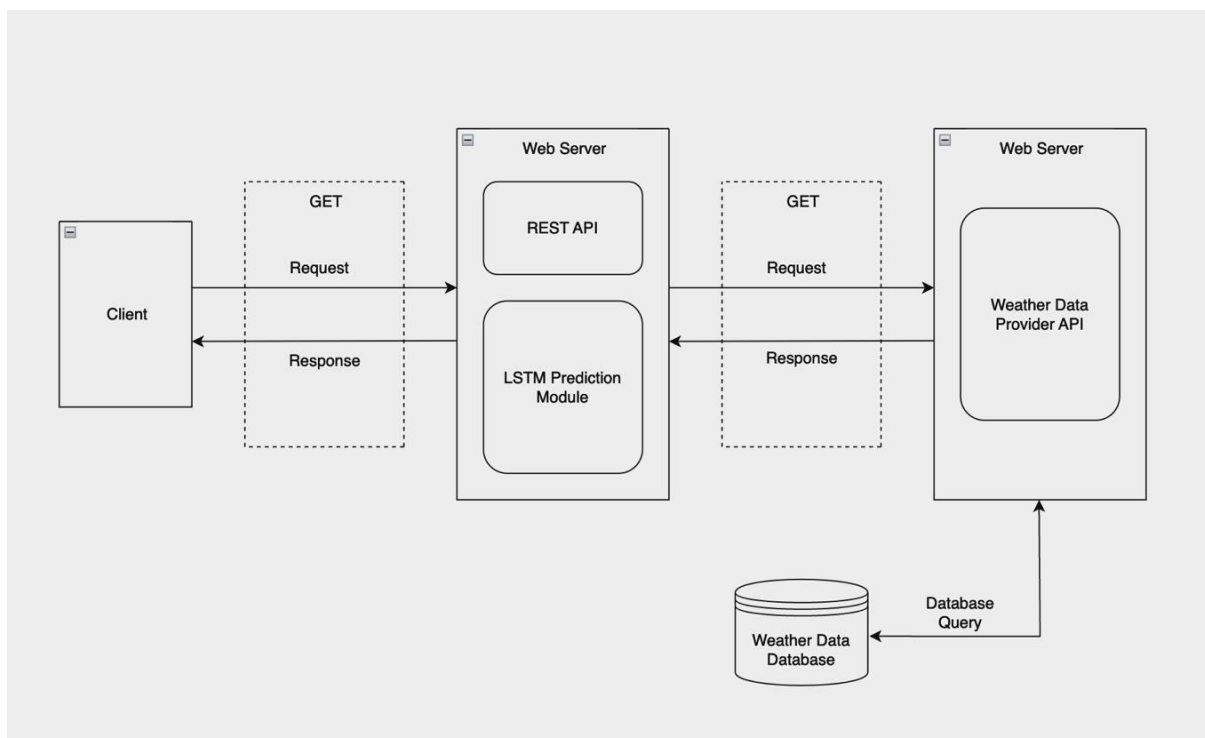


Рисунок 2. Архітектура системи

Запропонована архітектура системи забезпечує ефективний обмін даними між клієнтом, сервером прогнозування та постачальником історичних метеорологічних даних. Використання LSTM моделі в комбінації із технологією REST API створює гнучку, масштабовану та вискоєфективну систему прогнозування погоди.

Було проведено тренування та тестування моделі для отримання 48-годинного прогнозу на основі даних про 18 попередніх діб.

Для оцінки якості використовуються метрики Mean Absolute Error (MAE), Mean Squared Error (MSE), і Mean Absolute Percentage Error (MAPE) - це метрики, які використовуються для оцінки точності прогнозування або моделювання в наукових дослідженнях і аналізі даних.

Ми розглядаємо датасет, що представляє докладну інформацію про погодні умови в Києві, столиці України, з 1 січня 1881 року. Незважаючи на різноманіття джерел метеорологічної інформації, що доступні сьогодні, такий обширний історичний масив інформації може надавати в основному лише гідрометеорологічна служба. Це зумовлено історичними, організаційними та науковими особливостями збору та зберігання даних.

Нижче наведено графіки передбачень моделі для показника середньої температури на тренувальній та тестовій вибірках, а також значення метрик для кожного випадку.

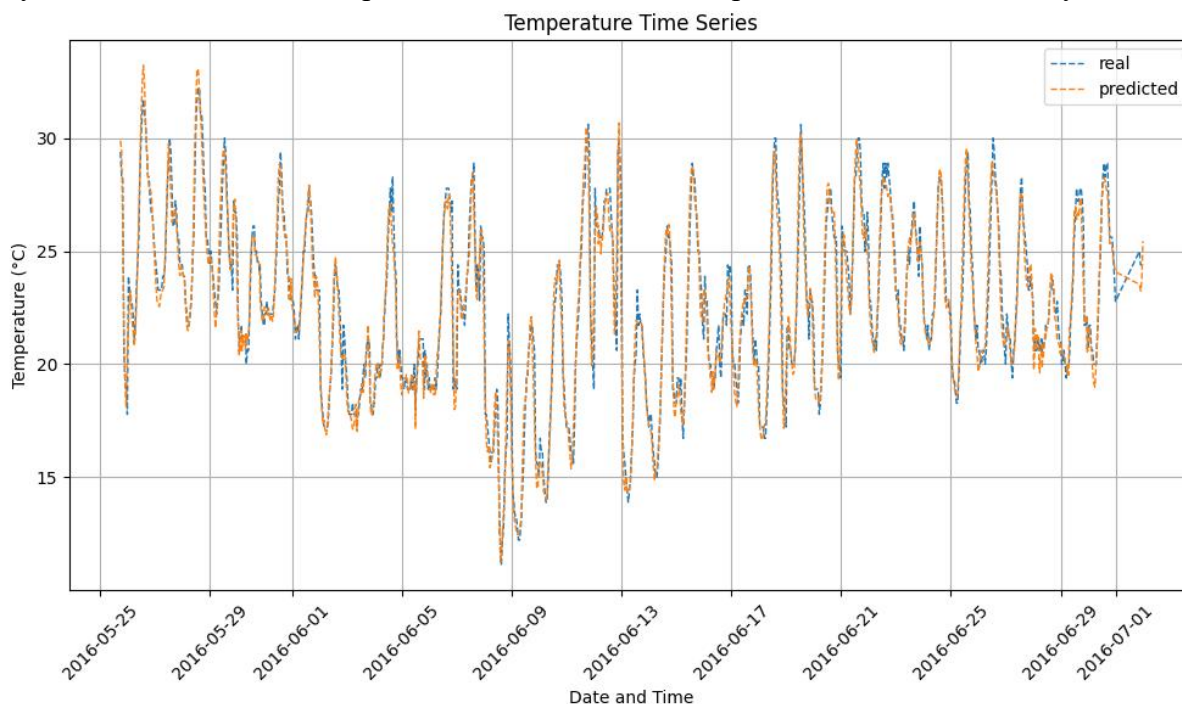


Рисунок 3. Порівняння графіків середньої температури для тренувальної вибірки

На рисунку 3 зображено порівняння графіків середньої температури для тренувальної вибірки (80% дата сету).

Показники метрик:

- MAE (Середня абсолютна похибка): 0,53;
- MSE (Середня квадратична похибка): 0,60;
- MAPE (Середня абсолютна похибка у відсотках): 2,9%.

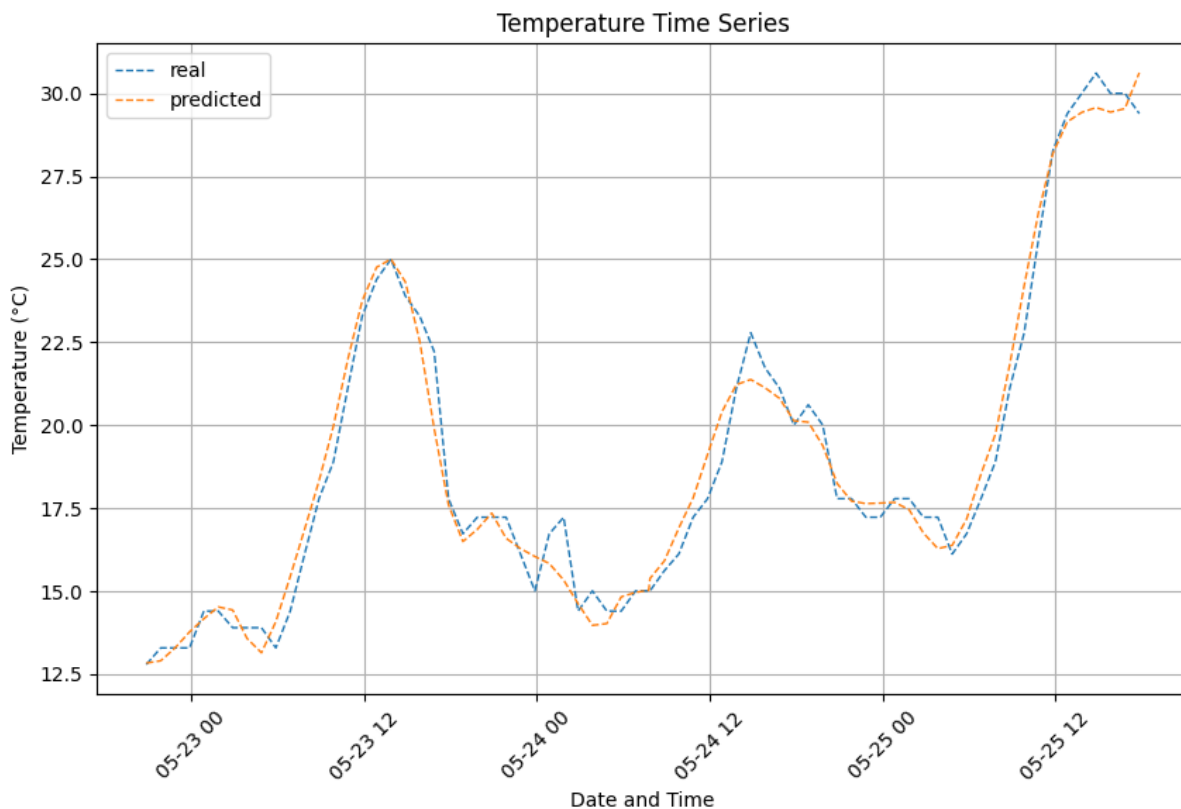


Рисунок 4. Порівняння графіків середньої температури для тестової вибірки

На рисунку 4 зображено порівняння графіків середньої температури для тестової вибірки (20% дата сету).

Показники метрик:

- MAE (Середня абсолютна похибка): 0,59
- MSE (Середня квадратична похибка): 0,82
- MAPE (Середня абсолютна похибка у відсотках): 3,5 %

5. ВИСНОВКИ

Розглядаючи реалізацію системи прогнозування метеорологічних умов на основі REST API та LSTM, можна відзначити ключову роль комбінації цих технологій. Інтеграція REST API забезпечує гнучкість та масштабованість системи, дозволяючи ефективно обмінюватися даними між її компонентами. Ця гнучкість стає особливо актуальною, оскільки система автоматично звертається до постачальника метеорологічної інформації через його власний API для отримання актуальних даних.

LSTM, в свою чергу, демонструє високу ефективність при аналізі часових рядів, зокрема в метеорологічному прогнозуванні. Ця модель враховує довгострокові залежності в даних, що підтверджується точністю прогнозів системи щодо погодних умов.

Архітектура системи розроблена таким чином, що вона легко адаптується до різних джерел даних, дозволяючи масштабувати систему. Крім того, завдяки її відкритості для інновацій, можливе впровадження нових методів аналізу даних або алгоритмів штучного інтелекту в майбутньому, що робить систему довгостроково стійкою до технологічних змін.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Han, J. M., Ang, Y. Q., Malkawi, A., & Samuelson, H. W. (2021). Using recurrent neural networks for localized weather prediction with combined use of public airport data and on-site measurements. *Building and Environment*, 192, 107601.
2. Sabzipour, B., Arsenault, R., Troin, M., Martel, J.-L., Brissette, F., Brunet, F., & Mai, J. (2023). Comparing a long short-term memory (LSTM) neural network with a physically-based hydrological model for streamflow forecasting over a Canadian catchment. *Journal of Hydrology*, 130380
3. Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Comput.* 9, 8 (November 15, 1997), 1735–1780.
4. Chen, J., & Huang, C. (2020). A Comprehensive Review of Weather Forecasting Using Machine Learning. *IEEE Access*, 8, 65288-65311.
5. LSTM: Long Short-Term Memory. (n.d.). Retrieved from https://www.tensorflow.org/guide/keras/rnn#lstm_long_short-term_memory
6. REST API Documentation. (n.d.). Retrieved from <https://developer.mozilla.org/en-US/docs/Glossary/REST>
7. Roy, A. (2019). Building a RESTful Web Service. Retrieved from <https://spring.io/guides/gs/rest-service/>
8. OpenWeather. (n.d.). Official Website. Retrieved from <https://openweathermap.org/>
9. Paramasivan, S. K. (2021). Deep learning based recurrent neural networks to enhance the performance of wind energy forecasting: A review. *Revue d'Intelligence Artificielle*, 35(1).

КЛАСТЕРИЗАЦІЯ ЗА ДОПОМОГОЮ OPTICS: АНАЛІЗ ТА ОПТИМІЗАЦІЯ З ВИКОРИСТАННЯМ ГРАФІКІВ ТА МЕТРИК

Іванюта О.О.¹, Недашківська Н.І.²

Національний технічний університет України «Київський політехнічний інститут
імені Ігоря Сікорського», Київ, Україна

¹ ivanyuta.olexandr@lil.kpi.ua, ² nedashkovskaya.nadezhda@lil.kpi.ua [0000-0002-8277-3095]

Розглядається задача кластеризації за допомогою OPTICS. Результати виведені на графіках досяжності. Проведено аналіз метрик якості, таких як Estimated number of clusters, Adjusted Rand Index (ARI), Adjusted Mutual Information (AMI) та Silhouette Coefficient, для кількісної оцінки точності та роздільності кластерів. На основі цих метрик визначено оптимальні моделі для кожного набору даних. Проведено експерименти з різними параметрами для визначення оптимальних значень. Розроблено програмне забезпечення для побудови кластеризації.

Ключові слова: OPTICS, DBSCAN, графік досяжності (reachability plot)

1. ВСТУП

В останні роки активно досліджується область алгоритмів кластеризації на основі щільності, які знаходять широке застосування у визначенні груп схожих об'єктів у великих обсягах даних для розпізнавання патернів та машинного навчання. Алгоритми кластеризації на основі щільності можуть обробляти зашумлені точки і виявляти кластери різної складної форми, але також зазвичай потребують значних часових витрат [1]. Представниками щільнісних алгоритмів є DBSCAN [2] та OPTICS [3]. Важливою властивістю багатьох наборів реальних даних є те, що їх внутрішня кластерна структура не може бути охарактеризована параметрами глобальної щільності. Для виявлення кластерів у різних регіонах простору даних можуть знадобитися різні локальні щільності. Основна ідея щільнісних алгоритмів кластеризації полягає в тому, що для кожної точки в кластері має існувати окіл заданого радіусу ϵ , який містить хоча б мінімальну задану кількість точок MinPts.

Останні публікації у цій області пропонують ряд інноваційних підходів та вдосконалень до алгоритму OPTICS [4–9]. Наприклад, в одному з досліджень вводиться новий метод кластеризації на основі щільності під назвою Fast Principal Component Analysis Pruning (FPCAP) [5]. Цей метод дозволяє ефективно прискорити виконання алгоритмів кластеризації, зокрема, вдосконалює алгоритми DBSCAN та BLOCK-DBSCAN [5].

З іншого боку, деякі дослідження акцентують увагу на важливості візуалізації та інтерактивного аналізу результатів роботи OPTICS. Зокрема, розроблено візуалізаційний інструмент VizOPTICS [6], який глибоко інтегрує людський та машинний інтелект для полегшення розуміння та використання OPTICS у виділенні значущих кластерів.

2. МЕТОДИ ТА МАТЕРІАЛИ

2.1 Основні означення

OPTICS (Ordering Points To Identify the Clustering Structure) – розширення DBSCAN. В OPTICS додатково будується графік досяжності: для кожної точки даних зберігається відстань до ядра та відстань досяжності, а також місце точки у відсортованій множині точок даних, необхідні для визначення належності до кластерів.

OPTICS базується на наступних означеннях [3].

ϵ -околом точки $p \in D$ називається множина $D_\epsilon(p) = \{q \in D \mid \text{distance}(p, q) \leq \epsilon\}$.

Параметр ϵ , строго кажучи, не є обов'язковим. Можна задати діапазон зміни значень ϵ , встановлюючи максимально можливе значення \max_eps цього параметру.

Точка $p \in D$ називається *основною точкою (core point) або ядром* якщо її ϵ -окіл містить принаймні MinPts точок, де ϵ і MinPts – задані.

MinPts – мінімальна кількість точок, потрібна для утворення кластера. Обирають $\text{MinPts} \geq m + 1$, де m – розмірність набору даних.

Точка називається *граничною (border point)*, якщо у її ϵ -околі точок менше ніж MinPts , але в ϵ -околі потрапляє основна точка. Точка називається *шумовою (noise)*, якщо у її ϵ -околі точок менше ніж MinPts і жодної основної точки туди не потрапляє.

Точка q *безпосередньо досяжна за щільністю* з точки p (позначимо $p \rightarrow q$) в множині точок D , якщо: 1) $q \in D_\epsilon(p)$ і 2) p – ядро, тобто $|D_\epsilon(p)| \geq \text{MinPts}$.

Умова 2) показує, що лише з основних точок інші точки можуть бути безпосередньо досяжними за щільністю.

Точка q *досяжна за щільністю (density-reachable)* з точки p в множині точок D , якщо існує послідовність точок q_1, \dots, q_n , $q_i \in D$, таких що $q_1 = p$, $q_n = q$ та $q_i \rightarrow q_{i+1}$:

$$p = q_1 \rightarrow q_2 \rightarrow \dots \rightarrow q_n = q$$

Відношення досяжності за щільністю не є симетричним в загальному випадку. Тільки основні точки (ядра) можуть бути взаємно досяжними за щільністю [3].

Точка q *зв'язана за щільністю (density-connected)* з точкою p в множині точок D , якщо існує точка $o \in D$, така що обидві точки p і q досяжні за щільністю з точки o .

Відношення зв'язності за щільністю симетричне.

На основі розглянутих означень, кластер – це множина з максимальної кількості зв'язаних за щільністю точок. Шумовими названо точки, які не містяться в жодному кластері.

Нехай D – множина точок даних, параметри ϵ і MinPts – задані.

Кластером C називається непорожня підмножина множини D , яка задовольняє наступним умовам [3]:

1) Максимальність: для кожних $p, q \in D$: якщо $p \in C$ і точка q досяжна за щільністю з p , то $q \in C$.

2) Зв'язність: для кожних $p, q \in C$: точка q зв'язана за щільністю з p .

Кластер складається не лише з основних точок (ядер), а також з граничних точок, які безпосередньо досяжні за щільністю з принаймні однієї основної точки цього кластеру.

2.2 OPTICS – покращений метод кластеризації

Алгоритм OPTICS вирішує основну слабкість DBSCAN – проблему виявлення кластерів даних з різною щільністю [9]. Результат кластеризації за алгоритмом DBSCAN дуже чутливий до значень початкових параметрів ϵ та MinPts , і різні значення цих параметрів можуть призводити до різних результатів кластеризації. Зокрема, в алгоритмі DBSCAN може бути важко знайти такі значення ϵ та MinPts , щоб правильно визначити всі кластери в даних.

Алгоритм OPTICS не є чутливим до радіуса ϵ , що робить його більш гнучким у виявленні кластерів. На рисунку 1 видно п'ять кластерів (A, B, C₁, C₂ і C₃). За відповідно підібраними значеннями ϵ_{ps} та MinPts, алгоритм DBSCAN може кластеризувати дані і отримати A, B, C, або кластеризувати так, щоб отримати C₁, C₂, C₃, але важко визначити параметри так, щоб отримати всі п'ять кластерів одночасно. У той час як алгоритм OPTICS, завдяки тому, що він не є чутливим до параметрів, може успішно виявити всі п'ять кластерів (A, B, C₁, C₂, C₃) одночасно.

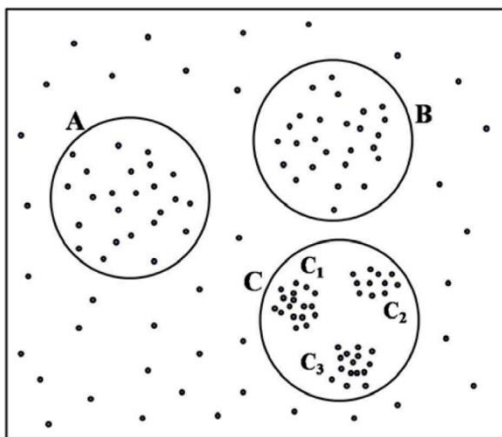


Рисунок 1. Кластери з різними параметрами щільності [3]

3. ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ АЛГОРИТМУ ТА ВИВЕДЕННЯ РЕЗУЛЬТАТІВ

Вхідними даними обрано два набори даних.

Для кожного з цих наборів було створено модель за допомогою алгоритму OPTICS. Параметри моделей налаштовані на оптимальні значення. На рисунку 2 продемонстровані результати роботи алгоритму.

Для першого набору даних, алгоритм визначив 2 набори даних та шум. Для другого набору – 6 кластерів та шум. З них повністю правильно визначено лише 1 кластер.

Кластери на графіках досяжності проявляються як долини на графіку. Глибина долин (тобто відстань досяжності між точками) вказує на щільність кластерів. Глибші долини означають щільніші кластери, і навпаки. На рис. 3.а видно дві глибоких долини A та B. Також на графіку присутній шум. На рисунку 3.б видно шість глибоких долин, які позначені латинськими літерами від A до F. Також на графіку присутній шум.

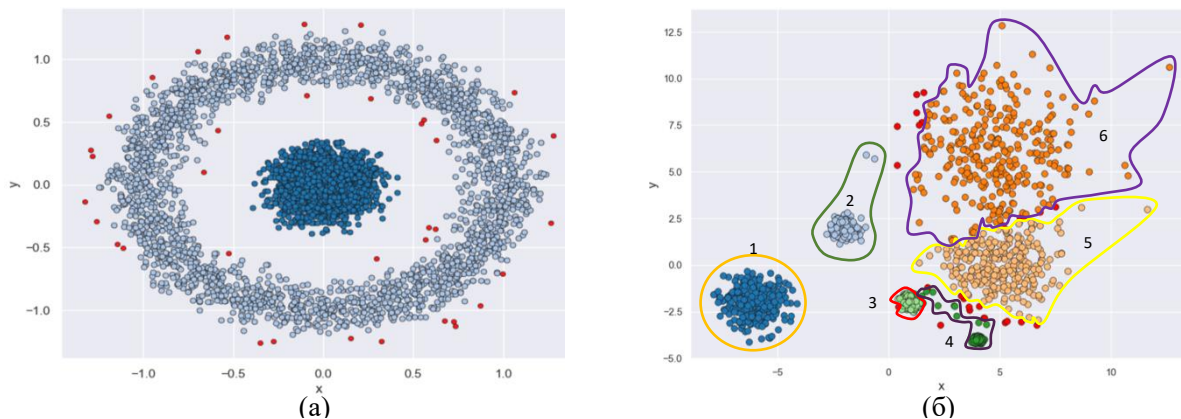
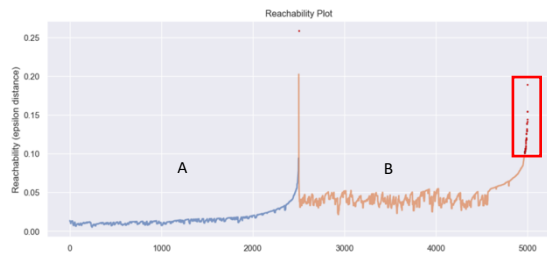
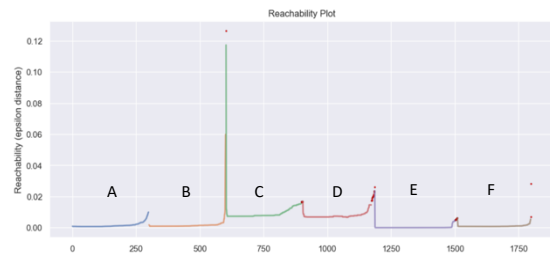


Рисунок 2. Результати кластеризації для першого набору даних (а) та для другого (б)



(a)



(б)

Рисунок 3. Графіки досяжності для першого(а) та другого(б) наборів даних

3.1. Результати перебору параметрів

На рисунку 4 зображені моделі з різними метриками. Чудовий результат показували моделі, які використовували DBSCAN для виділення кластерів.

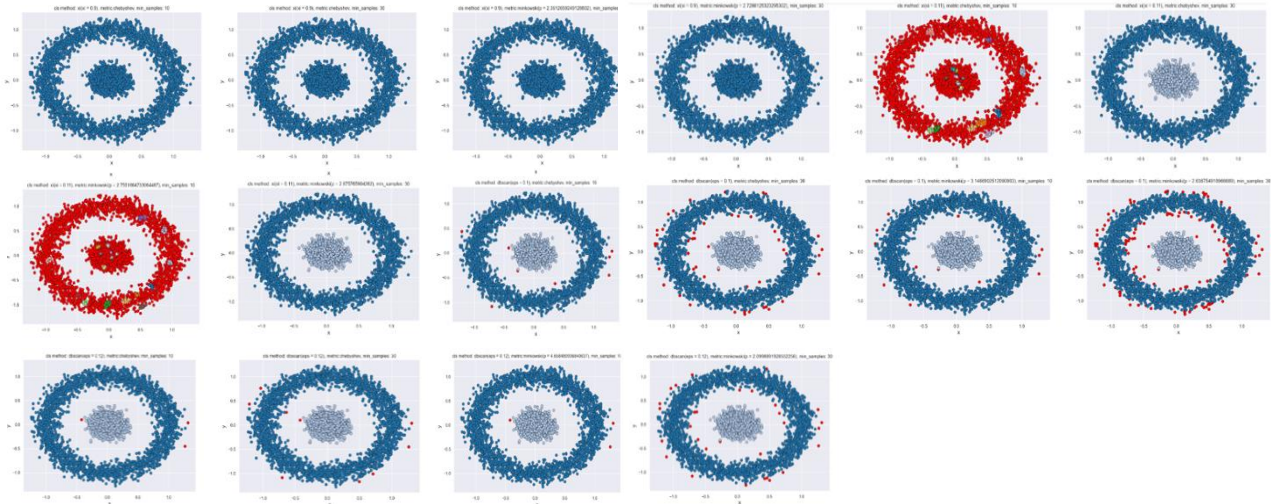


Рисунок 4. Графік альтернативних моделей для першого набору даних

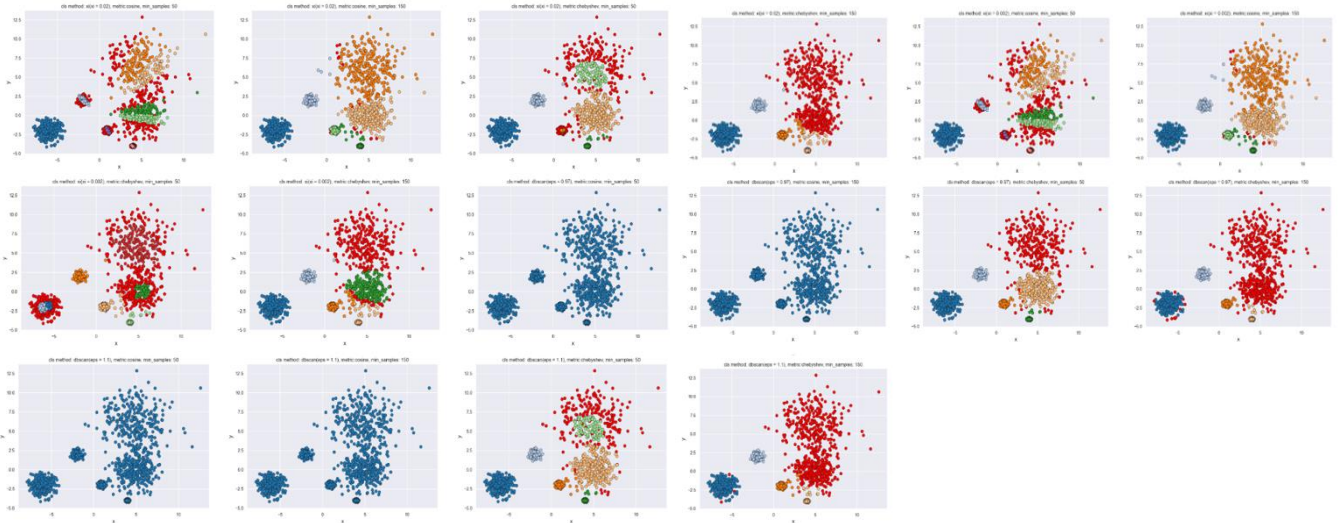


Рисунок 5. Графік альтернативних моделей для другого набору даних

Найгірші результати показали такі моделі, в яких χ_i дорівнює 0.9 (завелике значення), і моделі, в яких не використовувався DBSCAN та мінімальна кількість точок в кластері дорівнює 10.

При малих значеннях $min_samples$, у моделях, що не використовують DBSCAN, виділяють занадто багато кластерів, а в тих, що використовують виділяється максимум 3 кластери. Поєднання метрики cosine та великого значення ϵ призводить до того, що всі данні об'єднуються в один кластер. Інші моделі вийшли доволі хорошими. На цьому наборі даних, особливо при використанні DBSCAN, можна помітити, що алгоритм є дуже чутливим в залежності від поєднання різних параметрів.

3.2. Пошук найкращих параметрів

Було проведено дослідження, в якому було знайдено найкращі моделі за такими метриками:

1. Adjusted Rand Index (ARI) [10] – ARI вимірює подібність між справжніми мітками кластерів та прогнозованими мітками, враховуючи випадкові перестановки. Значення ARI від 0 до 1, де 1 вказує на ідеальне узгодження між справжніми та прогнозованими мітками
2. Adjusted Mutual Information (AMI) [11] – AMI знаходить ступінь взаємодії між справжніми та прогнозованими мітками. Значення AMI також варіюється від 0 до 1, де 1 вказує на повну взаємодію між мітками.
3. Silhouette Score [12] – Коефіцієнт силуету вимірює, наскільки добре відокремлені кластери в порівнянні з іншими кластерами. Значення варіюються від -1 до 1.

Для першого набору даних найкращим значенням мір якості відповідає одне й те ж розбиття. З рисунка 6 можна зробити висновок, що найкращою є модель з такими параметрами: $\chi_i = 0.11$, metric:chebyshev, $min_samples$: 30. Дана модель правильно визначила всі кластери для даного набору.

```
Best ARI value: 1.0 for model with parameters cls method: xi(xi = 0.11), metric:chebyshev, min_samples: 30
Best AMI value: 1.0 for model with parameters cls method: xi(xi = 0.11), metric:chebyshev, min_samples: 30
Best Silhouette score: 0.2837 for model with parameters cls method: xi(xi = 0.11), metric:chebyshev, min_samples: 30
```

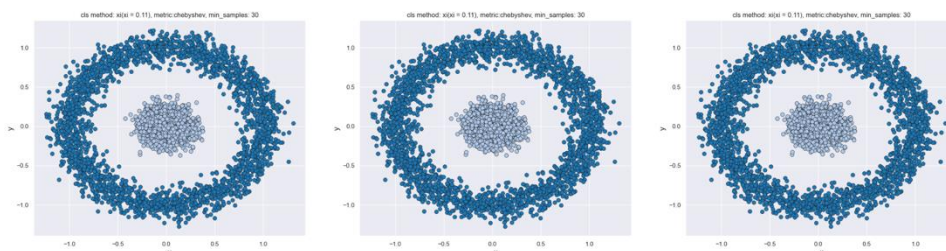


Рисунок 6. Найкращі значення кожної з метрик для першого набору даних та графіки відповідних моделей

Best ARI value: 0.9332 for model with parameters cls method: dbscan(eps = 0.97), metric:chebyshev, min_samples: 50
 Best AMI value: 0.9371 for model with parameters cls method: dbscan(eps = 0.97), metric:chebyshev, min_samples: 50
 Best Silhouette score: 0.7037 for model with parameters cls method: dbscan(eps = 0.97), metric:chebyshev, min_samples: 50

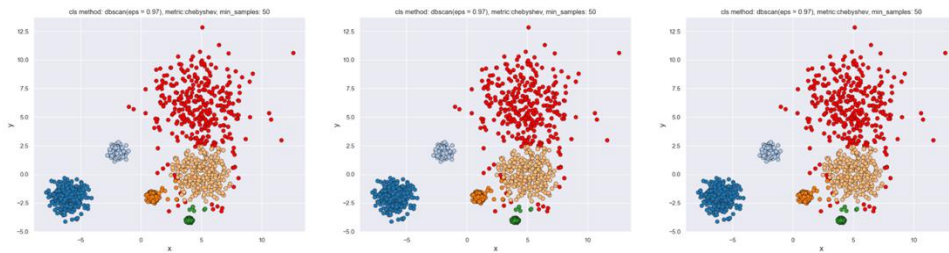


Рисунок 7. Найкращі значення кожної з метрик для другого набору даних та графіки відповідних моделей

З рисунка 7 можна побачити, що для другого набору даних найкращим значенням мір якості також відповідає одне й те ж розбиття. Проте ця модель визначає тільки 5 кластерів правильно, а шостий – як шум. Такі гарні результати метрик пов'язані з тим, що вони не відрізняють лейбл, який відповідає шумовим елементам, і враховують його як звичайний кластер. Тож найкращою є все таки наступна модель (рис. 8), яка має приблизно такі ж значення мір якості:

cls method: xi(xi = 0.06), metric:cosine, min_samples: 100
 Estimated number of clusters: 6
 Adjusted Rand Index: 0.926
 Adjusted Mutual Information: 0.919
 Silhouette Coefficient: 0.69

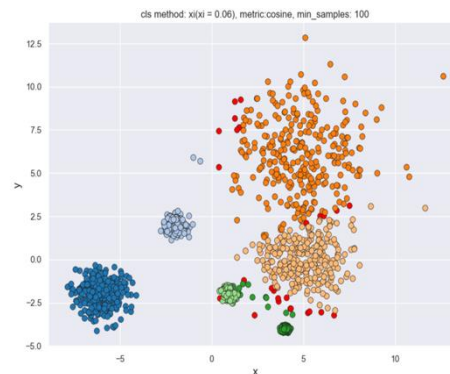


Рисунок 8. Графік моделі з хорошими значеннями метрик та шістьма визначеними кластерами

4. ВИСНОВКИ

Дослідження підтвердило ефективність OPTICS у визначенні кластерів та аналізі структури даних. Граф досяжності визначив межі кластерів та точки переходу. Оптимальні параметри налаштовані для кращого результату на кожному наборі даних. Метрики якості кластеризації оцінили точність та відокремленість кластерів, підтверджуючи успішне застосування OPTICS для аналізу даних.

Важливо врахувати, що деякі точки визначені як шум, що може вимагати додаткового дослідження. Загальний висновок полягає в тому, що OPTICS - потужний інструмент для кластеризації даних, проте його оптимальність залежить від конкретного контексту та завдань.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Feng L., Liu K., Tang F., Meng Q. GO-DBSCAN: improvements of DBSCAN.Algorithm based on grid, *Int. J. Comput. Theory Eng.* 2017. Vol. 9. P. 151–155.
2. Ester M., Kriegel H.P., Sander J., Xu X. A density-based algorithm for discovering. Clusters in large spatial databases with noise. *Second Int. Conf. Knowl. Discov. Data Min.* 1996. P. 226–231.

3. Ankerst M., Breunig M.M., Kriegel H.-P., Sander J. OPTICS: ordering points to identify. The clustering structure. *ACM SIGMOD Record*. 1999. Vol. 28. P. 49–60.
4. Schubert E., Gertz M. Improving the cluster structure extracted from OPTICS plots. *CEUR Workshop Proc.* 2018. Vol. 2191. P. 318–329.
5. Cheng D., Xu R., Zhang Bo, Jin R. Fast density estimation for density-based clustering methods. *Neurocomputing*. 2023. Vol. 532. P. 170–182.
6. Wu C., Chen Y., Dong Y., Zhou F., Zhao Y., Liang C.J. VizOPTICS: Getting insights into OPTICS via interactive visual analysis. *Computers & Electrical Engineering*. 2023. Vol. 107.
7. Wang J., Liu Z., Zhao Y., Xie Y., Xie Y. EAST-NBI experimental data processing method based on improved OPTICS algorithm. *Fusion Engineering and Design*. 2021. Vol. 172. P. 53–68.
8. Kamil I.S., Al-Mamory S.O. Enhancement of OPTICS' time complexity by using fuzzy clusters. *Materials Today: Proceedings*. 2023. Vol. 80. P. 2625–2630.
9. Grover N., A study of various fuzzy clustering algorithms. *International Journal of Engineering Research*. 2014. Vol. 3. P. 177–181.
10. Hoffman_M., Steinley_D., Brusco M. J. A note on using the adjusted Rand index for link prediction in networks. *Social Networks*. 2015. Vol. 42. P. 72–79.
11. Lazarenko D., Bonald T. Pairwise Adjusted Mutual Information. *NeurIPS 2021 Conference*. 2021. P. 53–59.
12. Shahapure K. R., Nicholas C. Cluster Quality Analysis Using Silhouette Score. *IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*. 2020. P. 124–127.

МОДЕЛЮВАННЯ СЕЙСМІЧНИХ ХВИЛЬ З ВИКОРИСТАННЯМ МАШИННОГО НАВЧАННЯ

Каніовська І.Ю.¹, Кавара А.О.²

Національний технічний університет України «Київський політехнічний інститут
ім. Ігоря Сікорського»

¹ ikaniovska@gmail.com, ² akavara2000@gmail.com

Відомо, що значна частка населення Землі проживає у сейсмічно активних районах. Приміром, на Тихоокеанському вогняному кільці (найбільш сейсмічно активна частина світу) розташовані такі країни, як Японія (із населенням близько 130 млн осіб), Філіппіни (100 млн), Індонезія (270 млн). Землетруси можуть нести значну небезпеку для життя населення, спричиняти руйнування будівель та інфраструктури. Саме тому важливим є покращення методів аналізу сейсмічних хвиль, зокрема застосовуючи методи машинного та глибокого навчання. Результатом дослідження є створення найбільш ефективної моделі, яка може відтворювати основні властивості сейсмічних хвиль.

Ключові слова: сейсмічні хвилі, аналіз часових рядів, машинне навчання, глибокі нейронні мережі.

1. ВСТУП

Землетруси несуть велику небезпеку для населення через масштаби потенційних руйнувань та раптовість настання цієї події. Однією із найбільших проблем є те, що цю подію неможливо передбачити завчасно, що значно зменшує можливості для створення антикризової стратегії запобігання важким наслідкам, таким, як знищення інфраструктури та людські втрати.

Із розвитком машинного та глибокого навчання, збільшенням обчислювальних потужностей, а також зі збільшенням кількості даних, яка накопичується дослідницькими станціями, є змога краще дослідити природу даного явища та більш ефективно оцінювати його основні характеристики. Проведення експериментів із моделюванням сейсмічних хвиль є важливим кроком для того, щоб наблизитися до можливості отримувати більш точні прогнози настання та сили майбутніх землетрусів.

2. МЕТОДИ ТА ПІДХОДИ ДО АНАЛІЗУ ТА МОДЕЛЮВАННЯ СЕЙСМІЧНИХ ХВИЛЬ

Основним джерелом дослідження сейсмічних явищ є збір та аналіз сейсмограм. Сейсмограма — це графік, який виводиться спеціальним приладом – сейсмографом. Даний графік визначає запис руху землі на вимірювальній станції як функції часу. Сейсмограми зазвичай реєструють коливання у просторі R^3 (координат x, y і z), причому вісь z перпендикулярна до поверхні Землі, а осі x і y паралельні поверхні. Енергія, виміряна на сейсмограмі, може бути результатом землетрусу або іншого джерела, наприклад вибуху.

До основних завдань до обробки цифрових сейсмограм відносять:

- дослідження характеристик, що залежать від часу, а саме:
 - виявлення імпульсів, їх фільтрацію, відновлення та симуляцію;
 - обирання фази землетрусу;

- поляризаційний аналіз;
- аналіз веспаграм (velocity spectrum analysis, спектральний аналіз швидкостей хвиль);
- формування векторів напрямку на основі вимірів набору хвиль.
- аналіз частотно-хвильових характеристик (f-k аналіз) та спектральний аналіз.

Основним вхідним параметром для визначення факту настання сейсмічної події є прибуття так званої Р-хвилі (primary or pressure wave, початкова хвиля або хвиля тиску). Особливістю Р-хвилі є те, що вона рухається найшвидше серед інших сейсмічних хвиль, тому під час певної сейсмічної події саме вона є першочерговим індикатором землетрусу.

Щодо виявлення імпульсів та появи Р- та S- (secondary waves) хвиль, то стандартний підхід передбачає обчислення ковзного середнього із різною довжиною вікон: короткострокове (STA, short time average) та довгострокове (LTA, long time average). Після цього обчислюється відношення сигнал/шум (SNR, signal-to-noise-ratio). Тоді сейсмічний сигнал виявляється після того, як SNR перевищує певне визначене значення. Перед підрахунком даної метрики хорошою практикою є використання фільтрів для очищення сейсмограми від шумів. Фільтрування також використовується для стандартизації вигляду сейсмограми та приведення її до одного зі стандартних виглядів [1].

Оскільки сейсмічним хвилям притаманні епістемічна невизначеність (тобто невизначеність, яка виникає через недостатні знання про систему, зокрема фізичні та геологічні особливості даних процесів) та алеаторична невизначеність (що виникає через помилки вимірювання), деякі моделі та алгоритми можуть мати обмеження та занадто сильні вимоги до розподілу даних, через що їхнє використання буде неефективним. Саме тому, при створенні моделі варто надати перевагу методам глибокого навчання перед традиційними методами машинного навчання. Зокрема, найбільш ефективними у даному випадку будуть наступні алгоритми, які можна додати до архітектури моделі навчання:

- на основі байєсовських технік (шар дропауту Монте-Карло, ланцюги Маркова у поєднанні із методом Монте-Карло, варіаційний вивід, варіаційні автокодувальники, навчання із підкріпленням та ін.)
- на основі ансамблів (глибокі ансамблі та їхні варіації) [2].

Ефективними також є згорткові нейронні мережі, генетичні та еволюційні алгоритми для аналізу [3].

3. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

У якості основного датасету було обрано STanford EArthquake Dataset (STEAD) [4]. Масив даних складається із майже 1.2 мільйона часових рядів, які відповідають більш, ніж 19000 годин запису сейсмічних сигналів. Дані складаються із двох частин:

- локальні хвилі землетрусів (тобто такі, які були записані на відстані не більш, ніж 350 км від гіпоцентру);
- шумові хвилі (тобто ті, які відповідають руху автомобілів, вибухам на поверхні, видобутку корисних копалин, та безпосередньо не відносяться до тектонічних процесів Землі).

Загалом дані були отримані із 2613 станцій, які розміщені на всіх континентах. Дані станції розміщені у сейсмічно активних районах, що забезпечує різноманітність вхідних даних по різним характеристикам (тривалість, магнітуда, глибина землетрусу тощо).

Приклад вхідних даних представлено на рис. 1.

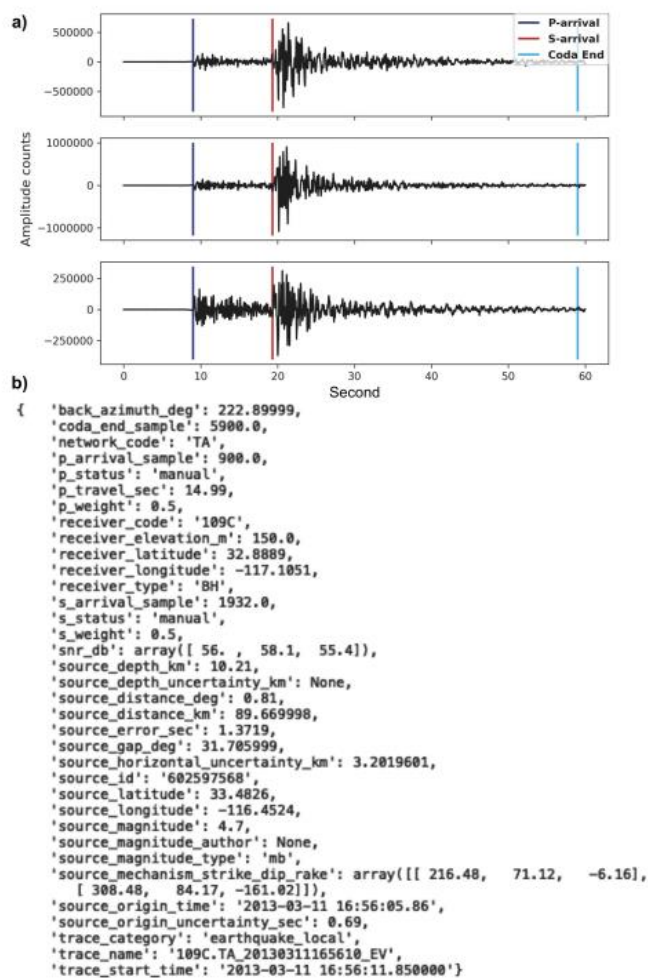


Рисунок 1. Приклад сейсмограми, що відповідає землетрусу. Секція а) часові ряди для поштовхів у напрямках схід-захід, північ-південь та для вертикальні поштовхів. Секція б) метадані сейсмограми

Власна модель прогнозування магнітуди землетрусів ґрунтується на моделі MagNet (Machine-Learning Approach for Earthquake Magnitude Estimation) [5]. Архітектура моделі представлена на рис. 2.

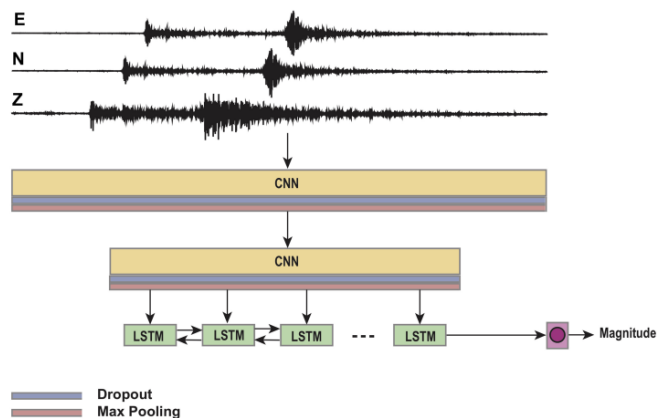


Рисунок 2. Архітектура моделі MagNet

На вхід подаються часові ряди для поштовхів усіх напрямків (схід-захід, північ-південь, ортогональні коливання). Далі дані обробляються за допомогою двох шарів згорткових нейронних мереж із дропаутом та шаром maxpooling 4*4 без функцій активації. Після цього іде шар зв'язних LSTM (long-short term-memory) комірок.

Для визначення часу прибуття P- та S-хвилі необхідна складніша модель із більшою кількістю шарів. За основу власної моделі була взята попередньо навчена модель EQTransformer [6]. Архітектура моделі представлена на рис. 3.

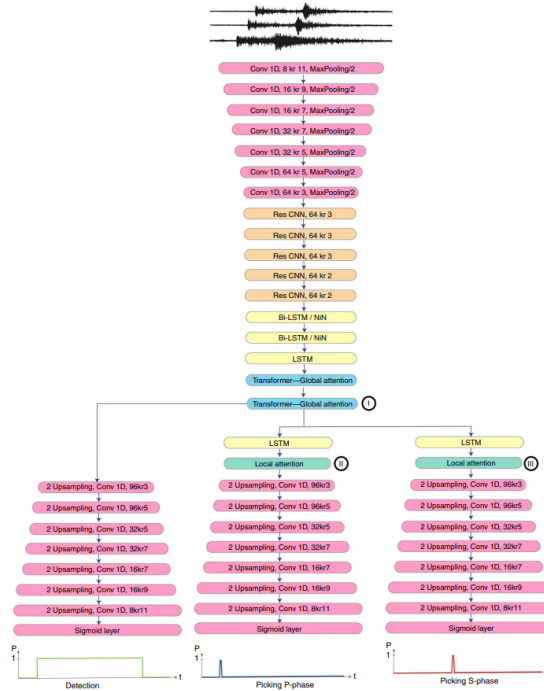


Рисунок 3. Архітектура моделі EQTransformer

Загальна структура мережі містить один глибокий кодувальник і три окремих декодувальники. Кодувальник приймає на вхід часовий ряд сейсмічного сигналу та генерує представлення високого рівня та контекстну інформацію про їх часові залежності. Потім цю інформацію використовують декодувальники для відображення ознак високого рівня для трьох послідовностей імовірностей, які пов'язані з існуванням землетрусу, P-фазою та S-фазою відповідно.

Значення основних метрик для отриманої моделі для визначення магнітуди землетрусу представлена у таблиці 1.

Таблиця 1. Метрики визначення якості моделі

Метрика	Значення
Співвідношення тренувальних та тестувальних даних	4:1
MAE (Mean Average Error)	0,1987
Стандартне відхилення MAE	0,2245
MSE (Mean Squared Error)	0,0899
Середня алеаторна невизначеність	0,0126
Середня епістемічна невизначеність	0,0679

Таким чином виявлено, що найвдаліша архітектура передбачає комбінацію шарів рекурентних та згорткових нейронних мереж. В подальшому можна використовувати моделі, які основані на архітектурі трансформера; байєсовські методи, що допоможуть визначати землетруси, які настають через одну й ту саму сейсмічну подію (наприклад, виверження вулкану).

4. ВИСНОВКИ

Землетруси – це небезпечне природне явище, яке впливає на життя населення Землі. Своєчасне повідомлення щодо настання землетрусів у певній місцевості має велику кількість переваг, оскільки допоможе знизити ризики та зберегти людські життя.

Основним носієм інформації про землетруси є сейсмограми. Правильна інтерпретація сейсмічних хвиль допомагає у подальшому вивченні даного природного явища. У ході дослідження було розглянуто основні завдання та принципи побудови моделей для проведення аналізу сейсмограм. У ході дослідження з'ясувалося, що модель, яка ґрунтується на шарах рекурентних та згорткових нейронних мереж, має непогані показники для поставленої задачі. Також було проведено експерименти щодо визначення магнітуди землетрусів на основі часових рядів сейсмічних сигналів, які підтвердили ефективність обраної стратегії.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Bormann P. Chapter 11. Data Analysis and Seismogram Interpretation / P. Bormann, K. Klinge, S. Wendt., 2009. – 102 с.
2. A review of uncertainty quantification in deep learning: Techniques, applications and challenges / [M. Abdar, F. Pourpanah, S. Hussain та ін.] // Information Fusion / [M. Abdar, F. Pourpanah, S. Hussain та ін.], 2021.
3. Research on Seismic Signal Analysis Based on Machine Learning / [Y. Xinxin, L. Feng, C. Run та ін.] // Special Issue Intelligent Computing and Remote Sensing / [Y. Xinxin, L. Feng, C. Run та ін.], 2022.
4. Mousavi, S. M., Sheng, Y., Zhu, W., Beroza G.C., (2019). STanford EArthquake Dataset (STEAD): A Global Data Set of Seismic Signals for AI, IEEE Access, doi:10.1109/ACCESS.2019.2947848
5. Mousavi, S. M., & Beroza, G. C. (2019). A Machine-Learning Approach for Earthquake Magnitude Estimation. Geophysical Research Letters.
6. Mousavi, S.M., Ellsworth, W.L., Zhu, W., Chuang, L, Y., and Beroza, G, C. Earthquake transformer—an attentive deep-learning model for simultaneous earthquake detection and phase picking. Nat Commun 11, 3952 (2020). <https://doi.org/10.1038/s41467-020-17591-w>

МОДЕЛІ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ ДАНИХ ДЛЯ ОЦІНЮВАННЯ ФІНАНСОВИХ МОДЕЛЕЙ

Коваленко О.М.¹, Гуськова В.Г.²

Національний технічний університет України «Київський політехнічний інститут
імені Ігоря Сікорського», Київ, Україна

¹ kovalenko.oleksandr@lil.kpi.ua, ² guskovavera2009@gmail.com

Робота присвячена дослідженню і створенню моделей та розробці ефективних методів і алгоритмів інтелектуального аналізу фінансових даних з задачею прогнозування і визначення аномалій

Ключові слова: математична статистика, теорія ймовірностей, нейронні мережі, моделі і методи інтелектуального аналізу даних

1. ВСТУП

Сучасний світ щоденно виробляє 2,5 мільярди гігабайт інформації на день, що накопичуються, зберігаються і потребують аналізу. Те ж стосується і економічних даних. Однак, зростання обсягів даних та їх складнощів створюють виклики для традиційних методів аналізу [1].

Відповідно, ІАД стає все більш актуальним і потужним інструментом для виявлення складних зв'язків, прогнозування тенденцій та виявлення аномалій у фінансових даних. Застосування новітніх методів дозволяє отримати нові інсайди і зробити обґрунтовані рішення, що сприяє підвищенню ефективності фінансового управління.

Однак, обробка та аналіз великого обсягу фінансових даних вимагають потужних обчислювальних ресурсів. В цьому контексті, хмарні технології надають універсальне та масштабоване середовище для зберігання, обробки та аналізу даних, що робить їх надзвичайно привабливими для фінансових аналітиків та дослідників.

У цій роботі міститься дослідження та розробка моделей і методів інтелектуального аналізу фінансових даних з використанням хмарних технологій. Конкретно, робота спрямована на створення ефективних алгоритмів та методів для оцінювання фінансових даних, прогнозування трендів та виявлення аномалій.

2. ОГЛЯД МОДЕЛЕЙ ТА МЕТОДІВ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ ДАНИХ

Методи інтелектуального аналізу даних використовуються для виявлення закономірностей і залежностей у великих обсягах неструктурованих даних.

Загальний результат ІАД можна охарактеризувати як знання щодо закономірностей і тенденцій, що повинно мати такі властивості:

- він відображає результати дослідження системи, відображаючи об'єктивну реальність.
- він представлений у зрозумілій людині формі, використовуючи загальноприйняті символи, поняття та природну мову.
- він компактний у своєму описі, що дозволяє його легко розуміти, інтерпретувати та використовувати.

Для створення моделі, яка б описувала та пояснювала закономірності даних, потрібно розуміти що дані можуть використовуватися для побудови просторових моделей і для побудови моделей часових рядів. Перший вид описує певну кількість процесів у конкретний момент часу t , а другий описує тільки один процес за інтервал часу t .

Часовий ряд – це послідовність числових показників, що упорядковані у часі і описують рівень стану і зміни досліджуваного об'єкту. Він характеризується своєю сезонністю, трендом, циклом та похибкою.

Поділяються на стаціонарні та нестаціонарні.

Стаціонарними називаються такі часові ряди, характер яких не змінюється з часом. Вимога до стаціонарного часового ряду полягає у присутності сталого середнього значення і в той самий час інші значення коливаються навколо цього середнього зі сталою дисперсією. Така особливість невласлива нестаціонарним часовим рядам, при цьому зберігаючи властивості сезонності та тренду.

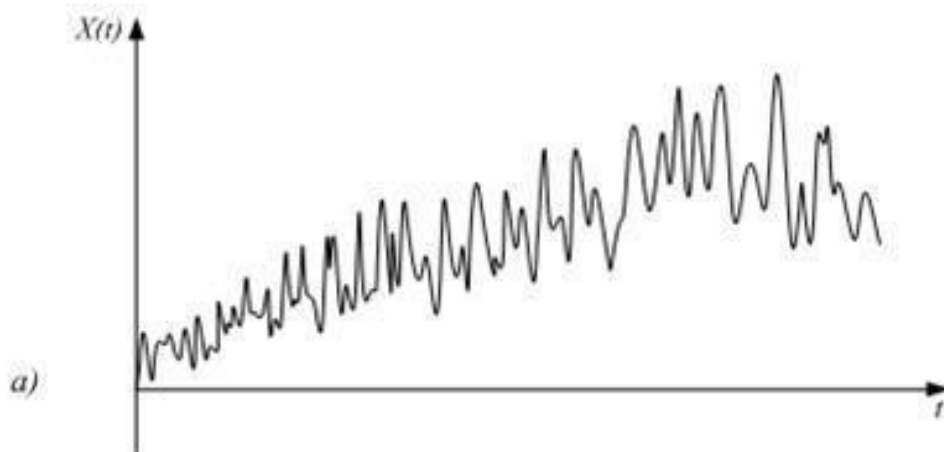


Рисунок 1. Вигляд нестаціонарного часового ряду

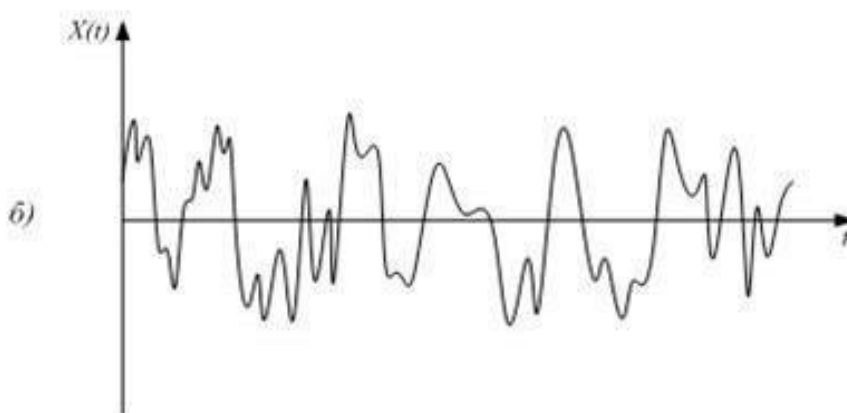


Рисунок 2. Вигляд стаціонарного часового ряду

Data Mining надає можливість знаходити нові гіпотези про поведінку невідомих, але реально існуючих залежностей в даних, створювати моделі, які можуть оцінити ступінь впливу факторів, що досліджуються.

Існує два загальних типи ІАД – на основі верифікації (verification-driven data mining) та на основі виявлення (discovery-driven data mining).

Аналіз даних на основі верифікації використовує структуровані запити (SQL) та багатовимірний, статистичний аналіз для отримання результатів. Це включає прогностичне моделювання, виявлення аномалій, аналіз зв'язків та сегментацію баз даних.

Інтелектуальний аналіз даних (ІАД) найчастіше розв'язує чотири основних завдання: асоціацію, кластеризацію, класифікацію і регресію.

Також існують ситуації нерівномірного розподілення даних у датасеті, коли з'являється необхідність застосовувати гібридний тип навчання, як з вчителем, так і без вчителя. Наприклад, кількість маркованих даних більша. В такому разі відбувається навчання з вчителем, будується аналітична модель, після цього відбувається навчання без вчителя з підкріпленням і побудовою аналітичної моделі та просто використовуючи навчання з підкріпленням [2].

Процес породження наявних даних є лінійним для стаціонарних часових рядів і зазвичай не мають тренду, або періодичної зміни середнього та дисперсії. Перевірити гіпотезу стосовно сталості середнього значення та дисперсії часового ряду можна виконати кількома способами. Ось один з найпростіших:

1. Перевірити значущість різниці двох середніх значень підмножин вибірки за критерієм перевірки гіпотези про рівність середніх двох нормально розподілених вибірок (z-критерій).
2. Перевірити сталість дисперсії. Наприклад, використовуючи F-критерій (критерій Фішера про відношення вибірових дисперсій).

Масштабуванням ознак у розрізі інтелектуального аналізу даних називається метод нормалізації незалежних ознак у фіксованому діапазоні. Цей процес є необхідним етапом попередньої обробки набору даних, бо забезпечує ефективне та швидке виконання використовуваного алгоритму, оскільки дозволяє алгоритму навчання не зважувати більші значення, що є затратним в плані обчислювальних можливостей.

Фінансові ринки є складними системами, де кожна зміна може мати значний вплив на ціни активів та рівень ризику. Врахування цих факторів та прогнозування їх впливу стає важливим завданням для інвесторів, трейдерів та фінансових установ. У цьому розділі ми розглянемо різноманітні моделі та методи, які допомагають управляти ризиками та приймати обґрунтовані рішення на фінансових ринках.

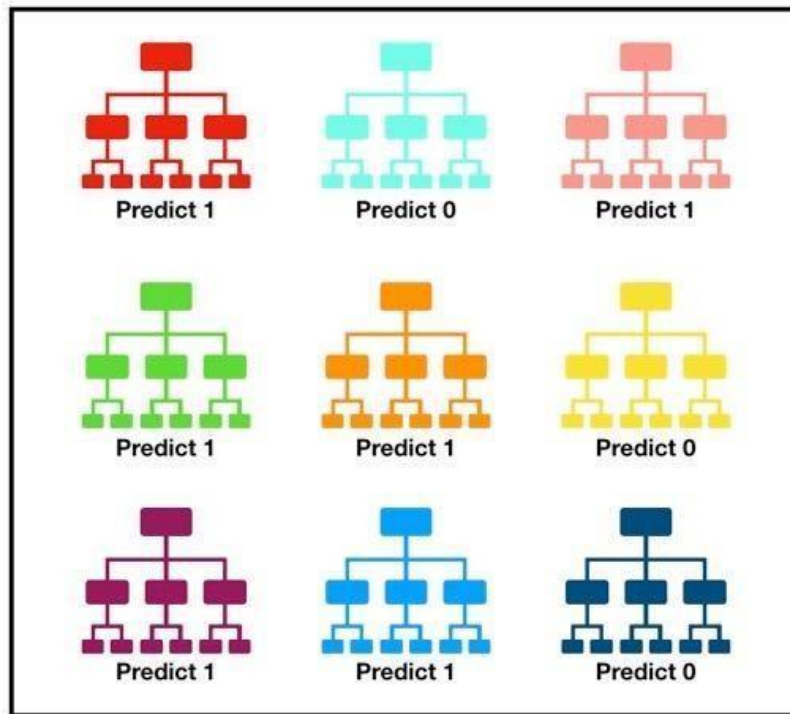
Передбачення майбутньої поведінки фінансових ринків є складною задачею, оскільки вони піддаються впливу багатьох непередбачуваних факторів, таких як економічні події, політична нестабільність та інші зовнішні чинники. Використання математичних моделей та аналітичних методів дозволяє нам підвищити точність прогнозів та зробити більш обґрунтовані рішення на фінансових ринках. Ці моделі базуються на статистичних методах, математичних алгоритмах та комп'ютерному моделюванні, що дозволяє аналізувати великі обсяги даних та виявляти складні залежності між різними факторами. У цьому розділі ми розглянемо основні математичні моделі та методи, їх переваги та обмеження, а також викладемо приклади їх застосування для прогнозування фінансових ринків

Велика частина машинного навчання зосереджена на класифікації - ми хочемо знати, до якого класу (групи) належить спостереження. Здатність точно класифікувати спостереження має велику цінність для різних бізнесзастосувань, таких як прогнозування того, чи купить певний користувач продукт, або передбачення того, чи буде заданий кредит невиконаним чи ні.

Випадковий ліс, як і його назва вказує, складається з великої кількості окремих дерев рішень, які працюють як ансамбль. Кожне окреме дерево випадкового лісу дає прогноз класу, і клас з найбільшою кількістю голосів стає прогнозом нашої моделі.

Основна ідея за випадковим лісом - проста, але потужна: оцінка від натовпу. З точки зору науки про дані, причина, чому модель випадкового лісу працює настільки добре, полягає в наступному:

Велика кількість майже некорельованих моделей (дерев) як комітет буде працювати краще, ніж будь-яка з окремих моделей-компонентів. [5]



Tally: Six 1s and Three 0s
Prediction: 1

Рисунок 3. Візуалізація методу випадкового лісу, що робить передбачення

Низька кореляція між моделями є ключовою. Так само, як і випадкові інвестиції з низькою кореляцією (такі як акції та облігації) поєднуються, утворюючи портфель, який перевищує суму його складових, некорельовані моделі можуть створювати прогнози ансамблю, які є більш точними, ніж будь-які окремі прогнози. Причина цього чудового ефекту полягає в тому, що дерева захищають одне одного від своїх індивідуальних помилок (якщо вони не постійно роблять помилки в одному напрямку). Хоча деякі дерева можуть бути неправильними, багато інших дерев будуть правильними, тому як група дерев вони здатні рухатися в правильному напрямку. Отже, передумови для успішної роботи випадкового лісу такі:

У ознак повинен бути певний справжній сигнал, щоб моделі, побудовані з використанням цих ознак, виконувалися краще, ніж простий випадковий вибір.

Прогнози (і, отже, помилки) індивідуальних дерев повинні мати низьку кореляцію одне з одним.

3. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Для аналізу було обрано вибірку щоденного курсу з 2010-01-01 до 2023-10-24 акцій успішної компанії, що займається виробленням електронних пристроїв та програмного забезпечення. Маємо 3475 записів про ці дні з такими ознаками:

- Open – ціна на початку торгового періоду
- Close – ціна в кінці торгового періоду
- High – найвища ціна, що була досягнута за даний період
- Low – найнижча ціна, що була досягнута за даний період
- Volume – кількість угод, укладених протягом торгового періоду

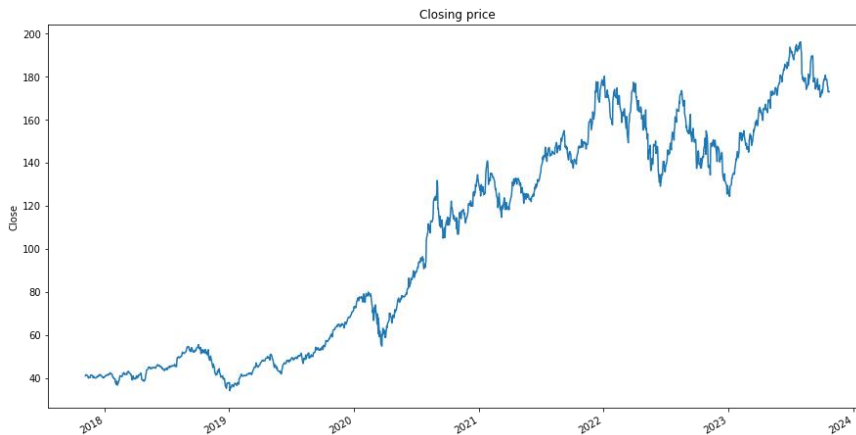


Рисунок 4. Close параметр відносно часу

Для прогнозування даного набору даних було створено 7 моделей.

1. Vanilla LSTM. Це стандартна реалізація мережі Long Short-Term Memory, що є варіацією рекурентної нейронної мережі і призначена для роботи послідовностями даних, такими як звук, текст, часові ряди. Основна ідея – відтворення LSTM без модифікацій або розширень. Має два шари, центральний та прихований, які оновлюються і передають інформацію в кожному кроці.

2. Stacked LSTM. Є модифікацією стандартної LSTM моделі, де декілька прихованих шарів розташовані один за одним (або один на одному – звідси і назва Stacked). Вони краще розуміють інгібиторні та зворотні залежності, мають більшу глибину, краще адаптуються до складних завдань та підходять для задач, де часові залежності є важливими.

3. Bi-directional LSTM. Чергова варіація LSTM, що дозволяє інформації переміщуватися в обох напрямках вздовж послідовності даних, на відміну від стандартного відтворення. Головна перевага – здатність враховувати контекст як попередньої так і наступної частини даних. Хоч дана модель і потребує більших обчислювальних потужностей, вона є досить універсальною та може використовуватися для різних задач.

4. ARIMA – модель часового ряду, яка використовується для аналізу та прогнозування часових рядів. Має такі характеристики:

(a) AR – авторегресія, залежність поточного значення часового ряду від попередніх.

(b) I – інтегрування, процедура диференціації, що призначена для перетворення ряду до стаціонарного, тобто позбавленого тренду та сезонності.

(c) MA – ковзне середнє, середнє значення змінних за певний період часу

5. SARIMA – модифікація ARIMA з додаванням компоненту сезонності. Така модель призначена для моделювання з вираженими сезонними паттернами.

6. Decision tree, дерево рішень, один з найбільш популярних алгоритмів Інтелектуального Аналізу Даних для вирішення проблем регресії та класифікації. Має структуру дерева, де внутрішні вузли представлені прийняттям рішень на основі конкретної ознаки, а кожен листок представляє прогнозоване значення. Ця модель легко інтерпритується та візуалізується для розуміння, як модель приймає рішення, корисна в визначенні ознак, проте попри все, може бути легко перенавченою

7. Random Forest – це ансамбльний алгоритм машинного навчання, що базується на деревах рішень. Така модель робить багато прогнозів для кожного прикладу даних і, в результаті, дані об'єднуються для отримання точного, зазвичай, результату. Однією з переваг можна назвати можливість працювати з великою кількістю ознак і даних. Random Forest є

популярним через високу точність, стійкість до перевантаження та можливість працювати з різними типами даних.

Результати якостей моделей

Таблиця 1. Результати навчання моделей на даних компанії_1

	R2	MSE	RMSE	MAE	MAPE
Vanilla LSTM	0,828	60,889	7,803	6,041	0,037
Stacked LSTM	0,889	39,221	6,263	5,454	0,035
Bi-directional LSTM	0,968	11,324	3,365	2,627	0,017
ARIMA	-2,507	0,053	0,229	0,197	0,039
SARIMA	-1,434	0,037	0,191	0,152	0,03
Decision Tree	-1,158	763,652	27,634	21,764	0,131
Random Forest	-1,131	754,069	27,46	21,572	0,13

Для порівняння візьмемо дані іншої компанії, що має суттєвий обвал акцій на початку 2022 року після різкого зростання. Надалі вартість то зростала, то спадала. Нижче на рис. 5 наведений графік вартості акції наприкінці торговельного періоду.

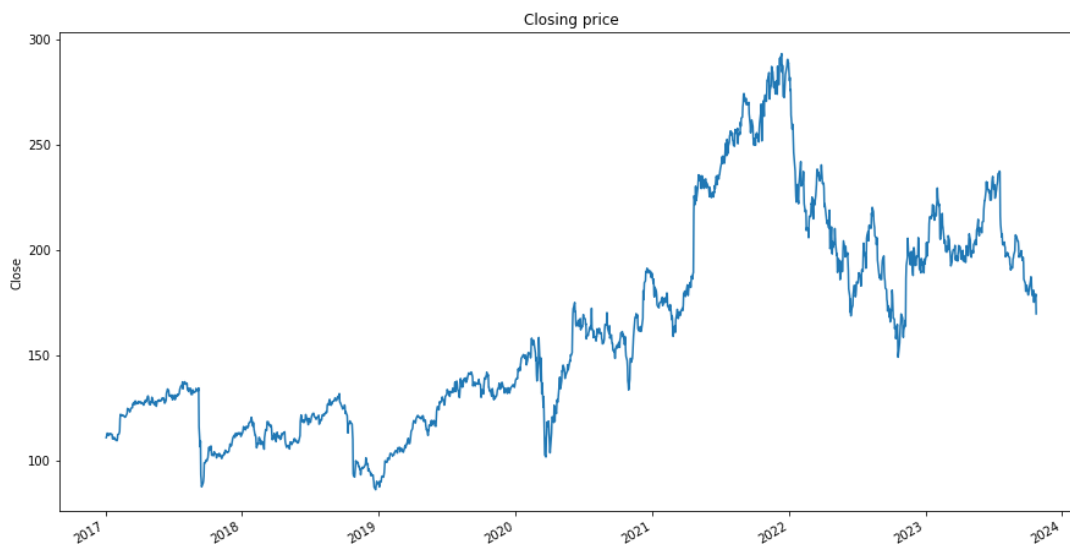


Рисунок 5. Вартість акцій компанії_2

Тепер побудуємо аналогічні моделі та поспостерігаємо за їх метриками.

Таблиця 2. Результат якості моделей компанії_2

	R2	MSE	RMSE	MAE	MAPE
Vanilla LSTM	0,926	22,065	4,697	3,689	0,019
Stacked LSTM	0,929	21,225	4,607	3,572	0,018
Bi-directional LSTM	0,939	18,154	4,261	3,167	0,016
ARIMA	-4,131	0,041	0,203	0,18	0,034
SARIMA	-2,704	0,03	0,172	0,154	0,029
Decision Tree	0,962	11,646	3,413	2,634	0,014
Random Forest	0,972	8,477	2,912	2,311	0,012

4. ВИСНОВКИ

Фінансові дані, попри загальну тенденцію до подібних поведінок, багато чим відрізняються поміж собою. В розглянутому випадку був приклад світової корпорації яка добре регулює стан своїх акцій і уміє справлятися з непередбачуваними ситуаціями, форс-мажорами державного рівня. Враховуючи постійний зріст та зрозумілі тренди, аналіз та дослідження цих фінансових даних було абсолютно безперешкодним та прогнозованим. В таких умовах досить легко обрати стратегію для подальших капіталовкладень, базуючись на простих і логічних методиках, які були наведені вище.

В іншому розглянутому випадкові ми маємо компанію з незрозумілими, на перший погляд, коливаннями вартості акцій. Більш того, модель Bi-directional LSTM, що чудово проявила себе в першому випадку, тут показала достатньо посередній результат.

Таким чином, можна дійти до деяких висновків аналізуючи наведені факти. При розробці моделей для Інтелектуального Аналізу Даних потрібно завжди брати до уваги декілька варіантів для його здійснення, зациклення лиш на одному, чи двох може мати серйозні негативні наслідки у вигляді поганого прогнозу чи оцінці ситуації. Компетентний та структурований підхід має неабияку роль для досягнення високого результату. Лише в стабільних умовах за відсутності зовнішніх чинників історичне моделювання дає точні результати. Для реального світу з сучасними проблемами необхідно завжди обирати комплекс заходів.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Методи інтелектуального аналізу даних URL: <https://buklib.net/books/24506/>
2. Ланде Д.В., Субач І.Ю., Бояринова Ю.Є. Основи теорії і практики інтелектуального аналізу даних у сфері кібербезпеки: навчальний посібник. — К.: ІСЗЗІ КПІ ім. Ігоря Сікорського», 2018. — 297 с.
3. Гороховатський В. О. Методи інтелектуального аналізу та оброблення даних : навч. посіб. / В. О. Гороховатський, І. С. Творошенко ; М-во освіти і науки України, Харків. нац. ун-т радіоелектроніки. – Харків : ХНУРЕ, 2021. – 92 с.
4. Kuznietsova N. V. Identification and dealing with uncertainties in the form of incomplete data by data mining methods. System research and information technologies. 2016. No. 2. P. 104. URL: <https://doi.org/10.20535/srit.23088893.2016.2.10> (date of access: 08.06.2023).
5. Understanding Random Forest URL: <https://towardsdatascience.com/understanding-random-forest-58381e0602d2>

СУЧАСНІ МОДЕЛІ ОЦІНЮВАННЯ ФІНАНСОВИХ РИЗИКІВ

Костенко М.О.¹, Кузнєцова Н.В.

Національний технічний університет України «Київський політехнічний інститут
ім. Ігоря Сікорського»

¹ kostenko.max@iitl.kpi.ua

Сучасні методи математичного моделювання нелінійних процесів охоплюють різноманіття підходів, а стандартним підходом вважається використання моделі ARIMA та її варіацій. Класичні регресійні підходи прогнозують цільову змінну лінійною комбінацією минулих значень цієї змінної. Тому доволі просто використовуються як з теоретичної, так і з обчислювальної точки зору завдяки простій структурі. Даний підхід обмежується складністю врахування великої кількості зовнішніх факторів через проблему мультиколінеарності, а також їх можливий нелінійний вплив. Нейронні мережі навчаються на досвіді і адаптуються до змін середовища, яке моделюється. Нейронні технології застосовуються для нелінійного моделювання, стійкі до інформаційних завад і здатні до узагальнення на основі історичних даних, а їх використання може покращити результат прогнозування. Для роботи з послідовностями представлені нейронні мережі, а також їх комбінації з моделями ARIMA. Це дозволяє вирішити поставлену задачу моделювання з урахуванням нелінійного або комбінованого впливу зовнішніх факторів. Для поліпшення точності прогнозів також представлений метод аугментації часових рядів за допомогою лінійної інтерполяції.

Ключові слова: фінансові ризики, системний підхід, часові ряди, ARIMA, NARX, DNN, LSTM.

1. ВСТУП

Моделювання і прогнозування нелінійних процесів є складним завданням, і ефективні методи варіюються в залежності від конкретного контексту та даних. Для досягнення точних та надійних результатів використовуються різні моделі. Зокрема, модель ARIMA (авторегресійна інтегрована змінна середня) використовується для моделювання та прогнозування стаціонарних часових рядів, зокрема враховуючи автокореляцію та інтегруючи диференціацію. Модель NARX (Nonlinear AutoRegressive with eXogenous inputs) дозволяє враховувати нелінійні взаємодії та вплив зовнішніх факторів на динаміку цін акцій. Ця модель дозволяє включати екзогенні змінні, такі як новини або інші фінансові показники. Нейронні мережі, такі як LSTM (Long Short-Term Memory), виявляються ефективними для моделювання складних та нестаціонарних залежностей в часових рядах цін акцій Tesla. LSTM може автоматично виявляти та враховувати довгострокові зв'язки, а також адаптуватися до змінних умов ринку.

Крім того, для підвищення точності прогнозів, доцільно також використовувати техніки аугментації часових рядів, що забезпечить більш якісне оцінювання моделі на невеликих об'ємах даних.

2. МОДЕЛІ ПРОГНОЗУВАННЯ НЕЛІНІЙНИХ ПРОЦЕСІВ

Прогнозування поведінки валют на фінансових ринках є завданням високої складності. Фінансові ринки визначаються надто багатофакторною природою, і побудова надійної математичної моделі для точного передбачення їхньої динаміки залишається викликом. Це діяльність, де завжди присутні ризики, і щоденно тисячі трейдерів та інвесторів стикаються з несприятливими обставинами. Для подолання цих проблем гравці фінансових ринків створюють різноманітні математичні моделі для прогнозування та аналізу нелінійної динаміки цих процесів.

Для розробки прогнозних моделей у нашому дослідженні ми використовуємо три різновиди: класичну модель ARIMA (Autoregressive Integrated Moving Average), NARX (Nonlinear AutoRegressive with eXogenous inputs) та LSTM (Long Short-Term Memory).

ARIMA (Autoregressive Integrated Moving Average): Модель ARIMA є ефективним методом для прогнозування часових рядів і була запропонована Боксом і Дженкінсом на початку 1970-х років [1]. Ця модель базується на регресії залежної змінної лише за її власними значеннями, поточним значенням та значеннями запізнення випадкової помилки. ARIMA використовується для перетворення нестационарних часових рядів в стаціонарні та ефективно моделює їхню динаміку.

NARX (Nonlinear AutoRegressive with eXogenous inputs): Модель NARX дозволяє враховувати нелінійні взаємодії та вплив зовнішніх факторів на динаміку часових рядів [2, 3]. Цей підхід використовує авторегресійну структуру, але також допускає включення екзогенних вхідних змінних, що може покращити прогнозування в умовах складних та змінних середовищ.

Модель NARX зазвичай приймає форму множини нелінійних різницевих рівнянь:

$$y(t) = f(y(t-1), \dots, y(t-n_y), x(t-d), \dots, x(t-n_x), \varepsilon(t-1), \dots, \varepsilon(t-n_\varepsilon)) + \varepsilon(t),$$

де $x(t)$, $t \geq 1$ – вхідні змінні системи,

n_x, n_y – максимальні вхідні та вихідні лаги відповідно,

n_ε – максимальний лаг білого шуму ε ,

$y(t)$ – досліджуваний вихідний процес, або вихідні змінні,

f – невідома нелінійна функція,

$\varepsilon(t)$ – білий шум, змінна, що враховує наслідки вимірювання шуму, похибок моделювання та невимірних порушень, вважається обмеженою ($|\varepsilon(t)| < \delta$) та некорельованою із вхідними даними та n_ε , та обчислюється залежно від процесу оцінки.

LSTM (Long Short-Term Memory): Модель LSTM є варіацією рекурентних нейронних мереж і здатна ефективно моделювати складні та довгострокові залежності в часових рядах [5]. Вона особливо ефективна для задач прогнозування, де важливі динамічні та нестандартні взаємодії між даними. LSTM може автоматично визначати та враховувати важливі патерни у великому обсязі даних.

Мережі з довго короткостроковою пам'яттю (Long Short Term Memory) – особливий вид рекурентних нейронних мереж (РНМ), здатних до навчання довгостроковим залежностям. Вони дають можливість отримати високоякісні результати на великій різноманітності проблем і в даний момент широко застосовуються для моделювання нелінійних процесів [6, 7].

LSTM спеціально спроектовані таким чином, щоб уникнути проблеми довгострокових залежностей. Запам'ятовувати інформацію на тривалий період часу – це практично їх типова поведінка. LSTM мають ланцюгову структуру як і класичні РНМ, але повторюваний модуль

має іншу структуру. Замість одного нейронного шару тут присутні чотири шари, причому вони взаємодіють особливим чином (Рис. 1).

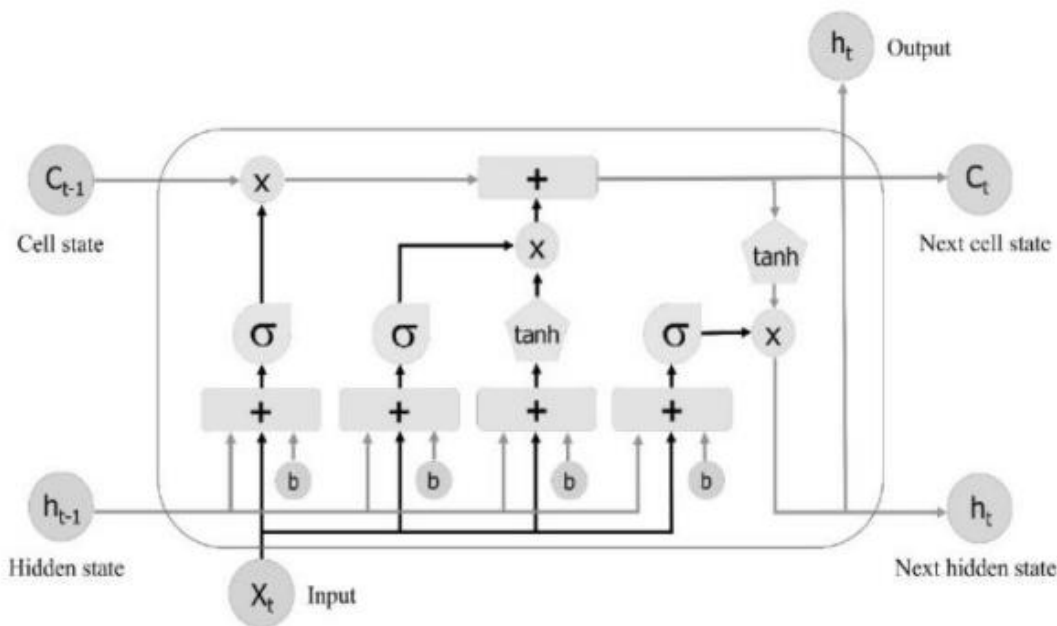


Рис 1. Структура LSTM

3. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Розглянемо функціонування наших розроблених моделей для прогнозу вартості акцій компанії Tesla – глобального гіганта у сфері електричних автомобілів та сонячних батарей. Tesla відома своєю революційною підходом до автомобільного виробництва, звертаючи увагу на стає покращення технологій і виробництва більш доступних електричних автомобілів. Крім того, вони мають величезний вплив на розвиток зеленої енергетики через виробництво сонячних батарей та систем для дому. В контексті прогнозування акцій Tesla, важливо визнати, що це завдання вкрай складне через багатофакторність та велику кількість невизначених змінних, що впливають на фінансові ринки. Tesla, як і багато інших технологічних компаній, піддається впливу різних чинників, таких як ринкова конкуренція, технологічні інновації, зміни у виробництві, регулююча політика та глобальні економічні умови.

Сучасні методи прогнозування акцій включають в себе використання алгоритмів машинного навчання, аналізу глибоких нейронних мереж, а також статистичних методів для аналізу ринкових тенденцій та динаміки. Проте, навіть з використанням передових технологій, точність прогнозування залишається високою ступенем невизначеності через непередбачувані зміни в економіці та інших сферах. Першою моделлю, яка використовувалась для прогнозування є модель ARIMA, для якої потрібно проаналізувати ряд на стаціонарність та обрати найкращу модель.

На основі результатів тестів на стаціонарність ряду, які включають зростаюче середнє та стандартне відхилення на графіку, стає очевидним, що наш ряд не є стаціонарним. Наступним кроком аналізу часового ряду є виконання його перетворення (диференціювання – перехід до попарних різниць сусідніх значень ряду), метою яких є зведення ряду до стаціонарного.

Модель ARIMA включає в себе три основні компоненти: авторегресійну (AR), інтегровану (I) та ковзну середню (MA). Компонента авторегресії AR враховує автокореляцію

між попередніми значеннями часового ряду, інтегрована компонента займається видаленням сезонності та трендів, а компонента ковзного середнього враховує кореляцію між спостереженнями та їхніми випадковими помилками.

Оптимальною моделлю виявилася модель ARIMA(1,1,0), а результати та графік вихідного часового ряду та прогнозу, побудованого за допомогою моделі ARIMA(1,1,0), зображено на рисунку 3.



Рисунок 3. Графік фактичних та прогнозних значень

У рамках цього дослідження ми спробуємо різні конфігурації моделі LSTM, і зробимо порівняльний аналіз їхньої ефективності в прогнозуванні майбутніх цін акцій Tesla. Ми також будемо оцінювати точність прогнозів та використовувати різні критерії адекватності, щоб визначити оптимальну модель для даного завдання. Порівняльне дослідження дозволить нам визначити, яка модель та конфігурація є найбільш підходящою для прогнозування цін акцій Tesla в реальних умовах фінансового ринку.

В нашому дослідженні надалі ми будемо використовувати нейронну мережу з наступною архітектурою (Рис. 4):

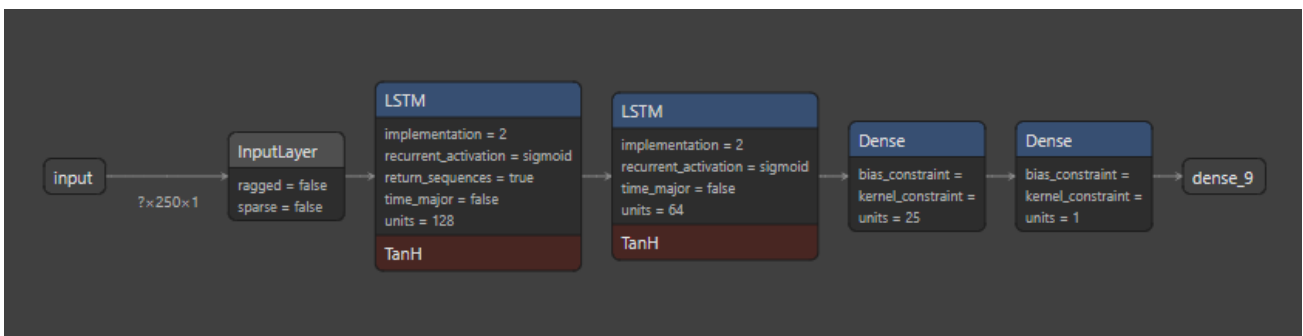


Рисунок 4. Архітектура НМ

Використання моделі нейронної мережі "LSTM-LSTM-Dense-Dense" для прогнозування часових рядів і послідовностей даних має численні переваги. Ця архітектура відзначається можливістю моделювання довгострокових залежностей в даних завдяки використанню LSTM-шарів. Додатково, вона дозволяє для більш точних та адаптивних прогнозів використовувати різні екзогенні фактори. Важливою перевагою є багаторівнева обробка і абстрагування інформації завдяки повторенню LSTM і Dense шарів. Ця гнучкість дозволяє адаптувати модель до конкретних завдань прогнозування.

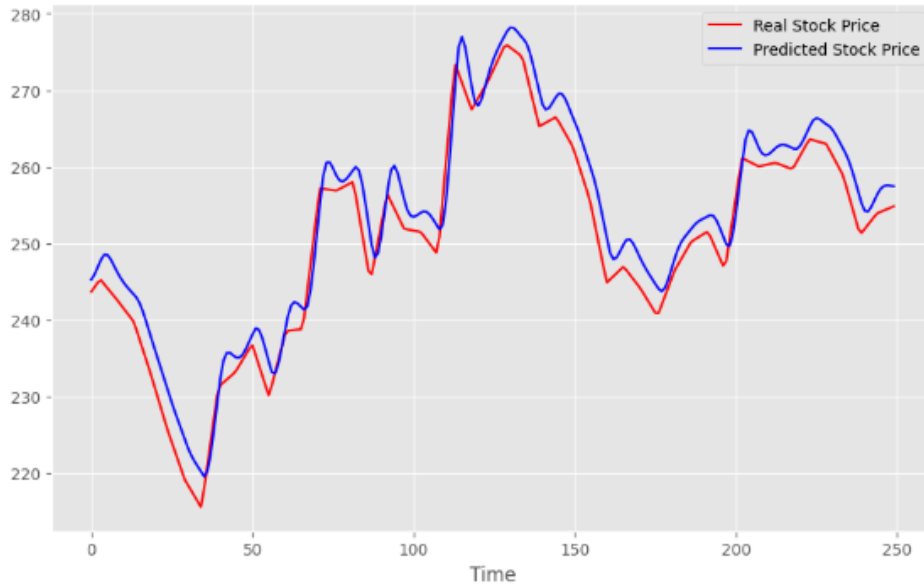


Рисунок 5. Графік фактичних значень та прогнозних за моделлю

Використаємо аугментацію для аналізу нашого часового ряду на прикладі NARX. TSAUG дозволяє створювати більше варіацій та аугментованих прикладів історичних даних для підвищення якості прогнозу. Отже, результати прогнозування після аугментації (Рис. 6):

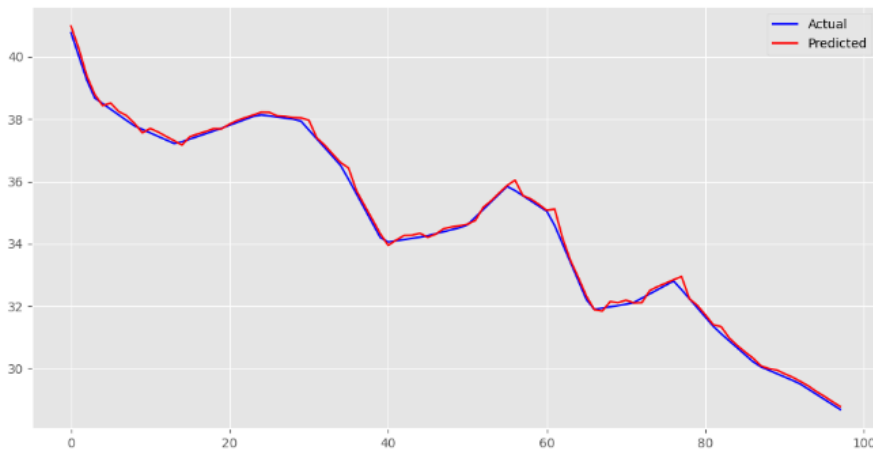


Рисунок 6. Графік фактичних значень та прогнозних за моделлю

Отримані результати свідчать про те, що модель прогнозування акцій Tesla є дуже точною і ефективною. Вона має дуже низькі помилки, високу точність і велику здатність пояснювати зміни в цінах акцій. Це може бути корисним для інвесторів та трейдерів для прийняття обґрунтованих рішень щодо купівлі або продажу акцій Tesla. Хоча початкова модель NARX також показувала чудові результати, в даному випадку вдалось ще покращити результати її роботи (Табл. 1):

Таблиця 1. Характеристики моделей

	ARIMA	LSTM	LSTM+NARX+TSAUG	NARX	NARX+TSAUG
MAE	0,19	10,7302	7,5464	4,224687	0,211251
MSE	0,224	137,504	91,2330	31,55268	0,165859
RMSE	0,3	12,04696	9,5516	5,617177	0,407258
R-Squared	0,9782	0,90434	0,9306	0,976779	0,9998808225637595

Моделі ARIMA та NARX дуже ефективно показують себе на даному набору даних, і їх прогнози можуть бути використані для подальшого прийняття рішення інвестором, але комбінація моделей LSTM+NARX+TSAUG та моделі NARX виявилась ще краще, а додаткове застосування лінійних інтерполяцій значно поліпшує точність прогнозу, і за результатами модель NARX+TSAUG є найкращою.

4. ВИСНОВКИ

Значна частина процесів на фондовому ринку, так само як і динаміка ціноутворення вартостей акцій компанії, є нестационарними часовими рядами, оскільки для них часто є характерними тренд, гетероскедастичність та сезонність, на них впливає велика кількість зовнішніх факторів та їх імовірнісні характеристики змінюються з часом, тобто є функціями часу.

Використання нестационарних даних часових рядів у фінансових моделях може призводити до побудови неякісних моделей та дає ненадійні результати прогнозів з великими похибками. Тому необхідно або зводити часовий ряд до стаціонарного вигляду, або застосовувати моделі, що здатні моделювати різницево-стаціонарні або інтегровані часові ряди. Серед таких можна виділити моделі типу Бокса-Дженкінса (ARIMA, SARIMA, з урахуванням зовнішнього впливу – ARIMAX, SARIMAX, нелінійні – NARMAX), моделі волатильності типу ARCH.

У ході нашого дослідження динаміки цін акцій компанії Tesla з початку 2020 року до жовтня 2021 року було проведено порівняльний аналіз різних моделей прогнозування. Результати показали, що ARIMA та ARIMAX моделі показали високу точність. Однак застосування нелінійної моделі NARMAX з урахуванням впливу зовнішніх факторів, дозволило досягти надзвичайно високої якості прогнозів. Ці результати свідчать про високу ефективність та надійність моделей NARMAX для прогнозування цін акцій Tesla в порівнянні з класичними ARIMA моделями.

Крім того, наше дослідження було збагачено за допомогою методів аугментації даних, які дозволили покращити якість прогнозів. Використання бібліотеки TSAUG перед моделлю NARX сприяло збільшенню обсягу доступних даних та покращило стійкість моделі до шумів і нерегулярностей в даних. Ця аугментація дала можливість створити різноманітні варіанти часового ряду, що сприяє кращому вивченню залежностей та динаміки цін акцій Tesla. Результати цього дослідження демонструють важливість використання аугментації даних як частину процесу прогнозування цін акцій та підкреслюють високу ефективність моделей NARMAX в поєднанні з цим підходом.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. П. І. Бідюк, В. Д. Романенко, та О. Л. Тимошук, Аналіз часових рядів. Київ, Україна: Політехніка, 2010.
2. José Maria P. Menezes Jr., Guilherme A. Barreto. Long-term time series prediction with the NARX network: An empirical evaluation. Neurocomputing, Volume 71, Issues 16–18, 2008, P. 3335-3343

3. R. Hyndman, G. Athanasopoulos, *Forecasting: Principles and Practice*. Melbourne, Australia: OTexts, 2013.
4. Shumway R., Stoffer D., *Time Series Analysis and Its Applications*. New York, USA: Springer, 2011.
5. S. Hochreiter, and J. Schmidhuber, “Long Short-Term Memory”, *Neural Computation*, vol. 9, no. 8, pp. 1735-1780, 1997.
6. F. A. Gers, D. Eck, and J. Schmidhuber, “Applying LSTM to Time Series Predictable Through Time-Window Approaches”, in *Proc. of International Conference on Artificial Neural Networks*, Vienna, 2001, pp. 669-676. doi: 10.1007/3-540-44668-0_93.
7. S. Hochreiter, Y. Bengio, and J. Schmidhuber, “Gradient flow in recurrent nets: the difficulty of learning long-term dependencies”. [Online]. Available: <http://www.bioinf.jku.at/publications/older/ch7.pdf> . Accessed on: Dec. 12, 2018.
8. “Understanding LSTM Networks”. [Online]. Available: <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>. Accessed on: March 9, 2018.
9. F. Chollet, and J. Allaire, *Deep Learning with R*. New York, USA: Manning, 2018.
10. I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, USA: MIT Press, 2016

СИСТЕМА ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ДЛЯ ВИБОРУ S-МОДЕЛЕЙ ЕКОНОМІЧНОГО ЗРОСТАННЯ

Кузьмінчук А.В.¹, Лопатін О.К.²

Національний технічний університет України «Київський політехнічний інститут
імені Ігоря Сікорського», Київ, Україна

¹ anatoliykozminchuk@gmail.com, ² lopatinalexey142@gmail.com

У цій роботі вперше досліджується двухфакторна агрегована S-тренд виробнича функція $y'(t)=P(t)*S(x(t),A,B,a,m,u)$ на основі емпіричних даних, а саме на прикладі індексу ВВП на душу населення Сінгапуру (1960-2022). Тут $y(t)$ вихідний ряд статистичних даних, $P(t)$ -коефіцієнт загальної факторної продуктивності, $y'(t)=S(x(t),A,B,a,m,u)$ детермінована функція, що апроксимує ряд $y(t)$, $S(x(t),A,B,a,m,u) = u + A(1 + B * \exp(-a(t - m)))^{-1}$. Позначимо через $G_y(t), G_p(t), G_s(t)$ річні швидкості зростання у відсотках вихідного ряду, TFP коефіцієнта і внеску фактора "нагромадженням капіталу" Має місце тотожність $G_p(t)=G_y(t)-G_s(t)$. У цій тотожності треба знати $G_s(t)$. Щоб знати $G_s(t)$, потрібно визначити $y'(t)=S(x(t),A,B,a,m,u)$ і отримати оцінку апроксимації $MARE(y(t),y'(t)) = (100\%)/N \sum_{t=1}^N (\frac{y(t)-y'(t)}{y(t)})$, критерій $MARE < 10\%$ вважається відмінним результатом, $10\% < MARE < 20\%$ вважається хорошим результатом.

Ключові слова: S-подібна виробнича функція, сукупна факторна продуктивність, ефективність інвестицій, принцип Чарльза Р. Халтена, фактор продуктивності, фактор нагромадження капіталу.

1. ВСТУП

У сучасній економічній теорії існує широкий спектр методів, які дозволяють розрахувати рівень сукупної факторної продуктивності (далі – СФП), що є одним з ключових показників ефективності виробництва як на рівні окремих фірм, так і на рівні галузей, регіонів та країн. Зростання капіталу, праці та технічного прогресу є трьома основними джерелами економічного зростання країни та регіону. Темпи зростання робочої сили обмежуються темпами зростання населення, особливо в індустріально розвинених країнах, де населення рідко зростає більш ніж на два відсотки на рік, навіть з урахуванням міжнародної міграції. Отже, темпи зростання капіталу (фізичного і людського) і технологічний прогрес є основними джерелами більшої частини економічного зростання.

Модель Солоу (модель Солоу-Свона) – модель екзогенного економічного зростання базується на роботах Солоу [5] і Свона [6], а також на роботі Солоу [7], в якій було введено поняття сукупної продуктивності факторів виробництва (TFP), що отримала назву залишкової Солоу. Модель Солоу вважається відправною точкою для всіх сучасних моделей економічного зростання.

Залишок Солоу все ще залишається, після багатьох десятиліть, робочою конячкою емпіричного аналізу економічного зростання. Було опубліковано тисячі сторінок досліджень, і щороку їх публікується все більше. Приведемо підбірку найбільш актуальних для побудови нових агрегованих виробничих функцій за останні п'ять років: Цонас, Майк Г., Полеміс,

Майкл Л., (2019), Цуніс Ніколас, Стедман Ян; (2021), Френсіс Девід К., Нона Каралашвілі, Хібрет Маемір, Хорхе Родрігес Мез (2020), Вілан Карл (2021); Дандан, Дуду (2020); Харб Жорж, Басіль Шарбе (2023): Узагальнення та подальший розвиток методів вимірювання сукупної факторної продуктивності. Бюро статистики праці США, (випуск 23 березня 2023 року) У цій статті визначено ключові терміни та поняття, які є ключовими для розуміння того, як Бюро статистики праці (БСП) розробляє показники продуктивності для різних рівнів американської економіки.

2. ВИРОБНИЧА ФУНКЦІЯ S-ТРЕНДУ

Нехай досліджується часовий ряд

$$y(t) = y(t_1), \dots, y(t_T). \quad (1)$$

Наприклад, це може бути ВВП на душу населення деякої країни. Для часових рядів прийнято розглядати його рівні як суміш чотирьох компонент – трендової, циклічної, сезонної та випадкової складових, які неможливо виміряти безпосередньо [1]

$$y(t_i) = T(t_i) + C(t_i) + S(t_i) + \varepsilon(t_i).$$

$T(t_i)$ – тренд, основна тенденція розвитку досліджуваного процесу в часі. Цей тренд є детермінованою складовою, незалежною від циклічних, сезонних та випадкових складових. Він може бути представлений у вигляді більш-менш гладкої кривої.

Компоненти часового ряду $T(t_i)$ не є спостережуваними. Вони є теоретичними величинами. Оцінка майбутніх членів ряду зазвичай робиться за допомогою прогнозної моделі. Прогнозна модель – це модель, яка апроксимує тренд. Ми обираємо S-образну криву Верхольста як модель прогнозування тренду (МПТ)

$$y'(t) = u + \frac{A}{1+B \cdot \exp(-a(t-m))} = S(t, A, B, a, m, u) \quad (2)$$

Наслідок 1. Тип прогнозної моделі можна встановити за графіком $(y(t), t)$ вихідного ряду. Отже, вихідні дані (1) мають бути апроксимовані МПТ (2).

Наслідок 2. Точність апроксимації ряду (1) оцінюється за критерієм MAPE.

Формулювання проблеми. Метою даної роботи є розробка нової виробничої функції вигляду $y'(t) = S(x(t), A, B, a, m, u)$, $t=1, \dots, T$.

У цьому випадку факторний показник $x(t)=t$ характеризує внесок компоненти "нагромадження капіталу" в економічний вихід. Це час із кроком в 1 рік (звичайний крок статистичних таблиць).

$$MAPE(y(t), y'(t)) = \frac{100\%}{N} \sum_{t=1}^N \left| \frac{y(t) - y'(t)}{y(t)} \right|,$$

де $y(t)$ – координати точкового графіка вихідного ряду, $y'(t)$ – координати, що будуються.

3. МЕТОДОЛОГІЯ: ГРАФО-АНАЛІТИЧНИЙ МЕТОД. РУЧНА ОПТИМІЗАЦІЯ

Діаграма технологічного укладу ВВП на душу населення в Сінгапурі 1960–2022.

Джерело даних (Табл. 1): https://www.google.ru/publicdata/explore?ds=d5bncppjof8f9_

Таблиця 1. ВВП на душу населення Сінгапуру в тисячах доларів (у поточних доларах США)

NO	Year	Y	NO	Year	Y	NO	Year	Y	NO	Year	Y
1	1960	4,28	17	1976	27,59	33	1992	161,36	49	2008	400,09
2	1961	4,49	18	1977	28,46	34	1993	182,90	50	2009	389,27
3	1962	4,72	19	1978	31,94	35	1994	215,52	51	2010	472,37
4	1963	5,11	20	1979	39,01	36	1995	249,15	52	2011	538,91
5	1964	4,86	21	1980	49,28	37	1996	262,33	53	2012	555,48
6	1965	5,17	22	1981	55,97	38	1997	263,76	54	2013	569,67
7	1966	5,67	23	1982	60,78	39	1998	218,29	55	2014	575,65
8	1967	6,26	24	1983	66,33	40	1999	217,97	56	2015	556,46
9	1968	7,09	25	1984	72,28	41	2000	238,53	57	2016	568,96
10	1969	8,13	26	1985	70,02	42	2001	217,00	58	2017	611,65
11	1970	9,26	27	1986	68,00	43	2002	221,60	59	2018	668,37
12	1971	10,71	28	1987	75,39	44	2003	237,30	60	2019	660,70
13	1972	12,64	29	1988	89,14	45	2004	276,08	61	2020	612,74
14	1973	16,85	30	1989	103,95	46	2005	299,61	62	2021	777,10
15	1974	23,42	31	1990	118,62	47	2006	337,68	63	2022	828,08
16	1975	24,90	32	1991	145,02	48	2007	394,33			

Побудова циклів S-трендів.

Наведемо в таблиці 2 розбиття вибірки на ділянки для моделювання, а також в таблиці 3 опишемо загальну структуру циклів.

Таблиця 2. Визначення циклів верхніми та нижніми асимптотами

Theta	1	2	3	4	5	6
Значення	4	58	262	221	555	958

Таблиця 3. Вигляд S-кривих

Цикл 1	Цикл 2	Цикл 3	Цикл 4	Цикл 5
$z_1(t) = \frac{T_2 - T_1}{D_1} + T_1$	$z_2(t) = \frac{T_3 - T_2}{D_2} + T_2$	$z_3(t) = \frac{T_3 - T_4}{D_3} + T_4$	$z_4(t) = \frac{T_5 - T_4}{D_3} + T_4$	$z_5(t) = \frac{T_7 - T_5}{D_5} + T_5$

Уточнення до таблиці 3: $z_i(t) = 1 + B_i \text{EXP}(-a_i * (t - m_i))$, $i = 1, \dots, 5$.

На рисунках 1 та 2 можемо спостерігати результати моделювання S кривих.

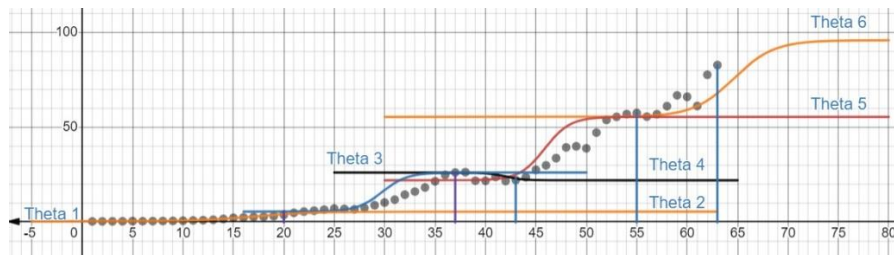


Рисунок 1. Вище наведено $S_i(t) = 1, \dots, 5$, визначені їхніми верхніми та нижніми асимптотами

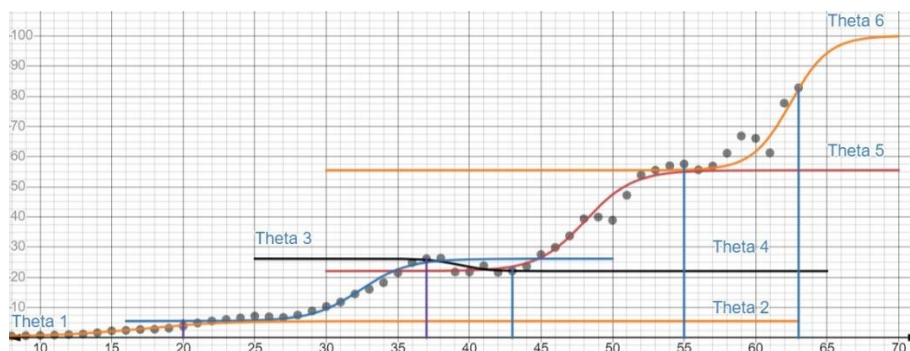


Рисунок 2. Вище наведено оптимізовані $S_i(t) = 1, \dots, 5$, визначені їхніми верхніми та нижніми асимптотами

4. ОГЛЯД РЕАЛІЗОВАНОЇ СППР

Для розв'язання задачі було обрано метод градієнтного спуску, а саме його модифікацію на основі частинних похідних параметрів. Модифікуємо метод так, щоб мінімізувати критерій різниці квадратів [12]:

Визначимо шукану функцію:

$$f(x_i) = \frac{A}{1 + B \exp(a * (m - x_i))}$$

Визначимо формулу різниці квадратів з огляду на поставлену задачу:

$$E = \sum_{i=1}^n (y_i - f(x_i))^2 = \sum_{i=1}^n \left(y_i - \frac{A}{1 + B \exp(a * (m - x_i))} - u \right)^2$$

Для використання градієнтного методу визначимо похідні для всіх параметрів
Визначимо ці похідні:

$$\frac{dE(A, B, a, m, u)}{dA} = \sum_{i=1}^n \left(\left(\frac{-2}{1 + B \exp(a * (m - x_i))} \right) * \left(y_i - \frac{A}{1 + B \exp(a * (m - x_i))} - u \right) \right)$$

$$\frac{dE(A, B, a, m, u)}{dB} = \sum_{i=1}^n \left(\left(\frac{2A \exp(a * (m - x_i))}{(1 + B \exp(a * (m - x_i)))^2} \right) * \left(y_i - \frac{A}{1 + B \exp(a * (m - x_i))} - u \right) \right)$$

$$\frac{dE(A, B, a, m, u)}{da} = \sum_{i=1}^n \left(\left(\frac{2AB(m - x_i) \exp(a * (m - x_i))}{(1 + B \exp(a * (m - x_i)))^2} \right) * \left(y_i - \frac{A}{1 + B \exp(a * (m - x_i))} - u \right) \right)$$

$$\frac{dE(A, B, a, m, u)}{dm} = \sum_{i=1}^n \left(\left(\frac{2aAB(m - x_i) \exp(a * (m - x_i))}{(1 + B \exp(a * (m - x_i)))^2} \right) * \left(y_i - \frac{A}{1 + B \exp(a * (m - x_i))} - u \right) \right)$$

$$\frac{dE(A, B, a, m, u)}{du} = \sum_{i=1}^n \left(-2 * \left(y_i - \frac{A}{1 + B \exp(a * (m - x_i))} - u \right) \right)$$

Наступним етапом потрібно визначити формат кроків ітерації:

$$\begin{cases} A_{n+1} = A_n + \lambda_n^1 \frac{dE(A, B, a, m, u)}{dA} ; \\ B_{n+1} = B_n + \lambda_n^2 \frac{dE(A, B, a, m, u)}{dB} ; \\ a_{n+1} = a_n + \lambda_n^3 \frac{dE(A, B, a, m, u)}{da} ; \\ m_{n+1} = m_n + \lambda_n^4 \frac{dE(A, B, a, m, u)}{dm} ; \\ u_{n+1} = u_n + \lambda_n^5 \frac{dE(A, B, a, m, u)}{du} , \end{cases}$$

де $\lambda_n^i, i = 1 \dots 5$, число менше 1 і більше 0, яке дозволяє впливати на розмір кроку ітерації.

При створенні до СППР було висунуто наступні умови: інтерактивність – система повинна мати інтуїтивно зрозумілий графічний інтерфейс користувача (GUI), який дозволяє користувачам взаємодіяти з системою за допомогою кнопок, меню, графічних елементів тощо; доступність – система повинна бути легкодоступною для користувачів, чим менше користувачу потрібно зробити для початку користування продуктом, тим більше ймовірності привернути його увагу; повнота – в системі має бути можливість провести повноцінний аналіз впливу: починаючи від завантаження даних для тренування моделі, закінчуючи використанням нових даних на створеній моделі та вивантаженням отриманих результатів.

Блок-схема алгоритму роботи з СППР зображена на рисунку 3.

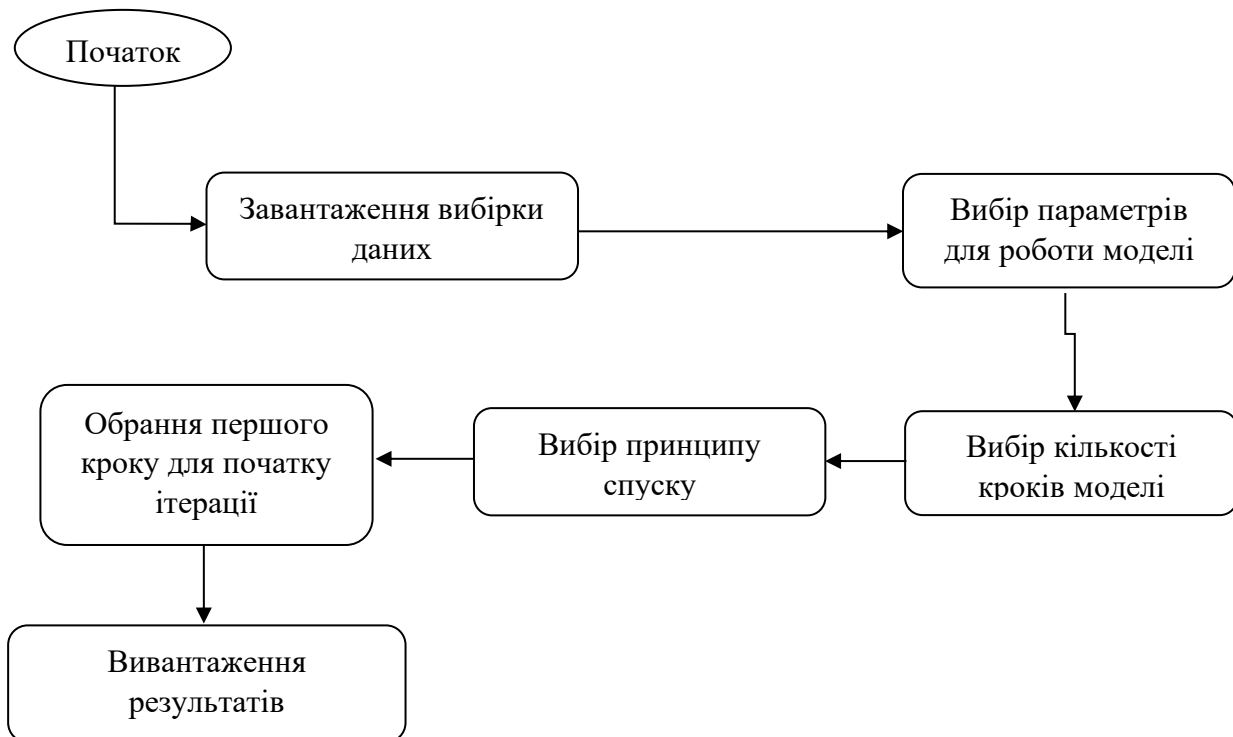


Рисунок 3. Блок-схема алгоритму роботи з СППР

Реалізацію вищезгаданих етапів роботи з СППР зображено на рисунках 4 та 5. На рисунку 4 зображено базовий інтерфейс додатку. На рисунку 5 зображено результат роботи

додатку. За наведеним рисунком можна побачити, що інтерфейс є інтуїтивно зрозумілим і СППР дозволяє провести аналіз, тобто вона відповідає висунутим до неї вимогам.

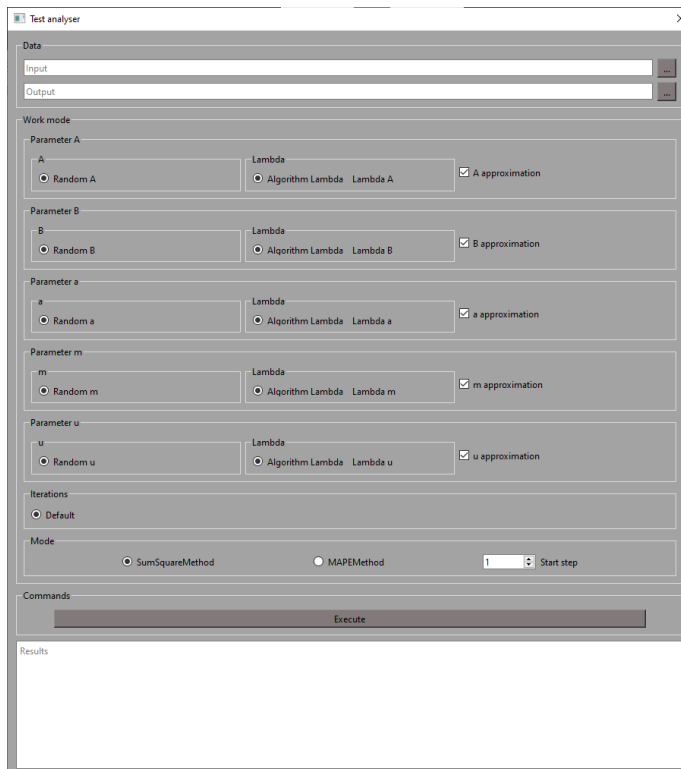


Рисунок 4. Інтерфейс СППР

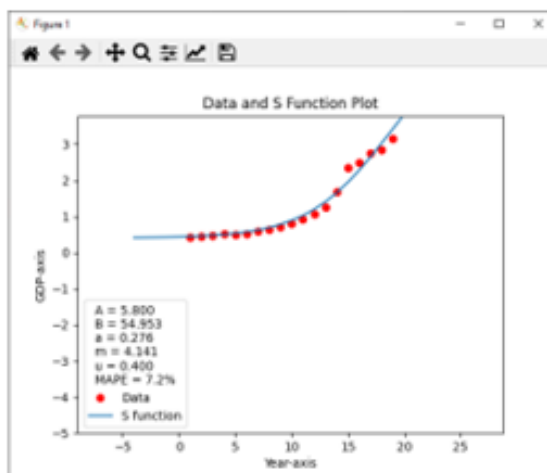


Рисунок 5. Інтерфейс результату роботи СППР

4. РЕЗУЛЬТАТИ РОБОТИ

Виконаємо обробку циклів вище за допомогою СППР, результати представлені на малюнку 5 та в таблиці 3.

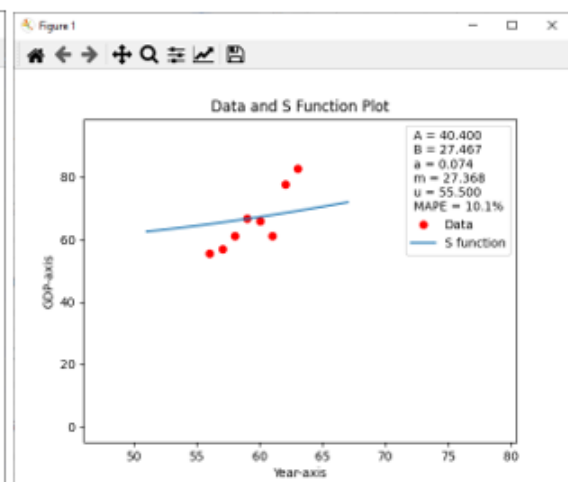
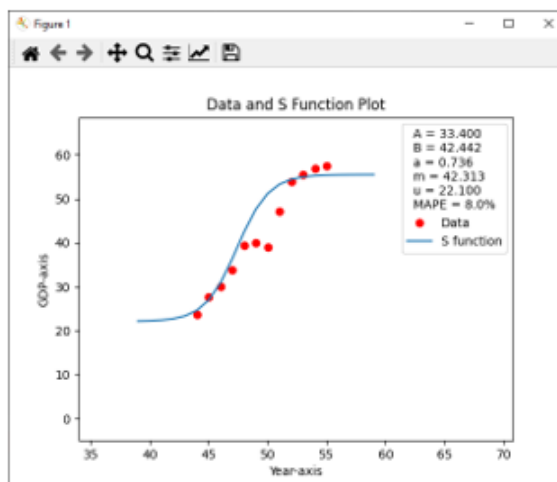
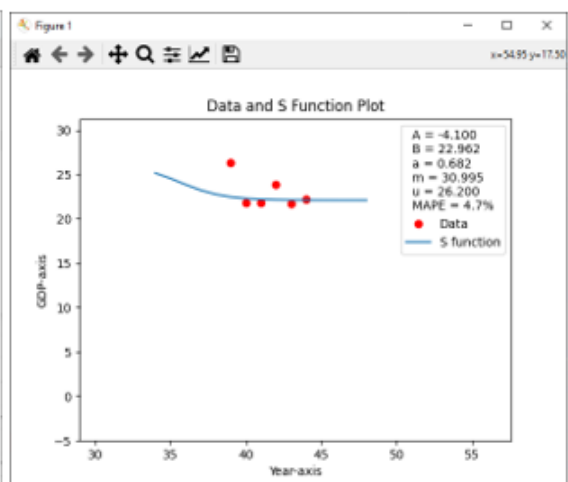
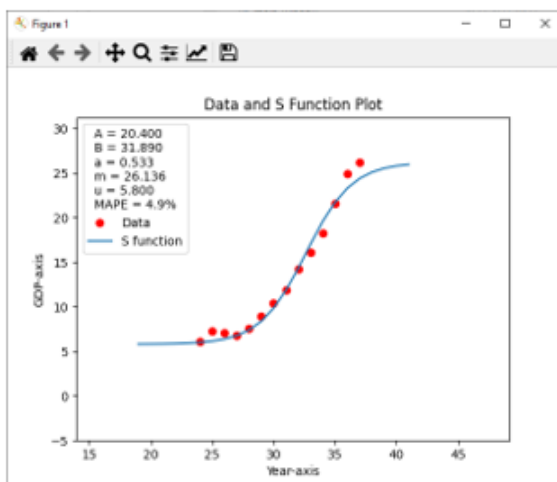
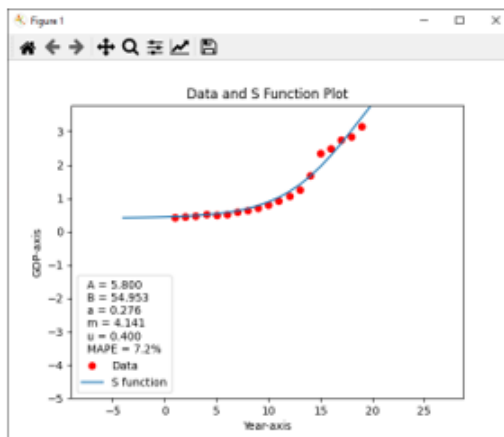


Рисунок 6. Результати обробки за допомогою СППР

Таблиця 4. Зведені результати роботи

Цикл 1				
	B	a	m	MAPE%
general	4	0,5	13	21%
selecting	4	0,35	13	8%
gradient descent	54,953	0,276	4,141	7,2%
Цикл 2				
	B	a	m	MAPE%
general	20	0,8	26	27%
selecting	30	0,6	26,7	8%
gradient descent	31,89	0,53	26,13	4,9%
Цикл 3				
	B	a	m	MAPE%
general	40	-1,2	45	9%
selecting	43	-1	43	5%
gradient descent	22,96	0,68	30,99	4,7%
Цикл 4				
	B	a	m	MAPE%
general	10	0,8	43	22%
selecting	6,4	0,6	45	6%
gradient descent	42,442	0,736	42,313	8,0%
Цикл 5				
	B	a	m	MAPE%
general	100	0,5	60	16%
selecting	6,2	0,75	60	6%
gradient descent	27,467	0,074	27,368	10,1%

5. ВИСНОВКИ

В рамках роботи було проведено роботу над побудовою S-кривої та розроблено СППР на його основі.

Порівняльний аналіз на основі рисунку 6 та таблиці 4 показав, що для розглянутого набору даних СППР прискорив обробку інформації та надав оптимальні результати.

Під час розробки СППР було висунуто вимоги до неї та обрано інструменти для створення. Приклад роботи з СППР та огляд інтерфейсу показують, що СППР відповідає висунутим до неї вимогам.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Бідюк, П. І., Романенко, В. Д., Тимошук, О. Л. (2010). Аналіз часових рядів: навчальний посібник. Київ, Київська область: Політехніка, 317 с.
2. Cıpra T. (2020). Time Series in Economics and Finance. Springer, 353 p

3. Dandan , Doudou (2020) Comparison and Analysis of Measurement Methods of Total Factor Productivity International Journal of Frontiers in Engineering Technology ISSN 2706-655X Vol.2, Issue 1: 18-30, DOI: 10.25236/IJFET.2020.020102
4. Harb Georges, Bassil Charbe (2023). TFP in the Manufacturing Sector: Long-Term Dynamics, Country and Regional Comparative Analysis, *Economies* 2023, 11(2), <https://doi.org/10.3390/economies11020034>
5. Solow R. M. (1956), A contribution to the Theory of Economic Growth, *The Quarterly Journal of Economics*, 70 (1), 65—94.
6. Swan T. W. (1956), Economic growth and capital accumulation, *The Economic Record*, 32(2), 334—361
7. Solow R. M. (1957) Technical change and the aggregate production function, *The Review of Economics and Statistics*, 39(3), \312—320.
8. Tsionas, Mike G., Polemis, Michael L., (2019). On the estimation of total factor productivity: A Bayesian non-arametric approach, *European Journal of Operational Research*, Elsevier, vol. 277(3), pp 886-902.
9. Tsounis Nicholas, Steedman Ian (2021), A New Method for Measuring Total Factor Productivity Growth Based on the Full Industry Equilibrium Approach: The Case of the Greek Economy *Economies* 2021, 9, 114, pp 1-21
10. The Bureau of Labor Statistics (BLS) produces measures of productivity for different levels of the U.S. economy. Total factor productivity (TFP) in the private nonfarm business sector (release March 23, 2023) USDL-23-0540 Technical information: (202) 691-5606 • Productivity@bls.gov • www.bls.gov/productivity Media contact: (202) 691-5902 • PressOffice@bls.gov
11. Wikipedia World Development Indicators." World Bank". https://www.google.ru/publicdata/explore?ds=d5bncppjof89_
12. Wikipedia Gradient descent https://en.wikipedia.org/wiki/Gradient_descent

РОЗРОБКА МОДЕЛЕЙ ОЦІНЮВАННЯ РИЗИКІВ ЗЕЛЕНИХ ПРОЕКТІВ

Кузнєцова Н.В., Шевчук О.С.¹

Національний технічний університет України «Київський політехнічний інститут
ім. Ігоря Сікорського»

¹ oleksii.shevchuk@ukr.net

Забруднення навколишнього середовища є однією з головних глобальних проблем сучасного світу, яка регулярно обговорюється у різних політичних та наукових колах. Світове товариство розуміє катастрофічність можливих наслідків, тому шукає варіанти для запобігання світової екологічної кризи. Розробляються еко-технології, впроваджуються відповідні закони, а також ведеться планування та імплементація зелених проектів. Ці проекти націлені на відновлення та збереження екосистеми, проте постає питання оцінки ефективності та ризикованості таких проектів. Виконуються дослідження щодо оцінки ризикованості зелених проектів як класичними методами, так і на основі нових розроблених методів. У роботі розглянуто класичні методи розв'язання задачі кредитного скорингу, розроблено моделі прогнозування ризиків конкретно зелених проектів

Ключові слова: оцінка ризиків, кредитний скоринг, зелені проекти.

1. ВСТУП

За час існування планети екосистема зазнавала значних змін, спричинених як зовнішніми чинниками, так і внутрішніми. Найбільш значущим внутрішнім чинником є вплив мешканців планети на її екосистему і основну роль у зміні екосистеми світу відіграє людина. Кількість населення неспинно зростає, що збільшує потреби у природних ресурсах. Століттями людство чинило негативний вплив на екосистему Землі. Однак зростає і рівень освіченості суспільства, що дозволило зрозуміти, що у боротьбі з природою переможцем людина бути не може. Усвідомивши небезпеку та критичність ситуації в екологічній сфері, найбільш розвинені країни світу почали вивчати всі аспекти даної проблеми та шукати способи її вирішення [1].

Новітній і на даний час найбільш перспективний спосіб вирішення глобальної екологічної проблеми – це розробка та впровадження зелених проектів. Чіткого формалізованого визначення зелених проектів немає, але усі варіанти визначення включають те, що дані проекти повинні бути спрямовані на зниження кількості споживання природних або енергетичних ресурсів або на їх відновлення. Крім цього, зелені проекти можуть приносити і економічні переваги та дивіденди, хоча це не є їх основною метою.

У зв'язку із появою зелених проектів постає питання коректної оцінки їх економічної та екологічної доцільності. Класичні методи та моделі оцінки проектів орієнтуються лише на економічну складову. У випадку зелених проектів такий підхід не буде повністю коректним. У такому разі, і рішення задачі кредитного скорингу зелених проектів доцільно переосмислити.

2. ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

Метою даної роботи є дослідження існуючих методів та моделей вирішення задачі кредитного скорингу, побудова моделей вирішення задачі оцінки та прогнозування ризиків зелених проектів. Результатом роботи буде розробка моделей аналізу та оцінки ризиків зелених проектів на основі математичних методів, а також порівнянні їх ефективності. Таким чином, об'єктом дослідження даної роботи є ризики інвестування у зелені проекти, а також способи їх аналізу та оцінки. Предметом дослідження є математичні методи та моделі аналізу та прогнозування ризиків зелених проектів.

3. ОГЛЯД МЕТОДІВ ОЦІНКИ РИЗИКІВ

Одним із класичних рішень задачі оцінки кредитних ризиків є регресійний аналіз. Регресійний аналіз передбачає пошук певних взаємозв'язків між залежною змінною, яку називають результуючою або цільовою та однією чи декількома незалежними змінними – предикторами. Рівняння регресії є зображенням зв'язку між цими змінними у вигляді математичної формули, а самі зв'язки можуть бути явними або неявними. Явні зв'язки, у більшості випадків, можуть бути завчасно описані за допомогою відомих формул. Неявний зв'язок не може бути переданий у вигляді формул, тому, перш за все, цей зв'язок потрібно знайти. Суть методу полягає у побудові рівняння регресії та пошуку його коефіцієнтів. Для цього вирішується система рівнянь за допомогою метода Крамера.

Для задачі класифікації, використовується логістична регресія. Її особливість полягає у використанні сигмоїдної функції:

$$p(x_i) = \frac{1}{1 + \exp(-f(x_i))}$$

Значення $p_i(x)$ часто інтерпретують як прогнозовану ймовірність того, що результат для заданих предикторів дорівнює 1. Отже, $1 - p(x)$ – це ймовірність того, що результат дорівнює 0. У цьому випадку 0 та 1 розглядаються як два різні класи, між якими розподіляються елементи.

Ще одним популярним методом вирішення задачі кредитного скорингу є метод дерев рішень. Він є одним із найбільш вживаних методів інтелектуального аналізу через свою простоту, вільність у неоднозначності та надійність, навіть при відсутності певних вхідних значень. Древа рішень можуть працювати як із дискретними, так і з неперервними величинами як цільової змінної так і предикторів.

Древа рішень складаються із декількох компонент, основними з яких є вузли та гілки. Самі вузли також поділяються на декілька типів: корінь, внутрішні вузли, листки.

Побудова дерева рішень відбувається у декілька кроків. Найважливіші з них:

- розбиття;
- зупинка;
- відсікання.

Задачею розбиття дерева є розділення батьківських вузлів на більш чисті дочірні вузли цільової змінної. Процес побудови моделі відбувається шляхом визначення найважливішого для вузла предиктора і подальшого розбиття множини на дві або більше категорій. Вибір найбільш значущого предиктора відбувається шляхом аналізу характеристик, пов'язаних зі ступенем "чистоти" отриманих дочірніх вузлів. Ці характеристики включають ентропію, індекс Джині, помилку класифікації, інформаційний вигравш та коефіцієнт вигравшу [2].

Процедура розбиття триває доти, доки не будуть виконані заздалегідь визначені критерії однорідності або зупинки. У більшості випадків не всі потенційні предиктори будуть використані для побудови моделі дерева рішень, але можливі випадки, коли певна вхідна

змінна може бути використана кілька разів на різних рівнях дерева рішень. До загальних параметрів, що використовуються в правилах зупинки, належать:

- мінімальна кількість записів у листку;
- мінімальна кількість записів у вузлі до розбиття;
- глибина (тобто, кількість кроків) будь-якого листка від кореневого вузла.

У деяких ситуаціях правила зупинки не працюють належним чином. Альтернативний спосіб побудови дерева рішень полягає в тому, щоб спочатку виростити велике дерево, а потім обрізати його до оптимального розміру, видаливши вузли, які надають менше додаткової інформації [3].

Метод випадкового лісу отримав свою популярність через зручність та адаптивність, що дозволяє йому вирішувати як задачі класифікації, так і регресії [4]. Основною перевагою методу є його здатність обробляти складні набори даних та запобігати перенавчанню. Випадкових ліс відноситься до ансамблевих методів і використовує бегінг при побудові моделі. Ансамбль складається з набору індивідуально навчених моделей, які потім об'єднуються в одну кінцеву модель. Дослідження показують, що ансамбль часто є більш точним, ніж будь-який з окремих моделей в ансамблі.

Бегінг (bagging) це один із методів створення ансамблів [4]. Бегінг означає бутстреп-агрегування, яке працює на основі концепції бутстреп-вибірок. Якщо початковий навчальний набір даних має розмір N і потрібно згенерувати m індивідуальних моделей для побудови ансамблю, тоді m різних навчальних наборів, кожен з яких має розмір N , генеруються з вихідного набору даних за допомогою вибірки із заміною. Множинні класифікатори, згенеровані в пакетах (bag), є незалежними один від одного.

Випадковий ліс відноситься до ансамблевих методів, тому, у випадку задачі класифікації, модель випадкового лісу буде складатися з колекції моделей дерев рішень, кожна з яких буде віддавати свій голос за обраний нею клас, під час прогнозування. Щоб досягти різноманітності серед базових дерев рішень, використовується підхід рандомізації, який добре працює з бегінгом.

Більш сучасний спосіб вирішення задачі кредитного скорингу – це бустингові методи. Градієнтний бустинг, так само як і випадковий ліс, відноситься до ансамблевих методів. Однак, якщо випадковий ліс використовує бегінг, то градієнтний бустинг використовує бустинг. У випадку бустингу, кожній вибірці присвоюються ваги з навчального набору даних. Якщо потрібно згенерувати m моделей, то вони генеруються послідовно так, щоб за одну ітерацію генерувалась одна модель. Для генерації класифікатора c_i ваги навчальних вибірок оновлюються на основі результатів класифікації класифікатора c_{i-1} . Моделі, згенеровані у випадку використання бустингу, залежать одна від одної [5].

AdaBoost – одна з модифікацій методів градієнтного бустингу, яка часто використовується для задач класифікації. Алгоритм AdaBoost отримує на вхід певну навчальну вибірку і викликає заданий слабкий алгоритм навчання багаторазово на кожній ітерації. Однією з основних ідей алгоритму є підтримка розподілу ваг D_t на навчальній множині. Спочатку всі приклади мають однакову вагу, але з кожною ітерацією вага неправильно класифікованих прикладів збільшується, таким чином слабкий "учень" змушений зосередитися на важких прикладах навчального набору. Задача слабого "учня" полягає у пошуку слабкої гіпотези h_t , яка б підійшла для розподілу D_t . Якість слабкої гіпотези вимірюється її похибкою. Після отримання слабкої гіпотези h_t , AdaBoost обирає параметр $\alpha_t = \frac{1}{2} \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$ і оновлює розподіл D_t так, щоб збільшити вагу прикладів, класифікованих неправильно, за допомогою гіпотези h_t , і зменшити вагу правильно класифікованих прикладів. Остаточна гіпотеза H є зваженою більшістю голосів з T слабких гіпотез.

4. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Для вирішення задачі кредитного скорингу зелених проектів було створено набір даних, шляхом комбінації двох окремих наборів – набору даних про кредитоспроможність компанії та набору даних про її вплив на навколишнє середовище. Таким чином, результуючий датасет складається з наступних двох:

- датасет кредитного рейтингу компаній та їх фінансових показників;
- датасет впливу діяльності компаній на навколишнє середовище.

Кредитний рейтинг компаній виражає здатність компанії погашати свої борги перед кредиторами і ґрунтується, переважно, на фінансовій звітності. Набір даних кредитного рейтингу компаній містить саме ці метрики, що вказують на економічне становище компанії.

Набір даних впливу діяльності компанії на навколишнє середовище характеризує вплив компанії на навколишнє середовище у грошовому еквіваленті.

Формування навчального набору здійснювалось на основі двох наборів даних за принципом еквівалентності назви компанії, та року, коли було проведено оцінку кредитного рейтингу та оцінку впливу на навколишній світ. У результаті було отримано датасет, що містить 847 відбитків даних, серед яких:

- 229 відбитків із рейтингом “А”;
- 534 відбитків із рейтингом “В”;
- 13 відбитків із рейтингом “С”;

У дослідженні було побудовано чотири моделі: логістичної регресії, дерев рішень, випадкового лісу та AdaBoost. Кожна з моделей будувалась для трьох різних комбінацій вхідних параметрів:

- набір даних лише із кредитним рейтингом компаній та їх фінансовими показниками;
- набір даних лише із показниками впливу компаній на навколишнє середовище;
- комбінований набір даних кредитного рейтингу та впливу компаній на навколишнє середовище.

Результати кожної моделі для кожного набору даних можна побачити у Таблицях 1-4.

Таблиця 1. Метрики моделі логістичної регресії

Фінансові навчальні дані			Дані впливу на навколишнє середовище			Комбіновані навчальні дані		
Точність	0,7164		Точність	0,6824		Точність	0,6824	
Зважений F1 score	0,6968		Зважений F1 score	0,5982		Зважений F1 score	0,6231	
F1-score			F1-score			F1-score		
“А”	“В”	“С”	“А”	“В”	“С”	“А”	“В”	“С”
0,4946	0,8049	0	0,1846	0,8029	0	0,2740	0,7970	0

Таблиця 2. Метрики моделі дерева рішень

Фінансові навчальні дані			Дані впливу на навколишнє середовище			Комбіновані навчальні дані		
Точність	0,8118		Точність	0,8059		Точність	0,8353	
Зважений F1 score	0,8118		Зважений F1 score	0,8063		Зважений F1 score	0,8387	
F1-score			F1-score			F1-score		
“А”	“В”	“С”	“А”	“В”	“С”	“А”	“В”	“С”
0,7273	0,8596	0	0,7207	0,8546	0	0,7719	0,8739	0,5

Таблиця 3. Метрики моделі випадкового лісу

Фінансові навчальні дані			Дані впливу на навколишнє середовище			Комбіновані навчальні дані		
Точність	0,8706		Точність	0,8294		Точність	0,9	
Зважений F1 score	0,8700		Зважений F1 score	0,8234		Зважений F1 score	0,8973	
F1-score			F1-score			F1-score		
“А”	“В”	“С”	“А”	“В”	“С”	“А”	“В”	“С”
0,7963	0,9043	1	0,7217	0,8797	0	0,8546	0,89258	0

Таблиця 4. Метрики моделі Ada Boost

Фінансові навчальні дані			Дані впливу на навколишнє середовище			Комбіновані навчальні дані		
Точність	0,8706		Точність	0,8177		Точність	0,9	
Зважений F1 score	0,8722		Зважений F1 score	0,8111		Зважений F1 score	0,8953	
F1-score			F1-score			F1-score		
“А”	“В”	“С”	“А”	“В”	“С”	“А”	“В”	“С”
0,8104	0,9009	1	0,7010	0,8714	0	0,8432	0,9283	0

Із наведених таблиць видно, що модель логістичної регресії показала найгірший результат із досить низькою точністю. Дана модель є єдиною, яка показала вищу точність на фінансовому наборі даних, ніж на комбінованому. Вона показала непогану точність для “В” рейтингу, але для “А” та “С” точність низька. Особливо важлива низька точність для “А” рейтингу, адже проблеми рейтингу “С” в більшій мірі пов’язані із низькою кількістю даних даного рейтингу в датасеті.

Модель дерева рішень показала значно кращі результати, ніж модель логістичної регресії незалежно від датасету. Точність визначення рейтингу “А” значно виросла на всіх наборах даних у порівнянні із регресією. Найкращі метрики модель показала на комбінованому наборі даних, навіть змогла визначити рейтинг типу “С”, але це більше випадковість. Найгірші результати моделі на наборі даних впливу на навколишнє середовище, але різниця незначна у порівнянні з фінансовим датасетом.

Результати моделей AdaBoost та випадкового лісу майже не відрізняється. Точність незначно краща у другій моделі, але це можна списати на статистичну похибку. В цілому, моделі перевершили за точністю попередні моделі регресії та дерева рішень. На наборі фінансових даних моделі можуть визначати рейтинг “С”, хоч їх частка в датасеті незначна. Найбільші показники точності та F1-score моделі показали саме на комбінованому наборі даних. Незначно гірші метрики моделі показали на фінансовому наборі даних, а навчальний набір даних впливу на навколишнє середовище виявився недостатньо інформативним, хоча точність також досить висока

5. ВИСНОВКИ

Питання оцінки ризиків зелених проектів стає все більш важливим. Кліматичні зміни та збільшення частотності стихійних лих стимулює людство переглядати стратегію поводження з природою планети. В результаті даного дослідження було розглянуто методи вирішення задачі кредитного скорингу та побудовано моделі для прогнозування ризиків зелених проектів. Було проведено порівняння створених моделей та визначено їх ефективність.

У випадку застосування моделі логістичної регресії додавання показників впливу на навколишнє середовище не дало покращень у прогнозуванні. Модель дерева рішень показала значно кращі результати, ніж модель логістичної регресії на усіх варіантах наборів даних, але вона все ще поступається ансамблевим моделям. Результати моделей AdaBoost та випадкового лісу майже не відрізняється. Точність незначно краща у другій моделі, але відмінністю фактично можна знехтувати.

Застосування екологічних метрик при оцінці перспективності та ефективності нових зелених проєктів, безумовно, буде важливим завданням для відбудови України. Вже зараз актуально впроваджувати зелені проєкти, накопичувати дані стосовно їх ефективності та ощадності, стимулювати збільшення таких проєктів шляхом інвестування і видачі вигідних позик та кредитних ліній. Тому, розпочате нами дослідження з метою оцінки можливості застосування класичних підходів та методів, а також розробки нових математичних моделей для оцінювання і врахування як фінансових так і екологічних показників при оцінювання зелених проєктів, безумовно, є початком і перспективних напрямом для подальших досліджень.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Сучасні екологічно чисті технології: Курс лекцій [Електронний ресурс] : навч. посіб. для здобувачів ступеня доктора філософії спеціальності 161 «Хімічні технології та інженерія» / КПІ ім. Ігоря Сікорського ; уклад.: В.М. Павленко, В.Ю. Тобілко, А.І. Бондарева. – Електронні текстові дані (1 файл: 0,945 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2021. – 78 с.
2. Patel N, Upadhyay S. Study of various decision tree pruning methods with their empirical comparison in WEKA. *Int J Comp Appl.* 60(12):20–25.
3. Hastie T, Tibshirani R, Friedman J. *The Elements of Statistical Learning.* Springer; 2001. p. 269-272
4. Кузнецова Н.В., Бідюк П.І. Теорія і практика аналізу фінансових ризиків: системний підхід, монографія. Київ 2020, 400 с.
5. Vrushali Y Kulkarni, Pradeep K Sinha. Random Forest Classifiers. *A Survey and Future Research Directions, International Journal of Advanced Computing, ISSN:2051-0845, Vol.36, Issue.1.* p.1144-1153
6. Euro 7 emission standards, *Insigh: from inferiup international limited*, 18 July 2023, P.8.
7. Євтух О. Типові ризики іпотечного капіталу та управління ними // *Вісник НБУ.* – № 11. – с. 43-46.
8. Грушко В.І., Пилипченко О.І., Пікус Р.В. *Управління фінансовими ризиками.* – Київ: Інститут економіки і права “Крок”, 2000. – с. 24.
9. Яворський Р. Розвиток банківської системи в Україні: Матеріали досліджень переможців всеукраїнського конкурсу “Економічні реформи в Україні”. – Київ, 1999. – с. 107.
10. Чайковський Я. Удосконалення методики комплексної оцінки кредитоспроможності позичальників // *Вісник НБУ.* – 2003. – № 11. – с. 30- 34.
11. Berry MJA, Linoff G. *Mastering Data Mining: The Art and Science of Customer Relationship Management.* New York: John Wiley & Sons, Inc; 1999.
12. Zibran MF. *Department of Computer Science. Diagnostic and Statistical Manual of Mental Disorders – Fourth Edition.* Alberta, Canada: Department of Computer Science, University of Calgary; 2012.
13. Freund Y., Schapire R. E. A Short Introduction to Boosting. *Journal of Japanese Society for Artificial Intelligence*, 14(5):771-780, September, 1999. 14 p.

АНАЛІЗ ВІДКРИТИХ ДАНИХ ПРО ЯКІСТЬ ПОВІТРЯ В МІСЬКОМУ СЕРЕДОВИЩІ ТА РОЗРОБКА ПРОГНОСТИЧНИХ МОДЕЛЕЙ ДЛЯ ПРОГНОЗУ ЗАБРУДНЕННЯ ПОВІТРЯ В МІСТІ

Луцкер Р.О., Гуськова В.Г.

Національний технічний університет України «Київський політехнічний інститут
ім. Ігоря Сікорського»

1. ВСТУП

У сучасному світі існує тенденція активного зростання урбанізації, призводячи до того, що понад 50% населення проживає великих містах. Великі міста, подібно великим пічам, викидають значні обсяги тепла, речовин і відходів у повітря, що призводить до забруднення атмосфери. Проблеми з якістю повітря, особливо гострі у місті Запоріжжя, індустріальному центрі України, стають серйозним викликом для здоров'я людей, оскільки токсичні гази негативно впливають на їхній організм, а індустріальні викиди обтяжують навколишнє середовище.

Для розв'язання проблем забруднення повітря необхідно впроваджувати комплексні заходи. Це включає розподіл міста на рівномірні райони зі станціями моніторингу якості повітря в реальному часі, хімічний аналіз повітря у різних районах та встановлення очисних споруд на критичних об'єктах. Забруднення може бути спричинене різними факторами, тому важливо проводити аналіз через різні організації, зосереджуючись на створенні комплексних систем моніторингу та прогнозування.

Наступним кроком є розробка систем прогнозування та оповіщення на основі актуальних даних про атмосферу. Для цього необхідні виміри, методи та моделі, включаючи метеорологічні та хімічні, забезпечуючи більш точне уявлення про розподіли динамічних та хімічних компонентів. Прогнозування повинно враховувати рівень хімічних елементів у минулому, використовуючи методи, такі як нейронні мережі та комбінування різних вхідних даних для досягнення максимально точного прогнозу рівня забруднення атмосфери.

2. ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

Мета роботи полягає в проведенні досліджень та розробці методів прогнозування забруднення атмосферного повітря з використанням програмної системи. Зазначена система має на меті створення моделі прогнозування за певним методом, надання двотижневого прогнозу та візуалізацію результатів за допомогою веб-інструментів. Для досягнення цієї мети перед розробником стоять завдання, такі як аналіз аналогів програмного забезпечення для визначення вимог, дослідження методів прогнозування часових рядів, розробка технічного завдання, вибір інструментів розробки, проектування структури програмного забезпечення, розробка самого ПЗ, його тестування, відлагодження та економічне обґрунтування розробки.

3. ОГЛЯД КЛАСИЧНИХ МЕТОДІВ ПРОГНОЗУВАННЯ ЯКОСТІ ПОВІТРЯ

Емпіричні моделі використовують прості алгебраїчні відношення для визначення різних величин, таких як концентрація забруднення. Наприклад, методика ОНД-86 використовує цей підхід для прогнозування рівнів забруднення повітря в Україні. Емпіричний

метод модової декомпозиції розкладає часовий ряд на прості складові, відомі як власні характерні форми.

Хоча безпосереднє використання емпіричних моделей є простим підходом, вони не враховують фізичний зміст і не забезпечують узагальнення при зміні вхідних даних.

Статистичні моделі є прогнозними моделями, які будуються на основі аналізу результатів спостережень за допомогою статистичних методів. Для їхньої побудови важлива значна кількість вимірювань, що вимагає витрат часу на отримання та оброблення даних.

У групі статистичних моделей виділяють регресійні та авторегресійні моделі. Регресійні моделі можуть бути простою лінійною, багатовимірною або нелінійною. Серед авторегресійних моделей найвідомішою є ARIMA (інтегрована модель авторегресії рухомого середнього).

Лінійна регресія дозволяє знаходити одне прогнозоване значення за моделлю, тоді як авторегресійна модель використовує результати попередніх спостережень як вхідні дані для рівняння регресії на наступних кроках. Загалом, в контексті прогнозування забруднення атмосферного повітря, такі моделі призначені для короткострокових прогнозів на декілька годин або діб.

Нейронні моделі складаються з нейронних мереж, які представляють собою розділ штучного інтелекту. Використання інструментів, аналогічних явищам у живих організмах, є основним принципом нейронних мереж. Важливою особливістю цих мереж є їхня здатність до паралельної обробки інформації всіма ланками, що значно прискорює процес обробки даних. При великому числі міжнейронних з'єднань мережа також виявляє стійкість до помилок, які можуть виникнути на деяких лініях.

Ще однією важливою характеристикою є здатність нейронних мереж до навчання та узагальнення накопичених знань. Мережа, яка пройшла тренування на обмеженій множині даних, може ефективно узагальнювати інформацію та демонструвати добрі результати на нових даних, які не використовувалися під час навчання.

3. ОГЛЯД МОДЕЛІ LSTM

Long Short Term Memory (LSTM) – це різновид рекурентної нейронної мережі. У RNN вихід з останнього кроку подається як вхід на поточний крок. LSTM була розроблена Hochreiter & Schmidhuber. Вона вирішила проблему довгострокових залежностей ШНМ, коли ШНМ не може передбачити слово, що зберігається в довгостроковій пам'яті, але може давати більш точні прогнози на основі нещодавньої інформації. Зі збільшенням довжини проміжку RNN не дає ефективної роботи. LSTM за замовчуванням може зберігати інформацію протягом тривалого періоду часу. Він використовується для обробки, прогнозування та класифікації на основі даних часових рядів.

Довготривала короткочасна пам'ять (LSTM) – це тип рекурентної нейронної мережі (RNN), яка спеціально розроблена для обробки послідовних даних, таких як часові ряди, мова і текст. LSTM-мережі здатні вивчати довгострокові залежності в послідовних даних, що робить їх добре придатними для таких завдань, як переклад мови, розпізнавання мови та прогнозування часових рядів.

Традиційний ШНМ має один прихований стан, який проходить через час, що може ускладнити для мережі навчання довгострокових залежностей. LSTM вирішують цю проблему шляхом введення комірки пам'яті, яка є контейнером, що може зберігати інформацію протягом тривалого періоду часу. Коміркою пам'яті керують три вентиля: вхідний ventиль, ventиль забування та вихідний ventиль. Ці вентиля вирішують, яку інформацію додавати в комірку пам'яті, видаляти з неї та виводити з неї.

Вхідний клапан контролює, яка інформація додається до комірки пам'яті. Клапан забування контролює, яка інформація видаляється з комірки пам'яті. А вихідний клапан контролює, яка інформація виводиться з комірки пам'яті. Це дозволяє LSTM-мережам вибірково зберігати або відкидати інформацію, коли вона проходить через мережу, що дозволяє їм вивчати довгострокові залежності.

LSTM має ланцюгову структуру, яка містить чотири нейронні мережі та різні блоки пам'яті, які називаються комірками.

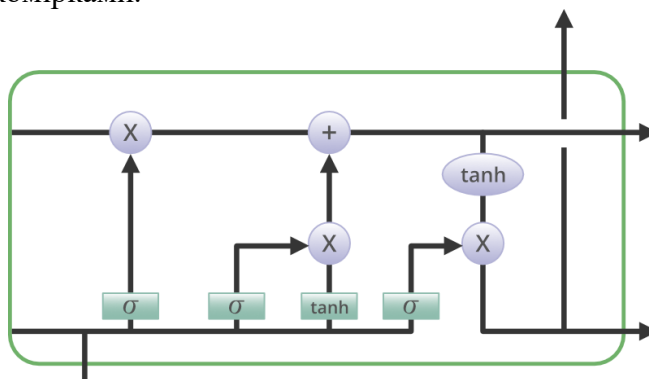


Рисунок 1. Структура нейронної мережі LSTM

Інформація зберігається в комірках, а маніпуляції з пам'яттю здійснюються за допомогою клапанів. Існує три види клапанів:

Клапан забування: Цей механізм дозволяє вилучити інформацію, яка вже не є корисною для стану комірки. Два входи, x_t та h_{t-1} , піддаються обробці клапаном, де їхні вагові матриці та зсуви визначають, яка частина інформації буде збережена або видалена.

Вхідний клапан: Вхідний клапан відповідає за додавання корисної інформації до стану комірки. Інформація регулюється сигмоїдною функцією, яка фільтрує значення, що повинні бути запам'ятовані. Функція \tanh генерує вектор, який містить всі можливі значення з попереднього вихідного стану та поточного входу.

Вихідний клапан: Вихідний клапан відповідає за вилучення корисної інформації з поточного стану комірки для представлення на виході. Генерується вектор за допомогою функції тангенса, а потім сигмоїдна функція фільтрує та визначає, які значення слід зберегти чи видалити, використовуючи входи h_{t-1} та x_t .

4. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Представлені результати використовують концентрацію частинок PM2.5, проте важливо відзначити, що аналогічні тенденції та результати можна спостерігати і для інших показників якості повітря. Ці дані служать лише експериментальним прикладом та ілюстрацією, але схожі закономірності відслідковуються і для інших забруднюючих речовин (табл. 1, рис. 2).

Таблиця 1. Результати прогнозування

Метод прогнозування	RMSE	MAE	Точність, %
Модель ARIMA	15,33	10,95	86,35
Штучні нейронні мережі	19,10	12,87	83,96
Рекурентні нейронні мережі	14,95	10,63	86,75
Нейронні мережі типу «Довга короткострокова пам'ять»	13,87	9,13	88,62

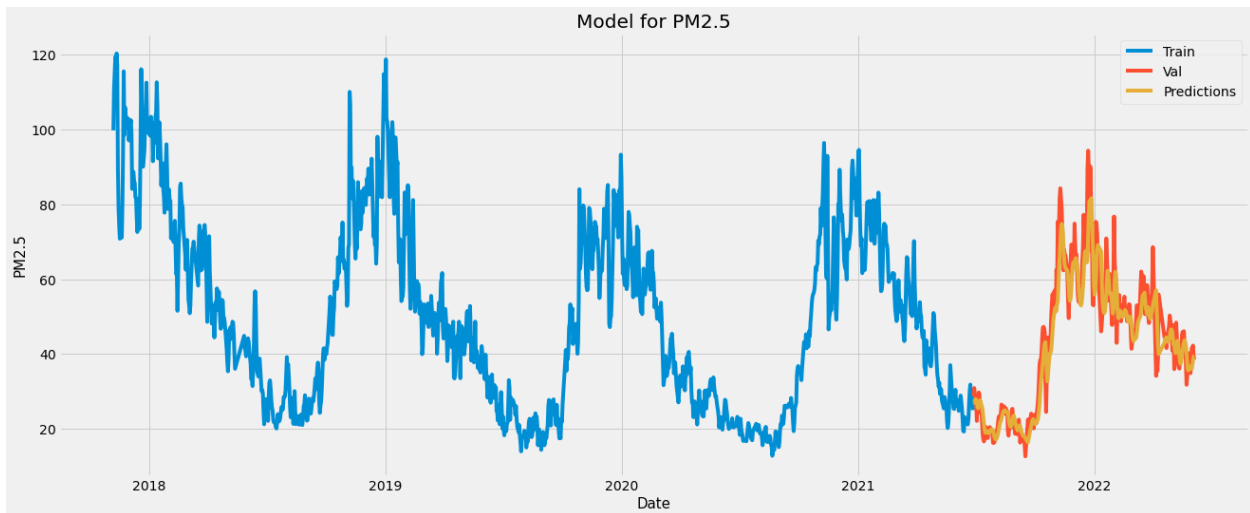


Рисунок 2. Результати прогнозування

Метод "Довга короткострокова пам'ять" (LSTM) виявився найефективнішим у порівнянні з іншими методами прогнозування, демонструючи найменше значення середньої квадратичної помилки (RMSE) та середньої абсолютної помилки (MAE), а також досягнувши найвищу точність прогнозування на рівні 88,62%. Рекурентні нейронні мережі та модель ARIMA також продемонстрували добрі результати, але в усіх трьох аспектах LSTM перевершила їх. Застосування штучних нейронних мереж виявилось менш ефективним у порівнянні з іншими методами, враховуючи значення RMSE, MAE та точність. Отже, отримані результати свідчать про високу ефективність методу LSTM у прогнозуванні якості повітря, роблячи його переважним вибором серед розглянутих методів. Нижче представлений графік прогнозування, отриманого за допомогою методу LSTM.

5. ВИСНОВКИ

У дослідженні порівняли різні методи прогнозування якості повітря за концентрацією частинок. Метод "Довга короткострокова пам'ять" (LSTM) виявився найефективнішим з найменшими помилками та точністю 88,62%. Рекурентні нейронні мережі та ARIMA показали гарні результати, але LSTM виявився кращим. Штучні нейронні мережі були менш ефективними. Узагальнено: LSTM – оптимальний вибір для прогнозування якості повітря з високою точністю. Графік прогнозування LSTM ілюструє його переваги. Це дослідження важливе для сфери моніторингу повітря.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Біляєв, М.М. Моделювання і прогнозування стану довкілля: підручник для студентів вищих навчальних закладів [Текст] / М. М. Біляєв, В.В. Біляєва, П.С Кіріченко. – Кривий Ріг : Вид. Р.А. Козлов, 2016. – 207 с.
2. Applications of Hilbert-Huang transform to non-stationary financial time series analysis [Text] / N. E. Huang, M. Wu, W. Qu, S. R. Long, S. S. P. Shen. // Applied Stochastic Models in Business and Industry. – 2003. – Vol. 19, № 3. – Pp. 245–268.
3. Moustiris K.P., Nastos P.T., Larissi I.K., Paliatsos A.G. Application of Multiple Linear Regression Models and Artificial Neural Networks on the Surface Ozone Forecast in the Greater Athens Area, Greece [Електронний ресурс] / K.P. Moustiris., P.T. Nastos, I.K. Larissi, A.G. Paliatsos – Режим доступу: <https://www.hindawi.com/journals/amete/2012/894714/>

ПОРІВНЯЛЬНИЙ АНАЛІЗ МОДЕЛЕЙ ДЛЯ МЕТОДІВ ПРОГНОЗУВАННЯ

Макухін Є.І.¹, Макаренко О.С.², Бідюк П.І.

Національний технічний університет України «Київський політехнічний інститут ім. Ігоря Сікорського»

¹ makukhin.yevhen@lil.kpi.ua, ² makalex51@gmail.com

Прогнозування є важливим елементом сучасного світу, де швидкі зміни і нестабільні умови вимагають точних та ефективних стратегій планування. Особливо це стосується прогнозування фінансових ринків. Створення якісної моделі є ключовою складовою якісного прогнозування. Метою роботи є порівняння різних моделей для методів прогнозування. Результатом роботи є програмне забезпечення що виконує прогнозування ціни акція і дозволяє порівняти результати різних моделей. У роботі використано теоретичні та емпіричні методи дослідження.

Ключові слова: моделі прогнозування, методи прогнозування, ARIMA, LSTM, N-BEATS, аналіз даних.

1. ВСТУП

На сьогоднішній день фондовий ринок є незамінним інструментом, що дозволяє фінансовим аналітикам, інвесторам та трейдерам інвестувати в цінні папери та оцінювати вартість активів. У той же час, для більш ефективних інвестицій користувачі повинні мати якісні прогнози. Для цього постає задача знаходження та використання методів та моделей для аналізу та побудови точних прогнозів динаміки ціноутворення об'єктів фондового ринку.

Знаходження відповідних моделей дозволяє зробити точний прогноз і допомогти в прийнятті рішення щодо інвестицій. Використання різних моделей дозволить знайти метод який найкраще підходить для відповідних даних. При цьому важливо враховувати усі особливості та властивості прогнозованих даних.

Для порівняння прогнозів було розглянуто математичні методи і machine learning моделі, що застосовують для вирішення практичних задач аналізу та прогнозування нестационарних процесів, а саме: модель авторегресії інтегрованого ковзного середнього ARIMA, LSTM, N-BEATS та ансамблеву модель.

2. МЕТОДИ ПРОГНОЗУВАННЯ

Метод ARIMA (Autoregressive Integrated Moving Average) є популярним і ефективним інструментом для прогнозування часових рядів. ARIMA включає в себе три основні складові: авторегресію (AR), інтегрування (I) та ковзне середнє (MA). ARIMA розшифровується як Авторегресійна інтегрована модель ковзного середнього. Вона належить до класу моделей, які пояснюють заданий часовий ряд на основі його власних минулих значень, тобто власних лагів і помилок прогнозування. Рівняння можна використовувати для прогнозування майбутніх значень. Будь-який "несезонний" часовий ряд, який демонструє закономірності і не є випадковим білим шумом, може бути змодельований за допомогою моделей ARIMA. ARIMA моделі задаються трьома параметрами порядку: (p, d, q), де

- p – порядок AR-члена;
- q – порядок члена MA;

- d – кількість диференціювань, необхідних для того, щоб зробити часовий ряд стаціонарним.

Авторегресія $AR(p)$ – регресійна модель, яка використовує залежний зв'язок між поточним спостереженням і спостереженнями за попередній період. Авторегресійний компонент ($AR(p)$) означає використання минулих значень у рівнянні регресії для часового ряду. $I(d)$ Інтегрування – використовує диференціювання спостережень (віднімання спостереження від спостереження на попередньому часовому кроці) для того, щоб зробити часовий ряд стаціонарним. Диференціювання передбачає віднімання поточних значень ряду від його попередніх значень d разів. $MA(q)$ Moving Average – модель, яка використовує залежність між спостереженням і залишковою похибкою від моделі ковзного середнього, застосованої до запізнених спостережень. Компонент ковзного середнього відображає похибку моделі як комбінацію попередніх членів похибки. Порядок q показує кількість членів, які включаються в модель. В даній роботі буде використаний метод `auto_arima` з бібліотеки `pmdarima` мови програмування Python для автоматизації процесу прогнозування ARIMA. Метод `auto_arima()` використовує покроковий підхід для перебору декількох комбінацій параметрів p, d, q і вибирає найкращу модель, яка має найменший AIC (інформаційний критерій Акаїке). Вона працює шляхом проведення тестів на диференціювання (наприклад, Квятковського-Філіпса-Шмідта-Шина, розширеного Дікі-Фуллера або Філіпса-Перрона) для визначення порядку диференціювання, d , а потім підбирає моделі в межах визначених діапазонів `start_p`, `max_p`, `start_q`, `max_q`. Якщо увімкнено опцію сезонності, `auto_arima` також намагається визначити оптимальні гіперпараметри P та Q після проведення тесту Канови-Хансена для визначення оптимального порядку сезонного диференціювання, D .

Мережі з довготривалою короткочасною пам'яттю (ДКЧП, англ. long short-term memory, LSTM), – це особливий тип рекурентних нейронних мереж, що здатний навчатися довготривалих залежностей. Вони були введені Хохрайтером і Шмідгубером (1997), і були вдосконалені і популяризовані багатьма людьми в наступних роботах. Вони надзвичайно добре працюють над широким спектром проблем і зараз широко використовуються. Як і більшість РНМ, мережа LSTM є універсальною в тому сенсі, що за достатньої кількості вузлів мережі вона може обчислювати будь-що, що може обчислювати звичайний комп'ютер, за умови, що вона має належну матрицю вагових коефіцієнтів, що може розглядатися як її програма. На відміну від традиційних РНМ, мережа LSTM добре підходить для навчання з досвіду з метою класифікації, обробки або передбачення часових рядів в умовах, коли між важливими подіями існують часові затримки невідомої тривалості. LSTM явно розроблені для того, щоб уникнути проблеми довготривалої залежності.

Блоки LSTM містять три або чотири «вентилі» (англ. gates), які вони використовують для керування потоком інформації до або з їхньої пам'яті. Звичайний блок LSTM складається з комірки, вхідних вентилів (Input Gate), вихідних вентилів (Output Gate) і вентилів забування (Forget Gate). Комірка запам'ятовує значення протягом довільних часових інтервалів, а три вентилі регулюють потік інформації в комірку і з неї. Вентилі забування вирішують, яку інформацію відкинути з попереднього стану, присвоюючи попередньому стану, порівняно з поточним входом, значення від 0 до 1. (Округлене) значення 1 означає збереження інформації, а значення 0 – її відкидання. Вхідні вентилі вирішують, які частини нової інформації зберігати у поточному стані, використовуючи ту ж систему, що і вентилі забування. Вихідні вентилі керують тим, яку інформацію з поточного стану виводити, присвоюючи інформації значення від 0 до 1, враховуючи попередній і поточний стан. Вибірковий вивід релевантної інформації з поточного стану дозволяє мережі LSTM підтримувати корисні, довгострокові залежності для прогнозування, як в поточному, так і в майбутньому часі.

Основна ідея застосування LSTM полягає в здатності мережі "вирішувати", яку інформацію тримати та яку забути, що дозволяє їй ефективно моделювати довгострокові залежності. Це здатність особливо корисна в задачах прогнозування часових рядів, обробці природної мови та інших завданнях, де важлива контекстуальна інформація. У Python для реалізації LSTM можна використовувати бібліотеки, такі як TensorFlow чи PyTorch.

N-BEATS (Neural Basis Expansion Analysis and Transformation System) - це архітектура нейронних мереж для задач прогнозування часових рядів, яка була представлена в статті "N-BEATS: Neural Basis Expansion Analysis and Transformation System" в 2020 році. Ця архітектура відрізняється своєю гнучкістю та здатністю адаптуватися до різноманітних завдань прогнозування. Ключові особливості N-BEATS:

- Підтримка декількох часових рядів: N-BEATS можна навчати на декількох часових рядах, кожен з яких представляє різний розподіл.
- Швидке навчання: Модель не містить жодних рекурентних шарів або шарів самоуваги – таким чином, швидше навчання та стабільний градієнтний потік.
- Багатогоризонтне прогнозування: Модель виробляє багатокрокові прогнози.
- Інтерпретованість: Автори розробили 2 версії моделі: загальну та інтерпретовану. Інтерпретована версія може виводити інтерпретовані прогнози щодо тренду та сезонності (рис. 1).

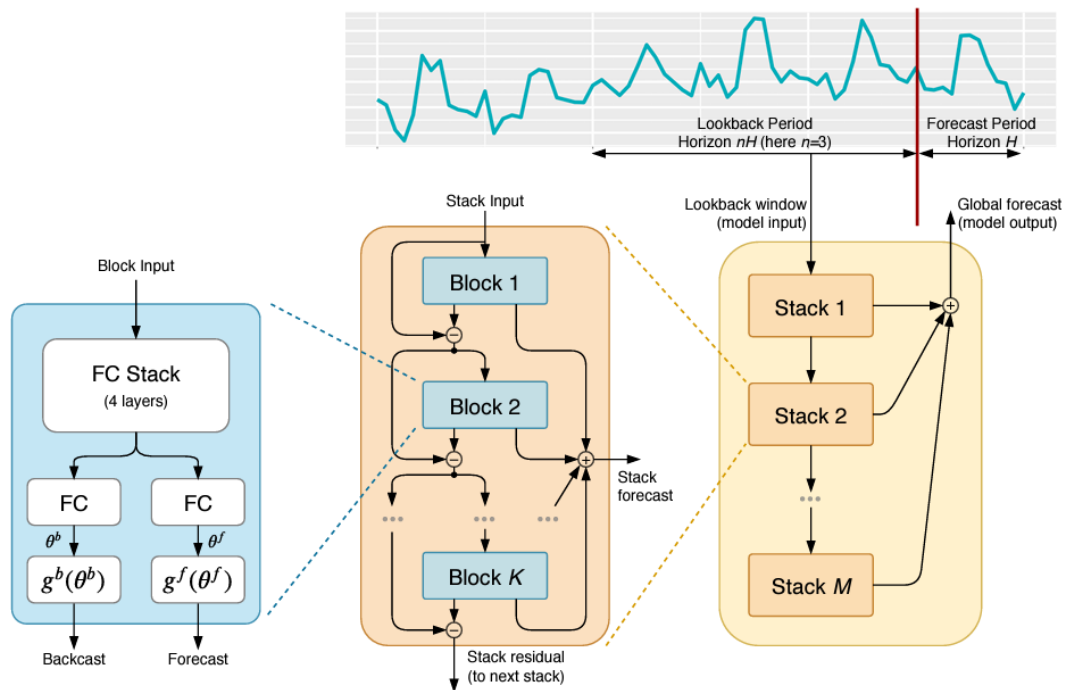


Рисунок 1. Запропонована архітектура моделі

Основні елементи:

- Блок (синій колір) – основна одиниця обробки.
- Стек (помаранчевий колір) – колекція блоків.
- Кінцева модель (жовтий колір) – сукупність стеків.

Ансамблеві методи використовують кілька алгоритмів навчання, щоб отримати кращу прогностичну ефективність, ніж можна було б отримати від будь-якого з складових алгоритмів навчання окремо. Для створення ансамблевих моделей буде використано комбінацію:

- Різних функцій втрат (MAE, MSE, MAPE)
- Випадково ініціалізованих моделей

Буде створено набір різних моделей, які намагатимуться моделювати одні й ті самі дані та функції для списку різних моделей, навчених з різними функціями втрат. Кожен шар в ансамблі моделей буде ініціалізовано випадковим нормальним (гаусівським) розподілом за допомогою Не нормальної ініціалізації, це допоможе оцінити інтервали прогнозування пізніше.

3. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Порівняльний аналіз моделей прогнозування буде здійснюватися на прикладі прогнозування вартості акцій компанії Apple.

При побудові ARIMA моделі потрібно вибрати параметри p, q, d . Для цього ми використовуємо `auto_arima` функцію з бібліотеки `rmadarma`, щоб дозволяє отримати найкращі параметри, навіть не будуючи графіки АКФ та ЧАКФ.

Після розділення вибірки на тестову та навчальну у співвідношенні 90 на 10 знайдемо оптимальну модель (рис. 2).

```
Performing stepwise search to minimize aic
ARIMA(1,1,1)(0,0,0)[0] intercept : AIC=9370.149, Time=0.52 sec
ARIMA(0,1,0)(0,0,0)[0] intercept : AIC=9376.414, Time=0.03 sec
ARIMA(1,1,0)(0,0,0)[0] intercept : AIC=9369.051, Time=0.08 sec
ARIMA(0,1,1)(0,0,0)[0] intercept : AIC=9368.795, Time=0.10 sec
ARIMA(0,1,0)(0,0,0)[0] intercept : AIC=9378.707, Time=0.03 sec
ARIMA(0,1,2)(0,0,0)[0] intercept : AIC=9370.375, Time=0.16 sec
ARIMA(1,1,2)(0,0,0)[0] intercept : AIC=9372.148, Time=0.45 sec
ARIMA(0,1,1)(0,0,0)[0] intercept : AIC=9371.671, Time=0.05 sec
```

```
Best model: ARIMA(0,1,1)(0,0,0)[0] intercept
Total fit time: 1.429 seconds
```

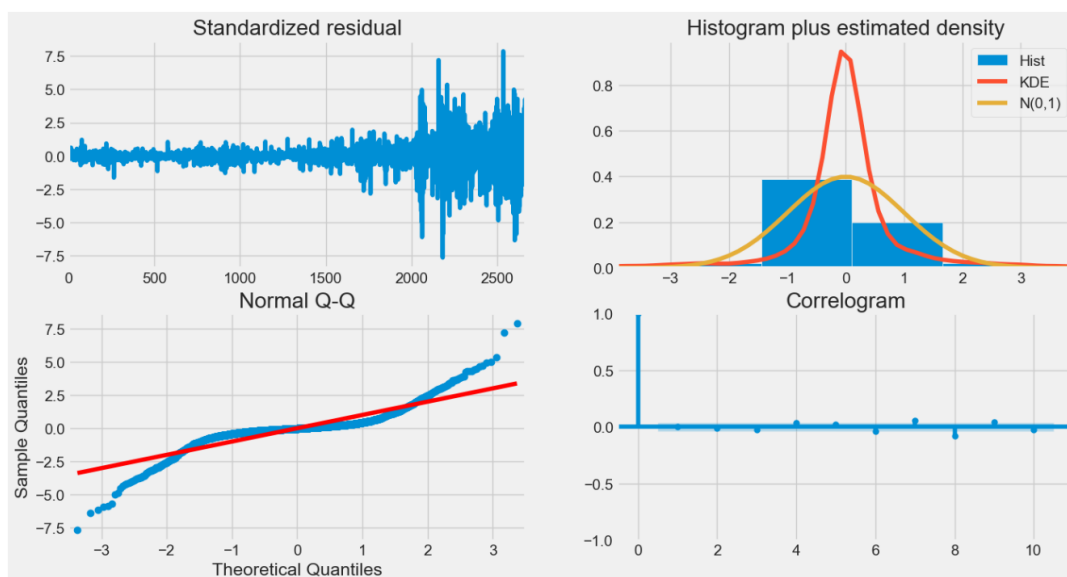


Рисунок 2. Результати пошуку оптимальної моделі

Оптимальною моделлю виявилася модель $ARIMA(0,1,1)$. Далі використаємо модель для прогнозування значень з 95% довірчим інтервалом і оцінимо модель (рис. 3). Для оцінки знайдемо значення MAE (середня абсолютна похибка), RMSE (середньоквадратична похибка), MAPE (середня абсолютна відсоткова похибка) і MASE (середня абсолютна масштабована похибка).

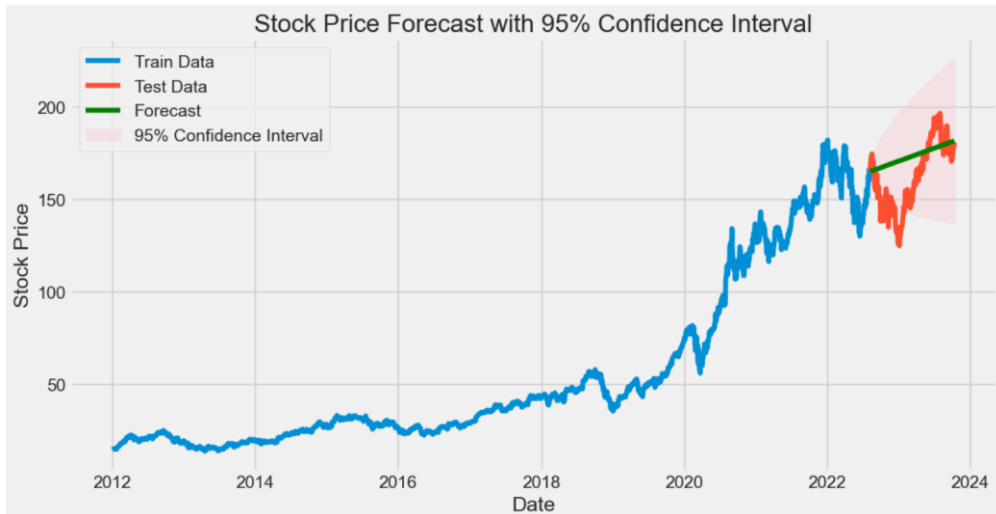


Рисунок 3. Результати прогнозу значень з довірчим інтервалом

Було отримано критерії якості прогнозу, що дорівнюють:

$MAE = 15,124319$, $RMSE = 18,654106$, $MAPE = 10,100647$, $MASE = 7,449288$.

Як можна помітити за результатами, модель має не досить точний прогноз.

Далі переходимо до моделей нейромереж. Спочатку розділимо часовий ряд за допомогою вікон. Вікно – це метод перетворення набору даних часового ряду в керовану навчальну задачу. Іншими словами, ми хочемо використовувати вікна минулого для прогнозування майбутнього. Будемо використовувати розмір горизонту 1 і розмір вікна 7. Потім перетворимо наші вікна на навчальні та тестові розбиття. Замість того щоб розбити на вікна існуючі навчальні та тестові вибірки, враховуючи природу розбиття на вікна (розбиття на вікна часто вимагає зсуву в певній точці даних), зазвичай краще спочатку розбити дані на вікна, а потім розділити їх на навчальні та тестові вибірки. В результаті для кожної з моделей маємо розділення вибірки на навчальну та тестову у співвідношенні 80 на 20.

Модель LSTM має функцію активації ReLu, 100 епох та розмір партії 128. За допомогою порівняння середньої абсолютної похибки знаходимо найкращу модель і прогнозуємо значення (рис. 4).



Рисунок 4. Результати прогнозу значень для моделі LSTM

Було отримано критерії якості прогнозу, що дорівнюють:

MAE = 2,206478, RMSE = 2,8834975, MAPE = 1,410206, MASE = 1,0534115

За результатами можна помітити що модель досить добре прогнозує значення ціни акцій.

Далі будемо оцінювати модель N-BEATS в якій для покращення продуктивності моделі збільшена кількість шарів у порівнянні з іншими аналогічними моделями. Почнемо з побудови блочного шару N-BEATS. Створимо основний будівельний блок для архітектури N-BEATS. Цього разу, оскільки ми будемо використовувати більшу архітектуру моделі, щоб забезпечити максимально швидке навчання моделі, ми налаштуємо наші набори даних за допомогою API `tf.data`. Налаштування гіперпараметрів для алгоритму N-BEATS візьмемо зі схеми на рисунку 5, що описує гіперпараметри, які використовуються для різних варіантів N-BEATS. Ми використовуємо N-BEATS-G, що означає загальну версію N-BEATS.

Table 18: Settings of hyperparameters across subsets of M4, M3, TOURISM datasets.

Parameter	M4						M3				TOURISM			
	Yly	Qly	Mly	Wly	Dly	Hly	Yly	Qly	Mly	Other	Yly	Qly	Mly	
N-BEATS-I														
L_H	1.5	1.5	1.5	10	10	10	20	5	5	20	20	10	20	
Iterations	15K	15K	15K	5K	5K	5K	50	6K	6K	250	30	500	300	
Losses	SMAPE/MAPE/MASE						SMAPE/MAPE/MASE				MAPE			
S-width							2048							
S-blocks							3							
S-block-layers							4							
T-width							256							
T-degree							2							
T-blocks							3							
T-block-layers							4							
Sharing							STACK LEVEL							
Lookback period							2H, 3H, 4H, 5H, 6H, 7H							
Batch							1024							
N-BEATS-G														
L_H	1.5	1.5	1.5	10	10	10	20	20	20	10	5	10	20	
Iterations	15K	15K	15K	5K	5K	5K	20	250	10K	250	30	100	100	
Losses	SMAPE/MAPE/MASE						SMAPE/MAPE/MASE				MAPE			
Width							512							
Blocks							1							
Block-layers							4							
Stacks							30							
Sharing							NO							
Lookback period							2H, 3H, 4H, 5H, 6H, 7H							
Batch							1024							

Рисунок 5. Рекомендовані налаштування гіперпараметрів моделі

Після налаштування гіперпараметрів, тепер, перш ніж ми створимо модель N-BEATS, нам потрібно пройти через два шари, які відіграють велику роль в архітектурі. Саме вони роблять можливим подвійне залишкове укладання N-BEATS:

- `tf.keras.layers.subtract(inputs)` – віднімає список вхідних тензорів один від одного
- `tf.keras.layers.add(inputs)` – додає список вхідних тензорів один до одного

Залишковий зв'язок (також званий "пропускним зв'язком") передбачає, що більш глибокий шар нейронної мережі отримує виходи, а також входи більш поверхневого шару нейронної мережі. У випадку N-BEATS архітектури використовуються залишкові зв'язки, які:

- віднімають зворотні виходи попереднього блоку від зворотних входів поточного блоку
- додають прогнозні виходи з усіх блоків разом у стек

В результаті для створення і навчання моделі робимо наступне:

1. Створимо екземпляр шару блоків N-BEATS за допомогою `NBeatsBlock` (це буде початковий блок для мережі, решта будуть створені як частина стеків)
2. Створимо вхідний шар для стека N-BEATS (для цього ми будемо використовувати `Keras Functional API`)
3. Зробіть початкові беккаст та прогноз для моделі з шаром, створеним у (1)

4. Створимо за допомогою циклу стеки блокових шарів
 5. Використаємо клас NBeatsBlock у циклі, створеному в пункті (4), для створення блоків, які повертають беккасти та прогнози на рівні блоків
 6. Створюємо подвійний залишковий стек, використовуючи віднімання та додавання шарів
 7. Об'єднаємо входи та виходи моделі за допомогою `tf.keras.Model()`
 8. Скопіюємо модель з оцінкою MAE та оптимізатором Adam
 9. Підготуємо модель N-BEATS для 5000 епох, і оскільки вона має таку кількість епох, ми використаємо декілька зворотних викликів:
 - `tf.keras.callbacks.EarlyStopping()` – зупиняє навчання моделі, якщо вона не покращує валідаційні втрати за 200 епох, і відновлює найкращі ваги, використовуючи `restore_best_weights=True`
 - `tf.keras.callbacks.ReduceLROnPlateau()` – якщо втрата валідації моделі не покращується протягом 100 епох, зменшити швидкість навчання у 10 разів, щоб спробувати допомогти їй зробити поступові покращення (чим менша швидкість навчання, тим менші оновлення намагається зробити модель)
- Таким чином, для цієї моделі маємо такі результати прогнозування (рис. 6).



Рисунок 6. Результати прогнозу значень для моделі N-BEATS

Було отримано критерії якості прогнозу, що дорівнюють:

$MAE = 2,1958368$, $RMSE = 2,8769698$, $MAPE = 1,4034737$, $MASE = 1,0483311$

За результатами можна помітити що якість прогнозу трохи покращилась, але не суттєво.

Наступний експеримент - створення ансамблю моделей. Ансамбль передбачає навчання та об'єднання декількох різних моделей.

Для створення наших ансамблевих моделей будемо використовувати комбінацію:

- Різних функцій втрат (MAE, MSE, MAPE)
- Випадково ініціалізованих моделей

По суті, ми створимо набір різних моделей, які намагатимуться моделювати одні й ті самі дані. Створимо функції для створення списку різних моделей, навчених з різними функціями втрат. Кожен шар в ансамблі моделей буде ініціалізовано випадковим нормальним (гаусівським) розподілом за допомогою Не нормальної ініціалізації, це допоможе оцінити інтервали прогнозування пізніше. Перевіримо для 5 ітерацій та 1000 епох. Це дасть нам 15 моделей (по 5 для кожної функції втрат). Однак, оскільки ми навчаємо 15 моделей, то отримаємо 15 наборів прогнозів. Замість того, щоб порівнювати кожен набір прогнозів з істиною в останній інстанції, візьмемо медіану.

Отже, маємо такі результати прогнозування:

MAE = 2,1159441, RMSE = 2,8000813, MAPE = 1,3533982, MASE = 1,0101889

Маємо графік прогнозу з довірчим інтервалом на рисунку 7.

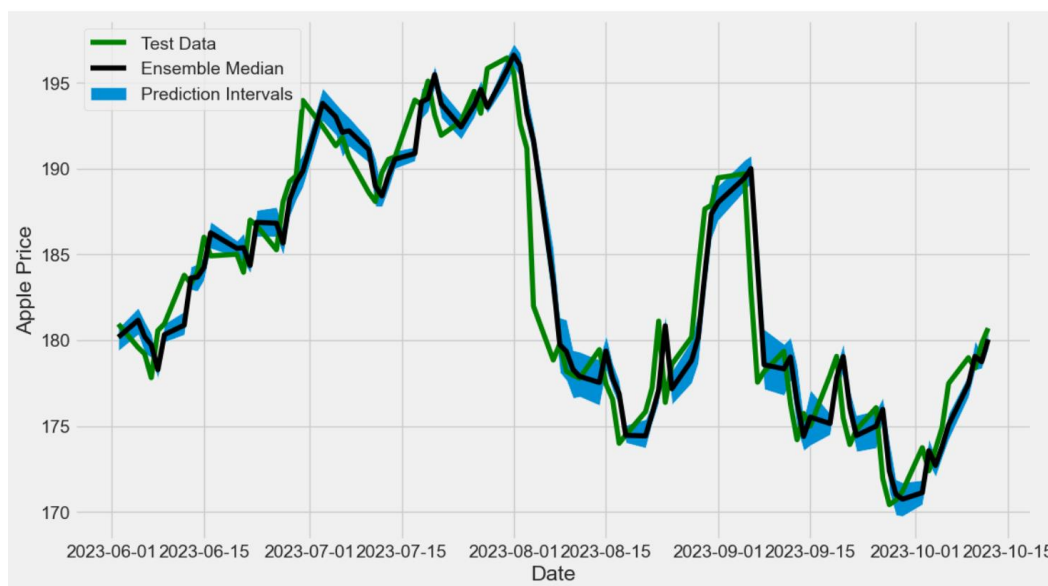


Рисунок 7. Результати прогнозу для ансамблю моделей

Тепер порівняємо результати кожної з моделей на рисунку 8.

	mae	mse	rmse	mape	mase
model_1_arima	15.124319	347.975677	18.654106	10.100647	7.449288
model_2_LSTM	2.206478	8.314558	2.883497	1.410206	1.053411
model_3_NBEATS	2.195837	8.276956	2.876970	1.403474	1.048331
model_4_ensemble	2.115944	7.840455	2.800081	1.353398	1.010189

Рисунок 8. Критерії якості прогнозу для кожної з моделей

Можна побачити, що найкращі результати має ансамблева моделей, хоча результати LSTM та N-BEATS майже аналогічні і усі 3 моделі мають достатньо високу якість прогнозування.

4. ВИСНОВКИ

На сьогоднішній день фінансові ринки стають все більш доступними для користувачів, тому виникає необхідність знаходження більш ефективних інструментів аналізу та прогнозування для подальшого інвестування.

Було проведено порівняльний аналіз методів прогнозування для нестационарних і нелінійних процесів фондових ринків та зроблено огляд моделей прогнозування, досліджено їх результати.

Було розглянуто математичні методи і machine learning моделі, що застосовують для вирішення практичних задач аналізу та прогнозування нестационарних процесів, а саме: модель авторегресії інтегрованого ковзного середнього ARIMA, LSTM, N-BEATS та ансамблеву модель.

Для порівняння моделей була реалізована програма що дозволяє побудову структури даних моделей, знаходження прогнозу та його результатів. У якості об'єкту експериментальних досліджень було обрано акції компанії Apple, та найкращі результати показала ансамблева модель машинного навчання, яка дозволила отримати дуже якісні прогнози. Інші моделі показали схожі результати, тому прогнози нейромереж можна вважати достатньо точними.

Результатом є реалізація та порівняння моделей прогнозування, що вирішують складні, але дуже важливі проблеми бізнесу, дають змогу ефективно вести інвестиційну політику та впроваджувати штучний інтелект в будь-якій сфері.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Бідюк П. І., Романенко В. Д., Тимошук О. Л. Аналіз часових рядів: навч. посіб. ННК «Інститут прикладного системного аналізу» Національний технічний університет України «Київський політехнічний інститут». 2010. 317 с.
2. Бідюк П. І. Економетричний аналіз часових рядів. Київ: Політехніка, 2007. 250 с.
3. Канторович Г.Г. Анализ временных рядов: лекционные и методические материалы. Москва: Экономический журнал ВШЭ, 2002. 129 с.
4. Бідюк П. І. Часові ряди: моделювання і прогнозування: монографія. Київ: ЕКМО, 2003. 144 с.
5. Магнус Я.Р., Катышев П.К., Песесецкий А.А., Магнус Я.Р. Эконометрика: Начальный курс: учеб. 6-е изд., перед. и доп. Москва: Дело, 2004. 576 с.
6. Вербик М. Путеводитель по современной эконометрике / пер. с англ. Банникова В.А.; научн. ред. и пред. Айвазяна С.А. Москва: Научная книга, 2008. 616 с.
7. Бокс Дж., Дженкинс Г. Анализ временных рядов, прогноз и управление / пер.с англ. Москва: Мир, 1974. 406 с.
8. Simonyan K. Very Deep Convolutional Networks for Large-Scal Image Recognition – Режим доступу до ресурсу:
<http://arxiv.org/abs/1409.1556>
9. Boris N. Oreshkin, Nicolas Chapados, Dmitri Carпов, Yoshua Bengio N-BEATS:Neural basis expansion analysis for interpretable time series forecasting – Режим доступу до ресурсу:
<https://arxiv.org/pdf/1905.10437>
10. Luke B. Godfrey, Michael S. Gashler Neural Decomposition of Time-Series Data for Effective Generalization - Режим доступу до ресурсу:
<https://arxiv.org/pdf/1705.09137>
11. Daniel L. Marino, Kasun Amarasinghe, Milos Manic Building Energy Load Forecasting using Deep Neural Networks - Режим доступу до ресурсу:
<https://arxiv.org/ftp/arxiv/papers/1610/1610.09460>
12. Soofi A.S., Liangyue C. Modelling and Forecasting Financial Data. Techniques of Nonlinear Dynamics. Boston: Springer US. 2002. 488 p.
13. Mujtaba S. M., Nadeem M. Analyzing Stock Markets using Data Warehousing. Journal of Independent Studies and Research. Jan. 2006. Vol. 4. P. 8.
14. Mondal D. A., Maji G., Goto T., Debnath N.C., Sen S. Data Warehouse Based Modelling Technique for Stock Market Analysis. International Journal of Engineering & Technology. 2018. Vol. 7 (3.13). P. 165-170.
15. Selva Prabhakaran ARIMA Model – Complete Guide to Time Series Forecasting in Python <https://www.machinelearningplus.com/time-series/arima-model-time-series-forecasting-python/>

ПІДХОДИ ДО ПРОГНОЗУВАННЯ ФІНАНСОВОГО СТАНУ ПІДПРИЄМСТВА ТА ОЦІНКИ ІНВЕСТИЦІЙНОЇ ПРИВАБЛИВОСТІ СІЛЬСЬКОГОСПОДАРСЬКОГО ПІДПРИЄМСТВА НА ОСНОВІ СТАТИСТИЧНИХ ДАНИХ

Мілявський Ю.Л.¹, Павлуша А.О.²

Національний технічний університет України “Київський політехнічний інститут ім. Ігоря Сікорського”

¹ yuriy.milyavsky@gmail.com, ² nata.pavelichanko@gmail.com

У роботі розглянуто проблему прогнозування фінансових показників (зокрема, прибутку) для сільськогосподарських підприємств на прикладі конкретного елеватора. Проведено аналіз фінансових даних, досліджено кореляційну матрицю, побудовано ряд регресійних моделей для прогнозування виручки, вибрано найкращу модель (з регресорами і трендом) на основі аналізу сукупності показників якості. На основі обраної моделі здійснено прогнозування. Запропоновано стартап проєкт для надання послуг з прогнозування фінансових показників сільськогосподарських підприємств для використання їх керівництвом, інвесторами та державою.

Ключові слова: фінансові показники, сільськогосподарське підприємство, часові ряди, авторегресія, показники якості моделей

1. ВСТУП

Прогнози оточують нас майже всюди: від всім відомого прогнозу погоди, до прогнозу рівня життя на Землі через 1000 років. Всі, хто користується прогнозом скаже, що це дуже зручно. Це можна помітити на прикладі прогнозування розвитку хвороб. Вчасно помітивши тенденцію розвитку захворювання, медики та вчені можуть попередити епідемію відкривши вакцину, чи ввівши рекомендації для людей по запобіганню хвороби.

Незважаючи на зручність прогнозів та існування багатьох підходів до прогнозування досі представники аграрного сектора в Україні запевняють, що неможливо побудувати прогноз на прибуток сільськогосподарського підприємства. Аргументують таку позицію тим, що на прибуток сільськогосподарського підприємства впливає врожай, а на врожай комбінація погодних умов, які майже неможливо спрогнозувати на рік вперед та більше.

Сільське господарство має вагомий вплив на українську економіку. Налагодження сільськогосподарської сфери таким чином, щоб українці мали можливість харчуватися продукцією власного виробництва, може сприяти пониженню цін на продукти для населення (оскільки власне вирощення вимагає менше затрат, ніж покупні у інших країн продукти), як наслідок це може спричинити підвищення рівня проживання в країні та стати одним із способів подолання бідності. Якщо після досліджень агропромислового комплексу знайти можливості працювати не лише на задоволення власних потреб, а й додати можливість експорту власної продукції за кордон, то це очевидне збільшення ВВП, що має безпосередній вплив на економіку держави. Науковці проводили багато досліджень, щоб вивчити стан аграрної сфери в Україні [1, 2]. Це важливо для розуміння необхідності прогнозів саме в цій сфері. Якщо застосована в цій роботі технологія виявиться ефективною, то в подальшому, держава може використовувати дану технологію при плануванні бюджету на наступний

календарний рік. Тобто, спеціалісти зможуть спрогнозувати прибутки визначених підприємств, та надати отримані результати державі, які вона в свою чергу може використовувати при плануванні витрат на новий рік.

2. ПІДХОДИ ДО ПОБУДОВИ ПРОГНОЗІВ ПРИБУТКУ ПІДПРИЄМСТВА

Для побудови прогнозів зручно використовувати часові ряди. Дані, що були отримані від підприємства змінюються з часом, тому даний підхід можливий в цьому випадку.

В роботі проводився аналіз роботи та побудова прогнозу виручки для підприємства «Уманьнасінтрав». Це елеватор в Уманському районі, Черкаська область. Дохід елеватора від переробки зерна в крупу, мучку та дрібку. При переробці використовуються різні технології, а саме тривалість лушення та різний рівень вологості. Підприємство обробляє твердозернові та м'якозернові типи зерна. Усього значень для аналізу 126. Дані зібрані помісячно за 10 років і 6 місяців. Частина даних представлена на рис. 1.

Тривалість лушення, с X1	Вологість, % X2	Тривалість відволожування, хв, X3	Тип зерна	Вихід крупи, %	Вихід мучки, %	Вихід дрібки, %	Виручка від крупи, грн	Виручка від мучки, грн	Виручка від дрібки, грн	Виручка із врахуванням витрат на сировину, грн	Необхідна кількість ел.ен, кВт	Необхідна кількість води, м3	Амортизація на вто, грн	Витрати на воду, грн	Витрати на електроенергію, грн	Виручка, грн	Виручка, млн грн
20	13	30	Твердозерний	97,2	2,1	0,7	14521680	78435	13072,5	10255687,5	74700	0	622500	0	253980	10001708	10,01
40	13	30	Твердозерний	96,8	2,1	1,1	14461920	78435	20542,5	10203397,5	74700	0	622500	0	253980	9949418	9,95
60	13	30	Твердозерний	95,9	2,6	1,5	14327460	97110	28012,5	10095082,5	74700	0	622500	0	253980	9841103	9,85
80	13	30	Твердозерний	93,6	3,9	2,5	13983840	145665	46687,5	9818692,5	74700	0	622500	0	253980	9564713	9,57
100	13	30	Твердозерний	92,7	4,1	3,2	13849380	153135	59760	9704775	74700	0	622500	0	253980	9450795	9,46
120	13	30	Твердозерний	90,9	4,5	4,6	13580460	168075	85905	9476940	74700	0	622500	0	253980	9222960	9,23
140	13	30	Твердозерний	89,2	5,9	4,9	13326480	220365	91507,5	9280852,5	74700	0	622500	0	253980	9026873	9,03
160	13	30	Твердозерний	86,9	6,9	6,2	12982860	257715	115785	8998860	74700	0	622500	0	253980	8744880	8,75
180	13	30	Твердозерний	84,1	8,8	7,1	12564540	328680	132592,5	8668312,5	74700	0	622500	0	253980	8414333	8,42
20	13,5	30	Твердозерний	97,1	2,2	0,7	14506740	82170	13072,5	10244482,5	74700	7,155172414	933750	457,931	255225	9988800	9,99
40	13,5	30	Твердозерний	96,9	2,1	1	14476860	78435	18675	10216470	74700	7,155172414	933750	457,931	255225	9960787	9,97
60	13,5	30	Твердозерний	95,8	2,8	1,4	14312520	104580	26145	10085745	74700	7,155172414	933750	457,931	255225	9830062	9,84
80	13,5	30	Твердозерний	93,8	3,8	2,4	14013720	141930	44820	9842970	74700	7,155172414	933750	457,931	255225	9587287	9,59
100	13,5	30	Твердозерний	92,9	4,1	3	13879260	153135	56025	9730920	74700	7,155172414	933750	457,931	255225	9475237	9,48
120	13,5	30	Твердозерний	90,7	5,1	4,2	13550580	190485	78435	9462000	74700	7,155172414	933750	457,931	255225	9206317	9,21
140	13,5	30	Твердозерний	89,3	6	4,7	13341420	224100	87772,5	9295792,5	74700	7,155172414	933750	457,931	255225	9040110	9,05
160	13,5	30	Твердозерний	86,7	7,1	6,2	12952980	265185	115785	8976450	74700	7,155172414	933750	457,931	255225	8720767	8,73
180	13,5	30	Твердозерний	84,3	8,7	7	12594420	324945	130725	8692590	74700	7,155172414	933750	457,931	255225	8436907	8,44
20	14	30	Твердозерний	97,6	1,9	0,5	14581440	70965	9337,5	10304242,5	74700	14,31034483	1245000	915,8621	256470	10046857	10,05
40	14	30	Твердозерний	97,1	1,9	1	14506740	70965	18675	10238880	74700	14,31034483	1245000	915,8621	256470	9981494	9,99
60	14	30	Твердозерний	96	2,8	1,2	14342400	104580	22410	10111890	74700	14,31034483	1245000	915,8621	256470	9854504	9,86
80	14	30	Твердозерний	94	3,7	2,3	14043600	138195	42952,5	9867247,5	74700	14,31034483	1245000	915,8621	256470	9609862	9,61
100	14	30	Твердозерний	93,2	3,8	3	13924080	141930	56025	9746535	74700	14,31034483	1245000	915,8621	256470	9507149	9,51
120	14	30	Твердозерний	90,9	5	4,1	13580460	186750	76567,5	9486277,5	74700	14,31034483	1245000	915,8621	256470	9228892	9,23
140	14	30	Твердозерний	89,8	5,5	4,7	13416120	205425	87772,5	9351817,5	74700	14,31034483	1245000	915,8621	256470	9094432	9,1
160	14	30	Твердозерний	87,3	6,7	6	13042620	250245	112050	9047415	74700	14,31034483	1245000	915,8621	256470	8790029	8,8
180	14	30	Твердозерний	84,8	8,8	6,4	12669120	328680	119520	8759820	74700	14,31034483	1245000	915,8621	256470	8502434	8,51

Рисунок 1. Отримані дані від підприємства

Для початку роботи виконано фільтрацію даних, що є важливим елементом дослідження. Підприємство під час надання даних повідомило про всі параметри, що можуть впливати на прибуток. Проте насправді не всі дані мають вплив на фінальний результат. Наприклад, було видалено з аналізу дані «тривалість відволожування», оскільки значення є константою. Це не буде впливати на зміну прибутку.

Для обробки даних було побудовано кореляційну матрицю залежності виручки від вологості та часу лушення (рис.2).

Correlation			
	BY MON	TIME	WET
BY MON	1.000000	-0.868251	0.109463
TIME	-0.868251	1.000000	0.000000
WET	0.109463	0.000000	1.000000

Рисунок 2. Матриця кореляції

Побудова матриці кореляції дозволила визначити важливу закономірність та побудувати припущення стосовно ефективності роботи підприємства. Оскільки залежність виручки від тривалості лушення має від'ємне значення, а залежність виручки від вологості зерна має

плюсовий показник, тобто це свідчить про обернену лінійну залежність для показників виручка-час та пряму лінійну залежність для показників виручка-вологість [4, 8]. Тобто при збільшення показника «вологість зерна» (можливі показники вологості 13%-16%, з кроком 0,5%) збільшується показник виручки. Також виручка збільшується при зменшенні часу лушення зерна (можливі показники 20с-180с з кроком 20с.). Таким чином, максимальну виручку можна буде отримати при вологості насіння 16% та за час лушення зерна 20с. Якщо повернемося до даних та перевіримо припущення шляхом пошуку максимального значення, отримаємо наступні результати, що наведено на рисунках 3 та 4.

Тривалість лушіння, с X1	Вологість, % X2	Тип зерна	Виручка, грн	Виручка, млн грн
20	16	М'якозерний	9890644,914	9,9
20	15,5	М'якозерний	9875540,345	9,88
20	15	М'якозерний	9860435,776	9,87
20	14,5	М'якозерний	9834126,207	9,84
20	14	М'якозерний	9796611,638	9,8
20	13,5	М'якозерний	9757229,569	9,76
20	13	М'якозерний	9743992,5	9,75
40	16	М'якозерний	9715099,914	9,72
40	15,5	М'якозерний	9673850,345	9,68
40	15	М'якозерний	9647540,776	9,65
40	14,5	М'якозерний	9628701,207	9,63
40	14	М'якозерний	9589319,138	9,59
40	13	М'якозерний	9570315	9,58
40	13,5	М'якозерний	9563009,569	9,57
60	16	М'якозерний	9519012,414	9,52
60	15,5	М'якозерний	9485232,845	9,49
60	15	М'якозерний	9479465,776	9,48
60	14,5	М'якозерний	9458758,707	9,46
60	14	М'якозерний	9454859,138	9,46
60	13,5	М'якозерний	9413609,569	9,42
80	16	М'якозерний	9378949,914	9,38
80	13	М'якозерний	9374227,5	9,38
80	15,5	М'якозерний	9369447,845	9,37
80	15	М'якозерний	9333800,776	9,34
80	14,5	М'якозерний	9290683,707	9,3
80	14	М'якозерний	9273711,638	9,28
80	13,5	М'якозерний	9254872,069	9,26
80	13	М'якозерний	9226695	9,23
100	16	М'якозерний	9192199,914	9,2

Рисунок 3. Пошук найбільшої виручки для показників обробки м'яких сортів зерна

Тривалість лушіння, с X1	Вологість, % X2	Тип зерна	Виручка, грн	Виручка, млн грн
20	15	Твердозерний	10067728,28	10,07
20	14,5	Твердозерний	10058226,21	10,06
20	15,5	Твердозерний	10056687,84	10,06
20	16	Твердозерний	10054984,91	10,06
20	14	Твердозерний	10046856,64	10,05
40	15	Твердозерний	10039715,78	10,04
40	16	Твердозерний	10038177,41	10,04
40	15,5	Твердозерний	10026807,84	10,03
40	14,5	Твердозерний	10017141,21	10,02
20	13	Твердозерний	10001707,5	10,01
20	13,5	Твердозерний	9988799,569	9,99
40	14	Твердозерний	9981494,138	9,99
40	13,5	Твердозерний	9960787,069	9,97
40	13	Твердозерний	9949417,5	9,95
60	16	Твердозерний	9933597,414	9,94
60	15,5	Твердозерний	9920360,345	9,93
60	15	Твердозерний	9908990,776	9,91
60	14,5	Твердозерний	9875211,207	9,88
60	14	Твердозерний	9854504,138	9,86
60	13	Твердозерний	9841102,5	9,85
60	13,5	Твердозерний	9830062,069	9,84
80	15,5	Твердозерний	9739212,845	9,74
80	15	Твердозерний	9720373,276	9,73
80	16	Твердозерний	9716967,414	9,72
80	14,5	Твердозерний	9677256,207	9,68
100	16	Твердозерний	9657207,414	9,66
100	15,5	Твердозерний	9643970,345	9,65
100	15	Твердозерний	9619528,276	9,62
80	14	Твердозерний	9609861,638	9,61

Рисунок 4. Пошук найбільшої виручки для показників обробки твердих сортів зерна

Проте припущення не дає чіткої впевненості в ефективності роботи підприємства після вибору даного сценарію роботи.

Так як отримані дані є часовим рядом, то на основі відфільтрованих даних було побудовано різні типи рівнянь авторегресії [3, 5, 6, 7]. В результаті порівняння оцінок було виявлено найкращу модель для подальшого прогнозування прибутку. Частина даних наведено в таблиці 1.

Таблиця 1. Показники, отримані для різних моделей авторегресії.

Модель	R^2	СКП	DW	СеКП	САП	САПП	U
AP(1)	0,301908	37,15439	1,807913	0,647216	0,537113	6,010733	0,983385
AP(4)	0,331257	31,80088	1,994592	0,650588	0,538119	6,037392	0,979757
AP(7)	0,351645	33,67155	2,109945	0,655933	0,655933	6,141945	0,976059
AP(11)	0,972906	1,351561	2,031354	0,296172	0,243550	2,736651	0,427008
AP(12)	0,972983	1,335510	1,968940	0,294263	0,240904	2,709157	0,423287
AP(14)	0,973359	1,309259	1,998872	0,298760	0,245208	2,459805	0,425987
AP(16)	0,974120	1,271503	1,989156	0,283336	0,232562	2,610530	0,399008
AP(17)	0,974058	1,269783	1,997185	0,288819	0,240220	2,695588	0,405153
AP(8) +регресори	0,999631	0,019121	0,268604	0,012783	0,009921	0,210940	0,018059
AP(8)+регре сори+тренд	0,999900	0,005179	0,929176	0,006644	0,005033	0,055541	0,009603

Для вибору найкращої моделі було розглянуто показники R^2 , середньо квадратична похибка статистична (СКП), коефіцієнт Дарбіна-Уотсона, показники якості такі, як середня квадратична похибка (СеКП), середня абсолютна похибка, середня абсолютна похибка в процентах, коефіцієнт Тейла [3, 6].

3. ОТРИМАНІ РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Для перевірки правильності роботи прогнозу, побудуємо прогноз на період з уже відомими даними. Тестовий прогноз продемонстрував високу якість, про що свідчить рис. 5.

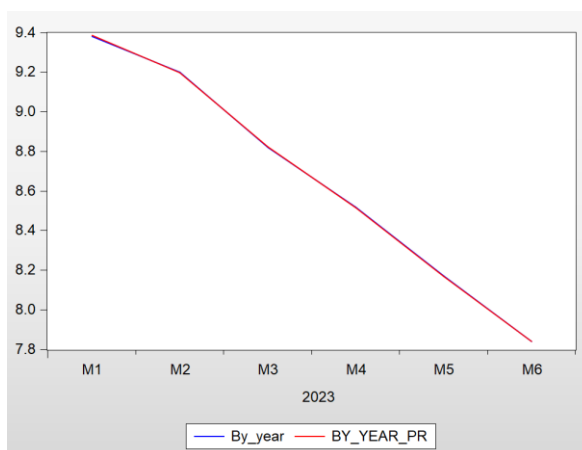


Рисунок 5. Графік реального та прогнозованого значення за 6 місяців

На графіку синім кольором позначено реальні дані, червоним прогнозовані. Як видно з рисунку, прогнозовані значення майже повністю відповідають реальним. Це свідчить про високу точність побудованого прогнозу.

Побудуємо динамічний прогноз, використовуючи найкращу модель, знайдену у результаті проведеного дослідження [8]. Обираємо саме динамічний прогноз, оскільки він дозволяє прогнозувати значення на обраний час. Перше значення прогнозу буде побудоване на основі реальних історичних даних, всі подальші будуються вже з урахуванням прогнозованих значень. Оскільки в результаті маємо надати інвестору прогноз мінімум на рік, вибираємо період 1 рік. Для наглядності результатів, наведемо отриманий прогноз разом з реальними даними за попередній період (таблиця 2).

Таблиця 2. Порівняння прогнозованої виручки з історичними даними за попередній період.

Місяць	Виручка, млн.грн (2022-2023)	Виручка, млн.грн (2023-2024)
Липень	8,49	8,63
Серпень	8,13	8,60
Вересень	7,8	8,56
Жовтень	9,9	8,52
Листопад	9,72	8,48
Грудень	9,52	8,44
Січень	9,38	8,40
Лютий	9,2	8,36
Березень	8,82	8,33
Квітень	8,52	8,32
Травень	8,17	8,27
Червень	7,84	8,23
Річний	105,49	101,14

4. ПРАКТИЧНЕ ЗАСТОСУВАННЯ ТА ВИСНОВКИ

Прогнозування широко використовується для соціальних досліджень, в метеорологічній сфері, в медицині, тощо. Проте дуже часто можна зустріти людей, які кажуть, що неможливо надати відносно точний прогноз для роботи деяких підприємств. В основному це стосується підприємств сільського господарства. Така позиція викликає труднощі при аналізі роботи підприємства. Також, відсутність прогнозів відлякують інвесторів від сільськогосподарської сфери.

В результаті дослідження було показано, що обрана модель та обраний метод прогнозування показали високу точність, що дозволяє робити прогноз на майбутнє. Отриманий прогноз пропонується представити потенційному інвестору. Таким чином людина, яку зацікавила робота конкретного підприємства може зробити висновок, як швидко інвестор зможе отримати прибуток з власної інвестиції, чи доцільно робити вкладення саме зараз в це підприємство, чи не будуть уже налаштовані процеси збитковими тощо. Також, в результаті проведеного аналізу роботи підприємства, в даному випадку це був елеватор, можна запропонувати інвестору чи керівнику підприємства сценарії роботи, при яких підприємство може отримувати більший прибуток, ніж для технології, яку використовують на проведення аналізу. Для підтримки точності прогнозу, рекомендується регулярно оновлювати історичні дані. В разі великого відхилення очікуваних результатів від реальних прогноз буде перероблено.

Під час вивчення потенційного ринку для стартапу, було виявлено три основні цільові аудиторії. Керівники підприємств могли б після замовлення послуги отримати аналіз роботи

їх підприємства, потенційний прибуток на період, що був обраний для прогнозування, а також можливі сценарії розвитку, для забезпечення більшого прибутку. Інвесторам, що замовили б аналіз підприємства, що їх цікавить, був би наданий схожий звіт, як керівникам підприємства, але інвестори могли б зробити висновок, чи цікавить їх дане підприємство з точки зору інвестування. Проектні організації пропонують багато послуг, в тому числі побудова нового підприємства та введення його в експлуатацію або оптимізація роботи вже існуючого підприємства. На початку роботи над подібними проектами компанії буде корисно отримати прогнози стосовно можливого прибутку, щоб зрозуміти доцільність побудови нового підприємства чи змін в роботі вже існуючого, та прогнози стосовно запитів на переробку та зберігання врожаю, щоб налаштувати логістичні процеси. Для подальшого розвитку використання технології можна також розглядати вектор взаємодії з подібними компаніями, з подальшою побудовою власної компанії з унікальним переліком послуг.

Таким чином, проведене дослідження продемонструвало можливість якісного прогнозування фінансових показників сільськогосподарського підприємства та виявило шляхи подальшого застосування отриманих прогнозів.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Засоба Сергій, *Розвиток агропромислового комплексу України*, Український журнал прикладної економіки та техніки, 2020, 402-409.
2. Агропромисловий комплекс України: сучасний стан розвитку та особливості. URL: <https://www.profbuild.in.ua/uk/stati-2/871-agropromislovij-kompleks-ukrajini-suchasnij-stan-rozvitku-ta-osoblivosti> (дата звернення 10.09.2023).
3. Бідюк П. І. Аналіз часових рядів (навчальний посібник) / Бідюк П.І., Романенко В.Д., Тимощук О.Л. // Аналіз часових рядів (навчальний посібник) — Київ: Політехніка, 2010. — 317 с.
4. Бідюк П. І. Системний підхід до побудови математичних моделей на основі часових рядів / П. І. Бідюк, І. В Баклан., В. М Рифа. // Системні дослідження та інформаційні технології. – 2002. – № 3. – 131 с.
5. Бідюк П. І. Часові ряди: моделювання та прогнозування / П. І. Бідюк, О. І. Савенков, І. В Баклан. – Київ : ЕКМО, 2004. – 144 с.
6. Enders W. Applied econometric time series / Enders W. – New York: John Wiley & Sons, Inc., 1995. – 434 p
7. Dehove A, Commault J, Petitclerc M, Teissier M, Macé J. Economic analysis and costing of animal health: a literature review of methods and importance. Rev Off Int Epizoot. (2012) 31:605–17. 10.20506/rst.31.2.2146
8. Berentsen P, Dijkhuizen A, Oskam A. A dynamic model for cost-benefit analyses of foot-and-mouth disease control strategies. Prev Vet Med. (1992) 12:229–43. 10.1016/0167-5877(92)90052-H
9. Gethmann J, Probst C, Sauter-Louis C, Conraths FJ. Economic analysis of animal disease outbreaks - BSE and Bluetongue disease as examples. Berl Munch Tierarztl Wochenschr. (2015) 128:478–82. 10.2376/0005-9366-128-478

ПОРІВНЯЛЬНИЙ АНАЛІЗ ТА ПОКРАЩЕННЯ МОДЕЛЕЙ ПРОГНОЗУВАННЯ ЦІН АКЦІЙ НА ФІНАНСОВОМУ РИНКУ

Муравльов А.Д., Гуськова В.Г.

Національний технічний університет України «Київський політехнічний інститут ім. Ігоря Сікорського»

1. ВСТУП

У динамічному середовищі фінансового ринку точне прогнозування цін на акції вже давно перебуває в центрі уваги інвесторів, аналітиків та дослідників. Безперервні припливи і відпливи ринкових тенденцій під впливом безлічі факторів – від економічних показників до геополітичних подій – роблять прогнозування цін на акції складним завданням. Оскільки фінансовий світ стає все більш взаємопов'язаним і керованим даними, попит на надійні моделі прогнозування зростає.

Метою є дослідження сфери прогнозування цін на акції через призму порівняльного аналізу та вдосконалення. Заглиблюючись у тонкощі існуючих моделей, ми розглядаємо різноманітні методології, що застосовуються для прогнозування цін на акції, та досліджуємо їхню ефективність у фінансовому ландшафті, що постійно змінюється. Основна мета полягає в тому, щоб виявити не лише сильні та слабкі сторони, притаманні сучасним моделям прогнозування, але й запропонувати вдосконалення, які можуть сприяти розробці більш точних та надійних інструментів прогнозування.

Шляхом всебічного вивчення домінуючих підходів, статистичних методів та алгоритмів машинного навчання, ця стаття має на меті висвітлити досягнення у сфері прогнозування фондових цін. Здійснюючи порівняльний аналіз, ми прагнемо виокремити ключові ідеї, які проливають світло на найефективніші стратегії прогнозування руху цін на акції. Крім того, дослідження шляхів вдосконалення існуючих моделей відкриває шлях до підвищення точності прогнозування та зменшення ризиків, пов'язаних з прийняттям фінансових рішень.

2. ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

Проблема дослідження:

Фінансові ринки є складними та волатильними системами, де ціни акцій піддаються впливу різних факторів, таких як економічні новини, геополітичні події, інтереси інвесторів та багато інших. Прогнозування цін акцій є критично важливим завданням для інвесторів, брокерів, фондових аналітиків та компаній, що публікують акції. Точні та надійні моделі прогнозування цін акцій можуть допомогти приймати кращі рішення з управління інвестиціями та ризиками.

Мета дослідження:

Метою є проведення порівняльного аналізу різних моделей прогнозування цін акцій на фінансовому ринку з метою вдосконалення їх точності та надійності. Основні завдання дослідження включають:

1. Вивчення і аналіз існуючих моделей прогнозування цін акцій, таких як моделі часових рядів, машинного навчання та штучні нейронні мережі.
2. Порівняльний аналіз різних методів та моделей на історичних даних ринку для визначення їхньої точності та відповідності реальним умовам.
3. Розробка покращень та оптимізація обраних моделей з метою зменшення помилок та покращення прогностичної здатності.

4. Практичне застосування найкращої моделі прогнозування на реальних даних ринку для перевірки її ефективності в реальних умовах.

Висновки та рекомендації для інвесторів та фінансових аналітиків щодо використання найкращої моделі прогнозування для прийняття рішень щодо інвестицій та управління ризиками на фінансовому ринку.

3. ОГЛЯД МЕТОДІВ ПРОГНОЗУВАННЯ

На сьогоднішній день існує низка сучасних методів прогнозування цін акцій, і вони можуть бути поділені на кілька груп в залежності від підходу та методології.

Методи машинного навчання:

– **Лінійна регресія:** Моделює залежність між ціною акцій та факторами, такими як обсяг торгівлі, індикатори ринку тощо.

– **Дерева рішень та випадкові ліси:** Використовуються для розробки моделей, які можуть враховувати нелінійні взаємозв'язки між факторами та цінами акцій.

RF (random forest) – це безліч вирішальних дерев. У задачі регресії їх відповіді усереднюються, в завданні класифікації приймається рішення голосуванням за більшістю. Всі дерева будуються незалежно за наступною схемою:

– Вибирається підвибірка навчальної вибірки розміру – по ній будується дерево (для кожного дерева – своя підвибірка);

– Для побудови кожного розщеплення в дереві переглядаємо `max_features` випадкових ознак (для кожного нового розщеплення – свої випадкові ознаки);

– Вибираємо найкращу ознаку і розщеплення по ній (за задалегідь заданим критерієм). Дерево будується, як правило, до вичерпання вибірки (поки в листі не залишаться представники тільки одного класу), але в сучасних реалізаціях є параметри, які обмежують висоту дерева, число об'єктів в листі і число об'єктів в підвибірці, при якому проводиться розщеплення

Зрозуміло, що така схема побудови відповідає головному принципу ансамблювання (побудови алгоритму машинного навчання на базі кількох, в даному випадку вирішальних дерев): базові алгоритми повинні бути хорошими і різноманітними (тому кожне дерево будується на своїй навчальній вибірці і при виборі розщеплення є елемент випадковості).

Чим більше дерев, тим краща якість, але час налаштування і роботи RF також пропорційно збільшуються. Часто при збільшенні кількості дерев якість на навчальній вибірці підвищується (може навіть доходити до 100%), а якість на тестовій вибірці виходить на асимптоту.

Метод випадкового лісу заснований на методі вирішальних дерев. Випадковий ліс - це безліч вирішальних дерев, а клас об'єкта, що проходить класифікацію, вибирається голосуванням більшістю.

Згенеруємо випадкову вибірку S розміру l по вихідній навчальній вибірці $D = \{x_i y_i\}_{i=1}^l$.

За вибіркою S індукувати неусічене дерево рішень T_i з мінімальною кількістю спостережень в термінальних вершинах рівним n_{min} , рекурсивно слідуючи до наступного підалгоритма:

- з вихідного набору p ознак випадково вибрати p ознак,
- з p ознак вибрати ознаку, яка забезпечує найкраще рішення,
- розщепити вибірку, відповідну до оброблюваної вершини, на дві підвибірки.

В результаті отримуємо ансамбль дерев рішень $\{T_i\}_{i=1}^B$.

Класифікація нових спостережень здійснюється наступним чином: нехай $\hat{y}_l(x) \in \{y_1, \dots, y_l\}$ – клас, передбачений деревом рішень T_i , тобто $T_i(x) = \hat{y}_l(x)$, тоді $\widehat{y}_{rf}^B(x)$ – клас, найбільш часто зустрічається в множині $\{\hat{y}_b(x)\}_{b=1}^B$.

– **Метод опорних векторів:** шукає оптимальну розділяючу гіперплощину, яка максимізує відстань між двома наборами даних різних класів.

Лінійна класифікація

Нехай є навчальна вибірка:

$$(\bar{x}_1, y_1), \dots, (\bar{x}_m, y_m), \bar{x}_i \in R^n, y_i \in \{-1, 1\}.$$

Метод опорних векторів буде класифікаційну функцію F у вигляді:

$$F(x) = \text{sign}(\bar{w} \cdot \bar{x} - b),$$

де $\bar{w} \cdot \bar{x}$ – скалярний добуток, \bar{w} — нормальний вектор до розділювальної гіперплощини, b – зсув. Ті об'єкти, для яких $F(x) = 1$, потрапляють в один клас, а об'єкти з $F(x) = -1$ – в інший.

Якщо навчальна вибірка містить два класи даних, які можна лінійно розділити, то ми можемо обрати дві паралельні гіперплощини, які розділяють два класи даних так, що відстань між ними якомога більша. Область, обмежена цими двома гіперплощинами, називається «розділенням», а максимально розділова гіперплощина це гіперплощина, яка лежить посередині між цими двома. Ці гіперплощини може бути описано рівняннями:

$$\begin{aligned} \bar{w} \cdot \bar{x} - b &= 1; \\ \bar{w} \cdot \bar{x} - b &= -1. \end{aligned}$$

З геометричної точки зору, відстанню між цими двома гіперплощинами є $\frac{2}{\|\bar{w}\|}$, тому для максимізації відстані між ними нам треба мінімізувати $\|\bar{w}\|$. Оскільки ми також маємо завадити потраплянню точок даних до розділення, ми додаємо таке обмеження: для кожного i ,

$$\begin{aligned} \text{або } \bar{w} \cdot \bar{x}_i - b &\geq 1 \text{ якщо } y_i=1; \\ \text{або } \bar{w} \cdot \bar{x}_i - b &\leq -1 \text{ якщо } y_i=-1. \end{aligned}$$

Ці обмеження стверджують, що кожна точка даних мусить лежати з правильного боку розділення. Ці дві нерівності можна записати як одну:

$$y_i(\bar{w} \cdot \bar{x}_i - b) \geq 1, \text{ для всіх } 1 \leq i \leq n.$$

З цього отримуємо задачу оптимізації “Мінімізувати $\|\bar{w}\|$ за умови $y_i(\bar{w} \cdot \bar{x}_i - b) \geq 1$ для $i=1, \dots, n$.”

Значення \bar{w} та b які розв’язують цю задачу — визначають класифікатор:

$$\bar{x} \rightarrow \text{sgn}(\bar{w} \cdot \bar{x} - b).$$

Очевидним, але важливим наслідком цього геометричного опису є те, що максимально розділова гіперплощина повністю визначається тими \bar{x}_i , які лежать найближче до неї. Ці \bar{x}_i називають опорними векторами.

Нелінійна класифікація

На практиці випадки, коли дані можна розділити гіперплощиною, або, як ще кажуть, лінійно, досить рідкісні.

У цьому разі чинять так: усі елементи навчальної вибірки вкладають у простір X вищої розмірності за допомогою спеціального відображення

$$\varphi: R^n \rightarrow X.$$

При цьому відображення φ вибирають так, щоб у новому просторі X вибірка була лінійно роздільна.

Класифікаційна функція F набуває вигляду

$$F(x) = \text{sign}(\bar{w} \cdot \varphi(\bar{x}) - b).$$

Вираз $K(\bar{x}_i, \bar{x}_j) = \varphi(\bar{x}_i) \cdot \varphi(\bar{x}_j)$ називається ядром класифікатора. З математичної точки зору **ядром** може слугувати будь-яка позитивно визначена симетрична функція двох змінних. Позитивна визначеність необхідна для того, щоб відповідна функція Лагранжа в задачі оптимізації була обмежена знизу, тобто задача оптимізації була б коректно визначена. Слід зазначити, що робота в просторі ознак високої вимірності збільшує похибку узагальнення опорно-векторних машин, хоча за достатньої кількості зразків цей алгоритм все одно працює добре.

Точність класифікатора залежить, зокрема, від вибору ядра. Найчастіше на практиці зустрічаються такі ядра:

– Поліноміальне. При його використанні обчислюються усі можливі комбінації вихідних ознак об'єкта до певної степені;

$$K(\bar{x}_i, \bar{x}_j) = (\bar{x}_i \cdot \bar{x}_j + C)^d.$$

– Радіальна базисна функція. При його використанні обчислюються усі можливі поліноміальні комбінації усіх степенів щодо вихідних ознак об'єкта.

$$K(\bar{x}_i, \bar{x}_j) = e^{-\gamma \|\bar{x}_i - \bar{x}_j\|^2}.$$

– Сигмоїдне

$$K(\bar{x}_i, \bar{x}_j) = \tanh(k\bar{x}_i \cdot \bar{x}_j + C).$$

Ядро пов'язано з перетворенням $\varphi(\bar{x})$ через рівняння $K(\bar{x}_i, \bar{x}_j) = \varphi(\bar{x}_i) \cdot \varphi(\bar{x}_j)$. Значення w також знаходиться в перетвореному просторі:

$$\bar{w} = \sum_i \alpha_i y_i \varphi(\bar{x}_i).$$

Скалярні добутки з w для класифікації, знов-таки, може бути обчислювано за допомогою ядрового трюку, тобто:

$$\bar{w} \cdot \varphi(\bar{x}) = \sum_i \alpha_i y_i \varphi K(\bar{x}_i, \bar{x}).$$

4. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Результати прогнозування за метриками представлено в таблиці 1.

Таблиця 1. Результати прогнозування за метриками

Метод	RMSE	MSE	MAE	MAPE	R2 Score
RF(10 trees)	3,62	13,152	2,86	0,0172	0,8525
RF(100 trees)	3,75	14,067	2,975	0,0178	0,8422
RF(1000 trees)	3,638	13,236	2,864	0,01718	0,8516
RF(10000 trees)	3,610	13,037	2,842	0,01705	0,853
SVM(linear)	2,97	8,84	2,35	0,014	0,9008
SVM(polynomial)	7,28	53,044	5,722	0,034	0,405
SVM(rbf)	3,69	13,66	2,931	0,0175	0,8468
SVM(sigmoid)	3679,8	13541002	2718	15	-151812

5. ВИСНОВКИ

Внутрішня структура моделей:

SVM: SVM будує лінійну або нелінійну гіперплощину, що розділяє дані, і залежно від типу задачі використовує різні ядерні функції для перетворення даних у вищорозмірний простір. Ідея полягає в тому, що дані можуть бути лінійно нероздільними у вихідному просторі, але можуть бути розділені гіперплощиною у вищорозмірному просторі.

Random Forest: Випадковий ліс складається з декількох рішачих дерев, які побудовані на основі підмножини даних та підмножини ознак. Кожне дерево незалежно приймає рішення, шляхом проходження вниз по дереву за допомогою послідовних тестів на ознаках. Остаточне рішення випадкового лісу визначається більшістю голосів дерев.

Обробка даних:

SVM: SVM вимагає числових значень ознак та масштабування даних. Категоріальні ознаки потребують попереднього перетворення в числові.

Random Forest: Випадковий ліс може працювати з числовими та категоріальними ознаками без попереднього масштабування даних.

Обидва методи мають свої переваги та обмеження, і вибір між ними залежить від характеристик даних, розміру вибірки та вимог до точності та швидкодії моделі. SVM часто ефективний для завдань з невеликими даними та коли важлива межа розділення класів. Випадковий ліс може бути корисним у великих наборах даних, коли потрібна висока точність та стійкість до перенавчання.

На датасеті, на якому порівнювалися моделі, Random Forest показав середні результати, помітна просадка точності при першому збільшенні кількості дерев, однак на більшій кількості точність підвищилась. Random Forest показав себе як більш стабільний варіант, однак класифікація за допомогою лінійного класифікатора SVM надала кращі результати.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Орельен Жерон. Прикладное машинное обучение с помощью Scikit-Learn ч TensorFlow. 2018. 688 с.
2. Случайный лес (Random Forest). URL: <https://dyakonov.org/2016/11/14/случайный-лес-random-forest/>.
3. Чистяков С.П. Случайные леса: обзор // Труды Карельского научного центра РАН. 2013. №1. С. 117-136.
4. Метрики в задачах машинного обучения. URL: <https://habr.com/ru/company/ods/blog/328372/>.
5. Ben-Hur, Asa; Horn, David; Siegelmann, Hava; Vapnik, Vladimir N. "Support vector clustering" (2001);"
6. Зайченко Ю.П. Основи проектування інтелектуальних систем. 2004. с.49-66
7. Random forest. URL: https://ru.wikipedia.org/wiki/Random_forest.
8. <https://www.kaggle.com/competitions/nlp-getting-started/overview>

МОДЕЛІ ОПТИМАЛЬНОГО РОЗПОДІЛУ ДАНИХ

Мухін В.Є., Яковлева А.П., Шмідт А.Є.¹

Національний технічний університет України «Київський політехнічний інститут
ім. Ігоря Сікорського»

¹ shmidt.anatolii@lil.kpi.ua

Мета дослідження полягає у розробці та оптимізації математичних моделей для ефективного розподілу даних у різноманітних структурах мережі, таких як ієрархічна, кільцева та решітчаста. Використовуючи теоретичні та емпіричні методи дослідження, надано новий підхід до оптимізації розподілу даних. Наукова новизна полягає у вдосконаленні математичних концепцій для різних типів мереж та впровадженні нової стратегії оптимізації. Застосування розроблених моделей може позитивно вплинути на широкий спектр областей, де важливий оптимальний розподіл даних, включаючи технологічні, комунікаційні та наукові сфери.

Ключові слова: розподіл даних, розподілені бази даних, математична модель, мережева структура, оптимізація.

1. ВСТУП

У контексті зростаючого обсягу даних у сучасному світі, ефективний розподіл і управління ними стає невід'ємною складовою оптимального функціонування різноманітних інформаційних систем. Особливо важливим стає це завдання в рамках різних мережевих структур, таких як ієрархічні, кільцеві та решітчасті мережі.

У цьому дослідженні ставиться за мету розгляд і оптимізацію математичних моделей для оптимального розподілу даних у зазначених типах мереж. Зосереджуючись на теоретичних аспектах та використовуючи математичні моделі, дослідження спрямоване на створення інноваційних підходів, призначених вдосконалити процеси управління даними в різноманітних мережах та сприяти підвищенню їхньої продуктивності.

2. МАТЕМАТИЧНІ МОДЕЛІ РОЗПОДІЛУ ДАНИХ

Перш за все, дамо визначення мережам, що будуть для яких побудуємо моделі.

Розглядатимуться розподілені бази даних. Розподілена база даних — сукупність логічно взаємопов'язаних баз даних, розподілених у комп'ютерній мережі. Логічний зв'язок баз даних в розподіленій базі даних забезпечує система управління розподіленою базою даних, яка дозволяє управляти розподіленою базою даних таким чином, щоб створювати у користувачів ілюзію цілісної бази даних [1].

Моделі побудовані для наступних типів (топологій) мереж: кільцевої, решітчастої, ієрархічної.

Кільцева топологія — це послідовний ланцюжок у замкнутому циклі. Дані переміщуються по кільцю в одному напрямку. Коли один вузол надсилає дані іншому, дані проходять через кожен проміжний вузол кільця, поки не досягнуть місця призначення. Проміжні вузли повторюють (ретранслюють) дані, щоб сигнал був сильним [2].

Плоска решітчаста топологія передбачає, що кожен її вузол сполучений із найближчим сусідом (Рисунок 1) [3]. У цій роботі моделюється багаторівнева плоска решітчаста мережа, де на кожному рівні 4 вузла.

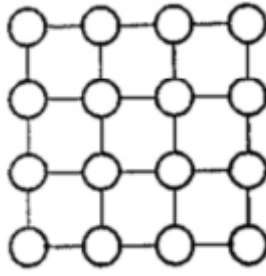


Рисунок 1. Плоска решітчаста мережа

Топологія дерева — це мережа, в якій кожен вузол пов'язаний з іншими в ієрархії. Її також називають ієрархічною топологією, оскільки в цій топології всі елементи розташовані як гілки дерева. У топології дерева будь-які два пов'язані вузли можуть мати лише одне взаємне з'єднання, отже, між ними може бути лише одне з'єднання [4].

Перейдемо до загальної моделі розподілу даних. Математична задача полягає у знаходженні мінімальної кількості серверів в заданій мережі при якій час відповіді теж буде мінімальний. Зі зміною типу мережі змінюється середній шлях повідомлення (кількість переходів між вузлами). Для спрощення моделі припускаємо, що дані розподілені між вузлами мережі рівномірно та сервери мають однакову пропускну здатність.

Для заданої задачі доцільно сформуванати наступне рівняння:

$$T(N_s) = \frac{N}{N_s} * T_s + L(N_s) * T_r.$$

$T(N_s)$ – це функція середнього часу відповіді від мережі в залежності від кількості серверів;

N_s – це кількість серверів;

N – це кількість записів в розподіленій базі даних;

T_s – це час пошуку на одному сервері;

$L(N_s)$ - це функція середнього шляху повідомлення в мережі в залежності від кількості серверів;

T_r – це час переходу повідомлення між серверами.

Це загальне рівняння, що описує загальну модель. Для кожного типу мережі необхідно визначити власну функцію середнього шляху $L(N_s)$.

Для кільцевої мережі визначаємо наступну функцію середнього шляху повідомлення ($\lceil \cdot \rceil$ -оператор округлення до меншого цілого числа):

$$L^1(N_s) = \frac{N_s \lceil \frac{N_s+1}{2} \rceil}{N_s}.$$

Розглядається багаторівнева решітчаста мережа, що на кожному рівні має 4 вузли. Для цієї мережі визначаємо наступне рівняння:

$$L^2(N_s) = \frac{N_s (\lceil \frac{N_s}{4} \rceil + 1)}{N_s - 1}.$$

Для ієрархічної мережі введемо параметр I , що визначатиме кількість підпорядкованих вузлів для кожного вузла вищого рангу. Визначимо рівняння:

$$L^3(N_s) = 2 * \log_I N_s.$$

3. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

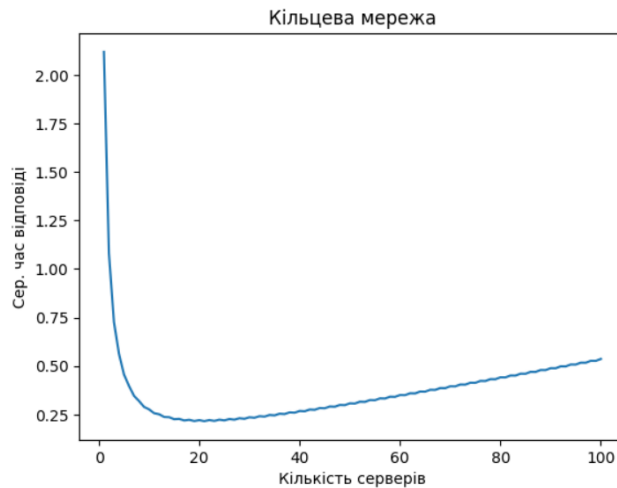
Для дослідження було створено програму, що реалізує описані математичні моделі. Програма створена у хмарному середовищі розробки Google Colab на мові програмування Python.

Для експерименту було використано наступні параметри: Кількість записів (N) – 1000; Час пошуку (T_s) – 0,0021 с; Час переходу (T_r) – 0,01 с.

Проведемо аналіз отриманих результатів кільцевої мережі (Рисунок 2). Середній час відповіді T зменшується до певного моменту, до оптимального значення кількості серверів N_s^* , а далі середній час збільшується. При збільшенні кількості серверів збільшується час передачі, тому після досягнення оптимального значення N_s^* сенсу додавати нові сервери немає.

Розглянемо графік на меншому відрізку (Рисунок 3) в який входить $N_s^*=21$. Бачимо, що для парної кількості серверів середній час відповіді T більший ніж сусідніх непарних значень. Це можна пояснити тим, що шлях запиту для непарної кількості серверів, наприклад 19, такий же як і для парної, наприклад 20, а якщо додати ще один сервер, то середнє навантаження зменшиться і відповідно середній час відповіді T .

Мінімальне значення середнього часу відповіді T^* досягається при N_s^* , $T^* = 0,2152$ с.



Оптимальна кількість серверів: 21

Рисунок 2. Результати кільцевої мережі

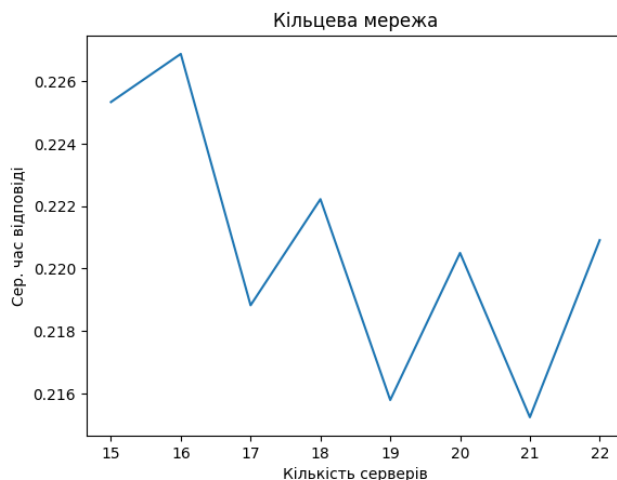
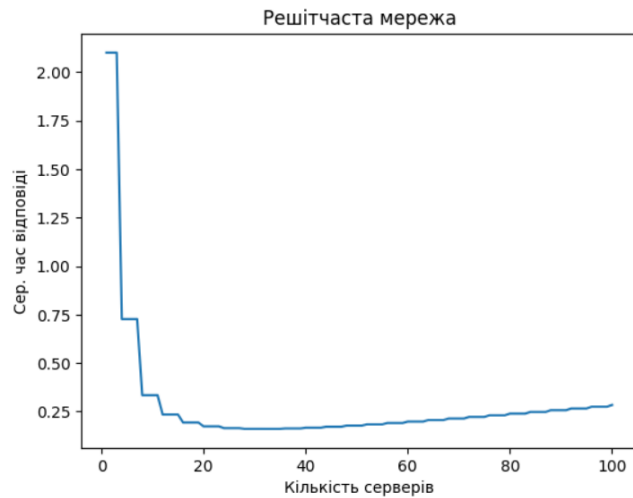


Рисунок 3. Результати кільцевої мережі на меншому відрізку

Проведемо аналіз отриманих результатів для решітчастої мережі (Рисунок 4). Середній час відповіді T зменшується до певного моменту, до оптимального значення кількості серверів Ns^* , а далі середній час збільшується.

Розглянемо графік на меншому відрізку (Рисунок 5) в який входить $Ns^*=32$. На цьому графіку краще помітно, що значення T є однаковим на одному рівні. Рівень складається з 4 серверів.

Мінімальне значення середнього часу відповіді T^* досягається при Ns^* , $T^* = 0,1606$. Порівняно з кільцевою мережею оптимальний середній час відповіді менший, але і серверів потрібно значно більше.



Оптимальна кількість рівнів мережі: 8

Рисунок 4. Результати решітчастої мережі

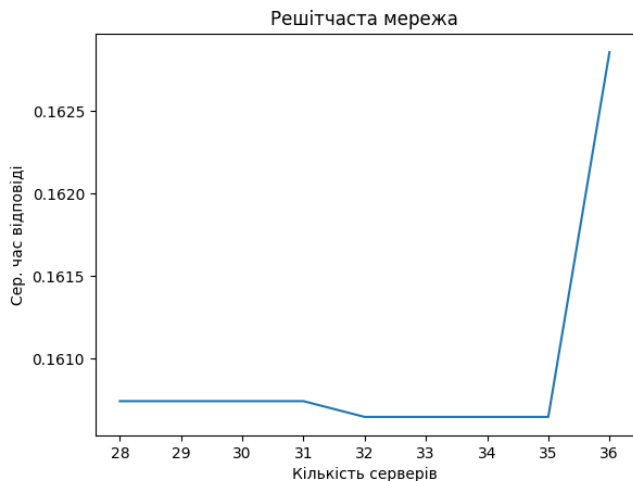


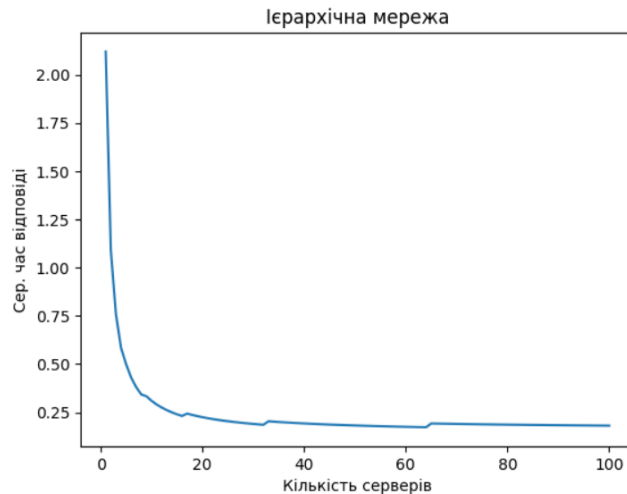
Рисунок 5. Результати решітчастої мережі на меншому відрізку

Проведемо аналіз отриманих результатів (Рисунок 6). Згідно встановлених параметрів сервери в ієрархії мають по два підпорядковані сервери меншого рангу ($I = 2$). Середній час відповіді T досягає оптимального значення кількості серверів Ns^* , а далі більше оптимального.

Розглянемо графік на меншому відрізку (Рисунок 7) в який входить $Ns^*=64$. На графіку помітно, що при збільшенні рівня ієрархії суттєво збільшується середній час відповіді T . Це

викликано тим, що збільшується кількість переходів які потрібно пройти для доступу до даних, що збільшує час передачі.

Мінімальне значення середнього часу відповіді T^* досягається при Ns^* , $T^* = 0,1728$. Порівняно з кільцевою мережею оптимальний середній час відповіді менший, але і серверів потрібно значно більше. Решітчаста мережа показала кращі результати по обом показникам.



Оптимальна кількість серверів: 64
 Оптимальна кількість рівнів мережі: 7

Рисунок 6. Результати ієрархічної мережі

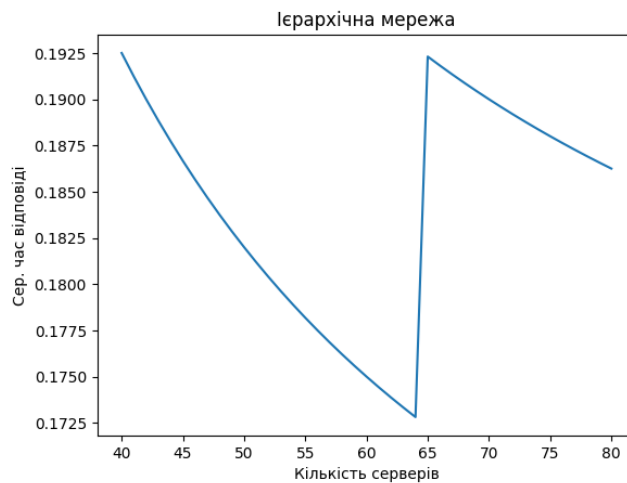


Рисунок 7. Результати ієрархічної мережі на меншому відрізку

У попередніх експериментах було проведено порівняння при наступних параметрах: кількість записів (N) – 1000; час пошуку (T_s) – 0,0021 с; час переходу (T_r) – 0,01 с. Для цих параметрів оптимальною за середнім часом відповіді виявилась решітчаста мережа.

Проведемо експеримент для вдвічі більшої кількості записів. Отже, параметри наступні: кількість записів (N) – 1000; час пошуку (T_s) – 0,0021 с; час переходу (T_r) – 0,01 с; кількість зв'язків ієрархічної мережі (I) – 2.

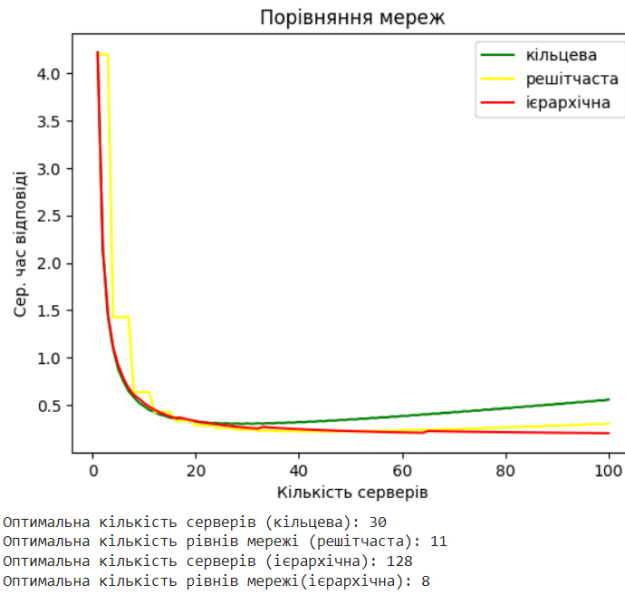


Рисунок 8. Результати для другого набору параметрів

Найменший оптимальний середній час відповіді $T^* = 0,1928$ досягається ієрархічною мережею, але за рахунок великої кількості серверів.

4. ВИСНОВКИ

У висновку важливо підкреслити, що використання математичних моделей є критично важливим для вирішення складних завдань в галузі оптимізації розподілу даних у серверних мережах. Математичні моделі надають структурований та науково обґрунтований підхід до вивчення складних систем, дозволяючи аналізувати взаємодію різних елементів та прогнозувати їхню поведінку.

В результаті дослідження вдалося провести експерименти та проаналізувати їх. Експерименти показали, що ієрархічна мережа найкраще проявляє себе при великій кількості даних або при малій швидкості обробки інформації. Кільцева та решітчаста мережі проявляють себе краще при менших об'ємах даних.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. К. Дж. Дейт Введение в системы баз данных / An Introduction to Database Systems. 8-ме вид. "Вильямс", 2005. 1328 с.
2. Networking complete. 2-ге вид. San Francisco : Sybex, 2001. 877 с.
3. Комп'ютерні системи : методичні вказівки до лабораторних робіт / укл.: Баловсяк С. В., Одайська Х. С. Чернівці : Чернівецький національний університет ім. Ю. Федьковича, 2021. 72 с
4. What is Tree Topology? Definition and Explanation - javatpoint. www.javatpoint.com. URL: <https://www.javatpoint.com/what-is-tree-topology> (дата звернення: 29.11.2023).к

ПРОГНОЗУВАННЯ КРЕДИТНОЇ СПРОМОЖНОСТІ КЛІЄНТІВ БАНКУ НА ОСНОВІ АНАЛІЗУ ФІНАНСОВИХ ДАНИХ

Петровський В.Є., Гуськова В.Г.

Національний технічний університет України «Київський політехнічний інститут ім. Ігоря Сікорського»

1. ВСТУП

У сучасних умовах економічної нестабільності та високого рівня конкуренції, важливість ефективного управління ризиками, пов'язаними з кредитуванням, набуває першочергового значення для фінансових установ. Одним із ключових аспектів цього управління є прогнозування кредитної спроможності клієнтів банку. Здатність ефективно визначити й управляти ризиками неплатоспроможності має велике значення для забезпечення стійкості та сталого розвитку фінансових установ.

Однією з перспективних стратегій є використання аналізу фінансових даних для створення точних та надійних моделей прогнозування кредитної спроможності. Розвиток технологій у галузі обробки даних та штучного інтелекту відкриває нові можливості для створення комплексних та інноваційних підходів до оцінки ризиків.

Дана наукова стаття присвячена вивченню сучасних методів та технік прогнозування кредитної спроможності клієнтів банку на основі аналізу фінансових даних. Автори ставлять за мету розглядати ключові аспекти цього процесу, включаючи вибір та обробку релевантних фінансових показників, застосування різноманітних методів аналізу, інтеграцію алгоритмів машинного навчання та оцінку впливу нестандартних даних на точність прогнозів.

Подальший розвиток ефективних стратегій прогнозування кредитної спроможності сприятиме покращенню якості управління кредитним портфелем, зменшенню ризиків та підвищенню фінансової стійкості банківських установ.

2. ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

У сучасному банківському секторі, де висока ступінь конкуренції та ризику вимагає від фінансових установ надзвичайної уважності до управління кредитним портфелем, прогнозування кредитної спроможності клієнтів є стратегічно важливим завданням. За останні роки, з розвитком технологій обробки даних та аналізу, виникає потреба у вдосконаленні методів прогнозування з метою забезпечення надійного управління ризиками.

Основною метою цього дослідження є розробка та впровадження ефективних методів прогнозування кредитної спроможності клієнтів банку на основі аналізу різноманітних фінансових даних. За допомогою сучасних підходів, включаючи алгоритми машинного навчання та статистичні методи, планується розробити моделі, які не лише точно визначатимуть ризик неплатоспроможності, але й дозволять уникнути надмірного відмовлення в кредитах солідним клієнтам.

Задачі дослідження включатимуть:

1. Визначення релевантних фінансових показників: Виділення ключових факторів, що впливають на кредитну спроможність, та їхнє визначення в контексті аналізу даних.
2. Розробка методів обробки та очищення даних: Створення ефективних та надійних методів для обробки фінансових даних, враховуючи можливі аномалії та неповноту інформації.
3. Вибір та оптимізація моделей прогнозування: Порівняння та оптимізація алгоритмів машинного навчання та статистичних методів для побудови точних та ефективних моделей прогнозування.

4. Інтеграція моделей в банківську практику: Розробка механізмів для ефективної інтеграції розроблених моделей в банківські системи та процеси управління кредитами.
5. Оцінка ефективності та надійності: Проведення комплексної оцінки розроблених моделей з метою визначення їхньої точності, чутливості та специфічності у прогнозуванні різних категорій клієнтів.

Це дослідження спрямоване на створення інноваційних підходів до прогнозування кредитної спроможності, які сприятимуть зниженню ризиків та підвищенню ефективності управління кредитним портфелем фінансових установ.

3. ОГЛЯД КЛАСИЧНИХ МЕТОДІВ ПРОГНОЗУВАННЯ КРЕДИТНОЇ СПРОМОЖНОСТІ КЛІЄНТІВ БАНКУ

1. **Multiple Linear Regression:** Метод Multiple Linear Regression (MLR) є класичним статистичним методом, що використовується для вивчення лінійних залежностей між залежною змінною та набором незалежних змінних. У контексті прогнозування кредитної спроможності, MLR дозволяє моделювати взаємозв'язок між різними фінансовими параметрами та ризиком неплатоспроможності.

Перевагами даного підходу є простота та легкість інтерпретації, а також відображення лінійних залежностей між змінними.

Обмеженнями є те, що модель працює краще в умовах лінійності, що може бути недостатнім для складних взаємозв'язків у фінансових даних.

2. **Gradient Boosting Regression:** Gradient Boosting є ансамблевим методом машинного навчання, який покращує прогнози, комбінуючи прості моделі, зосереджуючись на помилках попередніх моделей. У випадку прогнозування кредитної спроможності, Gradient Boosting може враховувати нелінійні та взаємодіючі ефекти між фінансовими параметрами.

Перевагами є те, що даний підхід добре вирішує проблему нелінійних взаємозв'язків та висока точність прогнозів. Обмеженнями є Схильність до перенавчання, що може виникнути при недостатньому контролі параметрів.

3. **Random Forest Regression.** Random Forest є іншим ансамблевим методом, який використовує декілька дерев рішень для отримання прогнозів. Кожне дерево обробляє підвибірки даних та випадковий вибір фіч, що робить модель менш схильною до перенавчання.

Перевагами є висока точність та стійкість до перенавчання, а також здатність обробляти великі набори даних з численними фічами.

Обмежена інтерпретованість порівняно з MLR та затратна обчислювально при великій кількості дерев.

4. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

За результатами дослідження було проведено ряд експериментів, що показали результати, представлені у таблиці 1.

Таблиця 1. Порівняльна таблиця методів: Multiple Linear Regression, Gradient Boosting Regression, та Random Forest Regression

Метрика	Multiple Linear Regression	Gradient Boosting Regression	Random Forest Regression
Середньоквадратична помилка (MSE)	0,2	0,05	0,04
Середня абсолютна помилка (MAE)	0,15	0,08	0,07
Коефіцієнт детермінації (R-squared)	0,3	0,9	0,92

5. ВИСНОВКИ

Multiple Linear Regression підходить для лінійних взаємозв'язків, але обмежений у моделюванні складних нелінійностей. Gradient Boosting Regression та Random Forest Regression мають високу точність, але Gradient Boosting потребує уважного налаштування, а Random Forest – ефективний при обробці великих обсягів даних. Всі методи досягають низьких помилок, проте Gradient Boosting та Random Forest найбільш точні за кількісними метриками. Їхня помилковість та дисперсія показують низький Bias та помірну Variance, вказуючи на бажаний баланс. Вибір методу залежить від конкретних вимог задачі: Multiple Linear Regression для простих сценаріїв, Gradient Boosting та Random Forest – для складних взаємозв'язків.

Загальною тенденцією є те, що для точних та складних прогнозів, особливо в умовах невизначеності, Gradient Boosting та Random Forest можуть бути більш ефективними методами порівняно з Multiple Linear Regression. Однак, враховуючи інтерпретованість та обмежену обчислювальну складність, Multiple Linear Regression може залишатися важливим інструментом у деяких сценаріях.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Christopher Pal, Mark Hall, Eibe Frank, Ian Witten. Data Mining: Practical Machine Learning Tools and Techniques, 4rd ed. / Morgan Kaufmann, 2016.
2. Jason Bell. Machine Learning: Hands-On for Developers and Technical Professionals / John Wiley & Sons, 2014.
3. Дивак М. П. Методичний посібник з дисципліни «Системний аналіз» / М. П. Дивак. – Тернопіль: ТАНГ. – 2004. – 136 с.

РОЗВ'ЯЗАННЯ ЗАДАЧІ ЗАПОВНЕННЯ ПРОПУСКІВ ДАНИХ АЛЬТЕРНАТИВНИМИ МЕТОДАМИ ПРИ СТВОРЕННІ ПРОГНОЗНИХ МОДЕЛЕЙ

Попов А.Ю.¹, Макаренко О.С.², Бідюк П.І.³

Національний технічний університет України «Київський політехнічний інститут ім. Ігоря Сікорського»

¹ popovandrii1403@gmail.com, ² makalex51@gmail.com, ³ pbidyuke_00@ukr.net

Наявність якісних даних виступає ключовим елементом для успішного моделювання у машинному навчанні. Пропущені дані можуть мати різні механізми утворення і правильна їх обробка здатна істотно покращити результати навчання. Метою роботи є дослідження впливу різних методів заповнення пропущених даних на прогнозні моделі з метою покращення їх роботи. Результатом роботи є програмне забезпечення для проведення дослідження на прикладі задачі прогнозування відтоку клієнтів. У роботі використано теоретичні та емпіричні методи дослідження.

Ключові слова: пропущені дані, методи заповнення, механізми пропущених даних, прогнозні моделі, машинне навчання, аналіз даних.

1. ВСТУП

У контексті машинного навчання якісні дані є запорукою якісного моделювання. Пропущені дані є однією з проблем, що зустрічаються найчастіше у аналізі даних, оскільки вони можуть утворюватися з найрізноманітніших причин, зокрема через технічні помилки, людський фактор, природу дослідження, тощо. Пропущені дані можуть серйозно впливати на точність та надійність моделей, тому аналіз та ефективне заповнення пропущених даних є важливим етапом попередньої обробки інформації.

Використання відповідних методів дозволяє зберегти цінну інформацію та уникнути втрати обсягу вибірки. Якісний аналіз пропущених даних допомагає уникнути викривлення результатів та покращити надійність прогнозних моделей, що базуються на машинному навчанні. Відповідна обробка пропущених даних сприяє створенню моделей, які краще враховують реальний характер даних, що у свою чергу підвищує ефективність та точність вирішення завдань машинного навчання.

2. МЕТОДИ ЗАПОВНЕННЯ ПРОПУЩЕНИХ ДАНИХ

Ключовим для класифікації методів заповнення є поняття механізму утворення пропущених даних. Механізм – це термін, що покликаний в загальному вигляді описати зв'язки між пропусками та наявними даними. За найпоширенішою класифікацією існує три механізми утворення пропущених даних:

1. MAR (англ. Missing at random) – припущення, яке передбачає, що ймовірність пропущених даних за змінною Y залежить від значення по одній або декількох інших ознак, але не залежить від значення самої змінної Y .

2. MCAR (англ. Missing completely at random) – ймовірність пропущених даних за змінною Y не залежить від значення по жодній іншій ознаці та не залежить від значення Y .
3. MNAR (англ. Missing not at random) – ймовірність пропущених даних за змінною Y залежить в тому числі і від значення самої змінної Y .

Методи заповнення пропусків умовно поділяються на традиційні та складні. Традиційні методи загалом широко поширені, прості для використання та виконують найважливіше завдання методів заповнення – створюють повний набір даних, що придатний до подальшого використання. Проте традиційні методи працюють лише на MCAR механізмі, інакше вони призводять до спотворення оцінок параметрів. Окремо традиційні методи можна розділити на методи видалення та заповнення даних.

Методи видалення видаляють дані з набору коли в них наявні пропуски. До традиційних методів видалення даних відносяться такі методи як аналіз повних спостережень та аналіз наявних спостережень. Метод аналізу повних спостережень є найпростішим з способів обробки пропущених даних, оскільки він видаляє всі записи, що мають пропуск по хоча б одній змінній. Цей метод часто призводить до спотворень даних та втрати великої частини даних навіть за невеликого відсотку пропусків. Метод аналізу наявних спостережень намагається зменшити втрату даних, видаляючи випадки на основі кожного окремого аналізу. Він є більш потужним, особливо коли змінні в наборі даних мають низькі або помірні кореляції. Унікальною проблемою цього методу є те, що кожне обчислене статистичне значення може базуватися на різних підмножинах даних.

До традиційних методів заповнення відносяться заповнення середнім, заповнення регресією, стохастичне заповнення регресією, група методів заповнення hot-deck, заповнення на основі останнього спостереження та інші. Основна особливість цих методів полягає у тому, що ці методи виконують одноразове заповнення і надалі заповнене значення розглядається як фактичне під час аналізу. Це часто призводить до недооцінювання помилки вибірки.

Заповнення середнім значенням використовує середнє значення наявних даних для заповнення усіх пропусків конкретної ознаки. Цей метод зменшує варіативність даних, що, в свою чергу, зменшує стандартне відхилення та дисперсію, а також амплітуди коваріацій та кореляцій. Метод заповнення регресією будує регресійне рівняння для кожної комбінації пропусків і заповнює відсутні значення передбаченими. Регресійне заповнення працює краще за заповнення середнім, але фактично стикається з прямо протилежною проблемою, оскільки воно заповнює дані ідеально корельованими показниками. Стохастичне регресійне заповнення – це покращений варіант регресійного заповнення, що має додатковий крок у вигляді доповнення кожного згенерованого значення нормально розподіленим залишком. Додавання залишків до заповнених значень відновлює варіативність даних, завдяки чому стохастична регресія не має тих проблем, які має звичайне регресійне заповнення. Крім того, стохастична регресія це один з небагатьох методів одиночного заповнення, що здатний створювати незміщені заповнення при MAR механізмі, завдяки чому він є одним з найбільш надійних з цієї категорії методів. Hot-deck заповнення – це група методів заповнення, основна ідея яких полягає у використанні в якості заповнення значень, що походять з схожих спостережень. Найбільш типовий варіант цього методу полягає у тому щоб для кожного пропуску взяти випадкове значення з підмножини повних спостережень, що мають схожі значення за певним набором змінних. Ці методи не погіршують варіативність даних до такого ж рівня, як інші методи одиночного заповнення, але вони можуть створювати зміщення у оцінках кореляцій та коефіцієнтів регресії.

На відміну від простих методів заповнення, складні методи надають можливість ефективно вирішувати проблему відсутності даних завдяки здатності заповнювати пропущені

дані без спотворення. Вони потребують MAR припущення для коректної роботи. До складних методів відносяться метод множинного заповнення, метод максимальної правдоподібності та інші.

Метод максимальної правдоподібності – це метод оцінювання значення змінної шляхом максимізації функції правдоподібності. Метою методу максимальної правдоподібності є визначення параметрів повного набору даних, які мають найвищу ймовірність створення наявних вибіркового даних. Вибіркова логарифмічна функція правдоподібності відіграє центральну роль у цьому процесі, оскільки вона кількісно визначає відносну ймовірність вибору множини значень з нормального розподілу з певним вектором середніх та матрицею коваріації. Сама оцінка значень змінних – це ітеративний процес, що перевіряє різні значення параметрів до тих пір, поки не знайде оцінки, які найбільш імовірно могли б створити такі дані. Варто відмітити, що метод максимальної правдоподібності не вирішує задачу пропущених даних у стандартному сенсі, оскільки він не створює безпосередні заповнення і не продукує повний набір даних як результат. Через це він не підходить до задач, де отримати повний набір даних є більш важливим результатом, ніж отримання точних оцінок даних.

Аналіз за допомогою методу множинного заповнення складається з трьох окремих етапів: фази заповнення, фази аналізу та фази об'єднання. Перша фаза заповнення – це двоетапна процедура, яка складається з кроку заповнення I та апостеріорного кроку P. На кроці I використовується алгоритм ідентичний до стохастичної регресії для побудови набору рівнянь регресії, що передбачають значення пропущених даних. Після цього на кроці P заповнені дані використовуються для отримання оцінок вектора середніх та коваріаційної матриці. Кінцевою метою етапу заповнення є створення m повних наборів даних, кожен з яких містить унікальні оцінки відсутніх значень. Створення унікальних заповнень потребує різних оцінок коефіцієнтів регресії на кожному I-кроці, а метою P-кроку є створення альтернативних оцінок вектора середніх та коваріаційної матриці (необхідні компоненти для рівнянь регресії на I-кроках). Багаторазове повторення цієї двоетапної процедури створює набір копій даних, кожна з яких містить унікальні оцінки відсутніх значень. Метою фази аналізу є аналіз заповнених наборів даних. На цьому кроці застосовуються ті самі статистичні процедури, які були б використані, якби дані були повними, але застосовуються вони до кожного заповненого набору окремо. Мета фази об'єднання полягає в тому, щоб об'єднати все в єдиний набір результатів. Метод множинного заповнення вважається одним з найбільш досконалих варіантів для розв'язання проблеми пропущених даних, оскільки він здатний генерувати коректні незміщені результати заповнення як для MCAR, так і для MAR механізмів пропусків, а також створювати повний набір даних, що є важливим фактором при виборі методу заповнення для роботи з наборами даних для навчання прогнозних моделей. Також метод множинного заповнення явно враховує невизначеність, пов'язану з відсутніми даними. Завдяки багаторазовому заповненню даних множинне заповнення дає оцінки змінних, які є середніми для ряду правдоподібних значень заміни, тому процес ніколи не довіряє єдиному набору заповнень.

3. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

У якості задачі прогнозування була обрана задача визначення утримання та відтоку клієнтів банку та відповідний набір даних клієнтів. Оскільки основною метою дослідження є робота з методами заповнення у поєднанні з прогнозними моделями, стартовий набір даних обрано повним. Це дозволяє отримати результати прогнозування використовуючи повний набір даних, після чого штучним чином ввести пропуски різного роду, опрацювати їх та провести навчання моделей на заповнених наборах. Також повнота стартового набору дозволяє точно порівнювати статистичні параметри по кожній змінній між повним та

заповненими наборами даних. Було досліджено набір даних по кожній змінній з метою попереднього визначення взаємозв'язків та особливостей даних.

Після завершення етапу дослідження даних наступним кроком було штучне створення пропусків. Набір даних було розділено на навчальну та тестову вибірки у співвідношенні 80 до 20. Штучні пропуски створювалися лише у навчальній вибірці, оскільки тестова вибірка повинна бути однаковою та повною заради більшої точності під час проведення порівняльної роботи. У роботі було створено ряд наборів даних з різними типами пропусків та різною їх кількістю, зокрема 4 набори даних, що мали виключно пропуски типу MCAR, 2 набори даних, що мали виключно пропуски типу MAR, 2 набори даних, що мають змішані пропуски типу MCAR та MAR, та 3 набори даних, що мають змішані пропуски з використанням пропусків типу MNAR. Зазначені набори даних дозволили перевірити результати роботи методів заповнення різного рівня складності в різних умовах, що не обов'язково відповідали цільовому механізму пропусків методу.

Було обрано 4 методи заповнення для дослідження: аналіз повних спостережень, заповнення середнім та дві реалізації методу множинного заповнення – мовою Python бібліотеки scikit-learn та мовою R бібліотеки MICE. За допомогою цих методів отримано 4 варіанти заповнених наборів даних для кожного вказаного набору даних: загалом 48 наборів. У якості алгоритмів прогнозування обрано 4 алгоритми: логістична регресія (LR), метод опорних векторів (SVC), метод випадкового лісу (RF) та LGBM.

Окремо перед початком навчання моделей з кожним створеним набором даних було проведено масштабування та обробку незбалансованості (оскільки цільова змінна мала відношення даних 80 до 20, що створювало додаткові труднощі у навчанні).

Після завершення попередньої обробки даних було запущено навчання обраних моделей на повному наборі. Основною метрикою для визначення якості моделі було взято метрику recall – оскільки коректне визначення позитивного результату (клієнт відмовиться) є найбільш важливим за сенсом поставленої задачі. Значення recall 70% було взято за мінімальне задовільне.

Таблиця 1. Результати навчання на повному наборі даних

	Accuracy	Precision	Recall	AUC
LR	0,727502	0,727573	0,727344	0,799392
SVC	0,816945	0,820831	0,810887	0,895356
RF	0,810966	0,821958	0,793896	0,891082
LGBM	0,819541	0,828747	0,805538	0,900956

Таблиця 2. Результати роботи моделей на тестових даних

	Accuracy	Precision	Recall
LR	0,725000	0,386397	0,679389
SVC	0,799500	0,492982	0,715013
RF	0,808500	0,508711	0,743003
LGBM	0,820500	0,530357	0,755725

Після цього було проведено навчання моделей на всіх створених наборах даних. За всіма проведеними дослідженнями було виявлено, що аналіз повних спостережень, незважаючи на його проблематику з статистичної точки зору, здатний створювати набори даних, навчання на яких може давати результати не гірші, ніж коли використовуються складні методи заповнення. Однак ці випадки є швидше випадковими, ніж постійними, тому надійність методу є доволі сумнівною. Результати роботи методу на даних з MCAR типом пропусків були змішані, на

даних з MAR типом пропусків були низькі, а на даних з MNAR та змішаним типом пропусків знову були змішані. Через цю загальну ненадійність можна зробити висновок, що цей метод не є найкращим вибором для вирішення подібних задач.

Заповнення середнім загалом показало гарні результати на більшості наборів даних, хоча зазвичай потребувало складних класифікаторів RF або LGBM. Цей метод працював гірше з методом опорних векторів та з логістичною регресією. Найменш якісні результати роботи були отримані на даних із змішаним типом пропусків, що включав MNAR категорію. Загалом цей метод, незважаючи на свою простоту, показав себе з гарної сторони і може використовуватись, якщо необхідно швидко вирішити проблему пропущених даних у контексті цієї задачі.

Множинне заповнення у реалізації IterativeImputer від Python бібліотеки scikit-learn показало незадовільні результати – моделі, навчені на даних, що були заповнені цим методом, часто не відповідали мінімальній необхідній якості класифікації. Найкраще цей метод працював з логістичною регресією та з методом опорних векторів – зокрема, логістична регресія неодноразово показувала помітно кращі результати за метрикою recall при використанні цього заповнення у порівнянні з стартовими повними даними – однак програвала по інших метриках, через що загальна якість моделі була нижчою. Моделі RF та LGBM стабільно мали низьке значення recall у поєднанні з цим заповненням. В той же час варто відмітити, що метрика precision цих моделей у поєднанні з цим заповненням мала суттєво кращі значення, ніж при навчанні на повному наборі даних. Через це можна зробити висновок, що цей складний метод заповнення не відповідав меті обраної задачі, однак може показати себе набагато краще у випадках, коли цільовою метрикою є precision. Також можна зазначити, що він є доволі специфічним, оскільки ефект підвищення precision або зниження recall не працює з кожною моделлю класифікації однаково, що ускладнює передбачуваність роботи цього алгоритму заповнення.

Множинне заповнення у реалізації від R бібліотеки MICE показало себе з найкращої сторони, забезпечуючи стабільно високі результати за всіма метриками, що були найбільш наближені до показників після навчання на повному наборі даних. Метод працював добре на всіх наборах незалежно від типу, комбінації та кількості пропусків, навіть добре показав себе при роботі з типом пропусків MNAR, до роботи з яким він за теорією не є пристосованим. Найкраще цей метод працював у поєднанні з алгоритмами RF та LGBM. Алгоритм RF у деяких випадках з використанням цього методу заповнення навіть показував кращі результати, ніж після навчання на повному наборі даних. Загалом, цей метод є найбільш надійним вибором серед досліджуваних для обробки пропусків будь-якого роду, хоча його використання може бути пов'язане з деякими додатковими складнощами через його належність до мови R.

4. ВИСНОВКИ

На сьогоднішній день в реальному світі дані часто є неповними і неідеальними, а для отримання прогнозів, що матимуть високу точність, необхідним є використання якісних навчальних даних. Різні методи заповнення пропущених даних демонструють різний рівень ефективності в залежності від контексту та характеру даних. На це, зокрема, впливають такі фактори як характер утворення пропусків або їх кількість.

У роботі було розглянуто як найпростіші традиційні методи обробки пропущених даних, серед яких аналіз повних спостережень, заповнення середнім, медіаною, інтерполяційні методи – так і складні методи, що враховують контекст та взаємозв'язки між даними – наприклад, множинне заповнення. Загалом, складні методи демонструють кращі результати, оскільки ставлять за мету збереження статистичних параметрів даних, але можуть бути більш

витратними з точки зору обчислювальних ресурсів, часу та простоти використання у порівнянні з традиційними методами.

Було реалізовано спеціалізовану інформаційну систему для проведення дослідження впливу методів заповнення на різні прогностичні моделі – вона складається з етапів дослідження та обробки вхідних даних, модуля заповнення даних, створення та навчання прогностичних моделей та візуалізації результатів. В роботі були досліджені методи аналізу повних спостережень, заповнення середнім та дві реалізації методу множинного заповнення. У якості об'єкту досліджень було взято набір даних банківських клієнтів та задачу прогнозування відтоку клієнтів. Найкращі результати були досягнуті з використанням реалізації R MICE алгоритму множинного заповнення.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Craig K. Enders. Applied Missing Data Analysis [Текст]: The Guilford Press; 1-е видання, 2010 – 20 с.
2. Donald B. Rubin. Inference and Missing Data [Текст]: Biometrika Vol. 63, No. 3. 1976 – 581-592 с.
3. Літл Р. Дж. А. Статистичний аналіз даних з пропусками [Текст] / Р. Дж. А. Літл, Д. Б. Рубін. – М.: Фінанси та статистика, 1991. – 336 с.
4. Schlomer G. L. Best practices for missing data management in counseling psychology [Текст] / Gabriel L. Schlomer, Sheri Bauman, Noel A. Card // Journal of Counseling Psychology. – 2010. – Vol. 57, No. 1. – P. 1–10.
5. Soley-Bori M. Dealing with missing data: key assumptions and methods for applied analysis [Текст] / Marina Soley-Bori // Technical Report. – 2013. – No. 4. –P. 1–20.
6. A comparative study of imputation methods for estimation of missing values of per capita expenditure in central java [Текст] / Y. Susianto, K. A. Notodiputro, A. Kurnia, H. Wijayanto // IOP Conf. Series: Earth and Environmental Science. – 2017. – 58. – P. 1–10.
7. Therese D. Pigott. A review of methods for missing data [Текст] / Therese D. Pigott // Educational Research and Evaluation. – 2001. – Vol. 7, No. 4. – P. 353–383.
8. Allison P. D. Missing data [Текст] / Paul D. Allison. – Mode of access: <https://statisticalhorizons.com/wp-content/uploads/2012/01/Milsap-Allison.pdf>.

СИСТЕМА АНАЛІЗУ ВПЛИВУ КЛАСТЕРИЗАЦІЇ НА ЯКІСТЬ РІШЕНЬ В МОДЕЛЯХ ШТУЧНОГО ІНТЕЛЕКТУ

Симонов Є.Д.¹, Макаренко О.С.

Національний технічний університет України «Київський політехнічний інститут ім. Ігоря Сікорського», Київ, Україна

¹ e.symonov@gmail.com

У роботі проведено емпіричний аналіз основних методів кластеризації, спрямований на визначення їхнього впливу на процес прийняття рішень у системах штучного інтелекту. В процесі дослідження виконувалось моделювання трьох підходів для визначення рівня впливу, таких як зміна моделі кластеризації, порівняння результатів моделей нейронних мереж з та без використання кластеризації, а також зміна кількості сформованих кластерів. Основною метою дослідження було підтвердження гіпотези про наявність впливу цих факторів на якість прийнятих рішень в системах штучного інтелекту. Результати аналізу стали основою для створення системи оцінки та порівняння отриманих результатів, що дозволило докладніше зрозуміти вплив методів кластеризації на ефективність систем штучного інтелекту.

Ключові слова: штучний інтелект, кластеризація, системний підхід, якість моделі, змінна кількості кластерів, текстовий аналіз.

1. ВСТУП

Моделі штучного інтелекту (ШІ) є невід'ємною частиною цифрової трансформації в науці, бізнесі та промисловості. Основою таких моделей є алгоритми, спрямованих на імітацію розумових процесів людського мозку. Такі мережі здатні навчатись на великих обсягах даних різних форматів, робити висновки та приймати рішення на основі попереднього досвіду. Крім того, вони взаємодіють з різними методами машинного навчання, такими як алгоритми класифікації, кластеризації та обробки природної мови, для виконання різноманітних завдань у сферах розпізнавання образів, аналізу даних та автоматизації процесів.

Одним з значущих варіантів використання моделей штучного інтелекту є обробка текстових даних та запитів. Такий варіант використання дозволяє взаємодіяти з навченими моделями за допомогою звичного формату – природньої мови.

Навчання мереж такого типу потребує обробки значних обсягів даних, і важливим етапом у цьому процесі є кластеризація, яка допомагає отримати вектори ознак переданих даних, на основі яких будуються прогнози.

Під час аналізу вхідних даних виникає питання про формат обраної моделі, оскільки різні типи даних вимагають використання різних моделей. Тому необхідно провести попередній аналіз ефективності та вибрати оптимальну модель. Також важливою є кількість вихідних кластерів, оскільки вона визначає ступінь деталізації та схожість даних для подальшого пошуку ознак.

Вирішення обох питань можливе за допомогою застосування системного підходу, який дозволяє аналізувати обидва аспекти та обрати оптимальне рішення для конкретного набору даних.

2. МЕТОДИКА ОЦІНКИ ВПЛИВУ

Методика оцінки впливу кластеризації на якість рішень в моделях штучного інтелекту (МШІ) полягає в тому, щоб оцінити, як розподіл даних на кластери впливає на точність, справедливість та інші важливі характеристики МШІ. Для цього можна використати наступні методи [1]:

Використання різних методів кластеризації для навчання однієї і тієї ж МШІ та порівняння результатів. Цей підхід дозволить оцінити, який метод кластеризації найкраще підходить для конкретної задачі та конкретної МШІ.

Використання одного методу кластеризації з різною кількістю вихідних кластерів та порівняння результатів. Цей підхід дозволяє вивчити, як різні рівні деталізації ознак впливають на точність та надійність прийнятих рішень. Шляхом аналізу різноманітних рівнів деталізації, можна визначити оптимальні параметри для покращення якості роботи системи, забезпечуючи більш точні прогнози та ефективні алгоритми прийняття рішень.

Використання кластеризованих даних для навчання МШІ та порівняння результатів з моделями, навченими на некластеризованих даних [1]. Такий формат дослідження дозволяє визначити, чи може кластеризація поліпшити точність, справедливість та інші важливі характеристики МШІ.

Також, варто розуміти, що під якістю МШІ мається на увазі міра того, наскільки добре модель виконує свою задачу. Вона визначається багатьма факторами, включаючи точність, швидкість, надійність, обґрунтованість, інформативність, гнучкість та ефективність [2].

Одним із варіантів дослідження запропонованих підходів – є використання системного підходу з побудовою спеціалізованої архітектури, яка дозволить проаналізувати результати в різних задачах. Запропонована система має наступну структуру (Рисунок 1):

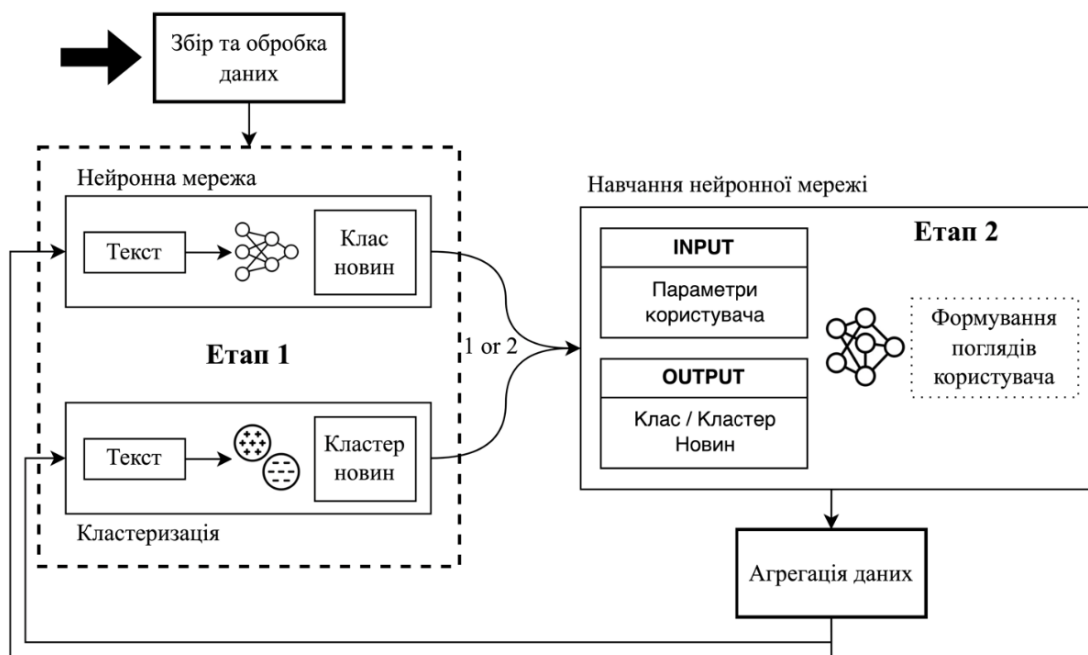


Рисунок 1. Структура системи дослідження

У структурі системи, відображеній на рисунку 1, наводиться підхід для оцінки якості МШІ з використанням методу кластеризації порівняно з аналогічними без нього. Перший етап у функціонуванні системи полягає у зборі, аналізі та обробці початкової інформації. Оскільки в даному дослідженні текст виступає як основний вид вхідних даних, слід зазначити, що без

попередньої обробки система виявлятиме значну кількість зайвих характеристик, таких як розділові знаки, слова у різних відмінках, числа тощо. Крім того, важливим етапом обробки текстових даних є їх перетворення у числові вектори, що виконується за допомогою спеціальної моделі TfidfVectorizer. Ця модель враховує як частоту використання конкретного слова у певному документі (TF – term frequency), так і інверсійну частоту використання слова у всьому наборі даних (IDF – inverse document frequency). Такий підхід допомагає виділити і враховувати важливі та рідкісні слова, що в свою чергу підвищує точність описуваного тексту та знижує вплив загальноновживаних слів.

Наступний етап дослідження полягає у порівнянні двох різних підходів. Перша варіація будує одну з кластеризаційних моделей (KMeans, Agglomerative Clustering, Gaussian Mixture чи Latent Dirichlet Allocation Model) та видає на виході вектори ознак кластерів. Другий підхід полягає у розробці нейронної мережі, яка використовує ті ж вхідні параметри, однак на виході надає точний номер об'єкту дослідження, відповідно до номерів у навчальній вибірці даних.

Результати роботи обох варіацій передаються як цільові значення в основну нейронну мережу, яка приймає параметри користувача на вхід та формує вектор ознак приналежності тематикам новин, найбільше відповідних конкретному користувачеві.

Побудувавши моделі на вхідних даних обох варіацій, ми маємо можливість порівняти результати на тестових вибірках, використовуючи метрики якості моделей. Цей етап у дослідженні дозволяє нам розуміти, яка модель найкраще відповідає задачі та потребам користувачів і визначити шуканий рівень впливу.

Така система дозволяє ретельно вивчати не лише вплив моделі кластеризації на результат, а й вплив кількості кластерів, так як цей параметр є одним із вхідних змінних системи. Отже, запропонована система дає можливість систематично оцінити всі раніше зазначені аспекти.

Описаний підхід до аналізу передбачає широкий спектр тестувань, де різноманітні комбінації вхідних параметрів дозволяють отримати комплексну оцінку впливу факторів на якість прийнятих рішень в моделях штучного інтелекту. Такий підхід є ключовим для розуміння взаємозв'язків між різними параметрами системи та їхнім впливом на кінцеві результати. Крім того, це дає можливість визначити оптимальні налаштування параметрів для покращення якості функціонування системи в різних умовах застосування.

3. ОПИС РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

Для дослідження було обрано набір даних "20newsgroups" [4] (датасет з бібліотеки sklearn.datasets), що включає інформацію про новини та їх тематику, і даних про користувачів соціальної мережі Twitter (нова назва "X"), які переглядали ці новини. Сполучення різних джерел даних створює складність та різноманітність вхідних даних. Вхідні дані представлені у форматі, що включає як текстові описи, так і числові та категоріальні дані.

Для дослідження впливу кластеризації на якість рішень в моделях штучного інтелекту, було задано початкову кількість кластерів рівною 20. Після виконання алгоритму, закладеного в систему, отримуємо такі результати (Рисунок 2):

Metrics	Basic	KMeans	Agglomerative	Gaussian	LDA
Accuracy Score	0.7294	0.8497	0.8425	0.8682	0.8316
Recall	0.7294	0.8497	0.8425	0.8682	0.8316
F1-Score	0.7818	0.8409	0.8418	0.8634	0.8294
Mean Absolute Error	0.9956	0.6517	0.6189	0.3811	0.3548
Mean Squared Error	4.3031	3.7551	3.1219	1.9918	0.9169
R-squared	0.4981	0.5596	0.5152	0.632	0.6429

Рисунок 2. Результати моделі ШІ з та без кластеризації

На рисунку 2 колонками виступають назви моделей, аналіз яких проводився системою. Назви рядків відповідають за метрики, розрахунок яких проводився.

У рамках дослідження було проведено аналіз декількох моделей, призначених для класифікації даних на основі параметрів користувача. Зокрема, порівнювалися модель базового типу (Basic), яка не використовує кластеризацію, з моделями, що включають в себе кластеризаційні підходи, такі як KMeans, Agglomerative Clustering, Gaussian Mixture та Latent Dirichlet Allocation Model [5].

Результати дослідження свідчать про високу ефективність поєднання нейронної мережі з моделю кластеризації Gaussian Mixture в контексті поставленої задачі:

Accuracy Score (Точність класифікації) [15]: Модель з використанням Gaussian Mixture має Accuracy Score на рівні 0,8682, що означає, що вона правильно класифікувала близько 86,82% прикладів. Це є найвищим результатом порівняно з іншими моделями і свідчить про її здатність точно передбачати класи на основі вхідних параметрів користувача

Recall та F1-Score [15]: побудована модель також володіє найвищими показниками Recall (0,8682) та F1-Score (0,8634), що вказує на високу здатність відшукати всі позитивні приклади та досягти балансу між точністю та повнотою класифікації.

Mean Absolute Error (Середня абсолютна похибка) [15]: побудована модель має один з найнижчих показників Mean Absolute Error (0,3811), що свідчить про мінімальну похибку у прогнозуванні.

Mean Squared Error (Середня квадратична похибка) [15]: Модель також має низький показник Mean Squared Error (1,9918), що підкреслює точність її прогнозів.

R-squared (R2) [15]: R-squared для даної моделі дорівнює 0.632, що є високим показником. Це свідчить про добру здатність моделі пояснювати зміну в залежній змінній та адекватність її прогнозів.

Для дослідження впливу кількості кластерів на якість рішень в моделях штучного інтелекту, повторно звернемось до створеної системи, але попередньо змінимо базові налаштування, а саме – вихідну кількість кластерів.

В рамках даного аналізу було виконано моделювання роботи системи зі зміною кластерів в діапазоні [3,50]. Дослідження більшої кількості кластерів є надлишковим так як основні тенденції впливу мали чітку ознаку до 50 кластерів, а зростання кількості кластерів збільшувало навантаження на обчислювальні ресурси. Найбільш показовими стали кластери 10, 15, 20 та 50. Цільовою метрикою дослідження було обрано – Accuracy Score.

Результати для 20 кластерів вже були проаналізовані та представлені раніше (рисунок 2). Точність для моделей складала (таблиця 1):

Таблиця 1. Точність нейронної мережі з моделями з 20 кластерами

N Clusters	KMeans	Agglomerative	Gaussian	LDA
20	0,8497	0,8425	0,8682	0,8316

Після експериментів з варіюванням кількості вихідних кластерів для обробки даних, ми отримали наступну статистику (таблиця 2):

Таблиця 2. Точність НМ з кластеризацією з різною кількістю вихідних кластерів

N Clusters	KMeans	Agglomerative	Gaussian	LDA
10	0,8436	0,8464	0,8507	0,8387
15	0,8409	0,8311	0,8486	0,8316
20	0,8497	0,8425	0,8682	0,8316
30	0,8612	0,8465	0,8721	0,8382
50	0,8518	0,8365	0,8557	0,8309

Як бачимо, в усіх проведених експериментах лідером виявилась нейронна мережа із застосуванням кластеризації за допомогою Gaussian Mixture. Візуалізуємо отримані результати (рисунок 3):

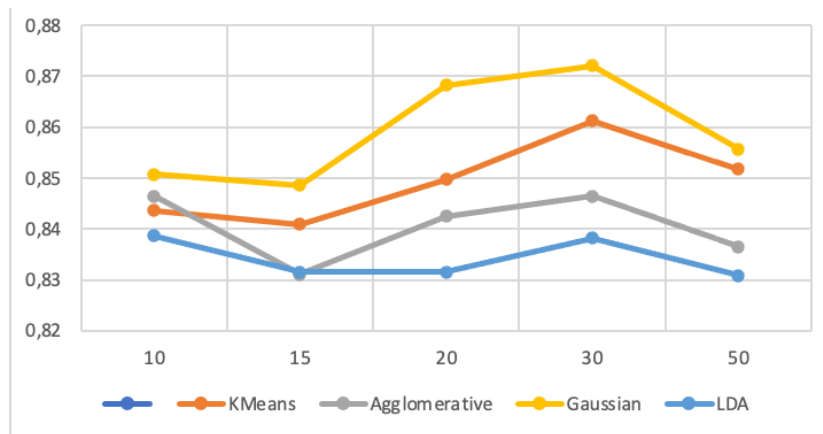


Рисунок 3. Візуалізація результатів експерименту

Спостерігаємо схожу динаміку для всіх побудованих моделей, що до 30 вихідних кластерів точність зростає і досягає свого максимуму, після чого спостерігається динаміка на спад.

Також, рисунок 3 дає можливість зрозуміти, що присутня шукана залежність між кількістю вихідних кластерів та точністю нейронної мережі.

4. ВИСНОВКИ

Розвиток нейронних мереж у сучасній науці та технологіях є важливим етапом в галузі штучного інтелекту. Ці мережі вимагають обробки величезного обсягу даних для ефективності та точності роботи. У цьому контексті методи кластеризації виявляються ключовими, сприяючи оптимізації навчання та аналізу нейронних мереж. В даному контексті дослідження впливу кластеризації на нейронні мережі визначає нові шляхи для досягнення оптимальної продуктивності та точності в роботі цих систем.

Створена система аналізує вплив методів кластеризації на ефективність рішень у моделях штучного інтелекту, розглядаючи різні підходи, типи кластеризації, різноманітність кількості формованих кластерів, а також порівнює нейронні мережі з та без застосування кластеризації.

У результаті цього дослідження було експериментально підтверджено, що кластеризація має вплив на якість прийнятих рішень у нейронних мережах (НМ). Модель без кластеризації в середньому показувала себе на 15% менше за вказаними показниками.

Окрім цього, підтверджено також вплив обраного методу кластеризації та кількості сформованих кластерів на якість роботи нейронних мереж. Серед розглянутих моделей найкраще себе показало поєднання нейронної мережі з моделлю кластеризації Gaussian Mixture

демонструє високу точність (Accuracy Score: 0,8682), recall та F1-Score (відповідно 0,8682 та 0,8634), низькі значення Mean Absolute Error (0,3811) і Mean Squared Error (1,9918) та високий показник R-squared (0,632). Також, серед розглянутого діапазону кількості кластерів, найбільш точною була модель з 30 кластерами.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. B. Li, J. Pei, J. Sun, and E. C. Chang, "A survey on clustering-based ensemble learning," ACM Computing Surveys (CSUR), vol. 50, no. 2, pp. 1-32, 2017.
2. Zhaohan Zhang, Mu Li, John Cavin, Katharine Flores, Rohan Mishra «A Fast and Robust Method for Predicting the Phase Stability of Refractory Complex Concentrated Alloys using Pairwise Mixing Enthalpy», 2022
3. Бідюк П.І. Проектування комп'ютерних інформаційних систем підтримки прийняття рішень: навч. посіб. / Бідюк П.І., Коршевнік Л.О.– К.: ННК "ІПСА" НТУУ "КПІ", 2010. – 340 с.
4. Відкритий датасет 20 NewsGroups [Електронний ресурс]: Режим доступу до ресурсу: https://scikit-learn.org/0.19/datasets/twenty_newsgroups.html
5. Моделі кластеризації [Електронний ресурс]: Режим доступу до ресурсу: <https://scikit-learn.org/stable/modules/classes.html#module-sklearn.cluster>
6. Nikhil Buduma. TensorFlow for deep learning—implementing neural networks, Released August 2016, Publisher(s): O'Reilly Media, Inc., ISBN: 9781491966198
7. Antonio Gulli, Sujit Pal, Deep Learning with Keras, Released April 2017, Publisher(s): Packt Publishing, ISBN: 9781787128422
8. Акишина А.А. Структура цілого текста / А.А. Акишина. – М. : Прогресс, 1979. – С. 24.
9. Вавіленкова А.І. Методи та алгоритми автоматизованого формування логіко-лінгвістичних моделей текстової інформації: дисертація на здобуття наукового ступеня кандидата технічних наук: 05.13.06 / НАН України, Інститут проблем математичних машин і систем. – Київ, 2011.– 155 с.
10. Stemming & lemmatization [Електронний ресурс] – Режим доступу до ресурсу: <https://nlp.stanford.edu/IR-book/html/htmledition/steming-andlemmatization-1.html>
11. Панкратова, П.І. Бідюк, М.Г. Рубець // Системні дослідження та інформаційні технології. — 2015. — № 3. — С. 7-19. — Бібліогр.: 13 назв. — укр.
12. К.Д. Паршак, ТЕКСТ ЯК ОБ'ЄКТ ЛІНГВІСТИЧНОГО ДОСЛІДЖЕННЯ, УДК 81.01:930.272
13. Ю.В. Рогущина, ЗАСОБИ ТА МЕТОДИ АНАЛІЗУ НЕСТРУКТУРОВАНИХ ДАНИХ, УДК 004.853, 004.55, <https://doi.org/10.15407/pp2019.01.057>
14. Негрей М.В., Гнот Т.В., Алгоритми Data Science у моделюванні бізнес-процесів, УДК 339.138: 004.89
15. How to use Metrics for Deep Learning [Електронний ресурс] – Режим доступу до ресурсу: <https://machinelearningmastery.com/custom-metrics-deep-learning-keras-python/>

СИСТЕМНИЙ ПІДХІД ДО АНАЛІЗУ КРЕДИТНИХ РИЗИКІВ В БАНКІВСЬКОМУ СЕКТОРІ

Сумін О.О.¹, Шубенкова І.А.

Національний технічний університет України «Київський політехнічний інститут
ім. Ігоря Сікорського»

¹ oleksandr.sumin@gmail.com

Кредитні ризиками є невід'ємною частиною діяльності фінансових установ. Оскільки ці ризики виникають внаслідок невизначеності, можна застосовувати системний підхід для ефективного управління ними. Впровадження системи оцінки та прогнозування ризиків може значно полегшити роботу аналітиків, сприяючи більш ефективному управлінню фінансовими ризиками в цілому. Результатом дослідження є розробка програмного продукту, що використовує методи машинного навчання та аналізу виживання для ефективної оцінки кредитного портфелю.

Ключові слова: системний аналіз, кредитні ризики, машинне навчання, моделі виживання.

1. ВСТУП

Дослідження процесу кредитування є надзвичайно важливим, оскільки цей фінансовий інструмент виступає ключовим фактором у розвитку економіки та суттєво впливає на фінансову стабільність підприємств і держави. З урахуванням постійного попиту на кредитування, прогнозування кредитних ризиків стає актуальною задачею для банків та інших кредитних установ.

Прогноз дефолту та кредитних збитків представляє собою ключовий складник кредитної політики банків та інших кредитних установ, оскільки він сприяє зниженню ризиків і підвищенню результативності процесу кредитування. Аналіз кредитних ризиків дозволяє банкам та іншим кредитним установам приймати обґрунтовані рішення щодо видання кредиту та встановлення відсоткових ставок в залежності від рівня ризику. Для цього при прогнозуванні дефолту та кредитних збитків використовуються різні методи, такі як статистичні моделі, машинне навчання та інші підходи.

2. ОЦІНЮВАННЯ КРЕДИТНИХ РИЗИКІВ

Збереження стабільності банківської системи сьогодні є найважливішим завданням для будь-якої світової економіки. З цією метою Міжнародним банком розрахунків (BIS) був створений набір правил, відомий як Базель III. Цей набір правил призначений для покращення оцінки та прогнозування ризиків банківськими установами, а також для зменшення можливих збитків під час кредитування. Базельські нормативи впливають значно на функціонування систем управління ризиками в Україні. Відповідно до методичних рекомендацій Базель III, Національний банк України, що виступає національним регулятором банківського сектору в Україні, ввів Постанову №351 "Про затвердження Положення про визначення банками України розміру кредитного ризику за активними банківськими операціями" (зі змінами).

Розрахунок значення кредитного ризику індивідуального позичальника, який визначається відповідно до Постанови №351, проводиться за визначеною формулою:

$$CR = PD * LGD * EAD,$$

PD – ймовірність дефолту;

LGD – збитки в разі дефолту;

EAD – експозиція під ризиком дефолту.

Зазначимо, що формула для розрахунку кредитного ризику, яка визначена в Постанові №351 від Національного банку України, відповідає консервативному підходу до оцінки кредитних ризиків. Цей підхід передбачає врахування можливості впливу кризових подій на ризик позичальника, і, таким чином, сприяє більш точному та обережному управлінню кредитними ризиками банків.

Згідно досліджень, кредитний ризик є одним з основних ризиків, пов'язаних з кредитуванням. Кредитний ризик може виникнути, якщо позичальник не зможе повернути позику вчасно або не зможе повернути її взагалі. Для зменшення кредитного ризику банки використовують такі методи, як аналіз кредитної історії позичальника, встановлення лімітів на кредитування, певних вимог до застави та інші. Також застосування методів аналізу великих даних може допомогти виявити потенційні ризики та запобігти ризикам надмірного кредитування. Дефолт є найбільш яскравим та серйозним ризиком як для кредиторів, так і для держави, оскільки масштаб несприятливих наслідків для обох сторін кредитних відносин є найбільшим у разі його виникнення. Однак, він може бути зменшений за допомогою правильного планування та управління ризиками, таких як аналіз кредитної історії позичальника, встановлення лімітів на кредитування, вимоги до застави та інше. Кредитний ризик тісно пов'язаний з кредитними збитками, за які можна прийняти різницю між усіма планованими грошовими потоками, які повинні були б отримати суб'єкт господарювання відповідно до договору, та всіма грошовими потоками, які суб'єкт господарювання очікує одержати (іншими словами, це сума всіх несплачених грошових обов'язків), знижених за допомогою дисконтування за ефективною початковою процентною ставкою.

Для початкового аналізу обрано логістичну регресію. Цей алгоритм добре підходить для бінарної класифікації та оцінки ймовірності події. Логістична регресія є ефективним інструментом для початкового аналізу даних та визначення впливу різних факторів на цільову змінну. Для розширення аналізу та підвищення точності моделей вирішено використати найвннй баєсівський класифікатор, що використовує теорему Баєса та припускає незалежність змінних. Він добре підходить для класифікації та прогнозування, особливо при роботі з великим обсягом даних. Дерева прийняття рішень є іншим ефективним інструментом для аналізу даних, спрямованим на побудову моделі для прогнозування цільової змінної на основі багатьох вхідних змінних. Цей метод дозволяє враховувати складні взаємозв'язки між змінними та поліпшує точність прогнозування. Для подальшого покращення ефективності моделей було вирішено використати модель беггінгу над випадковими деревами – RandomForest. Цей алгоритм використовує критерій Джині для мінімізації функції втрат, що допомагає побудувати точну модель та покращити якість розділення вузлів дерева. Для оцінки використовується модель бустингу – XGBoost. Цей алгоритм градієнтного бустингу на деревах відзначається високою точністю та здатністю поетапно покращувати модель, мінімізуючи середньоквадратичну помилку. XGBoost є потужним інструментом для роботи з великими обсягами даних та підвищення точності прогнозів.

У ході дослідження проведено тестування зазначених алгоритмів на вихідних даних для аналізу їхньої ефективності. Результати роботи алгоритмів представлено далі.

3. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Вихідними даними для аналізу було використано дані однорангової кредитної компанії Lending Club. Lending Club – це фінансова компанія, розташована у США, яка працює на

принципі однорангового позикодавця. Вони здійснюють посередництво між особами, які мають бажання інвестувати свої кошти, і тими, хто шукає кредит.

Так як класи в вибірці є незбалансованими, для підвищення якості класифікації впроваджено балансування класів цільової змінної за допомогою методу SMOTE. За основний критерій оцінки ефективності моделі був F1 для позитивного класу. Також для кожної моделі побудовано ROC-криву та основні метрики якості класифікації.

Таблиця 1. Результати прогнозування дефолту

Модель	Training Accuracy	Validation Accuracy
Logistic Regression	0,66	0,65
Gaussian Naive Bayes	0,64	0,68
Decision Tree	0,98	0,69
Random Forest	0,97	0,75
XGBoost	0,88	0,80

Найкращим варіантом виявилася модель бустингу, оскільки високі результати Decision Tree та Random Forest на тренувальній вибірці було отримано за рахунок перенавчання цих моделей, а зменшення кількості тренувальних записів не призводить до покращення результатів на валідаційній вибірці.

Для проведення оцінки очікуваного періоду настання дефолту використано оцінку Каплана-Майєра, котра є важливим інструментом в аналізі виживання та оцінці ризику. Спочатку проведемо аналіз та оцінку часу дожиття кредиту середньостатистичного позичальника. Це дозволить зрозуміти, як довго в середньому позичальники виконують умови перед настанням дефолту. Для більшої точності та розгалуженості оцінок, також проведемо дослідження в межах груп обумовлених оцінок кредитного рейтингу. Це допоможе з'ясувати, чи існують різниці у часі дожиття кредиту між різними групами позичальників з різними кредитними рейтингами.

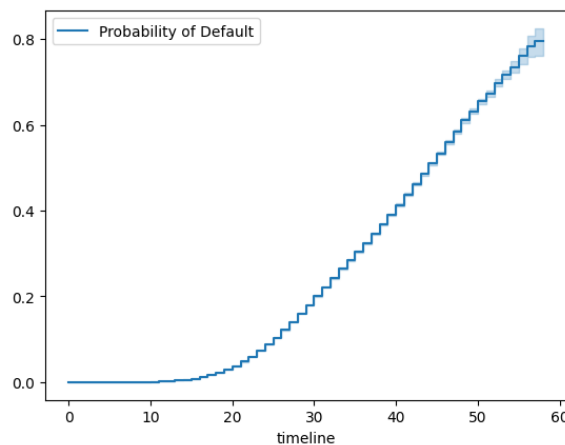


Рисунок 1. Ймовірність дефолту за період часу

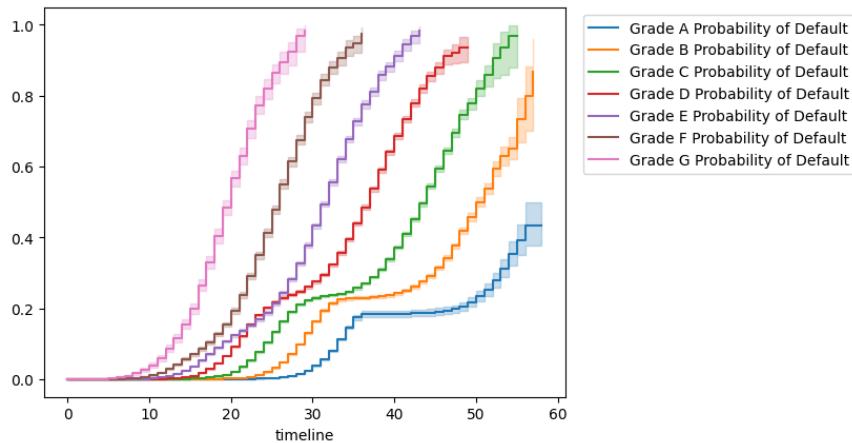


Рисунок 2. Ймовірність дефолту за період часу відповідно кредитного рейтингу

Для оцінки економічного ризику кредитного портфелю перейдемо до зведеної моделі розрахунку. У загальному випадку, функція втрат може бути визначена за такою формулою:

$$\mathcal{L} = S * \left(1 + InterestRate \frac{T}{12}\right) * \left(1 - \frac{E(\tau)}{T}\right),$$

$$E\mathcal{L} = \sum \mathcal{L}P\{default\},$$

- S – розмір виданої позики;
 $InterestRate$ – відсоткова ставка;
 $P\{default\}$ – ймовірність настання дефолту;
 $E(\tau)$ – очікуваний час настання дефолту (в місяцях);
 T – період кредитування (в місяцях).

$$E\{expected\ loss\} = E\mathcal{L} = \sum \mathcal{L}P\{default\},$$

$$E\{unexpected\ loss\} = 3\sqrt{E(E\mathcal{L}^2) + E^2(E\mathcal{L})}.$$

На прикладі деякої вибірки розрахуємо очікувані та неочікувані витрати, і порівняємо з реальними даними. Результати наведено в таблицях 2–3.

Таблиця 2. Тестування на вибірці розміром N=25000

Portfolio size	\$352 665 400,00
Expected loss	\$58 667 169,11
Unexpected loss	\$117 334 338,22
Real loss	\$51 916 256,83

Таблиця 3. Тестування на вибірці розміром N=50000

Portfolio size	\$704 084 850,00
Expected loss	\$117 426 534,61
Unexpected loss	\$234 853 069,21
Real loss	\$103 634 970,60

4. ВИСНОВКИ

У сучасному світі фінансові системи стають все складнішими та менш передбачуваними, що породжує потребу у розробці більш ефективних інструментів для аналізу та прогнозування

фінансових ризиків з метою якісного управління ними. Серед найбільш ефективних підходів до оцінювання ризиків на сьогодні вирізняється системний підхід, який можна вважати методологією досліджень, що полягає у розгляді об'єкта як цілісної системи елементів, взаємозв'язків та відношень між ними.

У ході дослідження було розроблено систему, яка забезпечує можливість прогнозування та оцінки ризику для певного кредитного портфеля. Ця система використовує різні методи, такі як машинне навчання та моделі аналізу виживання. Результати дослідження показали, що найбільш ефективним підходом для прогнозування ймовірності дефолту за кредитною лінією є використання градієнтного бустингу XGBoost. За допомогою цієї моделі машинного навчання та моделі Каплана-Майєра розроблено формулу для розрахунку оцінки очікуваного та неочікуваного ризику кредитного портфелю.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Про затвердження Положення про визначення банками України розміру кредитного ризику за активними банківськими операціями: Постанова правління Національного банку України №351 від 30.06.2016р. (зі змінами) URL: https://bank.gov.ua/ua/legislation/Resolution_30062016_351
2. Впровадження нових вимог до капіталу відповідно до норм Базеля та законодавства ЄС [Електронний ресурс]: Режим доступу до ресурсу: https://bank.gov.ua/admin_uploads/article/%D0%A7%D0%B0%D1%83%D1%81_pr_2017-07-19.pdf?v=4
3. Кузнєцова, Н. В. Порівняльний аналіз характеристик моделей оцінювання ризиків кредитування / Н. В. Кузнєцова, П. І. Бідюк // Наукові вісті НТУУ «КПІ» : науково-технічний журнал. – 2010. – № 1(69). – С. 42–53. – Бібліогр.: 10 назв
4. Бідюк, П. І. Система підтримки прийняття рішень для аналізу фінансових даних / П. І. Бідюк, Н. В. Кузнєцова, О. М. Терентьев // Наукові вісті НТУУ «КПІ» : міжнародний науково-технічний журнал, 2011. – №1. – С. 48-61. – Бібліогр.: 9 назв.
5. Бідюк П. І. Аналіз кредитоспроможності за допомогою методів інтелектуального аналізу даних / П. І. Бідюк, В. Г. Гуськова // Електронне моделювання. - 2019. - Т. 41, № 2. - С. 111-120.
6. Cao, Ricardo & Vilar, Juan & Devia Rivera, Andres E.. (2009). Modelling consumer credit risk via survival analysis. Sort: Statistics and Operations Research Transactions, ISSN 1696-2281, Vol. 33, № 1, 2009, pags. 3-30. 33.
7. Dirick, Lore; Claeskens, Gerda; Baesens, Bart (2017). Time to default in credit scoring using survival analysis: a benchmark study. Journal of the Operational Research Society, 68(6), 652–665. doi:10.1057/s41274-016-0128-9
8. Shen, Feng & Wang, Run & Shen, Yu. (2019). A cost-sensitive logistic regression credit scoring model based on multi-objective optimization approach. Technological and Economic Development of Economy. 26. 1-25. 10.3846/tede.2019.11337.

ІНТЕЛЕКТУАЛЬНІ ЗАСОБИ ПІДТРИМКИ АВТОМАТИЗАЦІЇ УПРАВЛІННЯ БІЗНЕС ПРОЦЕСАМИ

Тагільцев Д.І.¹, Мухін В.Є.²

Національний технічний університет України «Київський політехнічний інститут
імені Ігоря Сікорського», Київ, Україна

¹ dima.t16042001@gmail.com, ² v_mukhin@i.ua

Робота присвячена дослідженням у сфері підтримки автоматизації управління бізнес процесами. Розглянуті механізми та методи автоматизації бізнес процесів. Результатом роботи є моделі машинного навчання та нейронних мереж, які на основі датасету прогнозуються заробітну плату працівника.

Ключові слова: бізнес процеси, інтелектуальний аналіз даних, машинне навчання, нейронні мережі, регресія, випадковий ліс.

1. ВСТУП

В сучасному світі автоматизація бізнес-процесів стала невід'ємною частиною ефективного функціонування підприємств у різних галузях і сферах діяльності. Ця тема є дуже актуальною, оскільки сучасна економіка і технологічний прогрес швидко розвиваються, і автоматизація стає важливим інструментом для підтримки конкурентоспроможності та підвищення продуктивності бізнесу.

Спочатку важливо зрозуміти, що розуміється під автоматизацією бізнес-процесів. Це використання технологій та програмного забезпечення для автоматичного виконання завдань та операцій, які раніше виконувалися вручну. Автоматизація може охоплювати всі аспекти бізнесу, від управління запасами до обробки замовлень та взаємодії з клієнтами.

Однією з ключових переваг автоматизації є підвищення продуктивності. Системи автоматизації дозволяють виконувати рутинні завдання швидше і без помилок, що зменшує витрати часу та ресурсів. Наприклад, автоматизована система обробки замовлень може значно прискорити весь процес, починаючи від прийому замовлення до його відправлення клієнту. Це також допомагає знизити витрати на оплату праці, оскільки менше співробітників потрібно для виконання тих самих завдань.

Крім того, автоматизація підвищує якість виконання завдань. Вона допомагає уникнути помилок, які можуть виникнути через людський фактор, і забезпечує послідовність та стандартизацію процесів.

Для підприємств актуальність автоматизації також полягає в здатності підлаштовуватися під швидкий розвиток технологій та зміни вимог споживачів. Автоматизація дає змогу легко впроваджувати нові функції та процеси, а також швидко адаптувати бізнес-стратегію до зміни ринкових умов.

Звісно, автоматизація не є безперечною панацеєю і включає в себе свої виклики. Один із головних викликів полягає в витратах на впровадження автоматизованих систем. При цьому, на великих підприємствах ці витрати можуть бути виправдані завдяки підвищенню продуктивності та зменшенню витрат на довготермінові періоди. Однак для менших бізнесів і стартапів, внесення автоматизації може бути важким завданням.

2. МЕТОДИ ТА МОДЕЛІ ОРГАНІЗАЦІЇ, УПРАВЛІННЯ ТА АНАЛІЗУ БІЗНЕС ПРОЦЕСІВ

Інтелектуальний аналіз даних (ІАД) або Data mining представляє собою сучасний підхід до обробки та аналізу даних, який використовує технології штучного інтелекту (ШІ) та машинного навчання (МН) для виявлення шаблонів, прихованих залежностей і корисних знань у великих обсягах даних. Ця техніка дозволяє вдосконалювати процеси прийняття рішень, передбачати події, оптимізувати процеси та отримувати цінні інсайти з даних.

Data Mining є мультидисциплінарною галуззю, яка виникла і розвивається на базі досягнень прикладної статистики, розпізнавання образів, методів штучного інтелекту, теорії баз даних тощо (рис. 1). Звідси велика кількість методів і алгоритмів, реалізованих у різних діючих системах Data Mining. Багато з таких систем інтегрують у собі відразу кілька підходів. Проте, як правило, в кожній системі є якась ключова компонента.

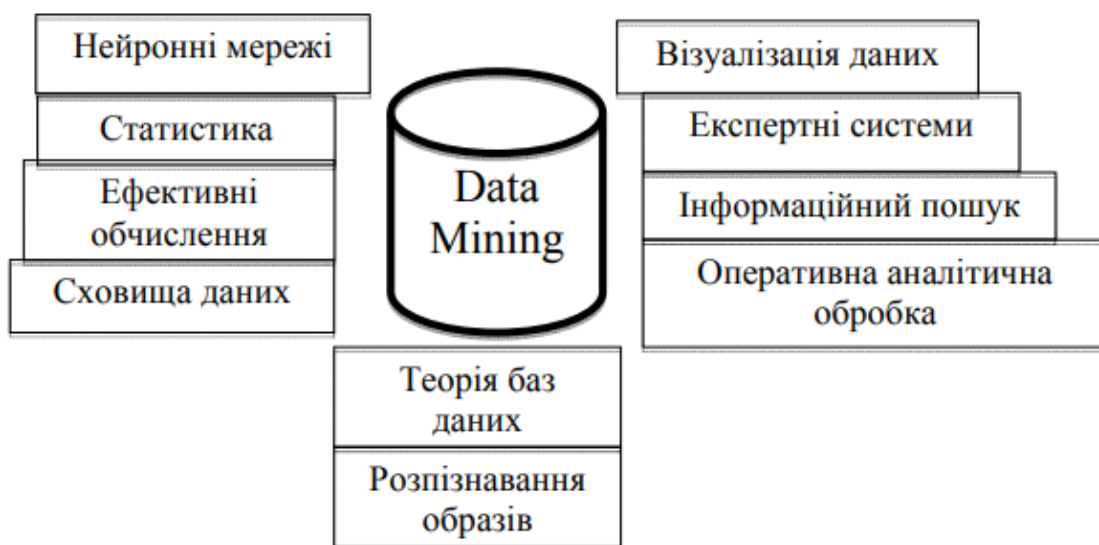


Рисунок 1. Datamining – мультидисциплінарна галузь

Етапами інтелектуального аналізу є:

1. Вивчення предметної області
2. Збір даних
3. Попередня обробка даних
4. Аналіз даних з метою виявлення патернів
5. Інтерпретація знайдених патернів (візуалізація, відбір корисних патернів відповідно до функції корисності)
6. Використання нових знань

Основними типами задач аналізу даних є задачі опису та задачі прогнозування, тобто:

- Класифікація – процес знаходження моделей чи функцій, які описують та розрізняють класи для прогнозування класу довільно заданого об'єкта з відомими атрибутами на основі навчаючої вибірки;
- Кластеризація – виявлення ознак, за якими можна буде здійснювати класифікацію, шляхом групування "схожих" між собою об'єктів, генерування міток класів на основі відстаней між об'єктами;
- Регресія – встановлення залежностей неперервних результуючих змінних від вихідних;

- Асоціація – пошук закономірностей між декількома подіями, що відбуваються одночасно;
- Прогнозування – оцінювання пропущених або майбутніх значень цільових чисельних показників;

Основними методами аналізу даних є:

- Множинний регресійний аналіз;
- Дерева рішень;
- Дискримінантний аналіз;
- Нейронні мережі;
- Аналіз часових рядів;
- Генетичні алгоритми;

Штучні нейронні мережі (НМ) – математичні моделі, а також їх програмні або апаратні реалізації, побудовані за принципами подання й обробки інформації у біологічних нейронних мережах – мережах нервових кліток живого організму.

Нейрон – це примітивний обчислювальний пристрій (або його модель), що має кілька входів і один вихід, і є основним обчислювальним елементом НМ (рис. 2).

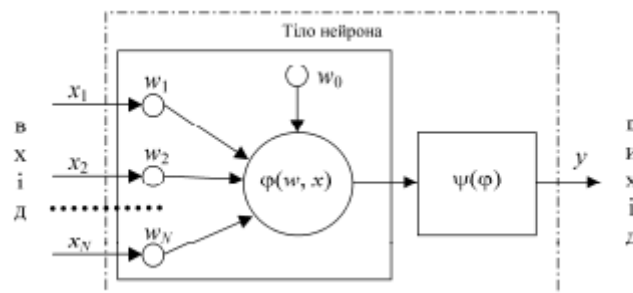


Рисунок 2. Схема штучного нейрона

На вхід одношарового нейрона надходить вхідний вектор – набір вхідних сигналів $x=\{x_j\}$, $j = 1, 2, \dots, N$, де N – кількість входів. Кожний вхідний сигнал x_j зважується (масштабується певним чином) відносно зіставленої йому ваги зв'язку (вагового коефіцієнта) w_j , яка моделює перетворення сигналу у синапсі (міжнейронному контакті).

Лінійна регресія - це статистичний метод, який використовується для моделювання та аналізу взаємозв'язку між залежною (відгук) та однією або декількома незалежними (предикторами) змінними. Технічно, лінійна регресія полягає в побудові математичної моделі, яка апроксимує лінійний зв'язок між цими змінними.

Лінійна регресія застосовується в широкому спектрі галузей і використовується для вирішення різних завдань, від прогнозування цін на нерухомість до дослідження впливу лікування на пацієнтів у медицині. Вона залишається потужним інструментом для аналізу даних і прийняття обґрунтованих рішень в сучасному світі.

Випадковий ліс (Random Forest) – це потужний ансамбльний алгоритм машинного навчання, який використовується для класифікації та регресії. Він базується на ідеї поєднання багатьох дерев рішень, щоб знизити перенавчання (overfitting) та покращити точність прогнозів. Випадковий ліс був запропонований Лео Брейманом і Адель Катлер в 2001 році і відтоді став дуже популярним в машинному навчанні.

Розглянемо основні характеристики та процес роботи випадкового лісу більш детально.

Випадковий ліс використовує не одне дерево рішень, а багато дерев. Кожне дерево побудоване на основі підвибірки (з повторенням) даних набору навчання та підвибірки атрибутів. Це робить кожне дерево унікальним (рис. 3).

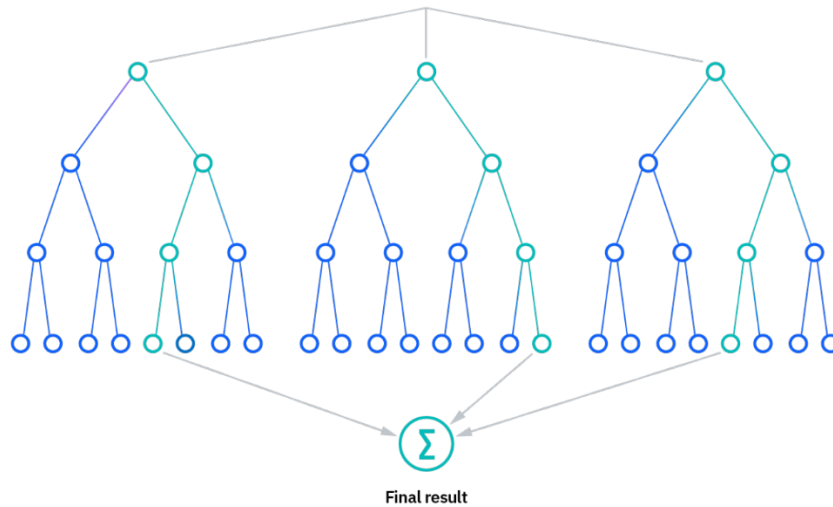


Рисунок 3. Схема випадкового лісу

3. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Для навчання моделей обрано датасет Employee Productivity and Satisfaction HR Data з ресурсу kaggle.com. Датасет представляє собою файл формату csv.

Датасет має 11 полів, кожне поле – певна відомість про людину (рис. 4).

Name	Age	Gender	Projects Completed	Productivity (%)	Satisfaction Rate (%)	Feedback Score	Department	Position	Joining Date	Salary
------	-----	--------	--------------------	------------------	-----------------------	----------------	------------	----------	--------------	--------

Рисунок 4. Поля датасету

Поле Salary – числове, але має дуже великі значення і відхилення. Застосуємо для нього механізм StandardScaler

Після отримання модифікованих даних розівемо їх на тренувальну та тестову вибірку у співвідношенні 8:2. Для цього виокремимо поле Salary та покладемо його в окрему змінну. Після чого застосуємо метод train_test_split та отримаємо X_train(тренувальні входні дані), X_test(тестові входні дані), y_train(тренувальний вихідний вектор), y_test(тестовий вихідний вектор).

Створимо 3 моделі для прогнозування рівня заробітної плати: модель багатошарової нейронної мережі, модель лінійної регресії та модель випадкового лісу. Оцінювати роботу моделей будемо за допомогою метрики середньоквадратичної помилки (MSE).

Після чого модифікуємо модель багатошарової нейронної мережі. Змінимо функції активації та додаємо дропаут.

На рис. 5 наведено порівняння результатів.

4. ВИСНОВКИ

В даній роботі були розглянуті інтелектуальні засоби для автоматизації управління бізнес процесами. В якості прикладу реалізації автоматизації було вибрано бізнес процес – прогнозування рівня заробітної плати працівника. Була освітлена тем актуальності автоматизації бізнес процесів та проведено аналіз механізмів її реалізації.

Вирішення задачі було реалізовано за допомогою алгоритмів машинного навчання, таких як: багатошарова нейронна мережа, лінійна регресія, випадковий ліс. Дані для навчання

моделей взяті з ресурсу kaggle. Дані були попередньо оброблені – видалені зайві поля, дискредитовані не числові поля.

Була реалізована модифікація багатошарової нейронної мережі з метою отримання кращих результатів її роботи.

В підсумку отримали що після модифікації багатошарова нейронна мережа ефективніша ніж модель лінійної регресії. Але все одно, для цієї задачі, найефективніша модель випадкового лісу.

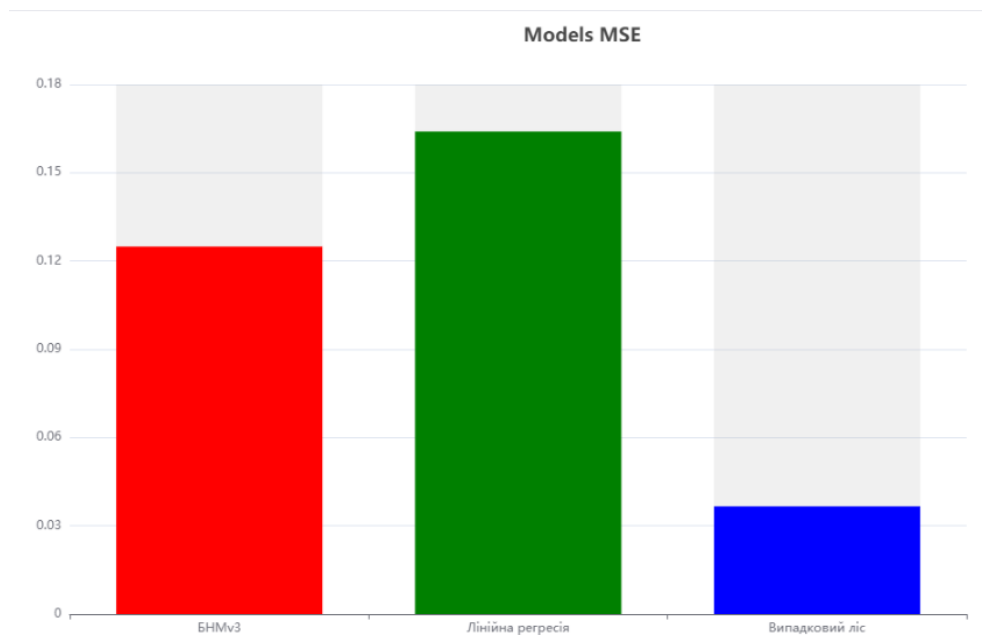


Рисунок 5. Порівняння MSE отриманих моделей

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Автоматизація управління бізнес-процесами підприємства як основа гнучкості його діяльності: характеристика систем управління. Веб сайт. URL:<https://doi.org/10.32782/2524-0072/2022-45-46> (дата звернення: 17.10.2023).

2. О.І. Черняк, П.В. Захарченко. Інтелектуальний аналіз даних. Київ, 2010, 841 с.

3. Інтелектуальний аналіз даних (Data Mining). Особливості технології Data Mining та її відмінності від інших методів аналізу даних. Веб сайт. URL: https://moodle.znu.edu.ua/pluginfile.php/486112/mod_resource/content/1/%D0%9B%D0%B5%D0%BA%D1%86%D1%96%D1%8F%201.pdf (дата звернення: 17.10.2023).

4. І.В. Лупан. Інтелектуальний аналіз даних Data Mining: навчально-методичний посібник. – Кропивницький, ФОП Піскова М. А., 2022. – 112 с.

5. Коефіцієнт кореляції Пірсона. Веб сайт. URL: https://uk.wikipedia.org/wiki/Коефіцієнт_кореляції_Пірсона (дата звернення: 17.10.2023)

6. Субботін С.О Нейронні мережі : теорія та практика: навч. посіб. / С. О. Субботін. – Житомир : Вид. О. О. Євенок, 2020. – 184 с.

7. Літнарівч Р.М. Побудова і дослідження математичної моделі за джерелами експериментальних даних методами регресійного аналізу. Навчальний посібник, МЕРУ, Рівне, 2011. –140 с.

8. Batch-normalization. Веб сайт. URL: <https://neerc.ifmo.ru/wiki/index.php?title=Batch-normalization> (дата звернення: 17.10.2023).

INTELLECTUAL DECISION SUPPORT SYSTEM FOR ESTIMATION OF FINANCIAL RISKS

Tymoshchuk O.L.¹, Levenchuk L.B.², Bidyuk P.I.³

National Technical University of Ukraine “Igor Sikorsky KPI”

¹ oxana.tim@gmail.com [0000-0003-1863-3095], ² lusi.levenchuk@gmail.com [0000-0002-8600-0890], ³ pbidyuke_00@ukr.net [0000-0002-7421-3565]

An intellectual decision support system is proposed the basic tasks of which are modeling and forecasting of financial processes and financial risk estimation. The system is developed on the basis of system analysis principles, i.e. it has hierarchic architecture, the possibility exists for identification and taking into consideration of some stochastic and information uncertainties, building alternatives for models and forecasts, and tracking the computing procedures during all stages of data processing. A modular architecture is implemented that provides a possibility for further enhancement and modification of the system functional possibilities with new forecasting and parameter estimation techniques, and financial risk models. The high quality of final result is achieved thanks to appropriate tracking of computing procedures at all stages of data processing: preliminary data processing, model constructing, and forecasts estimation. The system developed has good perspectives for the practical use. It is supposed that the system will find its applications as an extra tool for decision making when developing the strategies for financial risk estimation and management.

Keywords: financial risks, intellectual decision support system, mathematical modeling, operational risk estimation, system analysis approach.

1. INTRODUCTION

The problems of risk modeling and management are available practically in all areas of human activities. Especially urgent are the problems of mathematical modeling, estimation, and forecasting possible loss (the risks are characterized by the level of possible loss and its probability). These problems are facing banking institutions, insurance and investment companies, and production enterprises that are functioning in conditions of tough competition and unstable demand. To describe formally possible loss today there exists a set of ideologically different approaches based upon statistical methods and intellectual data analysis. To estimate market and some other types of risks it is possible to apply different versions of Value-at-Risk (VaR), methodology providing the possibility for reaching practically useful quality results [1, 2]. Also, wide applications have found the models of classification type such as logistic and linear regression, support vector machine, fuzzy logic, neuro-fuzzy models, Bayesian data analysis, theory of distributions, generalized linear models (GLM), and combinations of the methods mentioned [3–7].

Method selection for the formal describing of risk and risk estimation is mainly determined by availability of statistical data and instrumental platform for performing computations, staff qualification, and appropriate material support. Former experience of risk modeling and estimation illustrates that high quality of risk estimation results requires application of ideologically different approaches, comparison of the results achieved, and apply combining procedures of the results if

possible. Such approach to possible loss estimation can be implemented successfully using modern Intellectual Decision Support Systems (IDSS) based upon system analysis principles. IDSS belongs to the class of systems that can be quickly designed and implemented with software; it provides a possibility for implementing latest developments in the area of financial risk analysis with appropriate subsequent analytical support. IDSS design is based upon latest developments of system analysis ideas, information system theory, estimation theory, mathematical and statistical-and-probabilistic modeling and forecasting, intellectual data analysis, decision theory and other results of theory and practical data and expert estimates processing methods [8, 9].

The study considers the problem of constructing IDSS for solving the problems of estimation and forecasting financial risk loss using the possibility for application of alternative data analysis techniques, modeling and state and parameter estimation of the processes under study.

Problem statement. The purposes of the study are as follows: 1) performing analysis of the problems to be solved by IDSS, and the methods that could be used on the purpose; 2) to formulate requirements to the IDSS, and propose the system architecture; 3) to highlight the process of decision making with the system proposed; 4) to select some models for financial risk analysis; 5) to show an illustration of financial risk analysis using statistical data.

Requirements to the modern IDSS. Modern IDSS is complex computational system that has sophisticated functional possibilities with architecture of hierarchical type. The system should satisfy the following requirements: 1) to contain highly developed bases of data, knowledge, and necessary computational procedures; 2) the user interface should correspond the requirements of convenience, be simple and friendly, and adaptive to the user of various training level; 3) the hierarchical architecture of the IDSS and its functioning should correspond to the hierarchical way of thinking of decision making person (DMP) when making decision; 4) the system should be capable to accept new knowledge in the process of problem solving; 5) organization and the methods of computations, should correspond to the methods and requirements of DMP; 6) the quality of computations should correspond to the generally established norms and requirements; 7) the system should be able to provide user with all necessary forms of help and representation of intermediate and final results of computations; 8) IDSS should contain the means for data and knowledge exchange with other information processing systems using local and global nets. If the requirements mentioned are satisfied then the system efficacy is enhanced substantially, and its behavioral significance will be much higher for the company where it is used.

Alternatives generating with the IDSS. Generally, the decision making processes can be performed completely or partially in iteration cycles; IDSS may return retrospectively to the previous stages of data and knowledge analysis in automatic mode or on DMP initiative. The whole process of decision making and its implementation can be divided into stages shown below.

(1) – Analysis of the problem to be solved, collection of the necessary data and knowledge. At this stage all the information available is hired including possible former solutions in the past that may be useful for refining the problem statement, and remove the ineffective solutions.

(2) – Selection of the mathematical model class for formal description of the current problem statement, or some available tested models could be hired. The model can be selected in the class of linear/nonlinear models or in continuous/discrete time. In some cases simulation should be performed.

(3) – Constructing the model of the system (object) under study: estimation of the model structure and its parameters. It is always desirable to estimate several candidate models because alternative model structures provide the possibility for selecting the best one.

(4) – Quality (adequacy) analysis of the candidate models constructed and selection of the best one using statistical quality criteria. It is advisable to apply the candidates to solving the main

problem stated above, say, to forecasting selected processes. The forecasting results will show the actual possibilities of the model regarding actual problem solving.

(5) – Use the model selected for constructing forecasting function (when this is possible), and formulate decision making or control problem solving. Such results of model application are very useful for actual practical applications.

(6) – Alternatives generation using the model constructed and consequences estimation of its application. Here analytical quality criteria are to be applied for estimating possible results such as optimality criteria; usefulness functions; satisfaction of the requirements stated etc.

(7) – Generated alternatives analysis by the enterprise experts for whom the problem solving was performed. Now ultimate alternative selection is performed or all the alternatives are declined. Here it is possible to return to previous steps to get extra data and knowledge for generating additional alternatives.

(8) – Planning the actions for estimating financial, material and human resources necessary for implementing selected alternative. Determining the time interval (control horizon) necessary for implementing the decision selected.

(9) – Implementing the decision selected: monitoring the resources available and necessary for future steps of implementation; current monitoring of the situation with implementing the decision; registering the current, intermediate, and final results.

(10) – Application of possible quality criteria for estimating the results achieved: including expert and analytical estimates.

(11) – Final analysis of the results achieved by the company (client) experts; ultimate estimation of advantages and disadvantages of the alternative implemented; analysis of the current decision making and implementation process, and projections of possibilities for the future activities; generating report on the results achieved.

Mathematical methods to be used in IDSS. All mathematical methods to be used in IDSS are divided into two groups: 1) – the methods for general application that provide for functioning of the system; special methods that are directed to performing data and knowledge processing, alternative generating, selecting the best alternative, and forecasting the consequences of the alternative implementing.

The mathematical methods of general application are as follows: data collecting and preliminary processing (filtering, lost data imputation, extreme value processing, etc); effective organization of interaction IDSS – User; knowledge and data storing in the system bases for future usage; methods of data representation in convenient form for a user; the communication means for organizing communication with other information processing systems; logical control procedures.

The special mathematical methods are selected according to specific problem statement regarding data processing, model constructing, risk estimation, alternative decision generation etc. In most cases of IDSS development the necessity arises for usage of the following special methods: special procedures for preliminary processing of data and knowledge; the methods, methodologies of mathematical modeling on the basis of statistical data and expert estimates; forecasting methods based on the models constructed and forecast combining procedures; static and dynamic optimization (optimal control); the methods for quality analysis of computational procedures at all stages of IDSS hierarchy – appropriate sets of quality criteria: data quality, model adequacy, forecast quality, and alternative decision quality. All the methods and procedures mentioned above are well described in special modern publications. For example, time series modeling and forecasting methods are available in multiple references [9–11], and the area is progressing quickly. The IDSS designer should perform appropriate special literature review and select the best possibilities for his own application using system analysis approach to the system design.

IDSS architecture for financial risk analysis. IDSS architecture is represented by the large-scale system blocks and the links between them. It provides possibility for understanding the purpose of the system design, and its basic functions (Fig. 1).

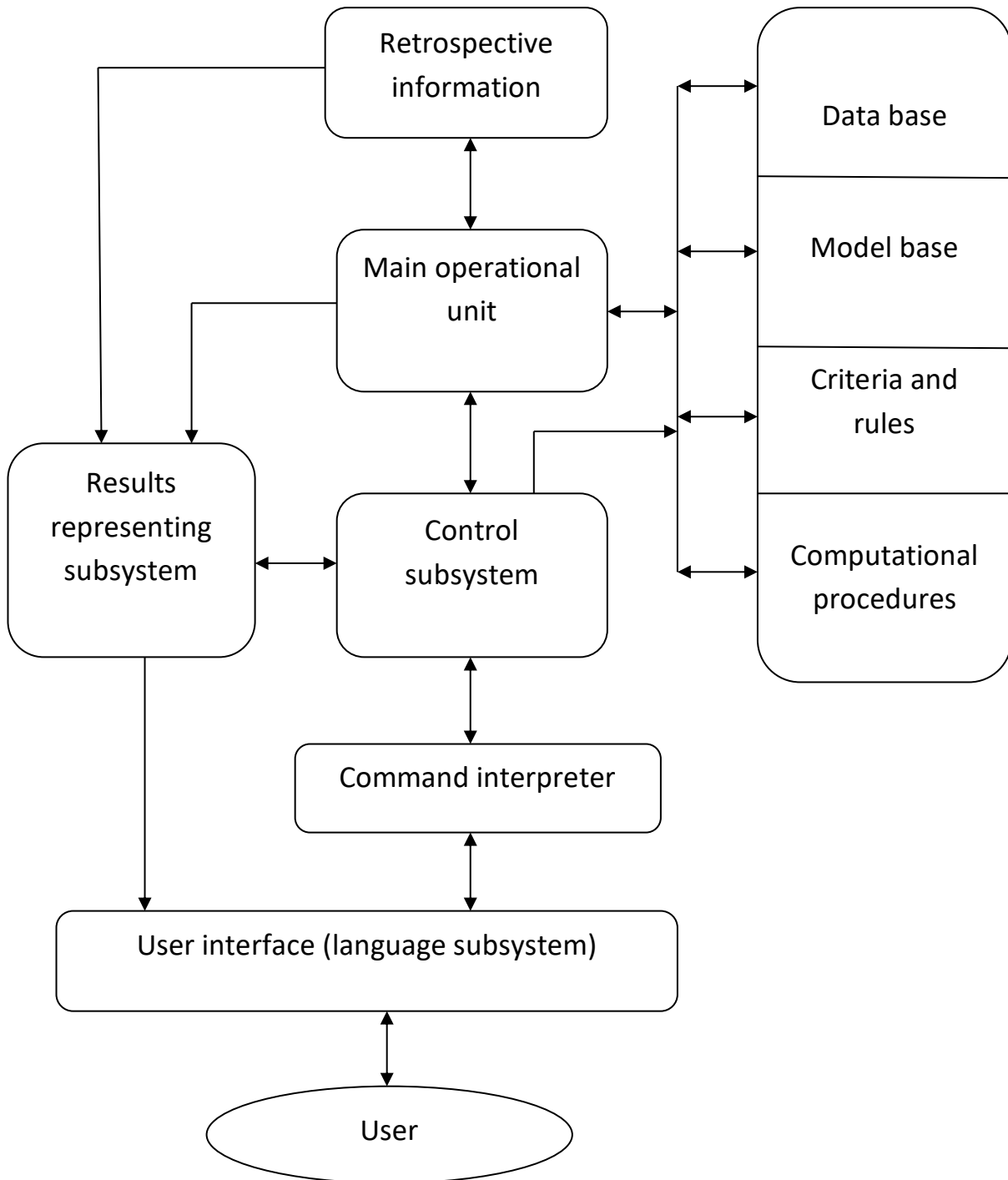


Figure 1. IDSS architecture for financial risks analysis

IDSS functioning is monitored by a user using the command interpreter. The commands are accepted by the main operational unit that coordinates functioning of all subsystems. The system interface is its basic element from the point of view providing the user with necessary interaction information. The architecture shown in Fig. 1, has general layout, and it can be used for solving many

similar problems of data preparing for modeling, mathematical model constructing, optimal state and parameter estimation, state forecasting, diagnostic of the processes under study, risk estimation, managerial decision support etc.

2. SOME MATHEMATICAL MODELS USED FOR FINANCIAL RISK ANALYSIS

Consider in short some types of mathematical models used in the IDSS proposed for operational risk analysis. Such risks are widely available at various companies and enterprises and are provoked by the inadequate or erroneous internal processes at enterprise, staff and system errors, or under influence of non-favorable external events. The most popular models are as follows: Generalized Linear Models (GLM), Bayesian networks, neuron and neuro-fuzzy models, multivariate distributions, Sb-AMA (Scenario-based Advanced Measurement Approach), and some others. GLM can be considered as extension of multiple linear regression with the following distinctive features: data distribution can be non-Gaussian and not necessarily continuous, for example, binomial; the forecasts are computed as linear combination of predictors linked to dependent variable via specific link function. GLM represent generalized class of statistical models that includes linear regression; the models of variance and covariance analysis; log-linear and nonlinear probit/logit models; normal, binomial, Poisson, and gamma distributions. Generally, dependent variable has exponential distribution of the form:

$$f(y, \theta, \phi) = \exp \left[\frac{y\theta - b(\theta)}{a(\phi)} + c(y, \phi) \right],$$

where, a, b, c are some functions corresponding to definite distribution; y is dependent variable; θ is canonic parameter; ϕ is variance parameter. Particular role plays the function, $b(\cdot)$, that describes correspondence between dependent variable mean and its variance.

Bayesian networks (BN) are statistical-and-probabilistic models that are represented in the form of directed acyclic graphs (DAG) where vertices are selected variables, and arcs show existing causal relations between the variables. Each variable is characterized by the complete finite set of mutually exclusive states. The relations between the BN variables are established by expert approach or special statistical-and-probabilistic tests if necessary data is available. The model parameters are conditional (for daughter variables) and non-conditional (for parent variables) probabilities of the variables values that are contained in corresponding tables [13].

The Sb-AMA approach suggests identification and selection of risk factors providing the possibility for scenario generating. This model is directed towards loss estimation that would happen in the future when some event is happening. Using historical data or expert estimate for each scenario is estimated frequency and size of loss. Here Monte Carlo simulation helps to estimate distribution of the loss without complicated mathematical computations. The estimation procedure is performed in assumption that the distribution family is known to which belongs the number of possible loss. The method is described in the following way: the random variable x , that characterizes the total loss that happened within the time interval t , can be defined by the sum:

$$x = \sum_{i=1}^{n(t)} L_i,$$

where, $n(t)$ is random value that characterizes the number of loss of specific type (here operational loss), that happened within the time interval t ; L_i is the set of random values that characterize the volumes of separate specific loss. The distribution of the random variable, x , provides the possibility

for achieving point loss estimation as well as compute quintile at given level what provides the possibility for estimating OpVaR (Operational Value-at-Risk) value.

3. CONCLUSIONS

The methodology was proposed for constructing IDSS directed for financial risks analysis and estimation on the principles of system analysis: hierarchical system architecture; identification and taking into consideration possible probabilistic-and-statistical uncertainties; application of optimization and adaptation procedures for model structure and parameter estimation; monitoring of computational procedures at all stages of data processing; alternatives generating regarding decision making. The IDSS designed has modular structure that provides the possibilities for modifying and expanding its functions with extra methods and techniques. The system provides the possibility for reaching high quality intermediate and final results thanks to applying appropriate quality criteria at each stage of computations. Practical applications of the IDSS proposed for operational and credit risk estimation gives an evidence for good perspectives of expanding its functional possibilities and achieving high quality results. In future studies it is planned enhancing data processing and model constructing automatizing level, the active usage of intellectual data processing techniques and models of Bayesian type, neural nets, combined models, and simulation procedures.

REFERENCES

1. Kumamoto H., Henley E. J. Probabilistic Risk Assessment and Management for Engineers and Scientists. – New York: IEEE Press, 1996. 612 p.
2. International Convergence of Capital Measurement and Capital Standards. A Revised Framework. Comprehensive Version. – Basel Committee on Banking Supervision, Bank for International Settlements. – Basel, 2006. 158 p.
3. Bidyuk P.I., Gozhyj O.P., Korshevnyuk L.O. Computerized Decision Support Systems. – Mykolaiv: Black Sea Petro Mogyla State University, 2012. 380 p.
4. Hoff P. A First Course in Bayesian Statistical Methods. – New York: Springer, 2009. 270 p.
5. Klugman S.A. Bayesian Statistics in Actuarial Science. – Boston: Kluwer Academic Publishers, 1992. 243 p.
6. Neil M. Using Bayesian networks to model expected and unexpected operational losses / Neil M., Fenton N.E., Tailor M. // Risk Analysis. 2005. p. 34-57.
7. Polovcev O.V., Bidyuk P.I., Korshevnyuk L.O. The System Approach to Modeling, Forecasting and Management of Financial processes. – Donetsk: Oriental Publishing House, 2009. 286 p.
8. Hollsapple C.W., Winston A.B. Decision support systems. – Saint Paul: West Publishing Company, 1996. 860 p.
9. Tsay R.S. Analysis of financial time series. – Hoboken: Wiley & Sons, Inc., 2010. – 715 p.
10. Bidyuk P.I., Romanenko V.D., Tymoshchuk O.L. Time Series Analysis. – Kyiv: Polytechnika, 2010. 317 p.
11. Gilks W.R., Richardson S., Spiegelhalter D.J. Markov chain Monte Carlo in practice. – New York: Chapman & Hall/CRC, 2000. 486 p.
12. Zgurovsky M.Z., Bidyuk P.I., Terentyev O.M. Method of constructing Bayesian networks based on scoring functions // Cybernetics and System Analysis, 2008, No.2, pp. 219-224.
13. Jensen F.V., Nielsen Th.D. Bayesian Networks and Decision Graphs. – New York: Springer, 2007. 457 p.

ЧАТ-БОТ ЯК СЕРЕДОВИЩЕ РОЗГОРТАННЯ СИСТЕМИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ: ПРИКЛАД ТЕЛЕГРАМ-БОТУ ПО НАДАННЮ РЕКОМЕНДАЦІЙ ЩОДО ВИБІРКОВИХ ДИСЦИПЛІН

Харабара Д.В.¹, Статкевич В.М.²

Національний технічний університет України «Київський політехнічний інститут ім. Ігоря Сікорського», Київ, Україна

¹ dharabara81@gmail.com, ² mstatkevich@yahoo.com [0000-0001-5210-9890]

В сучасному світі людство щодня стикається із сотнею рутинних дилем, вибір оптимального рішення в яких залежить від власного досвіду, рекомендацій оточуючих чи наявної інформації в мережі Інтернет. Зокрема із вибором оптимального варіанту зустрічаються й студенти, при необхідності визначення вибіркових дисциплін на наступний семестр. Задля полегшення та пришвидшення даного процесу можна використовувати Системи підтримки прийняття рішень (СППР). А в якості інтерфейсу ідеально підійде чат-бот платформа, яка є інтуїтивно зрозумілою для сучасного покоління. Метою роботи є спрощення процесу вибору вибіркових дисциплін, шляхом поєднання корпоративної автоматизації із СППР. Результатом дослідження є Telegram чат-бот із модулями надання рекомендацій та занесення оцінок. Перший із яких базується на рекомендаційній системі із використанням колаборативної фільтрації. А інший створений задля наповнення бази оцінок. У роботі було використано теоретичні та емпіричні методи дослідження.

Ключові слова: чат-бот, система підтримки прийняття рішень, рекомендаційна система, нода, вибіркові дисципліни.

1. ВСТУП

В наш час, епоху стрімкого розвитку технологій, багато компаній, організацій, установ в прагненні оптимізації своїх робочих процесів звертають неабияку увагу на комунікацію із своєю цільовою аудиторією, адже саме вона має неабиякий вплив на вектор розвитку компаній. Проте застарілі методи комунікацій: телефонні опитування, “гарячі лінії” і т. д. все більше відходять на другий план, а лідерство в напрямку швидкої та ефективної допомоги посідають чат-боти, адже зручніше написати своє питання зрозумілим повідомленням, без обмеження часових рамок і отримати одразу ж відповідь, ніж тривалий час очікувати поки ваш дзвінок прийме оператор.

В той ж час суспільство щодня стикається із сотнею рутинних дилем: який одяг сьогодні обрати, в залежності від погоди; який кінотеатр обрати для перегляду новинки; який курс валюти сьогодні, та як він змінювався за місяць. І це лише верхівка айсбергу буденних виборів, щодня кожен із нас робить безліч виборів, пошук найкращого варіанту в яких напяму залежить від нашого досвіду, власних переконань, чи рекомендацій друзів. А що якщо, досвіду немає і друзі не можуть нічого порекомендувати? Тоді на допомогу приходять всевітня павутина і години пошуку необхідної інформації, яка врешті-решт не дасть чіткої рекомендації, а просто поділиться із вами даними, які вам уже самим доведеться аналізувати і

робити вибір. І тут на допомогу можуть прийти ті самі чат-боти, які скорочують час для пошуку інформації, а для надання чітких рекомендацій можна використати рекомендаційну систему.

Зокрема здійснення вибору, фактично всліпу, зустрічається й в студентському житті, коли потрібно обирати вибіркові дисципліни на наступний семестр. Це питання неабияк ставить в ступор будь-якого студента, адже при виборі потрібно аналізувати рекомендації інших студентів, програму дисципліни та багато інших аспектів, тому значно простіше було б просто отримати рекомендацію опираючись на досвід попередників, без їх залучення до процесу.

Зважаючи на все вище сказане, очевидно, що для вирішення дилеми буденних виборів, зокрема вибору вибіркових дисциплін, потрібний системний підхід, який був би швидким та практичним у використанні, а його розробка вимагала невеликих витрат. Ідеальним підходом, який буде задовольняти всі вимоги, є розробка системи підтримки прийняття рішень на базі чат-боту із інтуїтивно зрозумілим інтерфейсом та простою логікою отримання вхідних даних для визначення рекомендацій.

2. МЕТОДИ ТА ПІДХОДИ ДО РОЗРОБКИ СППР

За одним із визначень, система підтримки прийняття рішень (СППР) – це інтелектуальна система, яка сприяє формуванню пропозицій, на основі аналізу релевантної інформації, отриманої з віртуального сховища даних, що консолідує велику кількість джерел [1]. Працює вона на базі різних підходів до аналізу даних користувача та предметної області, а основною задачею є надання інформації щодо оптимальних альтернатив, які можуть зацікавити користувача.

Якщо говорити про самі підходи до побудови СППР, то їх досить багато. Зокрема варто виділити [2]:

- рекомендаційні системи, засновані на колаборативній фільтрації – працюють на припущенні, що якщо смаки користувачів співпадали раніше, то їх вибір буде однаковим й у майбутньому;

- рекомендаційні системи, засновані на змісті – працюють на порівнянні смаків користувача та певних характеристик предмета;

- рекомендаційні системи, засновані на знаннях – працюють на основі деяких заздалегідь визначених правил та обмежень предметної області рекомендаційної системи та явних вимог користувача;

- демографічні рекомендаційні системи – користувачі діляться на групи, яким і надаються спільні рекомендації;

- рекомендаційні системи, засновані на корисності – для кожного предмета за спеціальною формулою розраховується його корисність для користувача;

- гібридні рекомендаційні системи – являють собою комбінацію декількох різних підходів.

Для вирішення поставленої задачі, а саме отримання рекомендацій щодо вибіркових дисциплін, чудово підходить підхід із використанням колаборативної фільтрації, адже саме він є автоматизацією традиційного підходу при виборі вибіркових дисциплін, під час якого студенти звертаються задля поради до своїх попередників.

Фактично, рекомендаційні системи, що засновані на колаборативній фільтрації, представляють собою матрицю, де по одній із осей розташовані користувачі, а по іншій предмети. А комірки самої матриці заповнені кількісними показниками відношення студента до даного предмету, в нашому випадку, це оцінка отримана із дисципліни. При цьому,

відсутність значень означає, що користувач, до цього моменту, ніяк не взаємодіяв із предметом (Таблиця 1). Відповідно ціллю системи є заповнення пустих комірок.

Таблиця 1. Приклад колаборативної фільтрації

	Предмет 1	Предмет 2	Предмет 3	Предмет 4
Студент 1	75	80	95	80
Студент 2	65	95	70	95
Студент 3	70	78	75	78
Студент 4	80	?	95	65

В загальному, для заповнення пустих комірок використовують два підходи:

- 1) на основі користувачів – серед усіх користувачів необхідно виділити n найбільш схожих на ключового та на основі їх оцінок зробити прогноз;
- 2) на основі предметів – серед усіх предметів необхідно виділити n найбільш схожих на ключовий та на основі їх оцінок зробити прогноз.

А для визначення схожості предметів чи користувачів використовують дві групи алгоритмів:

- алгоритми засновані на пам'яті;
- алгоритми засновані на моделі.

Зокрема при створенні СППР для визначення вибіркокових дисциплін доцільно використовувати підхід на основі користувачів та алгоритм визначення схожості заснований на пам'яті.

В якості пам'яті можна використовувати відомості щодо оцінок попередників. А схожість студентів, в такому випадку буде визначатися на основі оцінок їх обов'язкових дисциплін ключового користувача та попередніх студентів.

Якщо ж говорити про самі алгоритми, що засновані на пам'яті, то вони працюють шляхом розрахунку деякої міри подібності на основі тих даних, які наявні в системі (оцінок попередників), і подальшого її використання для розрахунку передбачення.

Серед найбільш застосованих мір виділяють косинус подібності, критерій кореляції Пірсона, скоригований косинус подібності, коефіцієнт Жаккара [3].

Для вирішення поставленої задачі було запропоновано використовувати саме косинус подібності, який представляє собою кут між векторами оцінок двох студентів, та задається наступною формулою [4]:

$$\text{cosineSimilarity}(u_i, u_j) = \cos(u_i, u_j) = \frac{\sum_{k=1}^n u_{i,k} u_{j,k}}{\sqrt{\sum_{k=1}^n u_{i,k}^2} \sqrt{\sum_{k=1}^n u_{j,k}^2}}. \quad (1)$$

А в якості математичного апарату для визначення прогнозованого значення оцінки було обрано агрегуючу функцію середнього зваженого [5]:

$$u_{i,k} = \frac{\sum_{u_j \in U'} \text{similarity}(u_i, u_j) u_{j,k}}{\sum_{u_j \in U'} |\text{similarity}(u_i, u_j)|}. \quad (2)$$

В результаті ми отримаємо СППР що базується на колаборативній фільтрації із використанням алгоритму визначення міри подібності, на основі пам'яті (косинус подібності) та агрегуючої функції середнього зваженого значення для визначення прогнозованої оцінки.

3. ПОСТАНОВКА ЗАДАЧІ ТА РОЗРОБКА ЧАТ-БОТУ

В рамках дослідження було прийнято рішення по розгортанню системи підтримки прийняття рішень на базі чат-боту в Telegram, оскільки чат-боти по своїй структурі є досить простими та інтуїтивно зрозумілими, а їх функціонування на базі месенджерів (Telegram, Viber, Facebook) робить їх використання зручним та швидким для будь-якого користувача.

Розробку самої бот-платформи із вбудованою рекомендаційною системою, було прийнято робити в хмарному середовищі розробки Corezoid, яке дає можливість створювати, редагувати та керувати процесами будь-якої складності. Corezoid це зручне рішення для розробки та подальшої підтримки невеликих ботів, адже він має вбудований набір нод, які фактично замінюють традиційні структури програмування (if...else, function(), function (return), setTime() і т.д.), що значно спрощує та пришвидшує розробку. Також дане хмарне середовище розробки має вбудовану технологію webhook, що в поєднанні із TelegramBotApi дає можливість в режимі реального часу відслідковувати повідомлення, що відправляються боту, та переадресовувати їх в логічне ядро.

Архітектура чат-боту складається із:

- 1) системи обробки повідомлень – відповідає за обробку вхідних та вихідних повідомлень;
- 2) модуля внесення оцінок – виконує задачу внесення нових оцінок із дисциплін;
- 3) модуля отримання рекомендацій – виконує безпосередній аналіз вхідних даних користувача, та надає альтернативи для вибору із певною мірою пріоритетності;
- 4) логічного ядра – відповідає за маршрутизацію користувача по боту;
- 5) сховищ даних – сховища дисциплін, студентів, викладачів, оцінок, які створені в State Diagram (вбудована в Corezoid діаграма станів).

Задля створення каналу комунікації було використано готовий програмний апарат від Telegram, а саме телеграм бот BotFather, який дає можливість створювати та налаштовувати ботів для різного функціоналу. Зокрема в рамках дослідження нас цікавить унікальний токен боту, який формується після його створення, із використанням якого, ми зможемо сформуванати шлюз для повідомлень, які будуть надсилатися боту, через webhook [6] (Рис. 1).

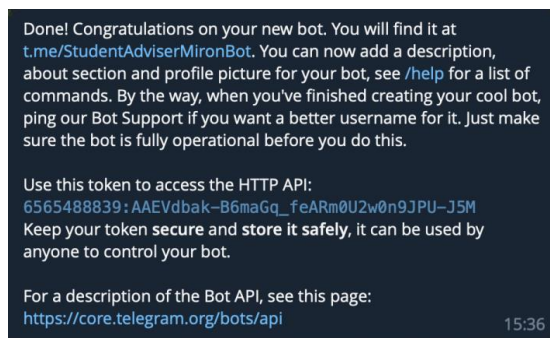


Рисунок 1. Згенерований токен боту

Результатом створення шлюзу став процес, в який переадресовуються всі повідомлення із боту.

Якщо говорити про Систему обробки повідомлень, то другою її складовою є інший процес, який, використовуючи метод `/sendMessage` [7], надсилає POST запит із повідомленням, яке транслюється в телеграм користувачу.

Задачу постійної актуалізації даних, задля отримання більш точних результатів, покликаний вирішити Модуль внесення оцінок. Доступ до даного модулю надається тільки

викладачам і тільки до конкретних дисциплін. Авторизація відбувається шляхом введення логіну та паролю, що зберігаються в Сховищі викладачів (Рис. 2).

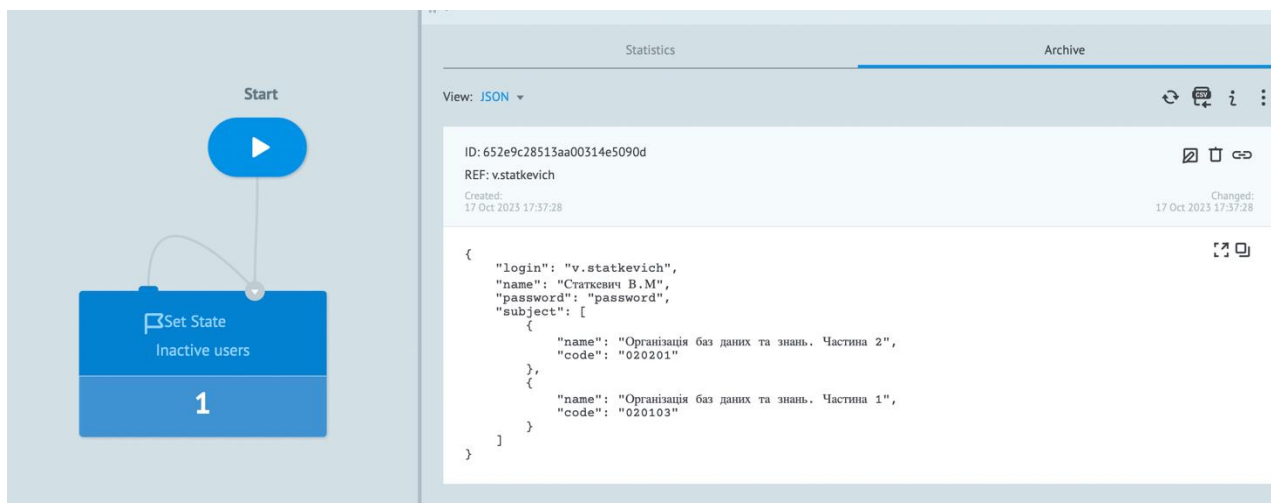


Рисунок 2. Сховище викладачів

Як зазначалось попередньо, в якості математичного апарату рекомендаційної системи є функція косинусу подібності (1) та функція середнього зваженого (2). Але, оскільки при аналізі подібності студентів до уваги будуть братися всі попередники, то було прийнято рішення додати також мінімальне значення подібності, яке відповідає за відсіювання найменш подібних попередників, за стартовий показник було прийнято взяти значення 0,995. Варто зауважити також, що дане значення буде динамічно зменшуватись, за умови відсутності подібних попередників. Структурну схему рекомендаційного модуля можна побачити на рисунку нижче (Рис. 3).

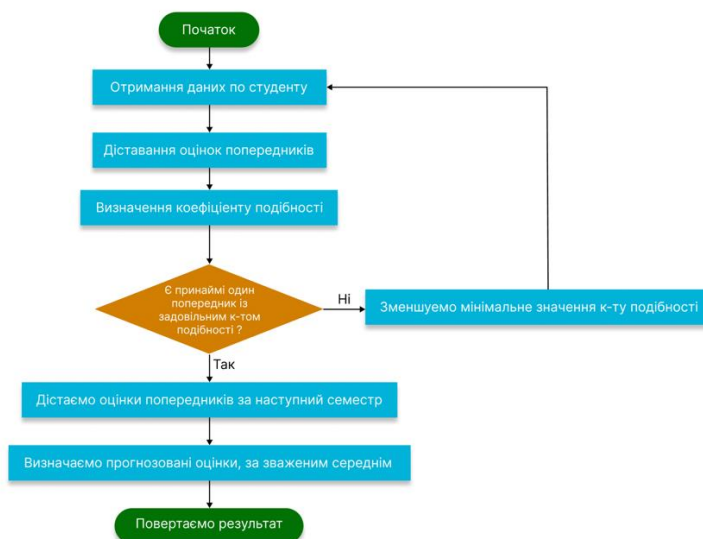


Рисунок 3. Структурна схема модулю отримання рекомендацій

З програмної сторони даний процес базується на двох основних функціях.

1. Функція визначення коефіцієнту подібності (див. формулу (1)).

```

function calc_coef_similar (std1_marks, std2_marks){
  let main_std_marks = std1_marks.filter(el => el.additional == false);
  let upper = 0;

```

```

let lower_1 = 0;
let lower_2 = 0;
main_std_marks.forEach(m => {
    upper = upper + parseInt(m.mark) * parseInt(std2_marks.find(el =>
el.subject == m.subject).mark);
    lower_1 = lower_1 + (parseInt(m.mark) * parseInt(m.mark));
    lower_2 = lower_2 + (parseInt(std2_marks.find(el => el.subject ==
m.subject).mark) * parseInt(std2_marks.find(el => el.subject ==
m.subject).mark));
});
let result = upper/(Math.sqrt(lower_1)*Math.sqrt(lower_2));
return result;
}

```

2. Функція обчислення прогнозованої оцінки (див. формулу (2)).

```

addition_subj.forEach(sbj =>{
    let upper = 0;
    let lower = 0;
    v_pred_stud_marks.forEach(std =>{
        if(std_marks.find(m => m.subject == sbj.code).mark != ""){
            upper = upper + (parseInt(std_marks.find(m => m.subject ==
sbj.code).mark) * data.similarity_coefficients.find(cof => cof.student ==
std.id).coef);
            lower = lower + data.similarity_coefficients.find(cof =>
cof.student == std.id).coef;
        }
    });
    let result = Math.round(upper/lower);
    predict_marks.push({
        'pred_mark': result,
        'subject_code': sbj.code,
        'subject_name': sbj.name
    });
});

```

Готовий та функціонуючий процес модулю надання рекомендацій зображений нижче (Рис. 4).

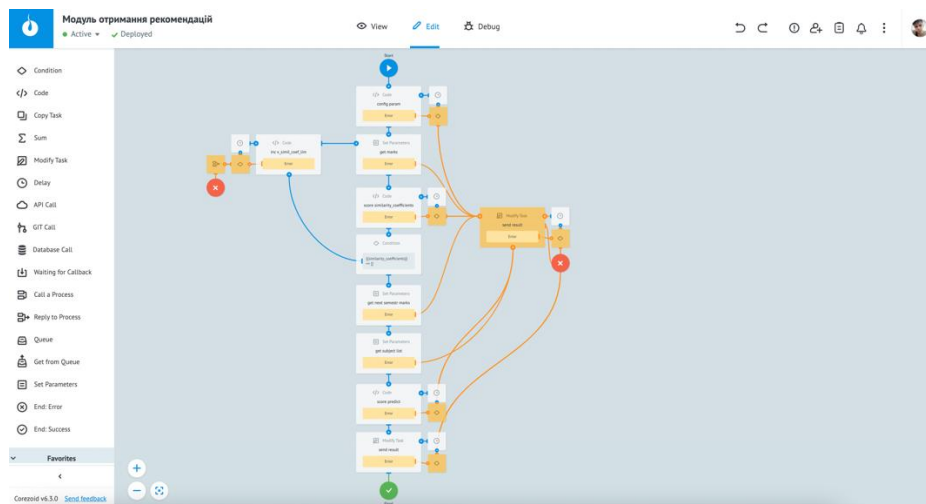


Рисунок 4. Процес модулю надання рекомендацій

4. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Результатом нашого дослідження стала повністю готова до використання система підтримки прийняття рішень для визначення вибірових дисциплін, яка має зручний та

зрозумілий інтерфейс, цілодобовий доступ та інтегрований модуль внесення оцінок, який дає можливість постійно актуалізувати дані, які в свою чергу покращують результати роботи СППР.

Тестування на базі тестової вибірки студентів, показало, що чат-бот із вбудованою рекомендаційною системою це чудовий підхід до вирішення дилеми вибору вибіркових дисциплін, оскільки повний процес отримання рекомендацій займає менше 5 хв. (Рис. 5), що в разі менше в порівнянні із традиційними підходами: пошуком інформації в інтернеті, самостійним аналізом досвіду попередників, аналізом програм дисциплін.

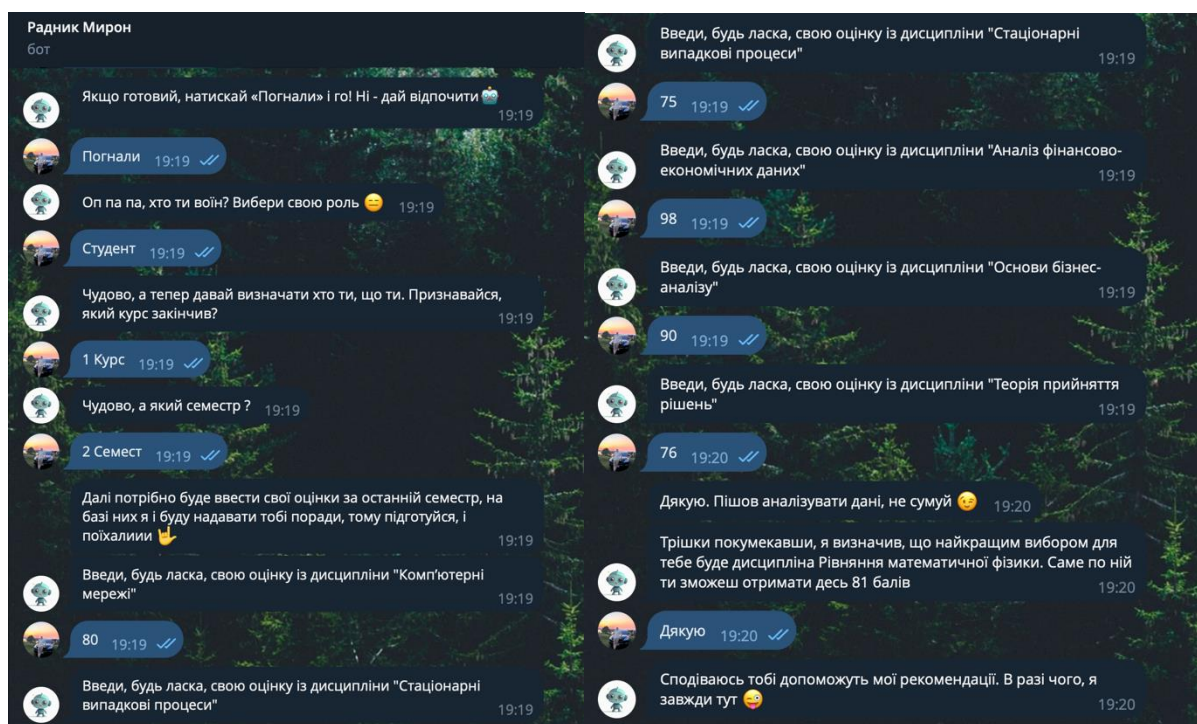


Рисунок 5. Приклад повного процесу отримання рекомендацій

Вагомим аргументом в сторону розробки подібних систем в середовищі Corezoid є те, що для розгортання подібних чат-ботів, після створення першого, достатньо скачати існуючий проект через інтерфейс, вставити його в новий обліковий запис, замінити токен каналу комунікацій, та оновити сховища даних. Без необхідності в зміні базових конфігурацій рекомендаційної системи, її розгортання може зайняти менше 15 хв.

5. ВИСНОВКИ

У наш час людство щоденно стикається із дилемою буденних виборів, зокрема перед кожним із студентів неодноразово поставало питання вибору дисциплін на наступний семестр.

Питання визначення вибіркових дисциплін не є складним, проте вибір найкращої альтернативи потребує досить багато часу, тому для спрощення даного процесу чудово підходить система підтримки прийняття рішень, яка базується на інтуїтивно зрозумілій платформі, та має змогу надавати рекомендації швидко та зручно.

В ході дослідження було розроблено саме таку рекомендаційну систему, на базі чат-боту. Бот платформа складається із Модуля внесення оцінок та Модуля отримання рекомендацій, а знайомий всім інтерфейс месенджеру Telegram дає можливість користувачам в будь-який час в режимі онлайн отримати рекомендацію.

Рекомендаційна система в даній платформі базується на колаборативній фільтрації із використанням алгоритму заснованого на пам'яті, де коефіцієнти подібності визначаються завдяки косинусу подібності, а прогнозування відбувається із використанням агрегуючої функції середнього зваженого.

Дослідження показало, що чат-бот це чудове середовище для розгортання рекомендаційних систем, метою яких є вирішення буденних дилем, адже воно відповідає всім вимогам кінцевого користувача: швидкість, зрозумілість, зручність.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ржеуський А.В., Кунанець Н.Е., Стахів М. Рекомендаційна система інформаційного обслуговування користувачів бібліотек. *Інформаційні моделі, системи та технології*: матеріали V наук.-техн. конф. (Секція 2. Інформаційні системи, м. Тернопіль, 1-2 лютого 2018 р.). Тернопіль : ТНТУ, 2018. С. 37.
2. Бідюк П.І., Коршевніук Л.О. Проектування комп'ютерних інформаційних систем підтримки прийняття рішень : навч. посіб. / Київ: ННК «Інститут прикладного системного аналізу» Національний технічний університет України «Київський політехнічний інститут», 2010. 340 с.
3. Пфанштиль И. Статистика YouTube 2019. Инфографика. URL: <https://rusability.ru/pfanshtil/statistika-youtube-2019-infografika/5fd296262dda593c3483e887>
4. Agarwal A., Chauhan M. Similarity Measures used in Recommender Systems: A Study // International Journal of Engineering Technology Science and Research IJETSR. Volume 4, Issue 6, 2017. P. 2394–3386.
5. Garcin F., Faltings B., Jurca R., Joswig N. Rating Aggregation in Collaborative Filtering Systems. *RecSys '09: Proceedings of the third ACM conference on Recommender systems*, ACM, New York, USA, 2009. P. 349–352.
6. Technical documentation. Corezoid actor engine. corezoid.com. URL: <https://doc.corezoid.com/docs>
7. Technical documentation. Telegram Bot API. telegram.org. URL: <https://core.telegram.org/bots/api>

АВТОМАТИЧНА СЕГМЕНТАЦІЯ ОБ'ЄКТІВ НА ЗОБРАЖЕННЯХ ЗА ДОПОМОГОЮ НАПІВКЕРОВАНОГО НАВЧАННЯ З ВИКОРИСТАННЯМ АКТИВНОГО НАВЧАННЯ ТА СТАБІЛЬНОЇ ДИФУЗІЇ

Шаповал Н.В., Крутий І.В.¹

Національний технічний університет України «Київський політехнічний інститут
ім. Ігоря Сікорського»

¹ fishingmaster871@gmail.com

Сегментація об'єктів в обробці зображень є критично важливим завданням, яке полягає у виділенні певних об'єктів на цифрових зображеннях. Враховуючи складність і мінливість даних зображень, це завдання стає складним, що вимагає передових методологій для ефективної сегментації. У цьому дослідженні підкреслюється використання напівкерованого навчання, доповненого методами активного навчання та інноваційним використанням моделей стабільної дифузії, для підвищення точності та ефективності автоматичної сегментації об'єктів. Метою дослідження є розробка більш надійного та адаптивного підходу до сегментації об'єктів, особливо у сценаріях, коли анованих даних недостатньо або коли зображення є дуже складними. Результатом цього дослідження є нова система сегментації об'єктів, яка поєднує напівкероване навчання зі стратегіями активного навчання та використовує стабільні моделі дифузії для покращення розмежування та розпізнавання об'єктів.

Ключові слова: сегментація об'єктів, напівкероване навчання, активне навчання, стабільна дифузія, обробка зображень, цифрова візуалізація.

1. ВСТУП

Сегментація об'єктів у цифровій обробці зображень, життєво важлива для цілого ряду застосувань від медичної діагностики до автономної навігації транспортних засобів, представляє складні виклики. Це завдання передбачає точну ідентифікацію і розмежування об'єктів у різноманітних і часто непередбачуваних середовищах зображень. Ускладнення виникають через такі фактори, як освітлення, перспектива і фоновий шум, що робить сегментацію об'єктів складною і багатогранною галуззю дослідження.

Напівкероване навчання, яке використовує як мічені, так і немічені дані для навчання, пропонує життєздатне рішення в сценаріях, де повністю мічені набори даних є дефіцитними або дорогими для придбання. Крім того, стратегії активного навчання підвищують ефективність і точність процесу навчання. Ці стратегії передбачають, що алгоритм навчання вибірково запитує мітки на найбільш інформативних точках даних, що демонструє покращення точності класифікації завдяки поєднанню методів активного навчання та напівкерованого навчання [4].

Подальшим досягненням у цій галузі є включення моделей стабільної дифузії. Отримані на основі принципів дифузійних процесів, ці моделі чудово справляються зі складними даними зображень.

Це дослідження має на меті систематично дослідити інтегрований підхід напівкерovanого навчання, активного навчання та моделей стабільної дифузії. Таким чином, воно прагне покращити сегментації об'єктів, зробити її більш надійною та пристосованою до широкого спектру завдань сегментації об'єктів, а також встановити нові стандарти точності та ефективності в цій важливій галузі обробки зображень.

2. МЕТОДОЛОГІЯ АВТОМАТИЧНОЇ СЕГМЕНТАЦІЇ ОБ'ЄКТІВ

Автоматична сегментація об'єктів на зображеннях, яка є важливим компонентом різних програм цифрової обробки зображень, може бути реалізована за допомогою комбінації методів напівкерovanого навчання, активного навчання та стабільної дифузії. Цей багатовимірний підхід спрямований на вирішення проблем, притаманних сегментації об'єктів, таких як неоднорідність зображень, зумовлена такими факторами, як освітлення, перспектива і фоновий шум.

Архітектура U-Net, яка є основною в роботі, є чудовою моделлю сегментації зображень. Це згортоква нейронна мережа, відома своєю ефективністю в сегментації біомедичних зображень, і вона була успішно адаптована для цілого ряду завдань сегментації зображень. До того ж U-Net розроблена для того, щоб добре працювати навіть з невеликою кількістю навчальних зображень. Вона може добре узагальнювати обмежені дані, що корисно для спеціалізованих задач, таких як виділення доріг, де мічені дані можуть бути дефіцитними і дорогими у виробництві. Завдяки своїй конструкції U-Net особливо добре навчається на невеликому наборі мічених даних, що є типовим сценарієм при роботі з супутниковими знімками високої роздільної здатності, де ручне нанесення міток є ресурсномістким.

В основі запропонованого підходу поєднання стабільної дифузії та напівкерovanого навчання з активним навчанням лежить використання як мічених, так і немічених даних. Ця стратегія ефективно максимізує корисність обмежених маркованих наборів даних, що є поширеною перешкодою в задачах сегментації зображень. Активне навчання ще більше вдосконалює цей процес, дозволяючи алгоритму навчання вибірково запитувати мітки на найбільш інформативних точках даних, тим самим оптимізуючи процес навчання і підвищуючи ефективність і точність сегментації.

Крім того, інтеграція моделей стабільної дифузії, які базуються на принципах дифузійних процесів, пропонує точний підхід до інтерпретації складних даних зображень. Ці моделі значно покращують здатність алгоритмів сегментації адаптуватися до складнощів реальних наборів даних зображень.

Поєднання стабільної дифузії та напівкерovanого навчання з активним навчанням потребує ретельної підготовки даних. Враховуючи різноманітність і складність зображень у наборах даних, було обрано напівкероване навчання (SSL), яке здатне навчатися як на мічених, так і на немічених даних. У SSL невелика частина даних позначається, щоб керувати процесом навчання, тоді як переважна більшість залишається не позначеною. Модель спочатку вивчає загальні риси з більшого немаркованого набору, а потім уточнює своє розуміння за допомогою меншого маркованого набору. Цей підхід має вирішальне значення для обробки широкого спектру варіацій і складнощів, притаманних даним зображень, що робить модель більш пристосованою до різних сценаріїв сегментації [3].

Під час тестування моделі методи активного навчання використовуються для ітеративного відбору найбільш інформативних точок даних для маркування. Цей підхід є особливо корисним для зменшення робочого навантаження при ручному маркуванні, одночасно покращуючи продуктивність моделі. Модель визначає найбільш невизначені зразки – ті, щодо яких вона має найменшу впевненість у своїх прогнозах, – і запитує їхні мітки.

Такий цілеспрямований підхід гарантує, що модель навчається на найскладніших і найрізноманітніших прикладах, тим самим підвищуючи її точність і узагальнюваність [2].

Моделі стабільної дифузії інтегруються з основною моделлю машинного навчання U-Net для покращення здатності моделі обробляти різні умови візуалізації. Ці моделі здатні генерувати синтетичні зображення, які імітують реальні сценарії, тим самим розширюючи навчальний набір даних ширшим діапазоном умов. Таке доповнення має вирішальне значення для навчання моделі розпізнавати і точно сегментувати об'єкти за різних умов освітлення, кутів і фонів. Стабільні моделі дифузії також допомагають у створенні крайніх випадків і рідкісних сценаріїв, які є важливими для тестування надійності моделі сегментації [4].

При оцінці ефективності моделей автоматичної сегментації об'єктів на зображеннях використовуються такі ключові метрики, як Binary Cross-Entropy Loss, Dice Coefficient, IoU (Intersection over Union) та Accuracy. Втрати двійкової перехресної ентропії є ключовим показником для кількісної оцінки точності піксельної класифікації, особливо ефективним для незбалансованих наборів даних, де домінують певні класи. Коефіцієнт Dice та IoU дають уявлення про просторове перекриття між прогнозами моделі та істинними даними, причому вищі значення вказують на більш точну сегментацію. Коефіцієнт Дайс, також відомий як оцінка F1 у задачах бінарної сегментації, вимірює співвідношення

$$2 * \frac{|X \cap Y|}{|X| + |Y|},$$

де X – це прогнозований набір пікселів, а Y – це істинні дані. IoU, або індекс Жаккара, обчислює відношення перетину до об'єднання прогнозованого і фактичного сегментів, забезпечуючи сувору міру точності сегментації. Точність, більш загальна метрика, оцінює загальну кількість правильно класифікованих пікселів, але може бути менш інформативною у випадках дисбалансу класів. Ці метрики в сукупності дають цілісне уявлення про роботу моделі, охоплюючи різні аспекти - від точності класифікації до точності меж сегментації. Використовуючи ці метрики, можна критично оцінити і вдосконалити ефективність своїх моделей сегментації при обробці різноманітних і складних даних зображень.

3. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Це дослідження фокусується на покращенні автоматичної сегментації об'єктів на зображеннях, використовуючи поєднання напівкерованого навчання з активним навчанням та стабільною дифузією. Використаний набір даних включав різноманітні зображення з високою роздільною здатністю. Набір даних містив широкий спектр зображень, що охоплював різні природні та міські ландшафти, з метою перевірити та оцінити можливості сегментації моделі.

Для більш детального розуміння після попередньої обробки зображень і масок, що включає в себе розділення даних на навчальні, валідаційні та тестові набори, зміна розміру, нормалізація та методи доповнення для покращення різноманітності набору даних. Після чого постає питання вибору підходящої моделі, у запропонованій роботі реалізовано архітектуру U-Net: U-Net обрано за її ефективність у сегментації зображень, особливо у сценаріях з обмеженою кількістю мічених даних.

Для оцінки ефективності на основі архітектури U-Net проводяться експерименти сегментації за допомогою напівкерованого навчання з використанням активного навчання, стабільної дифузії, а тоді комбінуються ці два методи та оцінюються результати.

Сегментації за допомогою напівкерованого навчання з використанням активного навчання: реалізовано загрузку як мічених, так і не мічених даних. Спочатку проводиться навчання моделі на початковому наборі мічених даних. Далі додається активне навчання, коли вибирається найбільш інформативні зразки з прогнозів, зразки, в яких модель найменш

впевнена, а після початкового навчання використовується модель для прогнозування на немічених даних. Відібрані зразки позначаються вручну і додаються до навчального набору даних. Модель перенавчається на цьому оновленому наборі даних. Цей процес повторюється, ітеративно покращуючи продуктивність моделі з кожним циклом. Ефективність такої моделі на вибраних даних в тому, щоб покращити модель, фокусуючись на складних випадках, що призводить до кращої продуктивності з меншою кількістю маркованих прикладів (Рис. 1).

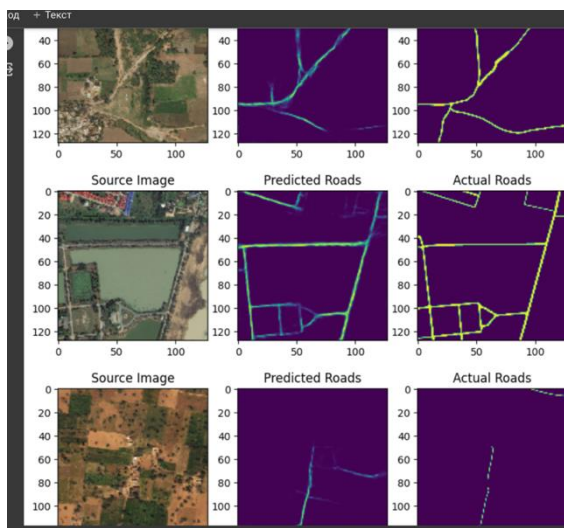


Рисунок 1. Результат детекції доріг з використанням сегментації за допомогою напівкерованого навчання з використанням активного навчання

Далі проводиться сегментація з використанням стабільної дифузії. Модель використовується для генерування варіантів зображень з вихідного набору даних, що передбачає зміну фонівих сцен, зберігаючи при цьому дороги незмінними. Ці згенеровані зображення потім додаються до навчального набору даних. Нова модель U-Net навчається на цьому розширеному наборі даних, який тепер містить як оригінальні, так і стабільні зображення, згенеровані дифузійною моделлю. Цей метод також покращив здатність моделі обробляти перешкоди та об'єкти, що перекриваються, на зображеннях (Рис. 2).

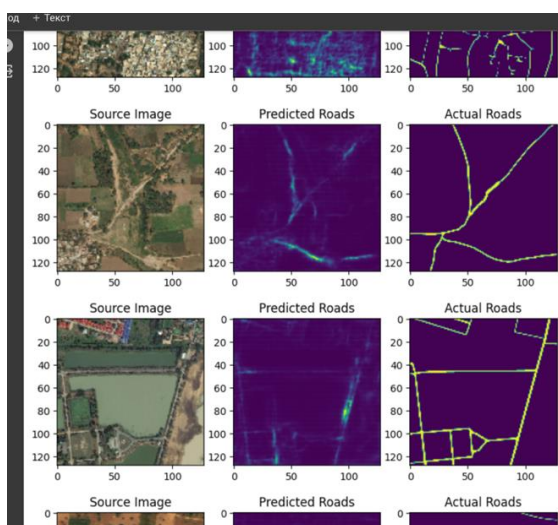


Рисунок 2. Результат детекції доріг з використанням сегментації за допомогою стабільної дифузії

Далі постає питання покращення роботи моделі шляхом об'єднання стабільної дифузії та напівкерованого навчання з активним навчанням. Головна роль і перевага стабільної дифузії, особливо для представленої задачі, складних даних і обмежена кількість сценаріїв. В тому що моделі стабільної дифузії використовуються для генерації синтетичних варіантів зображень, збагачуючи навчальний набір даних різноманітними фоновими сценами, зберігаючи при цьому послідовне представлення об'єктів. Ці додаткові синтетичні зображення покращують здатність моделі працювати зі складними об'єктами, що перетинаються на зображеннях, розширюючи сферу навчання та покращуючи адаптивність. Активне навчання допомагає вибрати найбільш інформативні та невизначені зразки з немаркованого набору даних. Ці зразки, після маркування, додаються до навчальної вибірки. Такий підхід забезпечує ефективне навчання, фокусуючись на складних прикладах, і призводить до підвищення продуктивності з меншою кількістю маркованих прикладів. І включаючи ці можливості, комбінування цих методів призводить до того, що стабільна дифузія збагачує процес напівконтрольованого навчання розширеним набором даних, що включає як оригінальні, так і синтетичні зображення. Потім застосовується активне навчання для подальшого уточнення точності моделі, зосереджуючись на найбільш невизначених зразках для маркування та перенавчання. (Рис. 3).

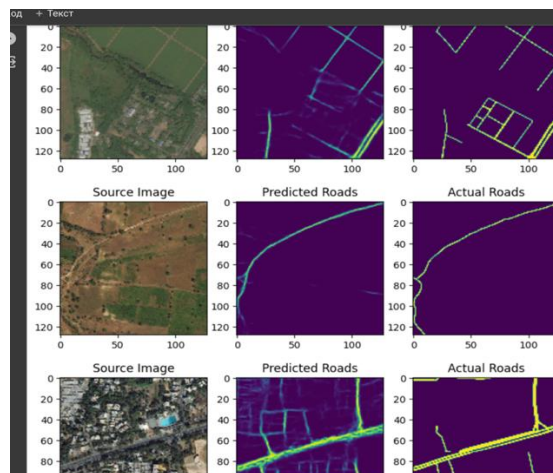


Рисунок 3. Результат детекції доріг з використанням сегментації за допомогою поєднання напівкерованого навчання з активним навчанням із стабільною дифузією

Сукупні результати були узагальнені в наступній таблиці (Таблиця 1), що представляє чітке порівняння між різними методами та їхніми відповідними метриками ефективності.

Таблиця 1. Порівняльний аналіз методів сегментації

Метод сегментації \ Метрики	U-Net	Напівкероване + активне навчання	Комбінований метод
Ассурасу	94,5 %	94,1%	95,5%
Dice Coefficient	0,69	1,05	1,12
Binary IoU	0,49	0,52	0,53

Результати підкреслюють чудову продуктивність комбінованого методу за всіма метриками. Зокрема, коефіцієнт Dice і бінарний IoU – ключові показники якості сегментації –

продемонстрували значне покращення, що ілюструє розширені можливості моделі для точного розмежування і сегментації об'єктів.

Крім того, було показано візуальне представлення сегментованих результатів (Рисунок 1 і Рисунок 2, Рисунок 3), що є чітким доказом ефективності сегментації моделі. Зображення підтвердили кількісні результати, продемонструвавши точність, з якою модель ідентифікувала та сегментувала об'єкти з різним фоном.

Поєднання методів демонструє значне покращення ключових показників ефективності, що свідчить про успішну інтеграцію методів. Результатом такого підходу є більш надійна, точна та ефективна модель сегментації, здатна обробляти різноманітні та складні дані зображень, що є особливо цінним у сегментації супутникових знімків високої роздільної здатності.

4. ВИСНОВКИ

Проведене дослідження ефективно демонструє переваги поєднання напівкерованого навчання, активного навчання та стабільної дифузії в контексті архітектури U-Net для сегментації зображень. Цей підхід є особливо цінним для вирішення проблем, пов'язаних з неоднорідністю зображень в задачах сегментації об'єктів. Інтеграція моделей стабільної дифузії доповнює набір даних синтетичними зображеннями, сприяючи адаптивності моделі до різних умов отримання зображень. Активне навчання відіграє вирішальну роль в оптимізації процесу навчання, забезпечуючи ефективність за рахунок фокусування на найбільш інформативних точках даних. Комбінована методологія не тільки максимізує корисність обмежених маркованих наборів даних, але й вводить ширший спектр сценаріїв, про що свідчить покращення ключових метрик, таких як точність, коефіцієнт Dice та бінарний IoU. Результати підтверджують ефективність цього підходу, що призводить до помітного підвищення продуктивності моделі, яка має вирішальне значення для моєї роботи з сегментації супутникових знімків високої роздільної здатності.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Fazakis, N. et al. (2019). Combination of Active Learning and Semi-Supervised Learning. [Електронний ресурс]. Режим доступу: <https://www.mdpi.com/1099-4300/21/10/988>.
2. Settles, B. (2009). Active Learning Literature Survey. University of Wisconsin-Madison. [Електронний ресурс]. Режим доступу: <https://research.cs.wisc.edu/techreports/2009/TR1648.pdf>.
3. Zhu, X., & Goldberg, A. B. (2009). Introduction to Semi-Supervised Learning. Synthesis Lectures on Artificial Intelligence and Machine Learning. [Електронний ресурс]. Режим доступу: https://www.academia.edu/36179630/Introduction_to_Semi_Supervised_Learning.
4. Ho, J., Jain, A., & Abbeel, P. (2020). Denoising Diffusion Probabilistic Models. Advances in Neural Information Processing Systems. [Електронний ресурс]. Режим доступу: <https://proceedings.neurips.cc/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf>.
5. He, H. et al. (2014). Automatic object segmentation of unstructured scenes using colour and depth information. IET Computer Vision. [Електронний ресурс]. Режим доступу: <https://ietresearch.onlinelibrary.wiley.com/doi/full/10.1049/iet-cvi.2013.0018>.

КЕРУВАННЯ ВІДНОСИНАМИ З КЛІЄНТОМ ЯК СЕРВІС В ГАЛУЗІ ЗАКЛАДІВ ОСВІТИ

Бабіч К.О.¹, Петренко А.І.²

Національний технічний університет України «Київський політехнічний інститут
імені Ігоря Сікорського», Київ, Україна

¹ babich912@gmail.com, ² tolja.petrenko@gmail.com [0000-0001-6712-7792]

Дослідження присвячено необхідності розробки та впровадження системи керування клієнтськими відносинами для приватних закладів освіти. Результати дослідження вказують на потребу сучасних шкіл, садочків, центрів раннього розвитку та інших закладів освіти в гнучкій та універсальній системі керування відносинами з клієнтами. Новизною є аналіз потреб бізнесу освіти в аспекті менеджменту взаємодії з клієнтами.

Ключові слова: клієнт, керування відносинами з клієнтом, CRM, освіта, заклади навчання, лояльність, аналіз.

1. ВСТУП

Сучасний конкурентний світ вимагає від закладів освіти не лише надання якісної освіти, а й активної участі в процесі формування позитивного сприйняття своїх клієнтів - учнів, студентів, батьків, та інших зацікавлених сторін. Керування відносинами з клієнтом (Customer Relationship Management, CRM) стає ключовим аспектом стратегії розвитку освітніх установ. У сучасному освітньому середовищі, де конкуренція між закладами освіти є надзвичайною, важливо визнати роль та значення керування відносинами з клієнтом (CRM) як стратегії підвищення якості освітніх послуг та забезпечення високого рівня задоволеності клієнтів. Це дослідження присвячене розгляду необхідності впровадження CRM-системи для керування відносинами з клієнтом в галузі освіти.

2. ЗНАЧЕННЯ КЕРУВАННЯ ВІДНОСИНАМИ З КЛІЄНТОМ В ОСВІТНЬОМУ СЕКТОРІ

В освітньому секторі, де конкуренція і вибір клієнта зростають, керування відносинами з клієнтом (CRM) стає визначальним елементом стратегії для закладів освіти. Забезпечення високої якості освіти вже не є єдиним критерієм успіху. Важливо будувати і утримувати позитивні відносини з учнями, студентами, їхніми батьками та іншими зацікавленими сторонами. Надалі ми розглянемо кілька ключових аспектів значення керування відносинами з клієнтом в освітньому секторі.

Задоволення та Утримання Клієнтів – CRM допомагає закладам освіти розуміти потреби та очікування своїх клієнтів, що включає учнів, студентів та їхніх батьків за допомогою збирання, систематизування, обробки та аналітики їхнього зворотнього зв'язку через ботів в різних соцмережах. При задоволенні цих потреб і врахуванні очікувань вищий рівень утримання клієнтів допомагає витримати конкуренцію та підвищити репутацію закладу.

Залучення та Рекрутинг – Ефективне керування відносинами з клієнтом сприяє активному залученню нових студентів та учнів через маркетингові розсилки на поштові скриньки та у вигляді смс-повідомлень. Позитивний досвід і взаємодія з потенційними клієнтами створюють підґрунтя для вибору конкретного закладу освіти.

Покращення Якості Освітніх Послуг – CRM допомагає закладам ефективно відстежувати академічний прогрес та вчительську ефективність через метрики. Аналіз зворотного зв'язку від клієнтів може бути використаний для постійного вдосконалення методів викладання та навчання.

Оптимізація Адміністративних Процесів – Використання CRM-системи дозволяє оптимізувати адміністративні процеси, такі як реєстрація, ведення документації та фінансовий облік. Це допомагає ефективніше використовувати ресурси адміністраторів та забезпечує більш зручну взаємодію з клієнтами.

Підвищення Лояльності та Ретенції – Ефективне керування відносинами з клієнтом сприяє створенню емоційного зв'язку із закладом освіти. Забезпечення позитивного враження і відношень допомагає підтримувати лояльність та зменшує ймовірність втрати клієнтів.

Аналіз та Використання Даних – CRM надає можливість збирати та аналізувати дані про клієнтів, що допомагає взяти до уваги індивідуальні потреби та адаптувати навчальні програми.

Керування відносинами з клієнтом в освітньому секторі є важливим фактором для успіху будь-якого закладу освіти. Врахування потреб та очікувань клієнтів, вдосконалення взаємодії та надання високоякісних послуг через використання CRM допомагають зберігати та залучати клієнтів, а також покращувати якість надання освітніх послуг.

3. ВПРОВАДЖЕННЯ CRM В ОСВІТНІХ УСТАНОВАХ

Важливо враховувати, що успіх впровадження CRM не тільки залежить від наявності системи, але й від того, як вона інтегрується в освітній процес та як користувачі використовують її можливості. Кожен заклад має унікальні потреби, і впровадження CRM-системи повинно враховувати ці особливості. Для початку проведемо порівняльний аналіз закладів освіти з та без CRM систем:

Таблиця 1. Порівняння закладів освіти з та без використання CRM.

Критерій	Без CRM	З CRM
Комунікація та зв'язок	Залежність від традиційних методів комунікації, що може бути менш ефективним у сучасному середовищі	Забезпечення ефективної комунікації між студентами, викладачами та адміністрацією. Можливість відстеження запитань та проблем.
Відстеження успішності студентів	Системи оцінювання та відстеження зведені до неефективних паперових носіїв.	Можливість відслідковувати академічні досягнення студентів, надавати персоналізовану підтримку та рекомендації.
Управління взаємодією з випускниками	Обмежені можливості для підтримки зв'язків зі випускниками та залучення їх в діяльність установи.	Забезпечення системи для взаємодії з випускниками, відстеження їх кар'єрного розвитку та отримання фідбеку.
Аналіз даних та прийняття рішень	Залежність від менш розвинутих систем для аналізу та відсутність централізованої бази даних.	Здатність аналізувати дані для прийняття стратегічних рішень та вдосконалення управління
Ефективність рекрутингу та утримання студентів	Залежність від традиційних методів рекрутингу та менш ефективного утримання студентів	Визначення стратегій рекрутингу, аналіз результатів та забезпечення задоволеності студентів
Адаптація до змін та інновацій	Процес опізнання та впровадження змін може бути ускладненим	Здатність швидко адаптуватися до нових технологій та інновацій у сфері освіти.

4. РЕЗУЛЬТАТИ ДОСЛІДЖЕНЬ

Дослідження в області керування клієнтськими відносинами в галузі освіти привело до визначення необхідності впровадження CRM-систем в закладах освіти через наступні переваги:

Стратегічне Значення CRM в Освіті – CRM визначає стратегічну орієнтацію освітніх установ на задоволення потреб та очікувань клієнтів, що включають студентів, учнів, та їхніх батьків.

Оптимізація Процесів Взаємодії – CRM допомагає оптимізувати процеси взаємодії, спрощуючи реєстрацію, адміністративні процедури та забезпечуючи швидкий доступ до необхідної інформації.

Покращення Якості Освітніх Послуг – Впровадження CRM сприяє постійному вдосконаленню якості навчального процесу через аналіз даних та зворотнього зв'язку від клієнтів.

Залучення та Утримання Студентів – CRM є ефективним інструментом для залучення нових студентів та утримання існуючих через створення персоналізованих та зручних умов.

Ефективна Комунікація – CRM дозволяє покращити комунікацію між вчителями, студентами, та батьками, забезпечуючи швидкий обмін інформацією та відстеження прогресу.

Роль CRM в Управлінні Ресурсами – Системи CRM допомагають ефективно використовувати ресурси, визначаючи пріоритети та оптимізуючи роботу персоналу.

Створення Позитивного Іміджу Закладу – Забезпечення високого рівня обслуговування та підтримки через CRM сприяє створенню позитивного іміджу установи серед клієнтів та громадськості.

Забезпечення Гнучкості та Адаптивності – CRM дозволяє швидко реагувати на зміни в потребах клієнтів та адаптуватися до нових вимог освітнього ринку.

Підвищення Лояльності та Ретенції – Ефективне впровадження CRM допомагає підтримувати лояльність студентів та зменшує ймовірність втрати клієнтів.

Постійне Вдосконалення через Дані та Зворотній Зв'язок - Використання даних з CRM та аналіз зворотного зв'язку дозволяє постійно вдосконалювати освітні процеси та відповідати на зростаючі вимоги клієнтів.

Ці пункти відображають ключові аспекти впливу керування відносинами з клієнтом на освітні установи та підкреслюють їх стратегічне значення в контексті сучасного освітнього середовища.

5. ВИСНОВКИ

Дослідження вказує на важливість керування відносинами з клієнтом в освітньому секторі та надає конкретні переваги ефективних стратегій у закладах освіти. Розуміння потреб та очікувань клієнтів, впровадження CRM, стратегії залучення та утримання, вимірювання задоволеності та результативності, а також ефективна комунікація є ключовими компонентами успішного керування відносинами в освіті. Дослідження підтверджує, що керування відносинами з клієнтом, впроваджене в освітніх установах, не лише покращує якість обслуговування, але і сприяє підвищенню рівня задоволеності клієнтів та позитивному розвитку установи в цілому.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. “Customer Relationship Management.” Wikipedia, November 23, 2023. https://en.wikipedia.org/wiki/Customer_relationship_management.

2. Gr. “Що Таке Управління Відносинами з Клієнтами Та Як Їх Організувати. - CRM Hugeprofit Blog.” CRM HugeProfit BLOG - ПРОСТИЙ ОБЛІК ФІНАНСІВ, ТОВАРІВ,

ПРОДАЖУ ТА ДОСТАВОК ДЛЯ МАЛОГО БІЗНЕСУ, July 27, 2023. <https://h-profit.com/blog/hto-takoe-upravlenie-vzaimootnosheniyami-s-klientami-i-kak-ih-organizovat/>.

3. Hashemi-Pour, Cameron, and Wesley Chai. "What Is CRM (Customer Relationship Management)?": Definition from TechTarget." Customer Experience, October 19, 2023. <https://www.techtarget.com/searchcustomerexperience/definition/CRM-customer-relationship-management>.

4. SoftwareSuggest. "30 Best CRM Software for Service Business in 2023." 30 Best CRM Software for Service Business in 2023. Accessed November 30, 2023. <https://www.softwaresuggest.com/crm-software/service-industry>.

5. Все що вам потрібно знати про CRM-системи. Accessed November 30, 2023. <https://business.dii.gov.ua/cases/prodazi/vse-so-vam-potribno-znati-pro-crm-sistemi>.

ТЕХНОЛОГІЇ WASM, WASI ТА WAGI ДЛЯ РОЗРОБКИ ТА РОЗГОРТАННЯ МЕРЕЖЕВИХ ЗАСТОСУНКІВ

Булах Б.В.¹, Бондаренко С.Д.²

Національний технічний університет України «Київський політехнічний інститут ім. Ігоря Сікорського», Київ, Україна

¹ bogdan.bulakh@gmail.com, ² stassss2009@gmail.com

Ця стаття зосереджується на потенціалі та перспективах застосування технологій WebAssembly (WASM), WebAssembly System Interface (WASI), та WebAssembly Gateway Interface (WAGI) у сфері веб-розробки та розгортанні мережесих застосунків. Аналізуючи ключові переваги цих технологій, такі як підвищення продуктивності, безпеки та гнучкості веб-додатків, стаття розглядає різноманітні приклади та сценарії їх застосування. Розглядаються практичні аспекти використання WASM, WASI та WAGI, включаючи їх вплив на розвиток цифрового веб-простору та можливості для інновацій у цій області.

Ключові слова: WebAssembly, WASM, WASI, WAGI, веб-розробка, мережесі застосунки.

1. ВСТУП

В останні десятиліття веб-технології пройшли шлях від простих статичних веб-сторінок до складних, високофункціональних мережесих застосунків, що змінило обличчя веб-розробки та взаємодії користувачів з цифровим контентом. Серед численних інновацій, що сприяли цьому розвитку, особливо важливу роль відіграє впровадження WebAssembly (WASM), яке відкрило нові можливості для розробників та користувачів. WASM, як низькорівнева бінарна інструкція, дозволяє запускати код на мовах, таких як C, C++, Rust, та інших, безпосередньо в браузері, що забезпечує значне покращення продуктивності та безпеки веб-додатків.

У цьому контексті, WASI (WebAssembly System Interface) та WAGI (WebAssembly Gateway Interface) є критично важливими доповненнями до екосистеми WebAssembly. WASI забезпечує стандартизований спосіб доступу до системних ресурсів, таких як файли та мережесі сокети, відкриваючи дорогу для більш широкого використання WASM не тільки у браузерах, але й у серверних застосунках. В свою чергу, WAGI пропонує легковаговий спосіб розгортання WebAssembly модулів як HTTP-сервісів, що спрощує розробку та розгортання мережесих застосунків, забезпечуючи високий рівень інтеграції та гнучкості.

Мета цієї статті полягає у дослідженні потенціалу та можливостей, які WASM, WASI, та WAGI пропонують для розробки та розгортання мережесих застосунків. Ми розглядаємо ключові переваги, які ці технології надають, такі як підвищення продуктивності, безпеки, та гнучкості веб-додатків, аналізуючи їхній вплив на ринок веб-розробок. Також ми зосереджуємось на практичних аспектах використання цих технологій, розглядаючи реальні приклади та кейси, які ілюструють їх ефективність у різноманітних сценаріях. В кінці, ми обговорюємо майбутнє та перспективи застосування WASM, WASI, та WAGI у розробці та розгортанні мережесих застосунків, оцінюючи їх потенційний вплив на подальшу еволюцію цифрового веб-простору [1, 2].

2. ОГЛЯД ТЕХНОЛОГІЙ

2.1. Що таке WASM (WebAssembly) і його переваги у веб-розробці

WebAssembly (WASM) є важливою інновацією у сфері веб-розробки, вносячи значні зміни у структуру та ефективність веб-додатків. Як низькорівнева бінарна мова, WASM дозволяє виконувати код швидше і ефективніше, ніж традиційні веб-технології, що є ключовим для розвитку складних веб-застосунків. Ця технологія відкриває нові горизонти для оптимізації продуктивності, дозволяючи розробникам створювати високопродуктивні веб-додатки з поліпшеною відгуковістю та користувацьким досвідом.

Основні переваги WASM у веб-розробці:

- Підвищення продуктивності: WASM дозволяє виконувати код на стороні клієнта набагато швидше, ніж традиційний JavaScript. Це забезпечує більшу швидкість відгуку веб-додатків та покращує користувацький досвід.
- Безпека: Завдяки своїй пісочниці, WASM забезпечує безпечне виконання коду, ізолюючи його від основної системи та браузера, тим самим знижуючи ризик безпеки.
- Мовна незалежність: WASM дозволяє розробникам використовувати мови програмування на свій вибір, не обмежуючись лише JavaScript. Це відкриває можливості для більш широкого використання мов програмування з високою продуктивністю та низькорівневим контролем.
- Універсальність: WASM може бути використаний у різних середовищах, включаючи мобільні пристрої та IoT пристрої, що робить його відмінним вибором для розробки переносних веб-додатків.
- Інтеграція з існуючими веб-технологіями: WASM легко інтегрується з існуючими веб-стандартами та API, дозволяючи розробникам створювати комплексні веб-застосунки, які використовують як WASM, так і JavaScript.

Незважаючи на те, що основна увага цієї статті приділяється WASI для розгортання мережових застосунків та її застосування для контейнеризації, важливо розуміти роль та внесок WASM, оскільки вона є фундаментальною для роботи WASI та WAGI [3, 4].

2.2. Роль WASI (WebAssembly System Interface) у розвитку мережових застосунків

WASI (WebAssembly System Interface) представляє собою інноваційний стандарт в екосистемі WebAssembly, який має за мету розширення можливостей та спрощення процесу розгортання мережових застосунків. WASI надає уніфікований інтерфейс для доступу до системних операцій, таких як робота з файлами, мережові запити, та інші ресурси, дозволяючи використовувати WebAssembly в широкому спектрі середовищ, включаючи серверні додатки.

Однією з ключових переваг WASI є його універсальність. Цей інтерфейс створює основу для запуску WebAssembly додатків в різних середовищах, від браузерів до обчислювальних хмар та IoT пристроїв. Це забезпечує високий рівень портативності застосунків, що значно спрощує процес їх розгортання та масштабування.

WASI впроваджує концепцію мінімально необхідних дозволів (principle of least privilege), яка забезпечує безпечний доступ до системних ресурсів. Кожен WASM модуль може мати доступ лише до тих ресурсів, які йому необхідні для виконання. Це значно зменшує ризики безпеки, пов'язані з несанкціонованим доступом до файлової системи або мережових ресурсів [5].

Завдяки WASI, розробники можуть легко розгорнути WebAssembly додатки в різних середовищах, уникнути необхідності залежати від конкретних особливостей операційних систем. Це полегшує інтеграцію додатків у різні платформи та екосистеми, забезпечуючи гнучкість та швидкість в розгортанні. WASI відкриває шлях до створення легковагових

контейнерів для WebAssembly додатків, що можуть виконуватися ізольовано від решти системи. Це створює можливості для більш ефективної контейнеризації та управління ресурсами, порівняно з традиційними підходами, такими як Docker контейнери. Незважаючи на свою новизну, WASI розроблено з урахуванням сумісності з існуючими веб-стандартами та технологіями. Це дозволяє інтегрувати WASI-сумісні додатки з існуючими системами та сервісами без значних змін в архітектурі.

WASI відкриває широкі можливості для практичного застосування. Від серверних додатків, що вимагають високої продуктивності, до мережесервісів з потребою в швидкому розгортанні та легкій вазі. Завдяки своїм перевагам, WASI може бути використаний для створення нового покоління мережесервісів, які відрізняються високою продуктивністю, безпекою та гнучкістю [3].

2.3. Застосування WAGI (WebAssembly Gateway Interface) для розгортання веб-додатків

WAGI (WebAssembly Gateway Interface) представляє собою ініціативу, що значно спрощує процес розгортання веб-додатків, розроблених з використанням WebAssembly. Ця технологія діє як міст між WebAssembly додатками та веб-середовищем, дозволяючи легко розгорнути WASM-модулі як частину веб-інфраструктури. WAGI забезпечує високий рівень інтеграції з існуючими веб-сервісами, що дозволяє використовувати WASM-додатки як ефективні, легковагові веб-сервіси.

Основна ідея WAGI полягає у створенні універсального інтерфейсу, який може обробляти HTTP-запити та відповіді, перетворюючи їх у формат, сумісний з WebAssembly. Це значно спрощує процес впровадження та розгортання веб-додатків, оскільки розробники можуть зосередитися на логіці додатку, а не на складностях інфраструктури.

Вагомою перевагою WAGI є його спроможність до швидкої адаптації та легкого впровадження у сучасні веб-архітектури, забезпечуючи гнучкість та швидкість розгортання додатків. Такий підхід дозволяє створювати веб-додатки, які легко масштабуються та підтримуються, відповідаючи сучасним вимогам до веб-розробки.

Отже, WAGI відкриває нові можливості для розробки та розгортання веб-додатків, надаючи розробникам потужні інструменти для ефективної роботи з WebAssembly у веб-середовищі. Його інтеграція з існуючими веб-технологіями та легкість використання роблять його привабливим вибором для розробників, які прагнуть максимально ефективно використовувати можливості WebAssembly у своїх проектах [5].

3. МОЖЛИВОСТІ ТА ПЕРСПЕКТИВИ

3.1. Покращення продуктивності та ефективності мережесервісів за допомогою WASM

Розвиток WebAssembly (WASM) відіграє вирішальну роль у підвищенні продуктивності та ефективності мережесервісів. WASM, як низькорівнева бінарна мова, забезпечує значне прискорення виконання коду порівняно з традиційним JavaScript, що є критично важливим для складних веб-додатків, які вимагають високої продуктивності. Ця технологія дозволяє виконувати в браузері код, написаний на таких мовах, як C, C++ чи Rust, що розкриває нові можливості для розробки потужних веб-додатків.

Використання WASM в мережесервісах не лише забезпечує вищу швидкість виконання, але й підвищує загальну ефективність застосунків, оскільки дозволяє оптимізувати використання ресурсів. Завдяки компактній бінарній формі, WASM-модулі завантажуються швидше, що знижує час відгуку для користувача. Також WASM підтримує багатопоточність

та паралельну обробку даних, що є важливим для додатків, що обробляють великі обсяги даних або вимагають складних обчислень.

Крім того, використання WASM дозволяє досягти вищого рівня безпеки. Вбудовані механізми ізоляції та обмеження доступу до системних ресурсів знижують ризик зловмисних атак і забезпечують більш безпечне середовище для виконання коду. Це особливо важливо в контексті розвитку хмарних технологій та IoT, де безпека даних є ключовим пріоритетом.

Окрім технічних переваг, WASM також впливає на розробку веб-додатків, оскільки дозволяє розробникам використовувати мови програмування з більш високою продуктивністю та контролем, ніж JavaScript. Це відкриває двері для нових підходів у розробці, дозволяючи створювати більш складні та високопродуктивні веб-додатки [6].

3.2. Інтеграція та універсальність застосунків з використанням WASI.

WASI, як ключовий компонент у веб-архітектурі, вносить значний вклад у інтеграцію та універсальність мережевих застосунків. Його здатність стандартизувати взаємодію WebAssembly додатків з системними ресурсами відкриває нові горизонти для розробників, дозволяючи їм створювати більш універсальні та легко інтегровані рішення. Використання WASI забезпечує уніфікований підхід до доступу до файлів, мережевих з'єднань та інших системних ресурсів, що суттєво спрощує процес розробки та розгортання застосунків у різноманітних середовищах. Це особливо важливо в контексті сучасних хмарних технологій та розподілених систем, де потреба в універсальних рішеннях, здатних легко інтегруватися в різні платформи та екосистеми, є вищою, ніж будь-коли.

Додатково, WASI сприяє універсальності застосунків, дозволяючи їм виконуватися на різних операційних системах без необхідності внесення змін у код. Це відкриває шлях для створення переносних додатків, що можуть працювати однаково ефективно як на серверах, так і на персональних комп'ютерах, мобільних пристроях або навіть в IoT пристроях. Така переносність та універсальність є критично важливими для розгортання складних мережевих систем, що вимагають високої гнучкості та масштабованості. Крім того, інтеграція застосунків, розроблених з використанням WASI, з існуючими системами та сервісами стає простішою та ефективнішою, оскільки WASI дозволяє виконувати код, оптимізований для різних середовищ, не втрачаючи при цьому на продуктивності чи функціональності.

Таким чином, WASI відіграє ключову роль у створенні більш гнучких та універсальних мережевих застосунків, що відповідають зростаючим вимогам ринку щодо інтеграції, масштабованості та ефективності. Його здатність адаптуватися до різних середовищ та платформ робить його ідеальним інструментом для розробки сучасних, високопродуктивних мережевих рішень, здатних задовольнити потреби як бізнесу, так і кінцевих користувачів [7].

3.3. Спрощення процесу розгортання за допомогою WAGI.

Роль WAGI (WebAssembly Gateway Interface) у спрощенні процесу розгортання веб-додатків є важливою інновацією, яка забезпечує розробникам потужні інструменти для більш ефективної роботи з WebAssembly. WAGI дозволяє легко інтегрувати WASM-модулі в мережеві застосунки, створюючи гладкий шлях для їх розгортання в різноманітних середовищах. Ця технологія практично видаляє бар'єри, пов'язані з розгортанням традиційних веб-додатків, забезпечуючи легкий та гнучкий механізм розгортання застосунків.

Завдяки WAGI, WebAssembly додатки можуть бути розгорнуті як ефективні веб-сервіси, які легко масштабуються та інтегруються з існуючими інфраструктурними компонентами та сервісами. Це означає, що розробники можуть швидко адаптувати та розповсюджувати свої додатки, використовуючи переваги високої продуктивності та безпеки, що надає WebAssembly, без потреби занурюватися в складні аспекти серверного програмування. WAGI

відкриває можливості для розгортання легковагових, але потужних веб-додатків, які відповідають сучасним вимогам до швидкості, ефективності та масштабованості.

Процес розгортання з WAGI стає значно простішим і більш універсальним, дозволяючи розробникам зосередитися на розробці бізнес-логіки, а не на технічних деталях розгортання. Це створює умови для більш швидкого та ефективного випуску продуктів на ринок, забезпечуючи розробникам перевагу в швидкісному та інноваційному цифровому середовищі. Таким чином, WAGI виступає не просто як інструмент для розгортання, але й як каталізатор для інновацій та розвитку в області мережевих застосунків, відкриваючи нові горизонти для веб-розробки та цифрової трансформації [7, 8].

4. ВИСНОВКИ

Аналізуючи розвиток і вплив технологій WASM, WASI, та WAGI на веб-розробку, ми бачимо їх значний потенціал для майбутнього. WebAssembly (WASM) підвищує продуктивність і безпеку веб-додатків, дозволяючи використовувати різні мови програмування для створення потужних веб-застосунків, сприяючи інноваційним підходам у веб-розробці. WASI розширює можливості WebAssembly, забезпечуючи доступ до системних ресурсів, що сприяє гнучкості розробки мережевих застосунків. WAGI спрощує розгортання WebAssembly додатків у веб-сервіси, знижуючи поріг входу для розробників.

Напрямок продовження дослідження є порівняння можливостей контейнеризації за допомогою Docker та WASI. Це дослідження дозволить оцінити ефективність і гнучкість обох технологій у контексті розгортання веб-додатків, а також їх вплив на безпеку, масштабованість та швидкість виконання. Зіставлення цих двох підходів до контейнеризації відкриває нові перспективи для розробки більш ефективних і безпечних мережевих застосунків, що є важливим для подальшого розвитку цифрових технологій.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. The .NET Tools Blog, "The Future of .NET with WASM". URL: <https://blog.jetbrains.com/dotnet/2022/12/15/the-future-of-net-with-wasm/> (дата звертання: 30.11.2023).
2. TheServerSide, "Why WebAssembly? Top 11 Wasm benefits". URL: <https://www.theserverside.com/tip/Why-WebAssembly-Top-Wasm-benefits> (дата звертання: 30.11.2023).
3. Red Hat, "Red Hat and WebAssembly". URL: <https://www.redhat.com/en/blog/red-hat-and-webassembly> (дата звертання: 30.11.2023).
4. Semaphore CI, "WebAssembly + Docker = Lightweight Containers". URL: <https://semaphoreci.com/blog/docker-wasm> (дата звертання: 30.11.2023).
5. Fermyon, "WebAssembly, WASI, and the Component Model". URL: <https://www.fermyon.com/blog/webassembly-wasi-and-the-component-model> (дата звертання: 30.11.2023).
6. WebAssembly System Interface (WASI). URL: <https://wasi.dev/> (дата звертання: 30.11.2023).
7. Deis Labs, "WAGI, the WebAssembly Gateway Interface". URL: <https://deislabs.io/posts/wagi-updates/> (дата звертання: 30.11.2023).
8. InfoWorld, "Understanding WAGI, the WebAssembly Gateway Interface". URL: <https://www.infoworld.com/article/3632865/understanding-wagi-the-webassembly-gateway-interface.html> (дата звертання: 30.11.2023).

ІНТЕЛЕКТУАЛЬНІ АГЕНТИ ДЛЯ СИСТЕМИ АВТОМАТИЗАЦІЇ УПРАВЛІННЯ БІЗНЕС-ПРОЦЕСАМИ У СФЕРІ ЛОГІСТИКИ

Вайнер Г.О.¹, Мухін В.Є.²

Національний технічний університет України «Київський політехнічний інститут
імені Ігоря Сікорського», Київ, Україна

¹ opalgeorgii@gmail.com, ² v_mukhin@i.ua

Робота має на меті покращити автоматизацію управління бізнес-процесами в логістиці за допомогою інтелектуальних агентів, забезпечуючи ефективніший розподіл ресурсів та прийняття рішень. Методи дослідження передбачають використання машинного навчання задля оптимізації процесів. Як результат буде побудована програма, яка зможе самостійно приймати найоптимальніше рішення для поставленої задачі. Новизна полягає в застосуванні інтелектуальних агентів для розв'язання конкретних завдань управління БП в логістиці, що раніше можливо не вирішувалися або вирішувалися менш ефективно. Практична значимість полягає у покращенні управління логістичними процесами, зменшенні часу і витрат реалізації для бізнесу.

Ключові слова: інтелектуальні агенти, автоматизація управління, бізнес-процеси, логістика, системи управління, інтелектуальні системи, технології логістики, інтеграція, дані логістики, ефективність, роботи з даними, автоматизація бізнес-процесів.

1. ВСТУП

У сучасному світі, де конкуренція на ринку надзвичайно жорстка, ефективне управління бізнес-процесами стає ключовим аспектом успіху для компаній. Інтелектуальні агенти, які базуються на передових алгоритмах штучного інтелекту, стають вирішальним елементом до оптимізації логістичних процесів.

Виникає необхідність ретельного вивчення і розробки інтелектуальних агентів, здатних адаптуватися до змінних умов, враховувати величезний потік даних, а також прогнозувати і реагувати на найрізноманітніші сценарії. Ця тема дозволяє зануритися у світ інновацій та високотехнологічних рішень для оптимізації логістичних бізнес-процесів [1].

Враховуючи динаміку ринкових змагань та постійно зростаючі вимоги споживачів, вивчення інтелектуальних агентів для автоматизації управління бізнес-процесами стає важливою стратегічною ініціативою. Ця тема розкриває можливості застосування передових технологій для оптимізації логістичних систем, що в свою чергу призводить до підвищення ефективності, зниження витрат і покращення якості обслуговування клієнтів. Вивчення інтелектуальних агентів у контексті логістики стає важливим етапом для компаній, які прагнуть бути попереду та забезпечити стійкість свого бізнесу в умовах постійних змін.

Розглядаючи роль інтелектуальних агентів у цьому контексті, можна відкрити нові можливості для підвищення ефективності, зниження витрат та забезпечення конкурентоспроможності в глобальному бізнес-середовищі [2].

2. ПОНЯТТЯ ІНТЕЛЕКТУАЛЬНОГО АГЕНТА ТА ЙОГО РОЛЬ В ЛОГІСТИЦІ

Інтелектуальний агент – це програмний об'єкт, який діє від імені користувача або іншого системного компонента, спроможний сприймати оточуючу інформацію, аналізувати її та приймати рішення на основі певних критеріїв. Інтелектуальні агенти зазвичай розробляються для вирішення конкретних завдань, використовуючи методи штучного інтелекту (машинного навчання, глибокого навчання, NLP (обробка природних мов) тощо), а також здатні взаємодіяти з іншими агентами та оточенням.

На рис. 1 показано простий приклад роботи інтелектуального агента:

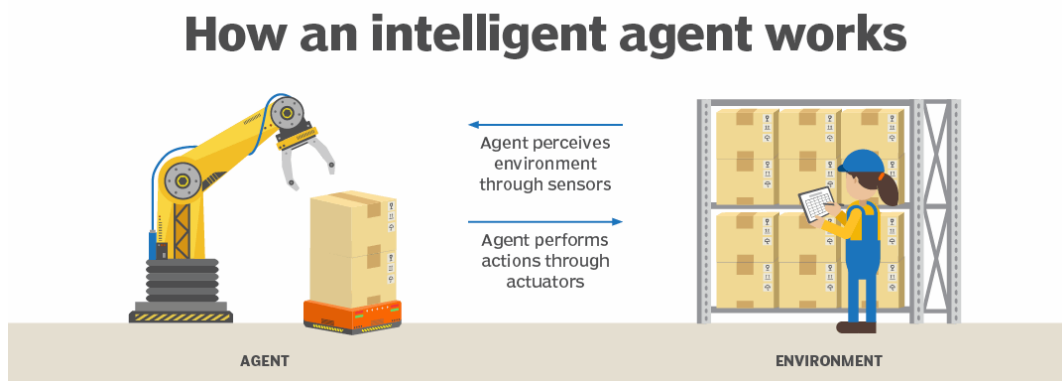


Рисунок 1. Простий приклад роботи інтелектуального агента

Пандемія COVID-19 викликала стрімку потребу для використання RPA (Robotic process automation) агентів. Програмне забезпечення RPA агентів було особливо цінним, коли багато підприємств відчували нестачу робочої сили, особливо враховуючи те, що логістичні підприємства зіткнулися зі зростаючим споживчим попитом і зростаючою кількістю замовлень через бум електронної комерції.

Популярність автоматизації процесів буде тільки зростати, і 90 відсотків великих світових компаній вже використовують RPA агентів [3, 4].

На (рис. 2) нижче можна побачити вражаюче зростання інвестицій у розробку ПЗ RPA у всьому світі за останні п'ять років: з 500 мільйонів доларів США у 2017 році до майже 2 мільярдів доларів США у 2021 році [5].

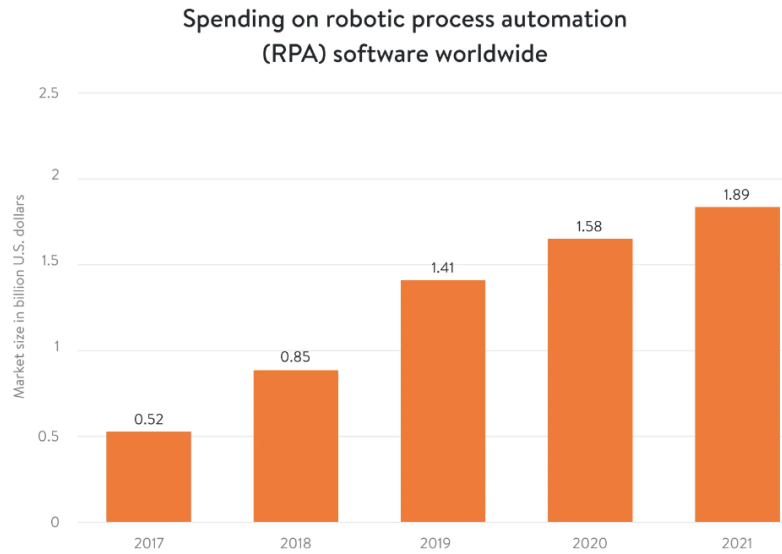


Рисунок 2. Світові витрати на ПЗ для інтелектуальних агентів

3. АНАЛІЗ ГОТОВИХ АГЕНТІВ ДЛЯ АВТОМАТИЗАЦІЇ БІЗНЕС-ПРОЦЕСІВ, ЇХ ПЕРЕВАГИ ТА НЕДОЛІКИ

Загально-готові рішення інтелектуальних агентів є варіантом, якщо ви хочете випробувати цю технологію та перевірити, чи вона відповідає вашим потребам.

5 головних переваг впровадження RPA агентів в вашу компанію (рис. 3):

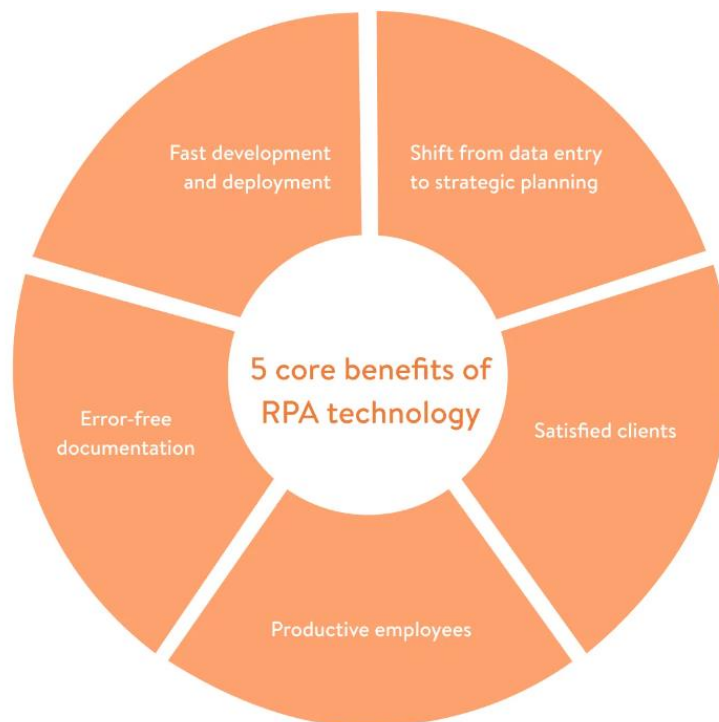


Рисунок 3. Головні переваги впровадження інтелектуальних агентів

Найпопулярнішими гравцями у світі RPA (Robotic Process Automation) є UiPath, BluePrism, Automation Anywhere і Kofax. Але є також деякі рішення RPA лише для сфери логістики. Проаналізуємо постачальників ПЗ для автоматизації БП у сфері логістики:

Shipamax

Shipamax пропонує програмне забезпечення для автоматизації БП з підтримкою ШІ для автоматизації введення даних у системи TMS (Transportation Management System) або ERP (Enterprise Resource Planning). Наприклад, програмне рішення Shipamax може отримувати дані з електронних листів і миттєво передавати їх до ERP [6, 7].

Переваги:

- Може витягувати всі типи даних із електронних листів, вкладень електронних листів, сканованих зображень і фотографій
- Здатний обробляти великі обсяги спеціалізованих логістичних документів
- Підвищена продуктивність завдяки інтеграції технологій AI і ML

Недоліки:

- Засоби автоматизації лише для логістичних процесів, пов'язаних із введенням даних
- Потрібен власний IT-відділ для процесу інтеграції з внутрішніми системами, які Shipamax не підтримує
- Покриває потреби лише експедиторів (той, хто супроводжує вантажі під час доставки у місце призначення, відповідає за них та стежить за збереженням цілісності вантажа під час транспортування)
- Немає безкоштовної пробної версії та є лише можливість замовити безкоштовну демо-версію програми

RPA Labs

RPA Labs – RPA агент з повним набором послуг доставки та логістики. Компанія пропонує три види ботів: боти для відповідей (чат-боти), боти для документів і боти для робочого процесу. Кожен тип бота має підкатегорії та може бути налаштований під потреби конкретної логістичної компанії. RPA Labs також обслуговує різні галузі логістики, такі як вантажні перевезення, інтермодальні перевезення (перевезення в контейнерах), 3PL і експедирування вантажів [8][9].

Процес обробки замовлення за допомогою RPA агента (бота) (рис. 4):

Processing an order with an RPA bot:



Рисунок 4. Обробка замовлення ботом

Переваги:

- Широкі функціональні можливості, включаючи введення даних, відстеження відправлень і виставлення рахунків

- Кожен програмний бот має вбудовану бібліотеку з логістичною термінологією для швидкої обробки конкретних даних ланцюга поставок
- Каталог програмних ботів для різноманітних цілей логістики
- Для обслуговування ботів не потрібна внутрішня технічна підтримка
- Оснащений технологіями AI та ML для виконання більш інтелектуальних завдань

Недоліки:

- Лише попередньо створені та навчені програмні боти з обмеженими можливостями налаштування
- Відсутність можливості безкоштовної пробної версії програмного забезпечення (ціна рішення залежить від кількості програмних ботів)
- Каталог ботів пропонує більше 20 ботів, але цього може бути недостатньо, щоб задовольнити всі потреби середніх і великих логістичних організацій

4. МЕТОДИ КЕРОВАНОГО МАШИННОГО НАВЧАННЯ (SUPERVISED LEARNING)

Регресія – це техніка, яка використовується для прогнозування значення (залежних) змінних на основі однієї або кількох (незалежних) змінних.

Найбільш часто використовувані методи регресії: лінійна регресія та логістична регресія. Обговоримо теорію, що лежить в основі цих двох методів [10].

Лінійна регресія

Лінійна регресія – це метод статистичної регресії, який використовується для прогнозного аналізу та показує зв'язок між безперервними змінними. Лінійна регресія показує лінійний зв'язок між незалежною змінною (вісь X) і залежною змінною (вісь Y), тому називається лінійною регресією. Якщо є одна вхідна змінна (x), така лінійна регресія називається простою лінійною регресією. А якщо є більше однієї вхідної змінної, така лінійна регресія називається множинною лінійною регресією. Модель лінійної регресії дає похилу пряму лінію, що описує зв'язок між змінними (рис. 5).

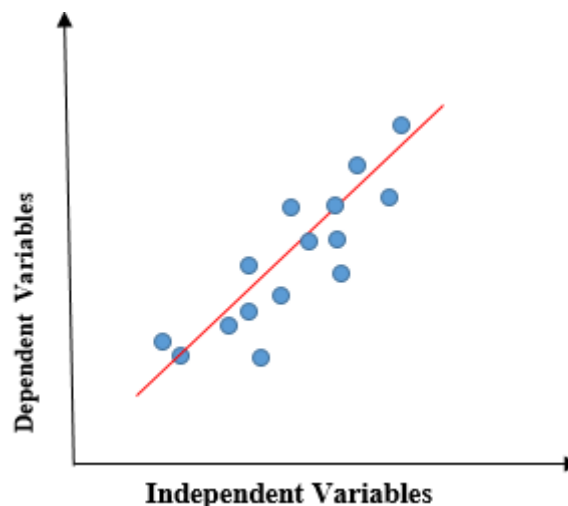


Рисунок 5. Принцип роботи лінійної регресії

Наведений вище графік представляє лінійну залежність між залежною змінною та незалежними змінними. Коли значення x (незалежна змінна) збільшується, значення y (залежна змінна) також зростає. На основі заданих точок даних ми намагаємося побудувати лінію (червона), яка найкраще моделює точки.

Для обчислення лінії лінійної регресії використовується традиційна лінійна функція.

$$y = mx + b \rightarrow y = a_0 + a_1x,$$

де y – залежна змінна, x – незалежна змінна, a_0 – зсув лінії по y , a_1 – коефіцієнт лінійної регресії.

Необхідність лінійної регресії

Як згадувалося вище, лінійна регресія оцінює зв'язок між залежною змінною та незалежною змінною. Давайте зрозуміємо це на простому прикладі:

Припустимо, що ви керуєте логістичною компанією і хочете передбачити час доставки вантажу на основі відстані між двома містами. У цьому випадку, відстань між містами є незалежною змінною, оскільки вона не залежить від часу доставки, і може бути виміряна безпосередньо. Час доставки вантажу є залежною змінною, оскільки він залежить від відстані та інших факторів, таких як тип транспорту, дорожні умови, режим руху тощо.

Використовуючи дані про відстань і час доставки з попередніх вантажів, ви можете побудувати модель лінійної регресії, яка допоможе вам передбачити час доставки вантажу для майбутніх вантажів, враховуючи відстань між містами. Це допоможе оптимізувати вашу логістичну діяльність, планувати маршрути та забезпечувати клієнтів більш точними термінами доставки.

У ході дослідження буде розроблено інтелектуальних агентів, що будуть взаємодіяти з бізнес-процесами логістичної сфери. Реалізовані алгоритми дозволяють оптимізувати розподіл ресурсів, вирішувати завдання координації та адаптації до змін в реальному часі. Застосування Python у розробці системи дозволяє легко розширювати та модернізувати функціонал для вирішення нових викликів у майбутньому [11].

Залишаються відкриті питання і завдання для подальших досліджень. Особлива увага приділяється системам управління, які повинні бути готові до непередбачуваних змін у логістичному середовищі, таких як підвищення обсягів вантажів, зміни в законодавстві або впровадження нових технологій. Дослідження мають вирішити питання адаптивності інтелектуальних агентів до динамічних змін і забезпечити стабільну ефективність системи в реальному часі [12, 13].

5. ВИСНОВКИ

Сучасні системи автоматизації в логістиці великою мірою спрощують та оптимізують бізнес-процеси. Однак, вони також стикаються з викликами, такими як висока вартість впровадження, потреба у великих інвестиціях в ІТ-інфраструктуру та інтеграцію, а також питаннями безпеки та конфіденційності даних. Зараз ключовим напрямком є інтеграція технологій штучного інтелекту та аналізу даних для створення інтелектуальних систем, які можуть ефективно працювати в умовах логістичних викликів та змін.

Загальний успіх впровадження автоматизації управління бізнес-процесами в логістиці залежить від того, наскільки вдало вдасться компенсувати недоліки та використовувати переваги цих систем в конкретних умовах та завданнях компанії.

Інтелектуальні агенти знаходяться в авангарді штучного інтелекту, дозволяючи машинам взаємодіяти зі світом і виконувати завдання автономно. Їхня здатність сприймати, розуміти та адаптуватися робить їх безцінними в різних галузях промисловості та застосуваннях. У міру того, як технологія штучного інтелекту продовжує розвиватися, інтелектуальні агенти стануть ще більш досконалими та інтегрованими в наше повсякденне життя, змінюючи спосіб роботи, спілкування та орієнтування у світі. Однак із цією зростаючою інтеграцією виникає потреба в етичних міркуваннях і відповідальному розвитку, щоб забезпечити використання розумних агентів на благо людства.

Зі збільшенням технологічного прогресу буде посилений розвиток інтелектуальних агентів. Надалі це призведе до створення складних пристроїв на основі штучного інтелекту, які вирішуватимуть поточні глобальні виклики.

Важливо мати на увазі, що для впровадження проектів машинного навчання компаніям необхідно мати можливість адаптуватися до цифрового середовища, що вимагає великих зусиль з боку компаній для оцифрування інформації, яку вони мають, формати, шаблони та інформацію, яку вони отримують.

Підсумовуючи, зараз компанії шукають ідеального союзника в технологіях для досягнення своїх цілей, спрямованих на ефективність, продуктивність і прибутковість. І беручи до уваги ситуацію, яка склалася в усьому світі, це ідеальний час для інвестування в технології автоматизації процесів задля зменшення витрат.

Вивчаючи роботу нейронних мереж та Machine Learning у цілому та використовуючи всі ці принципи компанії по усьому світу можуть автоматизувати свої бізнес-процеси.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Intelligent automation as a new era of RPA: веб-сайт. URL: <https://innovecs.com/blog/intelligent-automation/> (дата звернення: 10.10.2023).

2. Robotic process automation for logistics: let bots do all the monotonous work: веб-сайт. URL: <https://yalantis.com/blog/rpa-adoption-benefits-for-logistics/> (дата звернення: 10.10.2023).

3. Information Technologies of Formation of Intellectual Decision-Making Strategies under Conditions of Cognitive Failures / Lubomir Sikora, Natalia Lysa, Olga Fedevych, Mariya Navytka, Rostislav Tkachuk, Dronyuk Ivanna / Львів, LPNU 2020.

4. Improving data collection and processing efficiency by using hierarchies of intelligent agents in multi-agent systems / О. Е. Bezborodova / Penza, 2021.

5. Agents in Artificial Intelligence: веб-сайт. URL: <https://www.geeksforgeeks.org/agents-artificial-intelligence/> (дата звернення: 12.10.2023).

6. An Integrated Structured Analysis Approach to Intelligent Agent Communication / Hans Akkermans, Rune Gustavsson, Fredrik Ygge / Ronneby, Sweden, 1998.

7. Intelligent Buildings: Foundations for Intelligent Physical Agents / Francisco J. Mora Lizan, Carlos Rizo Maestre / Spain, 2017.

8. Artificial Intelligence in Logistics: збірка матеріалів компаніями DHL та IBM / Troisdorf, Germany, 2018. 1-45 с.

9. Intelligent Agents in Artificial Intelligence: веб-сайт. URL: <https://medium.com/@evertongomede/intelligent-agents-in-artificial-intelligence-6b5f4153ad72> (дата звернення: 13.10.2023).

10. Intelligent agents in Artificial Intelligence: веб-сайт. URL: <https://www.section.io/engineering-education/intelligent-agents-in-ai/#:~:text=The%20structure%20of%20intelligent%20agents,agent%20executes%20on%20this%20machinery> (дата звернення: 14.10.2023).

11. Artificial Intelligence in Logistics: збірка матеріалів компанією SSI Schäfer IT Solutions / Austria, 2018. 1-24 с.

12. Machine Learning – Fundamentals: веб-сайт. URL: <https://towardsdatascience.com/machine-learning-basics-part-1-a36d38c7916> (дата звернення: 15.10.2023).

13. Linear Regression in machine learning: веб-сайт. URL: <https://www.analyticsvidhya.com/blog/2021/06/linear-regression-in-machine-learning/> (дата звернення: 17.10.2023).

ІНСТРУМЕНТИ ТЕСТУВАННЯ ЦИФРОВОЇ ДОСТУПНОСТІ ВЕБ-САЙТІВ

Волосожар Д.В.¹, Кисельов Г.Д.²

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

¹ dasha1999.v2k@gmail.com, ² g.kyselov@gmail.com [0000-0000-0000-0000]

У даному дослідженні акцентується увага на підходах до вирішення завдань, пов'язаних з тестуванням цифрової доступності веб-сайтів. У роботі буде представлена математична модель цієї задачі, а також детально розглянуті різноманітні методи, які застосовуються для оцінки та поліпшення доступності веб-контенту.

Ключові слова: цифрова доступність, веб-сайт, тестування доступності, інструменти тестування, WCAG.

1. ВСТУП

У сучасному світі, де інтернет став неодмінною частиною нашого повсякденного життя, доступність веб-сайтів перестала бути просто бажаною характеристикою – вона стала необхідністю. Цифрова доступність означає, що веб-сайти, інструменти та технології розроблені таким чином, щоб ними могли користуватися всі люди, включаючи тих, хто має обмеження за здоров'ям, такі як порушення зору, слуху, руху або когнітивні обмеження. У зв'язку з цим, тестування доступності стає ключовою частиною процесу розробки веб-сайтів.

Основна мета цифрової доступності – забезпечити рівний доступ і рівні можливості для всіх користувачів інтернету. Відсутність доступності може не тільки ізолювати частину населення, але й становити юридичні ризики для організацій. Таким чином, розуміння і використання ефективних інструментів тестування доступності стає критично важливим для розробників веб-сайтів і власників бізнесу.

2. ОСНОВНІ ПРИНЦИПИ ЦИФРОВОЇ ДОСТУПНОСТІ

Цифрова доступність не є одномірною концепцією; вона охоплює широкий спектр практик і стандартів, спрямованих на забезпечення рівного доступу до веб-контенту для всіх користувачів. Найбільш значущим з цих стандартів є WCAG (Web Content Accessibility Guidelines), які є частиною ініціативи WAI (Web Accessibility Initiative) від W3C (World Wide Web Consortium).

WCAG: Основа для Доступності

WCAG визначає, як веб-контент повинен бути доступним для людей з різними обмеженнями. Ці настанови поділяються на три рівні відповідності: А (найнижчий), АА та ААА (найвищий). Вони охоплюють чотири основні принципи доступності:

1. **Сприйняття (Perceivable):** Інформація та компоненти інтерфейсу користувача повинні бути представлені таким чином, щоб їх могли сприймати всі користувачі. Наприклад, це означає надання текстових альтернатив для немовних зображень і забезпечення того, щоб аудіо- і відеоконтент був доступний у вигляді тексту.

2. **Операбельність (Operable):** Компоненти інтерфейсу та навігація повинні бути такими, щоб їх могли використовувати всі користувачі. Це включає забезпечення

функціональності за допомогою клавіатури, надання достатнього часу для читання та використання контенту, а також допомогу в уникненні і корекції помилок.

3. **Зрозумілість (Understandable):** Інформація та управління інтерфейсом користувача повинні бути зрозумілими. Це означає, що веб-сайти повинні мати чітку та зрозумілу мову, а також допомагати користувачам уникати та виправляти помилки.

4. **Надійність (Robust):** Контент повинен бути достатньо стійким, щоб міг надійно інтерпретуватися широким спектром користувацьких агентів, включаючи асистивні технології.

Застосування WCAG у процесі розробки веб-сайтів вимагає уважного планування та розуміння цілей доступності. Розробники повинні враховувати ці критерії на кожному етапі проектування та розробки, від структури сторінки та дизайну до вибору контенту та інтерактивних елементів [1].

3. КАТЕГОРІЇ ІНСТРУМЕНТІВ ТЕСТУВАННЯ ДОСТУПНОСТІ

Інструменти тестування доступності веб-сайтів можна умовно розділити на декілька категорій в залежності від їх функціональності та способу застосування. Кожна категорія відіграє унікальну роль у виявленні та виправленні проблем доступності.

Автоматизовані інструменти тестування

Автоматизоване тестування використовує програмне забезпечення для виявлення проблем доступності на веб-сторінках. Ці інструменти здатні швидко сканувати код сторінки і вказувати на потенційні проблеми, такі як відсутність альтернативних текстів для зображень, недостатній контраст кольорів, відсутність міток форм тощо.

- **Переваги:** Швидке виявлення очевидних проблем, можливість покриття великих обсягів контенту.
- **Обмеження:** Не може повністю замінити ручне тестування, оскільки деякі аспекти доступності вимагають суб'єктивної оцінки.

Ручне тестування

Ручне тестування включає в себе перевірку веб-сторінок людиною на відповідність стандартам доступності. Це може включати в себе використання сайту з позиції осіб з обмеженими можливостями, наприклад, за допомогою екранних читачів.

- **Переваги:** Дозволяє зрозуміти реальний досвід користувачів, виявляє проблеми, які не можуть бути знайдені автоматично.
- **Обмеження:** Часомісткий процес, вимагає високої кваліфікації тестувальників.

Використання екранних читачів

Екранні читачі – це спеціалізоване програмне забезпечення, яке перетворює текст та інші елементи веб-сторінок у мовлення або шрифт Брайля. Тестування за допомогою екранних читачів є важливим для забезпечення доступності веб-сайтів для користувачів з порушеннями зору.

- **Переваги:** Дає реальне уявлення про те, як користувачі з порушеннями зору взаємодіють з веб-сайтом.
- **Обмеження:** Вимагає спеціальних знань для ефективного використання.

4. ПОПУЛЯРНІ ІНСТРУМЕНТИ ТЕСТУВАННЯ ДОСТУПНОСТІ

Один з найвідоміших інструментів – це AXE Accessibility Checker. Це браузерне розширення, яке дозволяє розробникам легко перевіряти веб-сторінки на відповідність стандартам доступності. AXE може швидко ідентифікувати поширені проблеми, такі як відсутність альтернативних текстів для зображень або недостатній контраст кольорів.

Ще один важливий інструмент у сфері доступності – JAWS (Job Access With Speech), екранний читач, що широко використовується людьми з вадами зору. Використання JAWS для тестування дозволяє розробникам краще зрозуміти, як їхній веб-сайт буде сприйматися користувачами, що покладаються на аудіальне відтворення контенту.

WAVE (Web Accessibility Evaluation Tool) є ще одним корисним ресурсом. Це інструмент, що надає візуальне представлення проблем доступності на веб-сторінці. Він доступний як онлайн-сервіс та як розширення браузера і допомагає виявляти проблеми, які можуть бути неочевидними при звичайному огляді сторінки.

Google Lighthouse, ще один потужний інструмент, інтегрований у Chrome DevTools, надає розробникам змогу оцінювати якість веб-сторінок, включаючи аспекти доступності. Цей інструмент автоматично аналізує веб-сторінки і надає звіти з рекомендаціями щодо покращення.

Аналізатори контрастності кольорів важливі для переконання, що текст на веб-сайті читабельний для всіх користувачів, включаючи тих, хто має порушення зору. Ці інструменти дозволяють розробникам перевіряти, чи текст відповідає стандартам WCAG щодо контрастності кольорів.

Тепон.іо - це онлайн-інструмент, який надає розробникам можливість тестувати веб-сайти та веб-додатки на предмет відповідності стандартам доступності. Він надає звіти, які можуть бути інтегровані в процес розробки для забезпечення безперервного дотримання стандартів доступності.

Кожен з цих інструментів відіграє свою роль у забезпеченні того, щоб веб-сайти були більш доступними для всіх користувачів, незалежно від їхніх фізичних можливостей чи обмежень. Вибір правильного інструменту тестування і його ефективне використання може значно вплинути на рівень доступності веб-сайту [2].

5. ІНТЕГРАЦІЯ ТЕСТУВАННЯ ДОСТУПНОСТІ У РОЗРОБКУ ВЕБ-САЙТІВ

Важливість інтеграції тестування доступності у процес розробки веб-сайтів не може бути переоціненою. Це не лише про відповідність нормативним стандартам, але й про створення інклюзивного досвіду користування, який враховує потреби всіх користувачів, включаючи людей з обмеженими можливостями.

Успішне впровадження доступності починається з раннього етапу проектування. Вже на стадії дизайну необхідно мислити про такі аспекти, як візуальна контрастність, чіткість структури та забезпечення простоти навігації. Дизайнери та розробники повинні співпрацювати, щоб забезпечити, що інтерфейс є зрозумілим і доступним для всіх користувачів.

Під час розробки веб-сайтів критично важливо використовувати семантично коректний HTML, доступні форми, а також ARIA-мітки, щоб забезпечити, що сайт може бути легко використаний з екранними читачами та іншими асистивними технологіями. Тестування доступності повинно бути інтегрованим у процес розробки, а не вважатися додатковою чи

вторинною задачею. Це дозволяє виявляти та виправляти проблеми доступності на ранніх стадіях, замість того, щоб намагатися вирішити їх після запуску сайту.

Залучення кінцевих користувачів для тестування та зворотного зв'язку є ще одним ключовим елементом у процесі забезпечення доступності. Отримання відгуків від реальних людей, які можуть мати різні потреби та обмеження, допомагає уточнити та поліпшити веб-досвід для більш широкої аудиторії.

Інтеграція доступності у веб-розробку не лише покращує досвід користувачів з обмеженими можливостями, але й підвищує загальну якість та функціональність веб-сайтів, роблячи їх більш доступними та приємними для ширшого кола користувачів [3].

6. ВИСНОВОК

Огляд різних інструментів тестування доступності та їх застосування у веб-розробці підкреслює необхідність інклюзивного підходу до створення веб-сайтів. Доступність не є доповненням до веб-дизайну чи розробки; це основний аспект, що вимагає уваги з самого початку проекту та на всіх його етапах. Чи то використання автоматизованих інструментів, таких як AXE та Google Lighthouse, чи ручне тестування з допомогою екранних читачів, таких як JAWS, кожен інструмент відіграє свою роль у виявленні та виправленні проблем доступності. Забезпечення доступності веб-сайтів є не просто відповідністю нормативним вимогам, а й виявом поваги та уваги до потреб усіх користувачів. Це відкриває двері до створення більш інклюзивного цифрового середовища. У майбутньому, з розвитком технологій і зростаючим усвідомленням інклюзивності, можна очікувати подальшого розвитку інструментів тестування доступності та методів їх застосування.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Генрі, С. Л. "Основи веб-доступності". Ініціатива веб-доступності (WAI), 2021. URL: <https://www.w3.org/WAI/fundamentals/accessibility-intro/>. (дата звернення 18.11)
2. Колдвелл, Б., Купер, М., Рейд, Л. Г., і Вандерхайден, Г. "Рекомендації щодо доступності веб-контенту (WCAG) 2.1". W3C, 2018. URL: <https://www.w3.org/TR/WCAG21/> (дата звернення 20.11)
3. Проект A11Y. "A11Y Стильовий посібник". A11Y Project, 2020. URL: <https://a11y-style-guide.com/>. (дата звернення 28.11)

ГАРАНТУВАННЯ УНІКАЛЬНОЇ ДОСТАВКИ ПОВІДОМЛЕНЬ В РОЗПОДІЛЕНИХ СИСТЕМАХ

Гапонюк М.О.¹, Письменний І.О.²

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Інститут прикладного системного аналізу, Київ, Україна

¹ haroniuk.maksym@iit.kpi.ua, ² ihor.pismennyu@gmail.com

Концепція гарантії доставки повідомлення «точно один раз» є ключовим принципом у розподілених системах і чергах подій. Це означає, що повідомлення доставляється лише один раз і не більше, що має вирішальне значення в багатьох системах, де дублікати повідомлень можуть спричинити проблеми або призвести до неправильної обробки подій.

Ключові слова: розподілені системи, черги подій, доставка повідомлень.

1. ВСТУП

У розподілених системах, компоненти яких розміщені по різних мережах або місцях, підтримка цілісності даних і надійності системи має вирішальне значення. Черги подій є інструментами для зберігання та обміну потоковими даними або подіями між різними компонентами розподіленої системи [1]. Під час побудови розподілених систем з використанням черг подій, невід'ємною складовою надійної системи є гарантія, що жодне повідомлення не буде втрачено чи передано більше одного разу, навіть у разі несправності. Це спрощує проектування будь-яких розподілених систем. EOD (Exactly-Once Delivery) має сенс інтуїтивно: ми не хочемо повторювати або втрачати дані чи дії.

2. СТРАТЕГІЇ ДОСЯГНЕННЯ УНІКАЛЬНОЇ ДОСТАВКИ

В розподілених системах її компоненти можуть перестати працювати незалежно один від одного. Причини збоїв можуть бути найрізноманітнішими. Класична архітектура розподіленої системи [2], яка використовує черги подій, складається з трьох частин: виробник повідомлень (producer), черга подій (message broker), та споживач повідомлень (consumer).

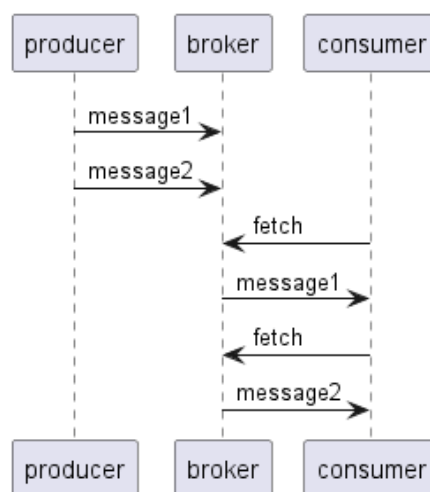


Рисунок 1. Приклад простої системи з використанням черги подій

У світі мікросервісів збої систем та мережі трапляються досить часто. Це призводить до того, що система в цілому може втратити консистентність. Існує чимало прикладів, де це не є критичною помилкою, наприклад, системи зі збору статистики відвідувань сайтів. В разі втрати декількох повідомлень в загальному - статистика буде вірною. Проте для більшості бізнес-процесів такий стан є недопустимим. Помилки можуть траплятись на різних етапах процесу передачі, збереження та обробки подій.

На рисунку 2а) зображено послідовність подій, при якій друге повідомлення не дійшло до споживача через помилку мережі. В такому випадку ми втрачаємо повідомлення. Проте ми можемо бути впевнені, що повідомлення буде надіслане не більше, ніж один раз. Даний підхід корисний, коли повторна обробка повідомлень є недопустимою для системи. мінімум оди раз

На рисунку 2б) зображено послідовність подій, у якій в разі відсутності підтвердження про отримання ми надсилаєм повідомлення ще раз. Даний підхід гарантує, що повідомлення точно буде доставлено до споживача. до того, що одне повідомлення буде надіслане і оброблене двічі.

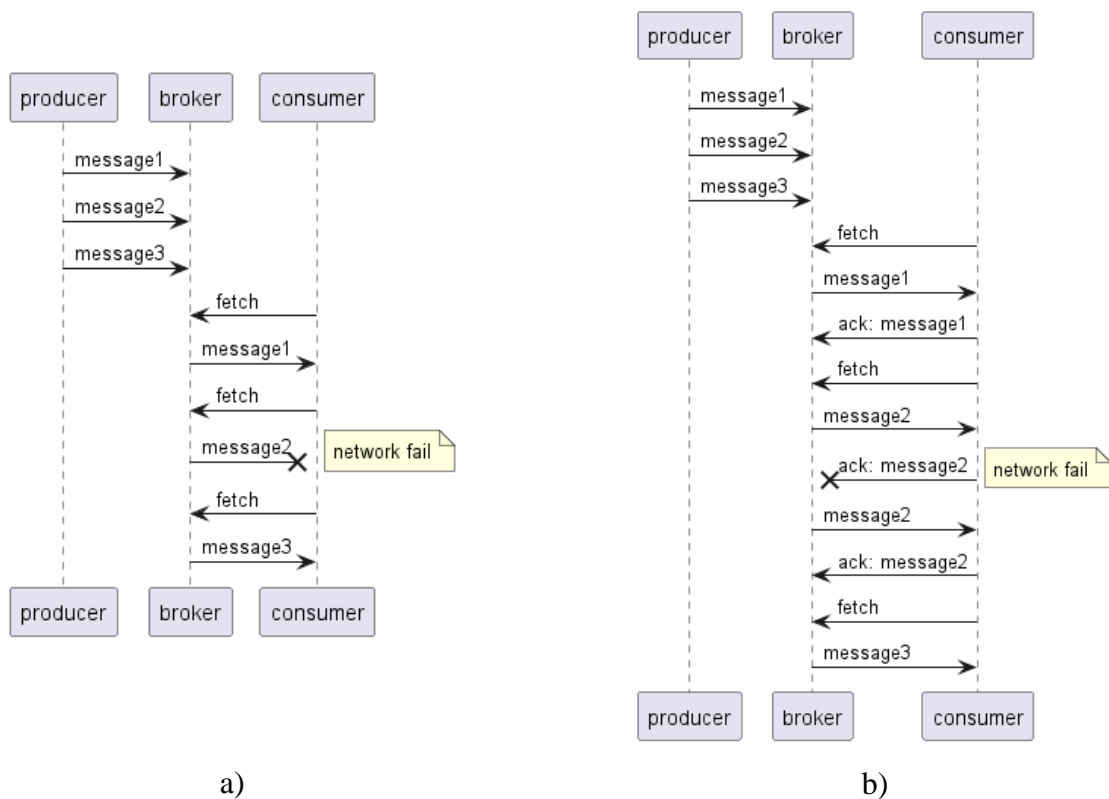


Рисунок 2. Приклади а) семантика «не більше одного разу», б) семантика «мінімум один раз»

Дублювання повідомлень нечасто трапляється в синхронних середовищах. Видавець може публікувати лише один набір даних за раз; публікація наступного набору даних лише після того, як буде отримано попередній, щоб гарантувати відсутність дублювання даних у брокері. На стороні підписника ми отримуємо один набір даних за раз. Тому не буде жодних «race conditions» [3], коли інформація отримується кількома підписниками.

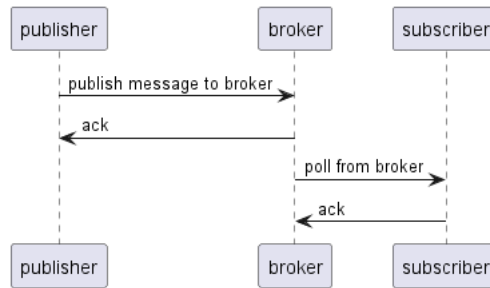


Рисунок 3. Синхронне середовище

У середовищі кількох видавців ми можемо бачити, як дублікати даних можуть виникати в системі повідомлень. Нехай кілька видавців публікують у черзі повідомлень. Якщо брокеру повідомлень не вдається надіслати підтвердження одному з видавців, оскільки воно було недоступне, тоді брокер повторить повідомлення кілька разів, щоб уникнути втрати повідомлення.

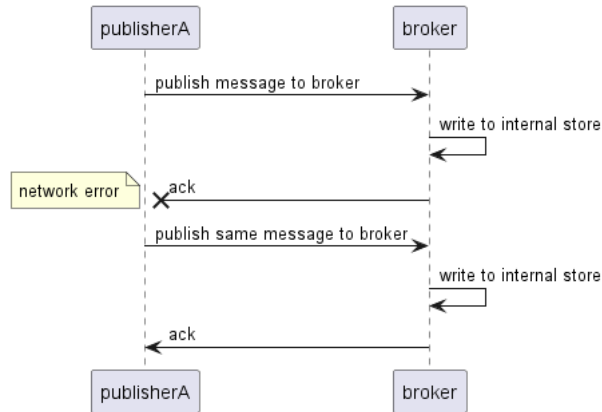


Рисунок 4. Помилка доставки підтвердження до видавця в асинхронному середовищі

Один із способів вирішити це – створити ідентифікатор кореляції [4]. Ідентифікатор кореляції буде ідентифікатором, який ідентифікує його ключ ідемпотентності в повідомленні. Тому, якщо система повідомлень отримала той самий ідентифікатор кореляції, вона знає, що він уже був отриманий раніше. Наприклад, Amazon SQS має ідентифікатор де дуплікації в черзі FIFO, який служить ідентифікатором кореляції. Однак він має час закінчення 5 хвилин. Якщо видавець надсилає друге повідомлення через 5 хвилин, брокер повідомлень не може гарантувати, що він відхилить повторне повідомлення.

З іншого боку, одноразової доставки майже неможливо досягти для кількох споживачів. Якщо брокер повідомлень видаляє повідомлення одразу після того, як підписник його отримав, існує ймовірність збою ще до того, як споживач обробить дані. У цьому випадку система не має знову доставляти повідомлення.

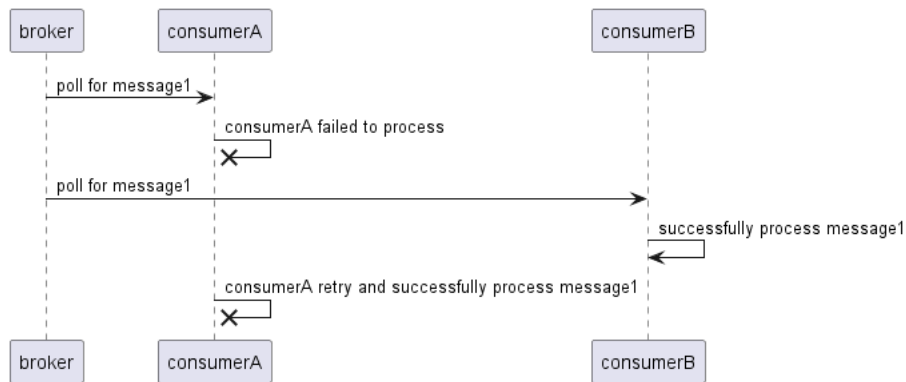


Рисунок 5. Повторна обробка повідомлення в асинхронному середовищі

Отже, можливе отримання повторюваних повідомлень точно після обробки. Розглянемо один сценарій: споживач А отримав повідомлення від брокера. Безпосередньо перед тим, як споживач А збирається видалити повідомлення, споживач В опитує повідомлення брокера. Отже, споживач В матиме те саме повідомлення, що й споживач А.

Одним з рішень [4] є створення тривалого сховища на стороні одержувача. Споживач може записати вхідне повідомлення на довговічне зберігання під час кожного опитування. Після успішного запису в пам'ять він видалить чергу з черги повідомлень. Надійне зберігання захистить від дублювання значень. Тим часом для обробки цих повідомлень буде виконуватись інший асинхронний фоновий потік обробки повідомлень. Ця концепція схожа на те, як різні системи баз даних використовують журнал попереднього запису для зберігання комітів.

Коли ми визначали обробку повідомлень, ключовим поняттям, яке ми слід використовувати – тривале збереження побічних ефектів. Ці побічні ефекти мають два види. По-перше, вони включають зміни в сховищі даних, що використовується процесом отримання для збереження свого стану. По-друге, побічними ефектами є також усі повідомлення, надіслані в результаті обробки вхідних повідомлень.

Щоб гарантувати одноразову обробку, ми можемо використовувати транзакції, які включають три операції:

- вилучити повідомлення з черги;
- змінити стан програми;
- поставити отримані повідомлення в чергу;
- видалити оброблене повідомлення з черги введення.

Транзакції гарантують, що включені операції є атомарними та довговічними. Це означає, що або всі три операції завершені, або жодна. Крім того, щойно транзакція буде прийнята системою, система гарантує, що вона не зникне в майбутньому. Транзакції реалізуються за допомогою концепції журналу – впорядкованої послідовності операцій, що зберігається на диску. Коли транзакція надсилається, її опис спочатку записується в журнал. Лише після того, як журнал стане довговічним (наприклад, шляхом запису контрольної суми та забезпечення очищення всіх буферів запису), операції фактично застосовуються до даних.

3. ВИСНОВКИ

Системи, які використовують розподілені черги подій повинні забезпечувати надійність доставки повідомлень у розподілених системах, в яких можуть виникнути різні ситуації, що ускладнюють процес доставки. Серед усіх складнощів, можна виділити наступні основні аспекти:

- гарантії доставки;

- передача повідомлень через ненадійні мережі, де можуть статися збої або втрати;
- механізми для повторного надсилання повідомлень у випадку невдачі першої спроби;
- можливість підтвердження, що повідомлення було успішно доставлено та оброблено;
- уникнення дублювання повідомлень в системі;
- забезпечення консистентності між різними частинами системи та співпраця між ними для досягнення консистентності системи.

Виявляється, що доказ неможливості доставки повідомлень один раз (на прикладі проблеми двох генералів [5]) базується на припущенні про асинхронну мережу, в якій повідомлення можуть затримуватися як завгодно довго. І навіть за цих припущень неможливість означає, що немає алгоритму, який завжди міг би гарантувати прогрес у досягненні консенсусу. На щастя, реальні мережі не є асинхронними. Вони більше схожі на напівасинхронні мережі, а це означає, що повідомлення доставляються з часом. Цієї гарантії достатньо для того, щоб низка консенсусних алгоритмів була доведена надійністю [4].

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. «What are Message Brokers? | IBM». Дата звернення: 25, Жовтень 2023. [Online]. Доступний у: <https://www.ibm.com/topics/message-brokers>
2. «Introduction to Message Brokers. What is a message broker? | by Hasitha Subhashana | Medium». Дата звернення: 25, Жовтень 2023. [Online]. Доступний у: <https://hasithas.medium.com/introduction-to-message-brokers-c4177d2a9fe3>
3. «Understanding Stream Processing And Apache Kafka | by Cory Maklin | Towards Data Science». Дата звернення: 29, Листопад 2023. [Online]. Доступний у: <https://towardsdatascience.com/understanding-stream-processing-and-apache-kafka-5610bc2d6fa3>
4. «Can Exactly Once Processing Queue in Real Systems even Exist? | edward-huang.com». Дата звернення: 25, Жовтень 2023. [Online]. Доступний у: <https://edward-huang.com/distributed-system/architecture/cloud/2021/02/15/can-exactly-once-processing-queue-in-real-systems-even-exist/>
5. «Two Generals' Problem. The Two Generals' Problem is an... | by Brian Kim | Medium». Дата звернення: 25, Жовтень 2023. [Online]. Доступний у: <https://brianjckim.medium.com/two-generals-problem-5eadb3f395ed>

РЕЗЕРВНЕ КОПІЮВАННЯ ДАНИХ ЯК ПОСЛУГА

Головін Б.О.¹, Булах Б.В.²

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Інститут прикладного системного аналізу, Київ, Україна

¹ b.holovin@gmail.com, ² bogdan.bulahk@gmail.com

Стаття присвячена дослідженню та аналізу резервного копіювання даних як послуги (Backup as a service, BaaS). У роботі описуються методи та стратегії використання таких типів додатків, а також проводиться порівняльний аналіз популярних платформ цього типу. Результати проведеного дослідження дозволили окреслити перелік переваг від використання резервного копіювання даних як послуги у забезпеченні безпеки та продуктивності при розробці додатків, підкреслюючи його наукову новизну та практичну значимість для індустрії програмного забезпечення.

Ключові слова: хмарні обчислення, сервіс, сервер, резервне копіювання.

1. ВСТУП

У сучасному інформаційному суспільстві, де обсяги даних є важливою складовою нашого повсякденного життя, забезпечення надійності та безпеки цих даних стає завданням важливим, як ніколи раніше [1]. Зокрема, у контексті стрімкого розвитку технологій хмарних обчислень та необхідності миттєвого доступу до інформації, тема резервне копіювання даних як послуга (Backup as a service – BaaS) набуває особливої актуальності.

Стаття робота присвячена вивченню, аналізу та розробку системи резервного копіювання даних як послуга. Вона охоплює різноманітні аспекти технічної реалізації, включаючи вибір оптимальних технологій, архітектурних рішень та методів забезпечення безпеки даних. Робота також присвячена аналізу існуючих рішень та їх порівняльній характеристиці.

2. КОНЦЕПЦІЯ BaaS

Це концепція, яка полягає в наданні послуги з автоматизованого резервного копіювання та збереження даних через мережу, зазвичай через хмарні обчислення. Замість того, щоб користувачам або підприємствам самостійно налаштовувати та управляти процесом резервного копіювання, вони можуть звернутися до постачальника послуг та отримати цю послугу як абонентське обслуговування, що дозволяє уникнути складнощів, пов'язаних з технічними деталями.

BaaS пропонує безпечне зберігання даних в хмарних сховищах, забезпечуючи їхню надійність та доступність. Ця інноваційна концепція вирішує проблеми безпеки та витрат на обладнання, забезпечуючи користувачам еластичність у виборі обсягу даних та легкий доступ до резервних копій в будь-який момент. У світлі постійного росту цифрової інформації та її важливості, BaaS стає ключовим інструментом для забезпечення безпеки та доступності даних [1].

3. ПОПУЛЯРНІСТЬ ТЕХНОЛОГІЙ

Використання BaaS вже досить популярне і послугами різних програм на основі цієї технології досить велика.

Декілька з найпопулярніших компаній у цьому сегменті включають:

1. Veeam відома своїми рішеннями для резервного копіювання та відновлення даних у віртуалізованих середовищах. Компанія надає послуги для підприємств різних розмірів.

2. Acronis є лідером у сфері резервного копіювання та кіберзахисту, надаючи широкий спектр послуг, включаючи BaaS.

3. Commvault пропонує рішення для управління даними та резервного копіювання як частину своєї обширної платформи. Компанія відома своєю гнучкістю та можливістю впровадження на будь-якому рівні складності.

4. Rubrik надає інтегровані рішення для управління даними та резервного копіювання в хмарному середовищі. Компанія відома своєю інноваційністю та здатністю забезпечувати захист даних у віртуалізованих та хмарних середовищах.

5. Backblaze є популярним провайдером BaaS для індивідуальних користувачів та невеликих підприємств.

Ці компанії виступають у ролі ключових гравців у сегменті BaaS, надаючи різноманітні рішення для задоволення різних потреб користувачів та підприємств у сфері збереження даних [5].

4. АНАЛІЗ ІСНУЮЧИХ ТЕХНОЛОГІЙ

Перелік популярних технологій

1. Amazon Simple Storage Service (S3) – це об'єктове сховище від Amazon Web Services (AWS).

- Переваги: Легка масштабуватися для зберігання великої кількості даних. Висока доступність через різні регіони та доступність даних. Безпека – дані в можуть бути зашифровані.
- Недоліки: Відносно велика вартість, може бути витратною, особливо при великому обсязі даних та великій кількості операцій [2].

2. Google Cloud Backup.

- Переваги: легка інтеграція з іншими сервісами Google Cloud Platform. Швидкість та продуктивність: Google має широкий географічний розподіл дата-центрів, що може підвищити швидкість доступу до даних.
- Недоліки: Для користувачів, які не мають досвіду з платформою Google Cloud, налаштування може бути складним [3].

3. Microsoft Azure Backup – це рішення для резервного копіювання даних у хмарі від Microsoft.

- Переваги: Легка інтеграція з іншими сервісами та продуктами Azure. Забезпечує можливість автоматичного керування резервними копіями.
- Недоліки: Вартість може бути високою, особливо при великій кількості даних [4].

Вибір між цими рішеннями повинен бути здійснений залежно від конкретних потреб вашої компанії або проекту, а також враховуючи бюджетні обмеження та рівень експертизи вашої команди з управління хмарними сервісами. Найкраще варто звернутися до офіційної документації та консультації з фахівцями для більш детальної інформації щодо конкретних можливостей та вартості цих рішень.

Вартісне порівняння

Таблиця 1. Вартісне порівняння

	AWS	AZURE	GOOGLE CLOUD
СХОВИЩЕ ОБ'ЄКТІВ (обсяг 100 ГБ, 10 ГБ записано (PUT), 90 ГБ читання (GET), 20 ГБ повернення, 200 ГБ сканування)	S3 standard: \$2.80 \$	Blob Storage standard: \$3.48 \$\$	Cloud Storage \$0.57 \$
ЗБЕРІГАННЯ ФАЙЛІВ (100 ГБ стандартного сховища, 1 000 ГБ нечастого доступу, 10 ГБ нечастих запитів на доступ)	Elastic File System standard: \$30.00 \$	Files standard \$56.16 \$\$	Filestore standard tier \$20 \$
БЛОКОВЕ ЗБЕРІГАННЯ (1 інсталяція, 100 ГБ, 2 знімки на день за наявності, 10 ГБ на кожен знімок)	Elastic Block Store general-purpose SSD: \$29.90 \$\$	Managed Disks standard \$11.12 \$	Persistent Disk standard \$43.00 \$\$\$

Кожна послуга пропонує безліч конфігурацій, опцій і варіантів зниження ціни, наприклад, передоплачені або зарезервовані ємності і більш низькі цінові рівні. Разом вони надають корпоративним клієнтам величезну гнучкість у налаштуванні продуктивності, доступності та надмірності сховища відповідно до вимог додатків і бюджету.

Як показують наші приклади, три основні постачальники є конкурентоспроможними. Якщо порівнювати ціни на Amazon S3, Google Cloud Storage та Azure, то значна різниця у вартості зазвичай пов'язана з особливостями того, як провайдери поєднують функції сховища та встановлюють ціни на основі параметрів використання, таких як запити на читання/запис. Покупці сховищ завжди повинні будувати детальні моделі ціноутворення, використовуючи доступні онлайн-калькулятори, перш ніж приймати рішення про покупку на основі вартості [6].

5. ЗАПРОПОНОВАНА ПРОГРАМНА АРХІТЕКТУРА

У системі зображеної на UML-діаграмі класів (Рис. 1) взаємодіють різні компоненти для надання різноманітних послуг користувачам [7]. Клієнтський інтерфейс дозволяє реєструвати облікові записи, входити в систему та відновлювати дані. API Gateway забезпечує аутентифікацію, генерацію токенів та контроль доступу. Система ідентифікації та аутентифікації виконує реєстрацію, перевірку ідентифікаційних даних та генерацію токенів.

Сервіс резервного копіювання включає створення резервних копій, налаштування регулярності та встановлення політики зберігання. Система зберігання даних обирає місце та механізм зберігання, забезпечуючи шифрування та захист. Модуль шифрування та безпеки реалізує шифрування, а мікросервіс безпеки контролює доступ, інтегрується з блокчейном та надає різні рівні захисту.

Компоненти моніторингу та аналітики слідкують за статусом резервних копій та аналізують дані для звітності. Управління ресурсами автоматично масштабує систему, а інтеграційні мікросервіси співпрацюють з іншими системами. База даних зберігає метадані та

налаштування користувачів, включаючи їх ідентифікаційні дані. Ці компоненти взаємодіють, забезпечуючи користувачам послуги реєстрації, аутентифікації, зберігання даних, резервного копіювання та безпеки.

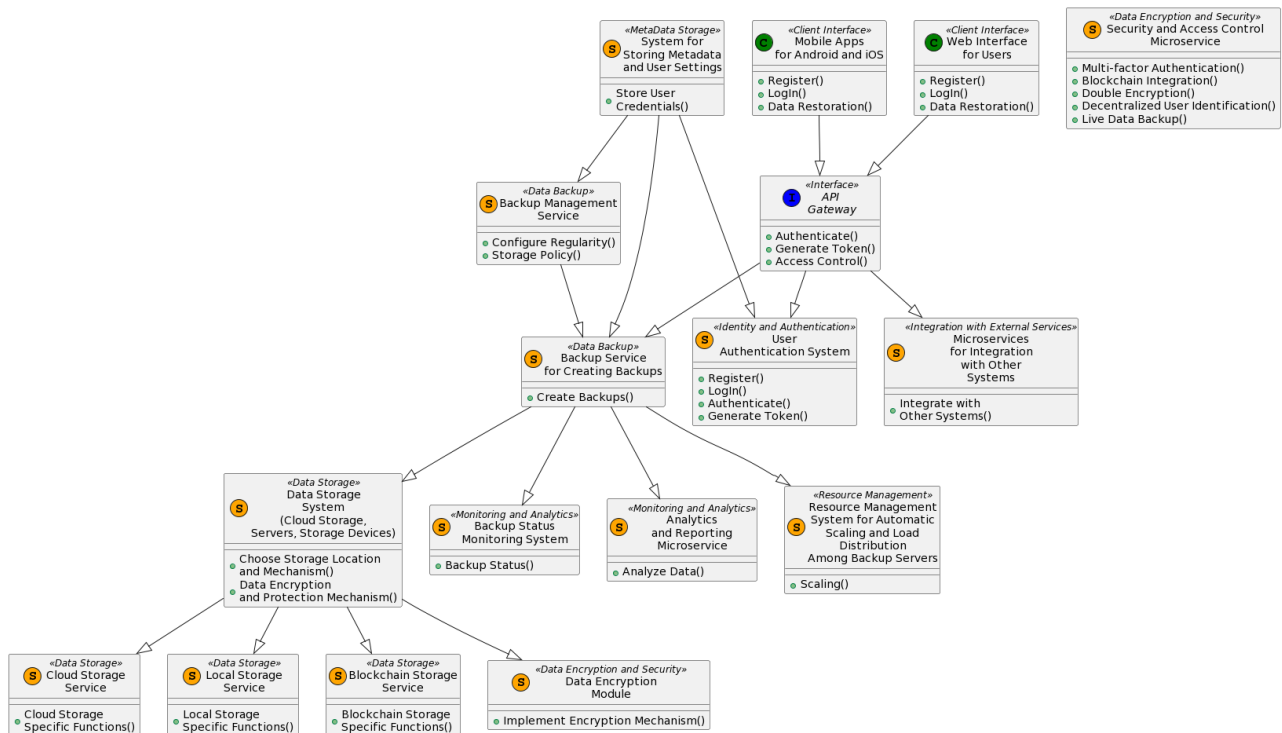


Рисунок 1. Приклад архітектури

Стек технологій які були обрані потрібно буде врахувати наступні технології та підходи:

- Spring Boot: для створення більш простого створення мікросервісів.
- Spring Cloud: надає набір інструментів та бібліотек для реалізації мікросервісної архітектури, включаючи інструменти для конфігурації, реєстрації, балансування навантаження та інші.
- Spring Security: Для реалізації системи ідентифікації та аутентифікації користувачів
- Spring Data: Для роботи з базою даних.
- Spring Cloud Netflix: для моніторингу, балансування навантаження та інших операцій.
- Spring Boot Actuator: для моніторингу та вимірювання продуктивності вашого додатку.
- Spring Cloud Config: Для централізованої управління конфігураціями мікросервісів.
- Spring Cloud Sleuth і Zipkin: Для трасування запитів та аналізу роботи мікросервісів у великій системі.
- MySQL, MongoDB
- Swagger : Для створення документації API.
- Docker та Kubernetes: Для контейнеризації та управління мікросервісами.
- Apache Kafka: Для обробки повідомлень між мікросервісами.
- Логування і моніторинг (наприклад, ELK Stack або Prometheus/Grafana): Для відстеження стану та продуктивності мікросервісів.

6. ВИСНОВКИ

Ця стаття досліджує та аналізує резервне копіювання даних як послугу (BaaS), розглядаючи методи, стратегії та платформи, пов'язані з цим видом послуги. У результаті проведеного дослідження виділено переваги використання BaaS у забезпеченні безпеки та

продуктивності в розробці додатків. Зокрема, висвітлено його наукову новизну та практичну значимість для індустрії програмного забезпечення.

Стаття акцентує актуальність резервного копіювання даних як послуги в сучасному інформаційному суспільстві, де зростає обсяг і важливість збереження та захисту даних. Особливий наголос робиться на перевагах VaaS, таких як легкий доступ до резервних копій та ефективно управління даними, особливо у контексті хмарних обчислень.

Детально проаналізовано популярні компанії, що пропонують послуги VaaS, такі як Veeam, Acronis, Commvault, Rubrik та Backblaze, розглядаючи їхні функції та популярність в індустрії. Це надає читачеві контекст щодо вибору підходящого рішення для конкретних потреб та обмежень.

У розділі, присвяченому аналізу існуючих технологій, висвітлено ключові рішення, такі як Amazon S3, Google Cloud Backup та Microsoft Azure Backup. Подано їхні переваги та недоліки, що дозволяє читачеві зробити обґрунтований вибір залежно від конкретних потреб та обставин.

Нарешті, надано приблизну архітектуру системи VaaS, використовуючи UML-діаграму класів. Зображено взаємодію різних компонентів, таких як клієнтський інтерфейс, API Gateway, сервіс резервного копіювання, система зберігання даних та інші. Це надає загальний огляд структури та функціоналу системи VaaS.

Напрямами подальших досліджень є реалізація послуги резервного копіювання та її порівняння з існуючими на ринку продуктами з урахуванням швидкодії, надійності та вартості. Важливим аспектом є розвиток інтеграцій між платформами VaaS та іншими інфраструктурними рішеннями, що сприятиме ще більш ефективному управлінню даними.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Remote backup service https://en.wikipedia.org/wiki/Remote_backup_service
2. Amazon S3. URL: <https://aws.amazon.com/s3>
3. Google Cloud Backup. URL: <https://cloud.google.com/solutions/backup-dr>
4. Microsoft Azure Backup. URL: <https://azure.microsoft.com/ru-ru/products/backup>
5. Cloud Backup Showdown. URL: <https://www.cloudwards.net/azure-vs-amazon-s3-vs-google-vs-backblaze-b2/>
6. Amazon S3 vs. Google Cloud Storage vs. Azure pricing, features . URL: <https://www.techtarget.com/searchstorage/feature/AWS-vs-Azure-vs-Google-pricing-and-features-compared>
7. Будівання UML діаграм. URL: <https://plantuml.com/>

МЕТОДИКА РЕДУКЦІЇ ОБСЯГУ ІНФОРМАЦІЇ В СИСТЕМАХ ОБРОБКИ ВЕЛИКИХ ДАНИХ

Дзиговський В.І.¹, Рогоза В.С.²

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

¹ dzyhovskiy.vladyslav@gmail.com, ² rosvetnik@gmail.com

Метою роботи є дослідження методик редукції обсягу інформації в системах обробки великих даних. Предметом досліджень є методи зниження розмірностей. Актуальність запропонованих методів виникає із необхідності ефективної редукції обсягу інформації в системах обробки великих даних. В сучасному світі об'єми інформації швидко зростають, і важливо мати засоби для точного відбору, стиснення та аналізу цих даних. Ці методики мають значення як у сферах бізнесу, де необхідно оптимізувати ресурси та приймати рішення на основі обмеженого обсягу інформації, так і у наукових дослідженнях, де зменшення шуму та видалення надлишкових даних дозволяють виявити суттєві зв'язки. Ці методи можуть знайти своє застосування у багатьох галузях, сприяючи оптимізації ресурсів та поліпшенню аналізу даних.

Ключові слова: редукція обсягу даних, зниження розмірності, перетворення матриць, простори високої та низької розмірностей, обробка та аналіз даних.

1. ОГЛЯД МЕТОДИК РЕДУКЦІЇ ОБСЯГУ ІНФОРМАЦІЇ

1.1. Метод головних компонентів

Метод головних компонентів (РСА) є одним із способів зменшити розмірність даних, втративши найменшу кількість інформації. Даний метод використовує ортогональне перетворення множини даних пов'язаних зі зміною у множину змінних без лінійної кореляції, які називаються головними компонентами. Зазвичай використовується два головних компоненти, щоб побудувати дані на площині, тобто двовимірному просторі, і візуально ідентифікувати кластери тісно пов'язаних точок даних (рис. 1) [1, 2].

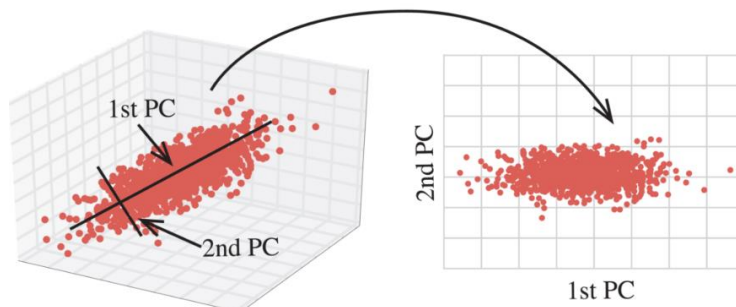


Рисунок 1. Редукція даних методом головних компонентів

Перевагами методу головних компонент є: ефективне зниження розмірності, виявлення найцінніших даних, видалення кореляції, низький вплив шуму, універсальність та простота реалізації.

Недоліками методу головних компонент є: складна інтерпретація даних, потреба у стандартизації даних, відсутність зв'язку між вихідними даними.

1.2. Лінійний розділювальний аналіз

Лінійний розділювальний аналіз (LDA) має на меті побудову та відокремлення одних класів від інших шляхом зниження розмірності. Результати можна використовувати як лінійний класифікатор, або щоб зменшити розміри даних перед їх класифікацією. LDA також дозволяє зручно візуалізувати дані, показуючи розміщення точок по класам та моделюючи різницю між ними (рис. 2) [3, 4].

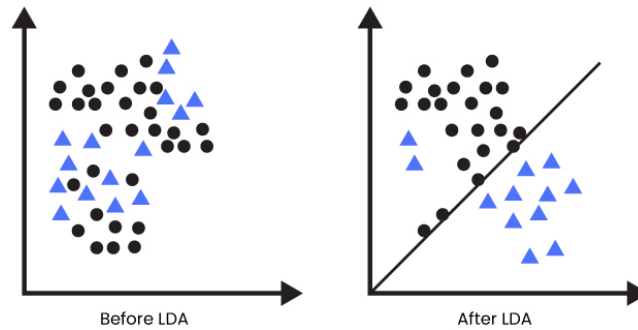


Рисунок 2. Розподіл даних двох класів лінійним розділювальним аналізом

Перевагами лінійного розділювального аналізу є: ефективність при малих спостереженнях, високий розподіл між класами даних, спрощення класифікації даних, проста інтерпретація даних.

Недоліками лінійного розділювального аналізу є: застосовність лише в задачах класифікації, низька точність вихідних даних, дані повинні бути визначеними та розподіленими між класами наперед, обчислювальна складність.

1.3. Метод незалежних компонент

Метод незалежних компонент (ICA) використовується для виявлення незалежних компонентів даних у суміші багатьох різних даних. Даний метод є розширенням методу PCA, який поділяє вхідні дані на фактори, які спричиняють спостереження.

Метод знижує розмірність даних за рахунок виділення найважливіших незалежних спостережень з суміші даних, відкидаючи менш важливі. Цей метод часто використовується в задачах розкладу або знаходження сигналів з однієї великої суміші таких сигналів (рис. 3) [5, 6].

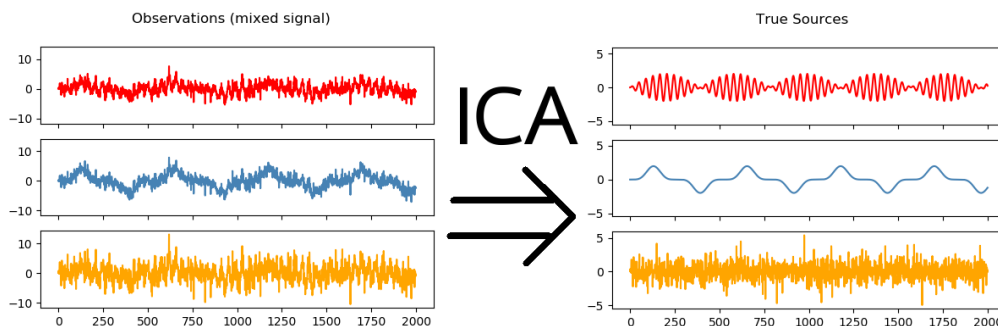


Рисунок 3. Пошук сигналів із суміші сигналів методом ICA

Перевагами методу незалежних компонент є: можливість визначити незалежні компоненти у суміші даних, застосування на практиці при роботі з сигналами, дуже низька чутливість до шуму.

Недоліками методу незалежних компонент є: метод ефективний лише, якщо вхідний сигнал є лінійною комбінацією, зниження розмірності лише наслідок відокремлення знайденого сигналу, чутливість до гіперпараметрів, обчислювальна складність.

1.4. Розклад невід'ємних матриць

Розклад невід'ємних матриць (NMF) – це метод аналізу даних і розкладу матриць на декілька додатних складових матриць, який може бути використаний для зниження розмірності і виділення прихованих структур у вихідних даних.

NMF допомагає знайти структури у вихідних даних шляхом розкладу вихідної матриці на добуток двох невід'ємних матриць меншої розмірності. Одна з цих матриць може розглядатися як набір базових шаблонів або тем, а інша – як коефіцієнти, що вказують на те, як ці шаблони комбінуються для створення вихідних даних (рис. 4) [7, 8].

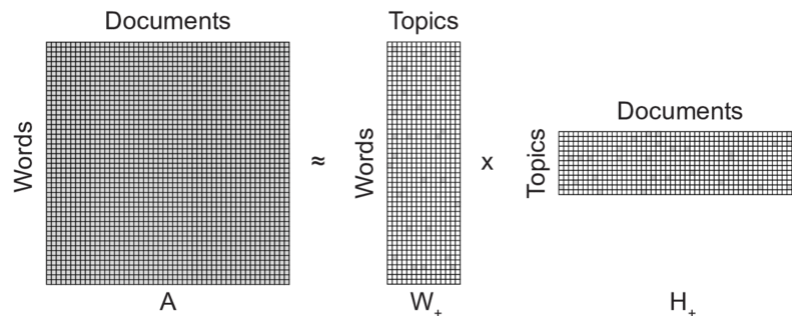


Рисунок 4. Загальний вигляд розкладу невід'ємних матриць

Перевагами розкладу невід'ємних матриць є: легко інтерпретовані результати, обмеження негативних значень, ефективність в аналізі тексту та обробці мов.

Недоліками розкладу невід'ємних матриць є: важкість розрахунку, можлива втрата точності та отримання занадто оптимістичних результатів, висока чутливість до початкових значень, наявність обмеження на тип даних.

1.5. Сингулярний розклад

Сингулярний розклад (SVD) дає чітке представлення матриці, і у той же час дозволяє відкинути менш важливі частини, залишивши близьке представлення бажаної розмірності. В основі метода лежить розклад вихідної матриці на добуток з трьох матриць, кожна з яких менша за розмірами від вихідної (рис. 5). Зменшення розмірності за допомогою SVD допомагає вирішувати проблеми з обробкою та аналізом великих обсягів даних, використанням менше пам'яті та обчислювальних ресурсів [1, 9].

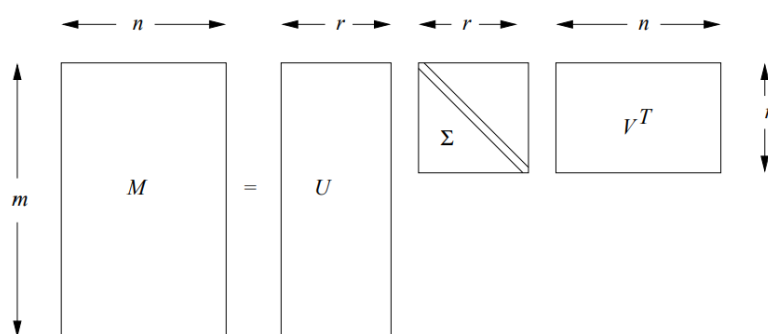


Рисунок 5. Загальний вигляд розкладу невід'ємних матриць

Перевагами сингулярного розкладу є: ефективність у зниженні розмірності, результати відсортовані відповідно до спадання дисперсії, зменшення шуму у даних, може використовуватися для апроксимації матриці, часто застосовується у стисненні комп'ютерних файлів та зображень.

Недоліками сингулярного розкладу є: складна інтерпретація отриманих даних, необхідність у попередній обробці даних, низька ефективність при роботі з розрідженими матрицями.

1.6. CUR-декомпозиція

Сингулярний розклад має недолік, коли вихідна матриця сильно розріджена при великій кількості даних. У такому випадку матриці сингулярного розкладу будуть занадто щільними, а це призведе до того, що виділити концепти буде складно.

Даний недолік вирішує CUR-декомпозиція, яка при розрідженій вихідній матриці буде надавати розрідженими матриці розкладу. Метод використовує підмножину рядків, стовпців і елементів оригінальної матриці для створення наближеної розкладеної форми. Рядки та стовпці обираються випадковим способом, але для надання пріоритету більш важливим елементам використовується норма Фробеніуса (рис. 6) [1, 10].

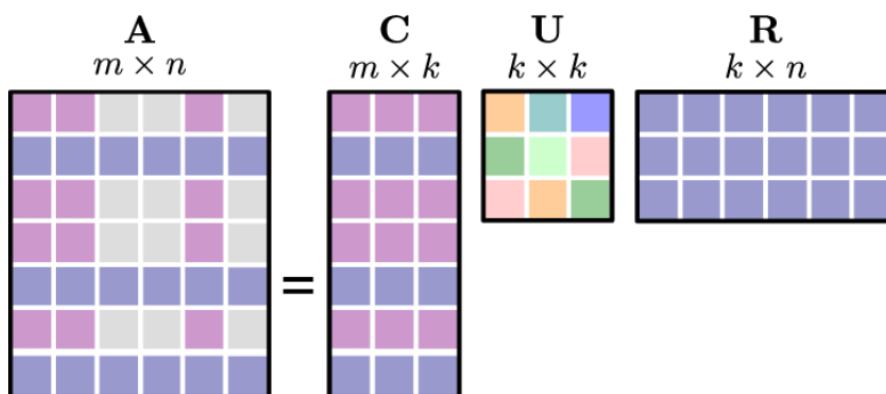


Рисунок 6. Загальний вигляд CUR-декомпозиції

Перевагами CUR-декомпозиції є: можливість вибору більш важливих даних, ефективність обчислення, висока інтерпретованість результатів, зменшене споживання обчислювальних ресурсів, ефективно працює з розрідженими матрицями.

Недоліками CUR-декомпозиції є: можлива втрата інформації, результати залежать від обраних важливих початкових значень, точність залежність від розмірності вибору.

1.7. t-розподілене стохастичне вбудовування сусідів

t-розподілене стохастичне вбудовування сусідів (t-SNE) – це метод для візуалізації даних, у якому дані високої розмірності перетворюються у низькорозмірний простір, зазвичай у двовимірний. На меті даного метода є збереження та візуалізація взаємозв'язків між даними, тобто перетворення простору так, щоб схожі дані залишалися близькими, а відмінні дані відділялися.

Метод t-SNE широко використовується для візуалізації даних, а також для перегляду та аналізу складних високорозмірних даних, таких як зображення, текстові або генетичні дані. Візуалізація, отримана через t-SNE, допомагає визначити групи або структури в даних, які важко виявити в початковому високорозмірному просторі (рис. 7) [11, 12].



Рисунок 7. Візуалізації даних методом t-SNE

Перевагами t-SNE є: збереження локальної структури, розпізнавання нелінійних даних, ефективна візуалізація та дослідження даних.

Недоліками t-SNE є: повільне обчислення даних, стохастичний характер може надавати різні результати при кожному запуску, можлива втрата даних, залежність від гіперпараметрів.

2. ВИСНОВКИ

У цій статті було розглянуто основні методики редукції обсягу інформації, а саме метод головних компонент, лінійний розділювальний аналіз, метод незалежних компонент, розклад невід'ємних матриць, сингулярний розклад, CUR-декомпозиція та t-розподілене стохастичне вбудовування сусідів. Було визначено їх принципи дії, сильні та слабкі сторони їх використання, за якими можна визначити ефективність кожного методу в тому чи іншому завданні.

У випадку розподілу даних за класами або незалежними зв'язками та їх візуалізації доречно буде використовувати лінійний розділювальний аналіз, метод незалежних компонент та t-розподілене стохастичне вбудовування сусідів. Дані методи є ефективними у дослідженнях та аналізу взаємозв'язків між даними, проте їх ефективність в редукції даних невисока, через часткову або повну втрату значень загальної інформації про спостереження.

Розклад невід’ємних матриць є ефективним у зниженні розмірності, але його реалізація на практиці занадто складна, а розрахунки є нестійкими, що впливає на точність досліджуваних даних.

Метод незалежних компонент, сингулярний розклад та CUR-декомпозиція є найкращим вибором для найбільш ефективної редукції обсягу інформації в системах обробки великих даних, оскільки їх технології направлені саме на зниження розмірності матриць зі збереженням загальної цінності отриманих даних відповідно до кожного спостереження. Реалізація та використання таких методик не є важкою та легко модернізувати під різні завдання, а відсутність пошуку зв’язків між даними дає найбільшу ефективність перетворення даних в багатовимірних просторах.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Leskovec J., Rajaraman A., Ullman J. Mining of Massive Datasets. 2014. P. 429–460.
2. Zheng A., Casari A. Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists. 2018. P. 128–145.
3. Linear Discriminant Analysis, Explained. Towards Data Science. URL: <https://towardsdatascience.com/linear-discriminant-analysis-explained-f88be6c1e00b> (дата звернення: 17.10.2023).
4. Is LDA a dimensionality reduction technique or a classifier algorithm? Towards Data Science. URL: <https://towardsdatascience.com/is-lda-a-dimensionality-reduction-technique-or-a-classifier-algorithm-eeed4de9953a> (дата звернення: 17.10.2023).
5. Introduction to ICA: Independent Component Analysis. Towards Data Science. URL: <https://towardsdatascience.com/introduction-to-ica-independent-component-analysis-b2c3c4720cd9> (дата звернення: 17.10.2023).
6. Independent component analysis: An introduction. Emerald Insight. URL: <https://www.emerald.com/insight/content/doi/10.1016/j.aci.2018.08.006/full/html> (дата звернення: 17.10.2023).
7. Non-Negative Matrix Factorization (NMF) for Dimensionality Reduction in Image Data. Towards Data Science. URL: <https://towardsdatascience.com/non-negative-matrix-factorization-nmf-for-dimensionality-reduction-in-image-data-8450f4cae8fa> (дата звернення: 18.10.2023).
8. Optimization and expansion of non-negative matrix factorization. BMC. URL: <https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-019-3312-5> (дата звернення: 18.10.2023).
9. Understanding Singular Value Decomposition and its Application in Data Science. Towards Data Science. URL: <https://towardsdatascience.com/understanding-singular-value-decomposition-and-its-application-in-data-science-388a54be95d> (дата звернення: 19.10.2023).
10. CUR Decompositions, Similarity Matrices, and Subspace Clustering. Frontiers. URL: <https://www.frontiersin.org/articles/10.3389/fams.2018.00065/full> (дата звернення: 19.10.2023).
11. What, Why and How of t-SNE. Towards Data Science. URL: <https://towardsdatascience.com/what-why-and-how-of-t-sne-1f78d13e224d> (дата звернення: 20.10.2023).
12. Introduction to t-SNE. DataCamp. URL: <https://www.datacamp.com/tutorial/introduction-t-sne> (дата звернення: 20.10.2023).

ДОСЛІДЖЕННЯ БІБЛІОТЕКИ REACT.JS ДЛЯ ПОБУДОВИ ВЕБ-ІНТЕРФЕЙСІВ ДЛЯ ЛЮДЕЙ З ОБМЕЖЕНИМИ МОЖЛИВОСТЯМИ

Забельський В.В.¹, Кисельов Г.Д.²

Національний технічний університет України «Київський політехнічний інститут
імені Ігоря Сікорського», Київ, Україна

¹ vovcou1999@gmail.com, ² g.kyselov@gmail.com [0000-0003-2682-3593]

Дослідження присвячено вивченню можливостей підвищення доступності веб-інтерфейсів для осіб із обмеженими можливостями. Застосування бібліотеки React.js для розробки інтерфейсів вивчено через використання різноманітних хуків та компонентів, спрямованих на полегшення взаємодії з користувачем. Результати дослідження вказують на ефективність застосування зазначених інструментів у створенні інклюзивних інтерфейсів. Новизною є розгляд практичних аспектів використання бібліотеки та її хуків для досягнення високого рівня доступності, що має вагоме значення в розробці доступних веб-додатків.

Ключові слова: веб-розробка, доступність веб-інтерфейсів, розробка доступних веб-інтерфейсів, обмежені можливості, бібліотека react.js, проблематика розробки доступних веб-інтерфейсів

1. ВСТУП

У сучасному цифровому світі, де технології невпинно розвиваються, важливість створення веб-інтерфейсів, доступних для всіх користувачів, набуває безпрецедентного значення. Зокрема, для людей із обмеженими можливостями важливо мати зручний та ефективний інтерфейс, що відповідає їхнім потребам, а отже збільшує і кількість потенціальних користувачів [1].

Ця стаття поглиблюється у сучасні підходи до проектування веб-інтерфейсів, зосереджуючись на використанні бібліотеки React.js. В цій статті досліджуються передові техніки, такі як розширені хуки та компоненти, спрямовані на поліпшення взаємодії та доступності. Аналіз впливу цих інструментів на процес розробки та якість веб-додатків є ключовим аспектом наших досліджень. Цей вступ розкриває необхідність інклюзивних підходів у веб-розробці, визначає мету та актуальність нашої роботи, підкреслюючи значущість створення доступних інтерфейсів для всіх шарів користувачів.

2. ОСНОВНІ ПРИНЦИПИ ДОСТУПНОСТІ ВЕБ-ІНТЕРФЕЙСІВ

В сучасному світі інтернет-технологій, проблеми доступності займають центральне місце. Навіть найпрогресивніший веб-інтерфейс втрачає свою цінність, якщо він недоступний для широкого кола користувачів, зокрема, для людей з обмеженими можливостями. Для забезпечення ефективної доступності веб-інтерфейсів важливо керуватися ключовими принципами та дотримуватися міжнародних стандартів, таких як Web Content Accessibility Guidelines (WCAG) [2–4].

Вловлюваність (Perceivable):

- Альтернативні тексти: Забезпечте альтернативний текст для зображень, щоб користувачі з вадами зору могли отримувати зміст.

- Підписи мультимедійних контентів: Надайте текстові підписи для аудіо та відео, щоб невидимі користувачі могли сприймати зміст.

Здатність до взаємодії (Operable):

- Навігація за допомогою клавіатури: Забезпечте можливість навігації та взаємодії за допомогою клавіатури для користувачів з моторними вадами.

- Достатній час: Дайте користувачам достатній час для сприйняття та реакції на контент.

Зрозумілість (Understandable):

- Послідовність та передбачуваність: Розміщуйте елементи на сторінці послідовно та передбачувано, щоб полегшити сприйняття для всіх користувачів.

- Допомога та підказки: Забезпечте доступ до допоміжних елементів та підказок для розуміння взаємодії та навігації.

Надійність (Robust):

- Сумісність із технологіями: Використовуйте технології, які підтримують стандарти доступності, для забезпечення надійності та сумісності з різними пристроями та платформами.

Стандарти доступності та їх важливість:

Впровадження принципів WCAG – це лише початок. Стандарти, такі як Section 508 в США чи Європейський стандарт EN 301 549, визначають юридичні вимоги до доступності для деяких проектів. Це стимулює розробників та підприємства дотримуватися найвищих стандартів.

Застосування принципів доступності та дотримання стандартів – це зобов'язання перед усією аудиторією, роблячи веб-сайти та додатки доступними для всіх. Такі кроки не тільки сприяють суспільству різноманітності, але й розширюють аудиторію, до якої можна звернутися.

Забезпечення доступності веб-інтерфейсів — це інвестиція у майбутнє, в якому висока якість взаємодії в Інтернеті є доступною для кожного, незалежно від їхніх індивідуальних можливостей чи обмежень [5–7].

3. ПРОЕКТУВАННЯ ДОСТУПНИХ ВЕБ-ІНТЕРФЕЙСІВ

Розробка доступних веб-інтерфейсів – завдання, що вимагає уваги до деталей та розуміння особливостей конкретних фреймворків.

Загальні принципи для всіх фреймворків:

1. Альтернативні тексти: Користувачі повинні мати альтернативні тексти для мультимедійних елементів.

2. Навігація: Структурована та передбачувана навігація полегшує взаємодію клавіатурою та іншими адаптивними технологіями.

3. Контрастність: Важлива контрастність тексту та фону сприяє зручності читання.

4. Послідовність та фокус: Правильна послідовність та візуальна підказка фокусу допомагають користувачам з інвалідністю.

5. Достатній час: Важливо надавати користувачам достатньо часу для взаємодії з контентом.

Загальні принципи для бібліотеки React:

1. Інклюзивні компоненти та їхні характеристики

Використання інклюзивних компонентів, які підтримують різні види взаємодії та сприяють доступності, стає стандартом у веб-розробці. Розглядаються такі характеристики, як семантична структура, правильне використання ARIA-атрибутів та забезпечення навігації за допомогою клавіатури.

2. Розширені хуки для доступності

Оглядаються розширені хуки, спрямовані на покращення взаємодії та доступності. Використання хуків, таких як `useFocus`, `useHover`, та `useAriaLive`, надає розробникам інструменти для ефективного керування фокусом, станом наведення та повідомленнями для користувачів [8].

3. Тестування та валідація

Досліджується роль тестування у забезпеченні доступності. Представляються підходи до автоматизованого валідаційного тестування за допомогою бібліотек, таких як `react-ally-testing`, та ручного тестування з використанням інструментів, які підтримують взаємодію із засобами читання екрану.

4. Забезпечення адаптивності та підтримки всіх засобів

Враховуються питання адаптивності та підтримки різних пристроїв та засобів взаємодії. Особлива увага приділяється мобільним платформам, де важливо забезпечити зручний доступ для всіх користувачів.

Але, у світі React, виникають і унікальні виклики, що варто враховувати. Такі проблеми в розробці на React, як:

1. Заголовки у React SPA (Single Page Application):

Створення SEO-оптимізованих та дружніх до відвідувачів заголовків у React SPA може бути викликом через відсутність традиційного оновлення `<title>`.

2. Втрата фокусу в React:

При переході між компонентами може виникати проблема втрати фокусу, особливо для користувачів, які використовують клавіатуру для навігації.

3. Інструментарій для фокусування:

Відсутність вбудованого інструментарію для ефективного керування фокусуванням може бути проблемою в складних взаємодіях.

4. Агіа атрибути для оптимізації доступності

Додавання `aria`-атрибутів у React-проекти може значно полегшити навігацію та взаємодію для користувачів із спеціальними потребами. ARIA (Accessible Rich Internet Applications) — це набір атрибутів, які надають додаткову інформацію елементам і дозволяють їм взаємодіяти із веб-додатками більш ефективно. Особливо `aria` атрибути важливі для скрінрідерів [9, 10].

Розробка доступних веб-інтерфейсів у React – це постійний процес, що вимагає уваги до деталей та використання правильних інструментів.

4. РЕЗУЛЬТАТИ ДОСЛІДЖЕНЬ

Дослідження в області проектування доступних веб-інтерфейсів у React привело до значущих висновків та розробки ефективних рішень для подолання специфічних викликів, які виникають у цьому фреймворку.

Загальні результати:

1. Підвищена доступність контенту: Застосування визначених принципів, таких як альтернативні тексти, контрастність та правильна навігація, сприяє підвищенню доступності веб-інтерфейсів для користувачів з обмеженими можливостями.

2. Оптимізація фокусування: Слід звертати увагу на фокусування та вирішувати проблеми втрати фокусу при взаємодії з React-компонентами, що покращить загальний досвід використання.

3. SEO-оптимізація заголовків.

Однією з найактуальніших проблем є недостатня підтримка стандартів доступності в ряді популярних бібліотек та фреймворків, зокрема, React.js. Відсутність вбудованих

інструментів для автоматичної генерації атрибутів ARIA, керування фокусуванням, та підтримки клавішних подій може впливати на користувачів, залежно від їхніх потреб.

Загальні результати досліджень також підкреслили важливість використання сучасних стандартів та практик розробки веб-інтерфейсів, зокрема, використання систем дизайну із вбудованими принципами доступності та впровадження адаптивного дизайну. Підтримка стандартів та використання сучасних технологій дозволяє створювати продукти, які будуть доступні для широкого кола користувачів [11, 12].

Виклики та потреба у подальших дослідженнях:

Попри досягнені успіхи, залишаються відкриті питання та завдання для подальших досліджень. Особливо актуальною є потреба у створенні універсальної бібліотеки-утиліти, яка впроваджуватиме інструменти для роботи з метатегами та заголовками, з фокусуванням, з ARIA-атрибутиками та іншим. Такий інструментарій сприятиме консистентності та ефективності розробки доступних веб-інтерфейсів у React, роблячи цей процес більш доступним для широкого кола розробників.

5. ВИСНОВКИ

Виготовлення доступних веб-інтерфейсів для осіб з обмеженими можливостями у фреймворку React виявилось завданням, яке вимагає уважності та системного підходу. Здобуті результати та висновки розкривають ключові аспекти цього дослідження.

Використання фреймворку React дозволяє:

1. Забезпечити доступність веб-додатків і реалізувати важливу соціальну задачу відповідальності та уваги до різних потреб користувачів.

2. React, як популярний фреймворк, може грати ключову роль у реалізації доступних інтерфейсів, використовуючи свої можливості та додаткові інструменти.

3. Створення бібліотеки-утиліт з готовими рішеннями з побудови доступних веб-інтерфейсів на React, сприятиме вирішенню конкретних проблем, в тому числі пов'язаних з фокусуванням та SEO.

Заключні думки:

Дослідження підтверджує, що розробка доступних веб-інтерфейсів у React є важливим завданням, що вимагає поєднання технічної компетентності та врахування потреб кінцевих користувачів. Вдосконалення доступності веб-середовища сприятиме створенню інклюзивного й рівноправного інтернет-простору для всіх.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Cooper, A., Reimann, R., & Cronin, D. (2007). *About Face 3: The Essentials of Interaction Design*. Wiley.

2. Tondreau, B. (2014). *Inclusive Design Patterns*. O'Reilly Media.

3. Henry, S. L., & Henry, S. L. (2017). *Just Ask: Integrating Accessibility Throughout Design*. ET\Lawton.

4. W3C Web Content Accessibility Guidelines (WCAG). Retrieved from <https://www.w3.org/WAI/WCAG21/quickref/>

5. Nielsen, J. (1999). *Designing Web Usability: The Practice of Simplicity*. New Riders.

6. *Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability* by Steve Krug. New Riders.

7. Hick's Law - Wikipedia. https://en.wikipedia.org/wiki/Hick%27s_law

8. Don Norman. (2002). *The Design of Everyday Things*. Basic Books.

9. ISO 9241-11:2018 - Ergonomics of human-system interaction -- Part 11: Usability: Definitions and concepts.

10. Accessibility Principles from the A11y Project. Retrieved from <https://a11yproject.com/principles/>
11. ARIA: Accessible Rich Internet Applications. Retrieved from <https://www.w3.org/WAI/standards-guidelines/aria/>
12. React Accessibility. Retrieved from <https://legacy.reactjs.org/docs/accessibility.html>

ПРОГРАМНИЙ СЕРВІС ДЛЯ РАЦІОНАЛЬНОГО УПРАВЛІННЯ ПЕРСОНАЛОМ КОМЕРЦІЙНОЇ КОМПАНІЇ

Кисельов Г.Д.¹, Гречко Д.М.²

Національний технічний університет України «Київський політехнічний інститут
ім. Ігоря Сікорського», Київ, Україна

¹ g.kyselov@gmail.com , ² anzh29@gmail.com

У сучасному бізнес-світі, де швидкість змін та високий рівень конкуренції є нормою, ефективне управління персоналом стає не просто важливим, а критично необхідним фактором успіху. Розвиток технологій відкриває нові горизонти в методах управління людськими ресурсами, дозволяючи компаніям бути більш гнучкими, адаптивними та ефективними. Інтеграція цих технологій у повсякденні управлінські процеси вносить зміни не тільки в способи роботи з персоналом, але й у саме бачення корпоративної культури та стратегії розвитку організації. Ця стаття зосереджується на розумінні того, як сучасні технологічні рішення та методології можуть трансформувати управління персоналом, при цьому не втрачаючи людський вимір, який залишається ключовим у будь-якій управлінській діяльності.

Ключові слова: управління персоналом, система, управління персоналом, ERD.

1. ВСТУП

У сучасному діловому світі, де керування людськими ресурсами стає дедалі більш складним і динамічним, потреба в надійній та інтуїтивно зрозумілій HR-системі є більшою ніж коли-небудь. Важливість ефективного управління персоналом не може бути недооцінена, оскільки воно впливає на кожен аспект організації – від загальної продуктивності до задоволеності працівників. З цією метою ми розробили HR-систему, яка використовує передові технології та практики проектування, щоб забезпечити максимальну ефективність та користувацький досвід.

Система побудована на сучасному стеку технологій, включаючи Vue.js для фронтенду та Golang для бекенду, які об'єднуються з PostgreSQL для створення потужної та масштабованої платформи. Vue.js пропонує гнучкий та реактивний інтерфейс користувача, забезпечуючи швидку та плавну взаємодію, яка є критичною для задоволення потреб кінцевих користувачів. Golang, з його перевагами у паралельній обробці та ефективності виконання, лежить в основі нашої бекенд-системи, забезпечуючи високу продуктивність та надійність.

Використання цих сучасних технологій дозволяє HR-системі досягти високого рівня масштабованості та гнучкості, необхідних для адаптації до зростаючих та змінюваних потреб бізнесу. Розроблено систему з урахуванням диференційованих ролей користувачів, надаючи адміністраторам, керівникам команд та звичайним користувачам унікальний доступ до функціоналу, відповідний їхнім потребам.

2. АРХІТЕКТУРА ПРОГРАМНОГО СЕРВІСУ

2.1. Вимоги до системи

Функціональні вимоги:

1. Управління персоналом:

- Зберігання детальної інформації про співробітників, включно з особистими даними, історією займаних посад, заробітною платою та історією винагород.
- Відстеження кар'єрного зростання та розвитку співробітників, управління цілями та амбіціями.

2. Заробітна плата та бонуси:

- Функціонал для розрахунку та управління заробітною платою, надбавками, бонусами.
- Можливість подання запитів на зміну заробітної плати з подальшим схваленням керівництвом.

3. Відпустки та відгули:

- Управління відпустками: реєстрація запитів на відпустку, відстеження балансу відпусток, схвалення та відхилення запитів.

4. Організаційна структура:

- Візуалізація ієрархії та структури компанії.
- Управління змінами у структурі команд, відділів та підрозділів.

5. Звітність та аналітика:

- Генерація звітів за різними параметрами, включаючи заробітну плату, ефективність та використання відпусток.
- Аналітичні панелі для оцінки даних про персонал та відповідних HR-метрик.

6. Менеджмент прав доступу:

Сервіс включає диференційовану систему прав доступу, що дозволяє адаптувати функціонал до ролей користувачів: адміністратора, керівника команди та звичайного користувача.

- Admin (Адміністратор): Має повний доступ до всіх модулів і функцій системи, включаючи управління користувачькими акаунтами, налаштування прав доступу, перегляд та редагування всіх даних.
- Team Lead (Керівник команди): Має розширені права для керування даними своєї команди, включаючи запити на заробітну плату, відпустки, оцінювання продуктивності та моніторинг KPI.
- User (Користувач): Має доступ до особистої інформації, може подавати запити на відпустку, переглядати свою заробітну плату та історію бонусів, а також оновлювати особисті цілі та амбіції.

2.2. Опис ERD (схема "сутність-зв'язок")

За встановленими умовами було розроблено ERD, яка стала основою для комплексної HR-системи, здатної адекватно відображати та обробляти широкий спектр HR-процесів. Ця ERD втілює в собі не тільки структуру зберігання даних, але й динаміку взаємодій між різними сутностями, такими як співробітники, відділи, команди та кар'єрні розвідки.

ERD – це аббревіатура від "Entity-Relationship Diagram" (схема "сутність-зв'язок"), яка є ключовим інструментом в області баз даних та інформаційних систем.

У взаємозв'язку з сучасними розробками, такими як PostgreSQL, яка слугує надійною платформою для зберігання та управління даними, ERD забезпечує швидкий доступ до інформації, високу конкурентоспроможність системи та можливість легкої інтеграції з іншими системами та сервісами [1].

Розробка фроненду з використанням Vue.js та бекенду на базі Golang гарантує, що користувачі отримують не тільки функціональний, але й естетично приємний інтерфейс, водночас забезпечуючи надійність та ефективність бекенд-операцій. Комбінація цих технологій дозволяє системі бути масштабованою, адаптивною до змінних потреб бізнесу та забезпечує високу продуктивність обробки даних.

ERD зображена на рис. 1.

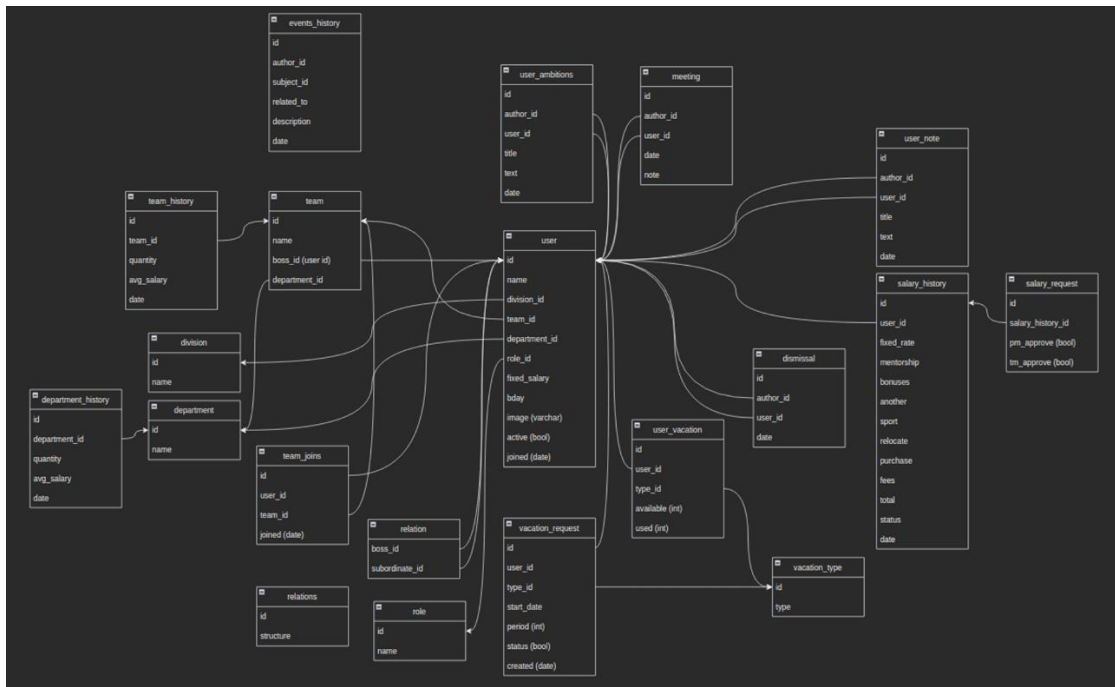


Рисунок 1. ERD

Значення сутностей системи на діаграмі:

1. Сутність **"User" (Користувач)** – Центральна сутність, яка містить інформацію про співробітників, включаючи ім'я, ID підрозділу, команди, відділу, роль, фіксовану зарплату, дату народження, зображення, статус активності та дату приєднання.
2. Сутність **"Division" (Підрозділ)** і **"Department" (Відділ)** – Представляють структурні елементи компанії, кожен з яких має унікальний ідентифікатор та назву. Співробітники (Users) асоційовані з конкретними підрозділами і відділами.
3. Сутність **"Team" (Команда)** – Описує команди в межах компанії, включаючи ідентифікатор команди, назву, ID керівника команди та ID відділу.
4. Сутність **"Role" (Роль)** – Визначає ролі співробітників у компанії, має унікальний ідентифікатор та назву ролі.
5. Сутності **"team_history"**, **"department_history"** – Містять історичні дані про команди та відділи, відповідно, включаючи кількість співробітників, середню зарплату та дату.
6. Сутність **"Relation" (Взаємозв'язок)** – Визначає відносини між співробітниками, зокрема керівника та підлеглих.
7. Сутності **"salary_history"**, **"salary_request"** – Записують історію зарплат та запити на зміну зарплати, включаючи інформацію про ставки, надбавки, бонуси тощо.
8. Сутність **"user_vacation"**, **"vacation_request"**, **"vacation_type"** – Управління відпустками, включаючи доступні та використані години, типи відпусток, статуси запитів на відпустку.

9. Сутність "meeting", "user_note" – Записи зустрічей та нотатки, пов'язані з конкретними користувачами, їх дати та текст.

10. Сутність "user_ambitions" – Цілі та прагнення співробітників, описані через заголовки та текст.

11. Сутність "events_history" – Історія подій, пов'язаних зі співробітниками, таких як дати та описи подій.

Ці сутності з'єднані між собою за допомогою зв'язків, що відображають структуру та процеси в організації. Зв'язки між сутностями забезпечують інтеграцію даних і можливість проводити комплексний аналіз. Наприклад, зв'язок між "User" та "Role" дозволяє визначити роль кожного співробітника, а зв'язок між "User" та "salary_history" дозволяє слідкувати за історією заробітної плати.

3. ІНТЕРФЕЙС ПРОГРАМНОГО СЕРВІСУ

3.1. Інтерфейс користувача

Профіль співробітника в HR-системі є централізованим джерелом для усієї важливої інформації, що стосується конкретного працівника [2]. Інтерфейс зображено на рис. 2. Це не тільки збірник основних персональних даних, але й аналітичний інструмент, який відображає динаміку змін заробітної плати, надбавок, бонусів та інших компонентів винагороди.

Профіль містить детальну інформацію, таку як стаж роботи в компанії, дату народження, організаційну роль та місце у структурі компанії. Також він надає інтуїтивно зрозумілі графіки та діаграми, що ілюструють місячні зміни в оплаті праці, включаючи основну ставку, надбавки за менторство, бонуси та інші виплати. Ця візуалізація допомагає виявляти тенденції та аналізувати зміни у заробітній платі, що може бути корисно для планування бюджету, проведення переговорів про підвищення зарплати та аналізу ефективності винагороди.

Крім того, профіль включає показники зростання зарплати працівника за рік та зміни в основній ставці, які виділені на профілі для швидкого доступу та оцінки. Ці показники є важливими для визначення загальної компенсаційної стратегії та для відстеження ефективності інвестицій у персонал.

Особистий профіль у HR-системі є потужним інструментом, який відіграє ключову роль в управлінні персоналом, дозволяючи HR-менеджерам та керівникам отримувати глибокий аналіз кожного співробітника індивідуально, що сприяє більш обґрунтованому прийняттю рішень.

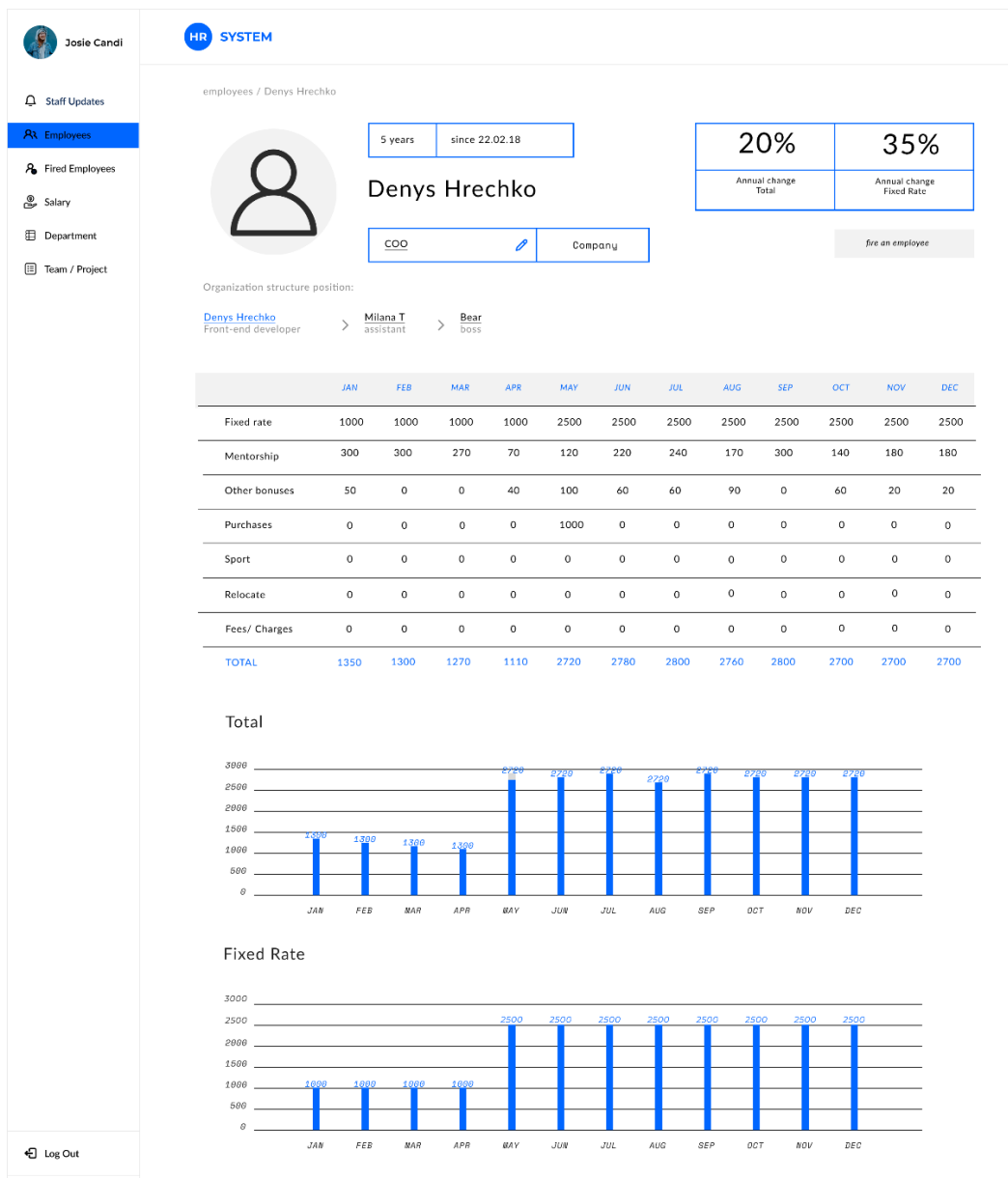


Рисунок 2. Інтерфейс користувача

3.2. Інтерфейс структури компанії

Ключовим елементом будь-якої сучасної HR-системи є її здатність надати чітке уявлення про структуру управління організацією. В нашій HR-системі це досягається за допомогою модуля "Структура компанії", який зображено на рис. 3, який візуалізує ієрархічні відносини між співробітниками, визначаючи їхні ролі та позиції в організаційній ієрархії. Цей інструмент не тільки допомагає у визначенні поточних зв'язків між різними рівнями управління, але й підтримує планування змін та розвитку внутрішньої структури компанії.

Використання графічних засобів для представлення структури компанії дозволяє легко ідентифікувати лінійних керівників, підлеглих та команди проектів, а також сприяє кращому розумінню внутрішніх процесів [3]. Наприклад, можна одразу побачити, хто є керівником певного відділу, кому підпорядковані певні співробітники, та як розподілені ролі в рамках проектної команди. Це створює прозорий та інтуїтивно зрозумілий досвід для управління

персоналом, що є надзвичайно важливим для великих організацій із складними структурами [4].

Також, цей інструмент підкреслює зв'язки між співробітниками та їх ролями в організації, що може бути використано для розробки планів кар'єрного зростання, навчальних програм та для оптимізації командної взаємодії. Централізоване управління даними про структуру компанії також полегшує аналіз внутрішніх потоків талантів та виявлення ключових позицій, що потребують зміцнення або розвитку [5].

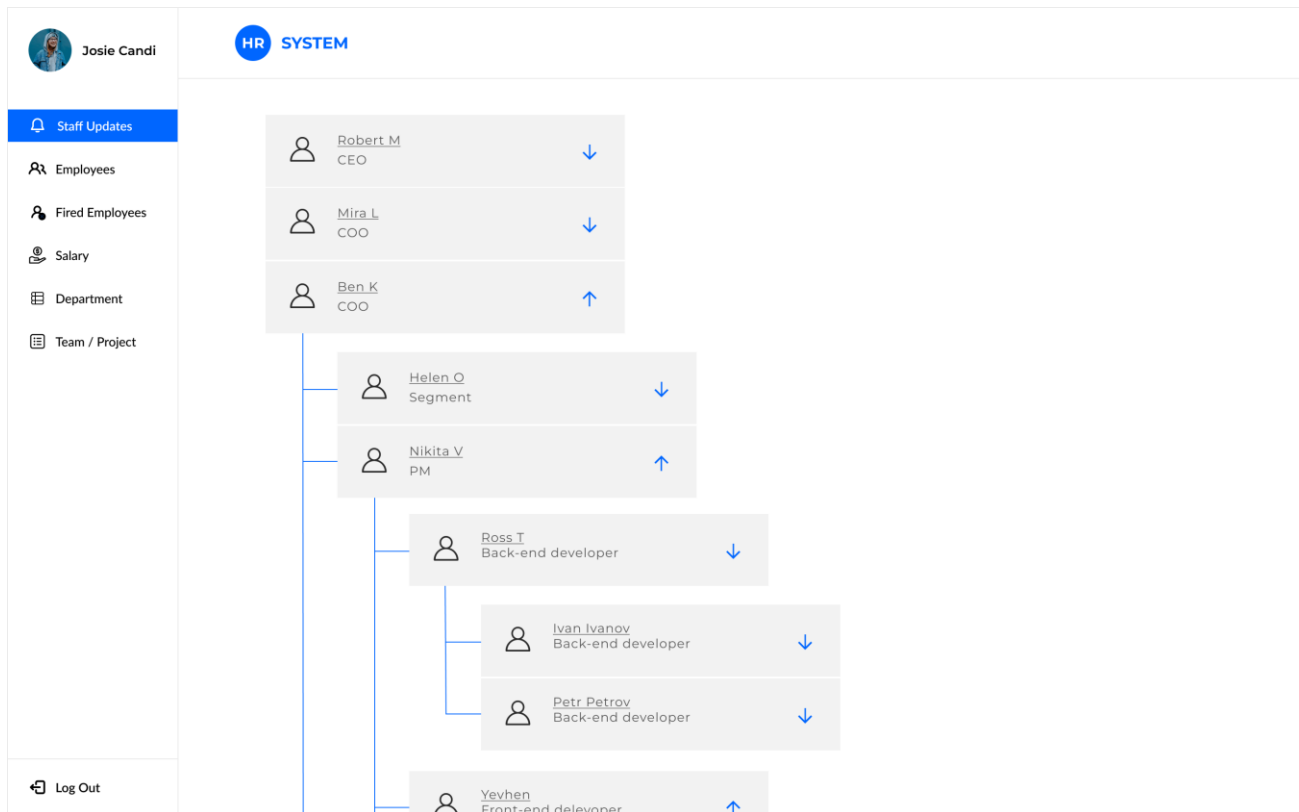


Рисунок 3. Інтерфейс структури компанії

4. ВИСНОВКИ

У світлі розвитку сучасних технологій та постійного вдосконалення методів управління персоналом, розробка і впровадження розробленої HR-системи у комерційній компанії може налагодити менеджмент персоналу та бути одним з основних факторів розвитку компанії. Використання передових технологій, таких як PostgreSQL, Vue.js, та Golang, у комбінації з добре спланованою ERD (Схемою "сутність-зв'язок"), зрозумілим інтерфейсом, забезпечило створення системи, яка не тільки відповідає поточним потребам ринку, але й готова до майбутніх викликів.

Система управління персоналом має ряд функцій, які корелюють з деякими конкретними методологіями управління персоналом. Ось як це виглядає:

1. Компетентнісний підхід: Система підтримує ідентифікацію та розвиток ключових навичок та компетенцій працівників, особливо через індивідуалізовані профілі та нотатки про розвиток співробітників.

2. HR-бізнес партнерство: Система дозволяє HR-фахівцям тісно співпрацювати з різними департаментами та командами, забезпечуючи взаєморозуміння та підтримку бізнес-цілей.

3. Agile HR: Гнучкість системи, яка дозволяє швидко реагувати на зміни, адаптувати структури команд та відділів, а також легко управляти відсутностями співробітників, відповідає принципам Agile.

4. Evidence-based HR: Система дозволяє збирати та аналізувати дані про співробітників, що сприяє прийняттю обґрунтованих рішень у сфері управління персоналом.

5. Performance Management: Можливість ведення нотаток та оцінка продуктивності працівників через систему відповідає основним принципам управління продуктивністю [7].

6. Талант-менеджмент: Система підтримує ідентифікацію, розвиток та утримання талантів, особливо через індивідуалізовані профілі та кар'єрне планування.

7. Lean HR: Автоматизація процесів, зокрема у сфері заробітної плати та управління відпустками, сприяє оптимізації ресурсів та підвищенню ефективності HR-функцій [6].

8. Transformational HR: Система може сприяти інноваціям та змінам в організації, забезпечуючи платформу для адаптації та розвитку організаційних структур [8].

Система надає платформу, яка відповідає багатьом сучасним підходам управління персоналом, забезпечуючи гнучкість, ефективність та стратегічне планування в управлінні людськими ресурсами.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. "PostgreSQL Official Website." URL: <https://www.postgresql.org/> (Дата звернення: 18.11.2023).

2. "User Interface (UI) Design." Interaction Design Foundation. URL: <https://www.interaction-design.org/literature/topics/ui-design> (Дата звернення: 18.11.2023).

3. "Agile HR: A Comprehensive Guide." Pelago Health. URL: <https://www.pelagohealth.com/resources/hr-glossary/agile-hr/> (Дата звернення: 18.11.2023).

4. Mishra, Paritosh. "Research on the construction and implementation of intelligent human resource management system." Routledge Focus on Business and Management, 2021, vol. 12. (Дата звернення: 18.11.2023).

5. Meijerink, Jeroen. "Design and implementation of human resource information management system." The International Journal of Human Resource Management, 2021, vol. 6. (Дата звернення: 18.11.2023).

6. "Lean HR: Streamlining Processes and Procedures." AIHR. URL: <https://www.aihr.com/blog/lean-hr/> (Дата звернення: 18.11.2023).

7. "Performance Management - Overview, Components, Importance." Investopedia. URL: <https://www.investopedia.com/terms/p/performance-management.asp> (Дата звернення: 18.11.2023).

8. "HR Transformation: Reimagining Human Resources." Gartner. URL: <https://www.gartner.com/en/human-resources/topics/hr-transformation> (Дата звернення: 18.11.2023).

РОЗРОБКА БІБЛІОТЕКИ КОМПОНЕНТІВ ДЛЯ ВЕБ ІНТЕРФЕЙСІВ МЕДИЧНИХ ПЛАТФОРМ З ВИКОРИСТАННЯМ БІБЛІОТЕКИ REACT.JS

Кисельов Г.Д.¹, Коваль Д.О.²

Національний технічний університет України «Київський політехнічний інститут
ім. Ігоря Сікорського», Київ, Україна

¹ kyselov@gmail.com [0000-0003-2682-3593], ² koval.diana@lil.kpi.ua

Незважаючи на прогрес у галузі інформаційних технологій, медична сфера досі істотно відстає у впровадженні сучасних підходів до створення користувацьких інтерфейсів. Більшість наявних медичних систем мають застарілі, неінтуїтивні інтерфейси, що ускладнює роботу лікарів та погіршує досвід пацієнтів. Ця проблема значно загострилася під час пандемії COVID-19. Існує нагальна потреба у створенні сучасних медичних платформ зі зручними інтерфейсами. Реалізація спеціалізованої бібліотеки React компонентів може суттєво пришвидшити розробку таких систем та поліпшити досвід взаємодії в медичній сфері. Обґрунтування необхідності розробки такої бібліотеки представлено в даній роботі.

Ключові слова: UX, ReactJS, охорона здоров'я, медичні інформаційні системи, адаптивний інтерфейс, цифровізація медицини, компоненти.

1. ВСТУП

Сучасні медичні інформаційні системи та платформи вимагають створення інтуїтивних та зручних веб інтерфейсів як для лікарів, так і для пацієнтів. Швидкість розробки, можливість повторного використання коду та гнучкість інтерфейсу є ключовими факторами успіху таких рішень. Однією з найпопулярніших технологій для створення сучасних веб додатків є фреймворк React – JavaScript для побудови користувацьких інтерфейсів.

React дозволяє будувати швидкі односторінкові застосунки з можливістю багаторазового використання компонентів інтерфейсу. Це ідеально підходить для розробки складних медичних платформ, де потрібно відображати великі обсяги даних в інтерактивному режимі. Крім того, екосистема React містить багато готових бібліотечних інтерфейсів користувача (User Interface – UI) з широким спектром компонентів.

Проте більшість існуючих бібліотек не враховують специфіку саме медичних додатків. Розробка власної бібліотеки React компонентів, оптимізованих для охорони здоров'я, дозволить значно пришвидшити створення якісних медичних платформ та сфокусуватися на бізнес-логіці.

У цій статті детально розглянемо процес створення спеціалізованої React бібліотеки UI компонентів для медичних веб додатків та особливості її застосування.

2. ПРОБЛЕМИ ВПРОВАДЖЕННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ В МЕДИЦИНІ

Незважаючи на стрімкий розвиток інформаційних технологій, медична галузь істотно відстає у їх впровадженні. За даними дослідження Emerging Healthcare Technologies (2020),

лише 20% лікарень у США використовують хмарні технології, 25% – телемедицину, а 45% – штучний інтелект.

Це значно повільніше, ніж в інших галузях. Наприклад, за даними Flexera (2021), 93% компаній вже використовують хмарні сервіси [1]. Така відсталість медицини уповільнює розробку сучасних медичних інформаційних систем з інтуїтивним та зручним інтерфейсом.

Багато лікарень досі використовують застаріле програмне забезпечення зі складними неінтуїтивними інтерфейсами, що ускладнює роботу медиків. Поширені електронні медичні картки (EHR) часто мають незручну навігацію, складний пошук та введення даних [2].

Це призводить до зниження продуктивності лікарів та підвищення ймовірності помилок. За дослідженням Stanford Medicine (2017), лікарі витрачають в середньому по 2 години на роботу з EHR на кожну годину безпосередньо з пацієнтом [3].

Проблема зручності користувацького інтерфейсу особливо загострилася під час пандемії COVID-19, коли потреба в ефективних цифрових інструментах різко зросла. Усунення затримок у впровадженні новітніх технологій, зокрема створення сучасних медичних платформ, є критично важливим для розвитку галузі.

Існує нагальна потреба в модернізації інтерфейсів медичних інформаційних систем шляхом використання сучасних підходів на кшталт React, що дозволить підвищити ефективність медичних працівників та рівень обслуговування пацієнтів.

3. ОБҐРУНТУВАННЯ ВИБОРУ REACTJS ЯК БАЗИ ДЛЯ МЕДИЧНОЇ ПЛАТФОРМИ

ReactJS є одним з найпопулярніших і затребуваних фреймворків для розробки сучасних веб додатків. За даними Statista [4], у 2022 році React використовували 63% розробників веб додатків. Його можливості ідеально підходять для створення складних медичних платформ:

- Висока продуктивність. React використовує віртуальну об'єктну модель документа (Document Object Model – DOM, що дозволяє оптимізувати рендеринг інтерфейсу. За тестами UI Electric, React демонструє до 32% вищу продуктивність за інші фреймворки. Це критично важливо при роботі з великими обсягами медичних даних.
- Гнучкість. Можливість створення повторно використовуваних компонентів дозволяє легко масштабувати інтерфейс під різні потреби. Компонентний підхід React спрощує розробку адаптивних інтерфейсів.
- Спрощення розробки. В React всі компоненти містять логіку, стан та стилі в одному файлі (JSX), що полегшує розуміння і підтримку коду. За опитуванням Clever-Cloud, 84% розробників вважають React простішим у використанні за інші фреймворки.
- Екосистема. React має величезну спільноту (понад 1 млн розробників) та широкий набір готових рішень, що прискорює створення складних додатків.
- Сумісність. React компоненти можуть взаємодіяти з іншими бібліотеками, сервісами та API за допомогою REST та GraphQL. Це дає гнучкість при інтеграції з медичними системами.

ReactJS поєднує швидкість, масштабованість, гнучкість та зручність підтримки коду. Це робить його ідеальним вибором як основу для створення сучасних медичних платформ, що повинні ефективно працювати з великими обсягами даних та складним інтерфейсом.

4. ВИКОРИСТАННЯ ДОСВІДУ КОРИСТУВАЧА

Вимірювання ефективності користувацького досвіду (User Experience – UE) є важливим етапом у розробці будь-якої платформи, особливо в медичній галузі. Адже від дизайну і зручності інтерфейсу може залежати здоров'я пацієнтів.

Для вимірювання UX ефективності розробленого продукту використовують як кількісні, так і якісні методики [5]. До кількісних належать вимірювання показників конверсії, швидкості виконання завдань, кількості помилок тощо. Серед якісних – опитування користувачів, аналіз емоційних реакцій, спостереження за роботою з інтерфейсом [6].

Для React бібліотеки компонентів доцільно використовувати як кількісні метрики – час завантаження сторінки, кількість кліків для виконання певного завдання, так і якісні – задоволеність користувачів, емоційні реакції під час тестування.

Оскільки бібліотека компонентів React призначена для медичних платформ, критично важливо забезпечити високу ефективність UX. Адже помилки в інтерфейсі можуть призвести до неправильного лікування чи діагностики.

Для вимірювання UX ефективності доцільно використовувати інструменти, такі як Google Analytics [7], Hotjar [8], Mixpanel [9]. Вони дозволяють збирати кількісні дані про поведінку користувачів, а також записувати їх взаємодію з інтерфейсом для подальшого аналізу.

Окрім цього, варто регулярно проводити А/В (порівняння двох варіантів вебсторінки) тестування різних версій компонентів, щоб виявляти найбільш ефективні рішення. Такий підхід дозволить оптимізувати UX поступово, на основі реальних даних. Комбінація кількісних і якісних методів вимірювання UX дозволить створити React компоненти максимально зручними і ефективними для медичних платформ. А це, в свою чергу, сприятиме підвищенню якості медичних послуг та безпеки пацієнтів.

5. БАЛАНСУВАННЯ ПОТРЕБ КОРИСТУВАЧІВ

При розробці бібліотеки компонентів для медичних платформ важливо враховувати потреби різних груп користувачів: лікарів, пацієнтів, адміністраторів та інших. Їх вимоги до інтерфейсу можуть суттєво відрізнятись, тому UX дизайнерам доводиться шукати компроміси.

Лікарі потребують швидкого доступу до медичних даних пацієнтів та зручних інструментів для ведення історій хвороб. Пацієнти ж більше цінують простоту навігації, доступність інформації та конфіденційність.

Одне з можливих рішень – створити різні інтерфейси для лікарів і пацієнтів. Це дозволить оптимізувати UX під конкретну аудиторію. Проте підтримка двох окремих інтерфейсів потребує більших ресурсів на розробку та тестування.

Тому часто обирають створення єдиного інтерфейсу з адаптивним UX дизайном. Наприклад, для лікарів передбачити більш детальні медичні дані та спеціалізовані інструменти, а для пацієнтів – спрощену термінологію і можливість приховати складну інформацію.

При розробці React компонентів для медичних додатків доцільно реалізувати такі можливості:

- Перемикання між режимами "Лікар" та "Пацієнт";
- Відображення різних наборів даних для лікарів і пацієнтів;
- Спрощені тексти та термінологія для пацієнтів;
- Додаткові медичні інструменти для лікарів;
- Налаштування конфіденційності і доступу до даних.

Такий підхід дозволить створити універсальну бібліотеку компонентів з адаптивним інтерфейсом, яка задовольнятиме потреби різних груп користувачів медичних додатків.

Окрім лікарів і пацієнтів, варто враховувати вимоги компаній-замовників та інвесторів. Часто вони фокусуються на метриках залученості користувачів, конверсії та зростання аудиторії. Ці цілі не завжди збігаються з UX оптимізацією для лікарів і пацієнтів.

Тому важливо знайти баланс між потребами всіх користувачів, надаючи пріоритет якості медичних послуг та безпеці пацієнтів. Гнучка React бібліотека компонентів з адаптивним інтерфейсом стане оптимальним рішенням.

6. ВИКОРИСТАННЯ ШТУЧНОГО ІНТЕЛЕКТУ У МЕДИЦИНІ

Штучний інтелект (Artificial Intelligence – AI) відкриває значні можливості для розвитку сучасних медичних інформаційних систем та покращення їх інтерфейсів. AI дозволяє аналізувати величезні масиви медичних даних для виявлення прихованих закономірностей, підтримки прийняття клінічних рішень, прискорення діагностики тощо [10].

Наприклад, компанія Arterys використовує AI для аналізу серцевого магнітно-резонансного зображення та оцінки функції лівого шлуночка серця. Це дозволяє значно прискорити діагностику серцевих захворювань [11].

За даними Stanford Medicine (2020), AI система CheXNet демонструє точність у діагностиці пневмонії на рентгенівських знімках на рівні 96,6% – порівняно з 79% у лікарів-рентгенологів [12].

За даними Philips (2022), 67% лікарів зазначають, що на ефективність використання AI в клінічній практиці значно впливає якість інтерфейсу користувача [13].

7. ПЕРЕЛІК ПРОПОНОВАНИХ КОМПОНЕНТІВ БІБЛІОТЕКИ

Для прискорення розробки медичних додатків на React пропонується наступний набір компонентів:

- Картка пацієнта з основною інформацією;
 - Картка лікаря з необхідною інформацією;
 - Історія хвороби з хронологією записів;
 - Календар призначень і прийомів;
 - Графіки медичних показників;
 - Рецепти та призначення;
 - Модальні вікна попереджень і повідомлень;
 - Форми введення медичних даних;
 - Таблиці для відображення аналізів і досліджень;
- Елементи навігації і фільтрації даних.

Кожен компонент матиме налаштовувані властивості, щоб гнучко адаптувати інтерфейс під потреби користувачів.

Приклад створення медичного інтерфейсу на React з власними компонентами (Рис. 1):

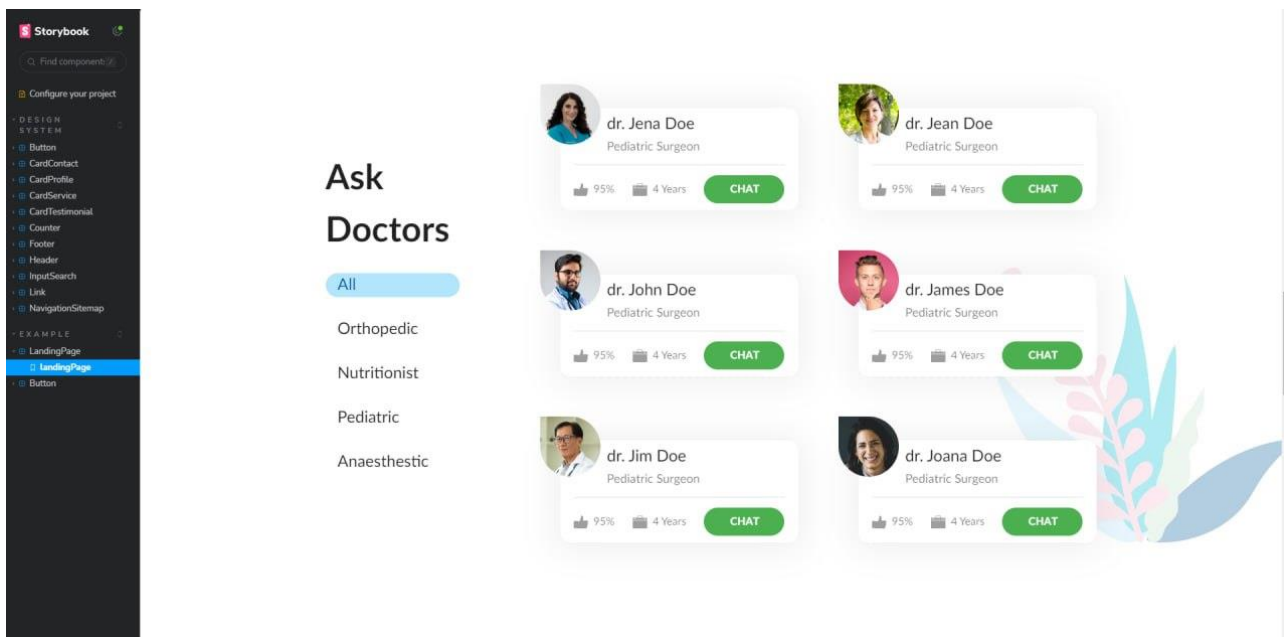


Рисунок 1. Інтерактивна картка лікаря, фільтри

Такий підхід дозволяє швидко створювати медичні інтерфейси, комбінуючи готові оптимізовані компоненти.

8. ВИСНОВКИ

Розробка спеціалізованої бібліотеки React компонентів для медичних веб додатків є перспективним напрямком, що дозволить значно пришвидшити створення сучасних медичних інформаційних платформ.

Незважаючи на зростання обсягів медичних даних та широке впровадження інформаційних технологій в інших галузях, охорона здоров'я досі істотно відстає у цифровізації. Більшість наявних медичних систем мають застарілі, неінтуїтивні інтерфейси, що ускладнює роботу лікарів та погіршує досвід пацієнтів.

Існує нагальна потреба у створенні сучасних медичних платформ зі зручними веб інтерфейсами. React дозволяє ефективно і швидко будувати такі інтерфейси з можливістю багаторазового використання компонентів. Розробка власної бібліотеки React елементів інтерфейсу спеціально для медичних додатків оптимізує цей процес.

Така бібліотека повинна враховувати потреби різних груп користувачів, зокрема лікарів, пацієнтів, адміністраторів. Використання адаптивного підходу дозволить створити універсальні компоненти з гнучкими налаштуваннями інтерфейсу. Обов'язковим є тестування UX ефективності за допомогою кількісних та якісних методів.

Застосування ReactJS та спеціалізованої бібліотеки компонентів дасть змогу прискорити розробку сучасних медичних платформ з інтуїтивними веб інтерфейсами. Це сприятиме підвищенню ефективності медичних працівників, поліпшенню обслуговування пацієнтів та розвитку галузі в цілому.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Flexera, “State of the Cloud Report”, 2021. URL: <https://www.flexera.com/about-us/press-center/flexera-releases-2021-state-of-the-cloud-report> (дата звернення 08.11.2023).
2. The Office of the National Coordinator for Health Information Technology, “Improving Clinical Workflow in EHRs”, 2019.

URL: <https://www.healthit.gov/sites/default/files/page/2021-02/ONC-LEAP-in-Health-IT-SEN-FY2021.pdf> (дата звернення 08.11.2023).

3. Stanford Medicine, “Alleviating physician burnout through EHR optimization”, 2017. URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8068431/> (дата звернення 08.11.2023).

4. Statista React usage. URL: <https://www.statista.com/statistics/1124699/worldwide-developer-survey-most-used-frameworks-web/> (дата звернення 13.11.2023).

5. NNGroup, “Quantitative vs. Qualitative Usability Testing”. URL: <https://www.nngroup.com/articles/quant-vs-qual/> (дата звернення 11.11.2023).

6. “System Usability Scale”. URL: <https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html> (дата звернення 13.11.2023).

7. Google analytics tool. URL: <https://analytics.google.com/analytics/web/> (дата звернення 13.11.2023).

8. Hotjar analytics tool. URL: <https://www.hotjar.com/> (дата звернення 13.11.2023).

9. Mixpanel analytics tool. URL: <https://mixpanel.com/> (дата звернення 13.11.2023).

10. Accenture, “AI: Built for Healthcare”, 2021. URL: <https://www.accenture.com/au-en/insights/health/artificial-intelligence-healthcare> (дата звернення 08.11.2023).

11. Arterys, “AI Assisted Cardiac MRI Software”. URL: <https://www.arterys.com/cardio-radiology-ai-platform> (дата звернення 08.11.2023).

12. Stanford Medicine, “AI system outperforms radiologists in diagnosing pneumonia”, 2020. URL: <https://news.stanford.edu/2017/11/15/algorithm-outperforms-radiologists-diagnosing-pneumonia/> (дата звернення 08.11.2023).

13. Philips, “Survey: AI in Radiology”, 2022. URL: <https://www.philips.com/a-w/about/news/archive/standard/news/articles/2022/20220714-philips-future-health-index-2022-identifies-radiology-leaders-as-the-true-early-adopters-of-ai-and-predictive-analytics-in-healthcare.html> (дата звернення 08.11.2023).

РОЗРОБКА WORDPRESS ПЛАГІНА ДЛЯ КЕРУВАННЯ ФАЙЛОВОЮ СИСТЕМОЮ

Кисельов Г.Д.¹, Таран А.І.²

Національний технічний університет України «Київський політехнічний інститут
ім. Ігоря Сікорського», Київ, Україна

¹ g.kyselov@gmail.com [0000-0003-2682-3593], ² anastasiataran2000@gmail.com

WordPress залишається найбільш популярною та універсальною системою управління контентом (CMS), яка ідеально підходить для широкого спектру проектів – від особистих сайтів до об'ємних порталів та інтернет-магазинів. Його популярність базується не лише на стандартному наборі функціоналу, але й на можливості розширення завдяки плагінам. Проте, в сучасному інструментарії для веб-розробки, такому як WordPress, виникає необхідність в більш "розумному" керуванні файлами та структурами відображення. Розробка плагіну, спрямованого на автоматизацію цих процесів, є вкрай актуальною у сучасному цифровому середовищі. Такий плагін дозволить вирішити поточні виклики і потреби користувачів, пропонуючи ефективний та інтуїтивно зрозумілий інструментарій для управління файлами та їхнього представлення. Ідея та актуальність розробки WordPress плагіна для керування файловою системою представлено у даній роботі.

Ключові слова: WordPress, CMS, плагін, веб-розробка, файловий менеджер, керування файловою системою, оптимізація та продуктивність, UX/UI веб-інтерфейсу.

1. ВСТУП

За останні роки веб-розробка стала неабияк активним та популярним напрямком, привертаючи увагу великого кола людей завдяки своїй доступності та зручності у створенні різноманітних веб-проектів. Однак із зростанням складності проектів та вимог до їхньої динамічності з'являється необхідність у високоефективних інструментах для управління контентом.

Із величезною кількістю проектів та вимогами щодо оперативності впровадження змін, системи управління контентом (CMS) стали визначальним елементом веб-розробки. Однією з найпопулярніших та універсальних CMS є WordPress [1]. Його популярність ґрунтується не лише на широких можливостях базового функціоналу, але й на можливості його розширення завдяки плагінам [2].

Незважаючи на успішність та гнучкість WordPress, сучасний веб-розвиток вимагає більш "розумного" керування файлами та структурами відображення. Саме ця прогалина стає об'єктом уваги даного дослідження. На сьогоднішній день відсутність інтелектуальної системи керування файлами у WordPress створює реальну проблему для тих, хто прагне ефективно та продуктивно управляти вмістом своїх веб-проектів.

У цьому контексті запропоновано розробку комерційного плагіну, спрямованого на раціоналізацію та автоматизацію управління файловою системою WordPress-сайтів. Цей плагін ставить за мету вирішення поточних викликів та потреб користувачів, надаючи

ефективний та інтуїтивно зрозумілий інструмент для керування файлами та їхнього візуального представлення.

Очікується, що розроблений плагін значно підвищить продуктивність користувачів, спростивши процес управління контентом на їхніх WordPress-сайтах. Це дослідження націлене на вироблення високоякісного та ефективного рішення, яке зможе забезпечити нові можливості та покращити функціональність одного з найпопулярніших інструментів веб-розробки – WordPress.

2. АНАЛІЗ ФУНКЦІОНАЛУ ТА ПЕРСПЕКТИВИ ВДОСКОНАЛЕННЯ ПЛАГІНА WORDPRESS

2.1. Аналіз поточного стану плагіна WordPress

На сьогоднішній день плагін для керування файловою системою вже успішно реалізований та функціонує як повноцінний продукт. Його базовий функціонал, такий як завантаження файлів, введення даних про рік створення та авторів завантажених матеріалів, вибір категорій, до яких відносяться завантажені матеріали і відображення цієї інформації на веб-сайтах, повністю відповідає вимогам. Проте, оскільки мета полягає в створенні комерційного продукту, який має задовольняти високим стандартам та вимогам користувачів, існуючий функціонал визнається базовим і вимагає подальших вдосконалень та розширень.

Мета дослідження – забезпечити користувачам більш інноваційний і ефективний інструмент для управління файловою системою WordPress, тому поточний плагін планується вдосконалити та розширити з урахуванням сучасних вимог та тенденцій у сфері веб-розробки.

2.2. Вимоги до плагіна WordPress

Реалізація вискоефективного та функціонального плагіна для системи управління контентом WordPress передбачає врахування ряду ключових вимог [3], які сприяють оптимальній взаємодії з платформою та забезпечують використання користувальницького досвіду. Розглядаються наступні ключові аспекти для розширення функціоналу існуючого плагіна WordPress.

2.2.1. Створення динамічного списку полів даних про файл

Однією з ключових вимог до комерційного плагіна є можливість створення та редагування динамічного списку полів даних для кожного файлу. Це означає, що користувач повинен мати можливість гнучко визначати та налаштовувати атрибути, що стосуються конкретного файлу, такі як заголовок, опис, ключові слова та інші метадані.

Ключові аспекти створення динамічного списку полів:

- Гнучкість в налаштуваннях: кожен файл може вимагати унікального набору даних. Забезпечення гнучких налаштувань для користувачів дозволить їм визначити, які саме поля даних є важливими для їхнього контенту. Наприклад, для зображення може бути важливими інші атрибути, ніж для документа або аудіофайлу.
- Динамічність та розширюваність: створення механізму для динамічного додавання нових полів інформації без потреби великих змін у вихідному коді плагіна забезпечить розширюваність [4]. Це дозволить враховувати майбутні зміни та потреби користувачів, не впроваджуючи великі оновлення.
- Індивідуальні налаштування поля: кожне поле даних повинно мати свої власні налаштування. Наприклад, текстове поле може потребувати встановлення обмежень на кількість символів, тоді як поле зображення може вимагати визначення розмірів чи дозволу на високоякісні формати.

- Підтримка різних типів даних: система повинна бути готовою обробляти різні типи даних, такі як текст, числа, дата та інші. Це дозволить користувачам пристосувати плагін для різноманітних веб-проектів та зберігати різні види інформації про файли.
- Відображення та організація інформації: створення ефективного інтерфейсу для відображення та організації цих полів даних на адміністративній панелі дозволить користувачам швидко знаходити та редагувати необхідні атрибути для кожного файлу.

Реалізація динамічного списку полів даних – це ключовий компонент для забезпечення гнучкості та індивідуального підходу до керування файловою системою в межах WordPress-сайту.

2.2.2. Додавання функціоналу для різного відображення файлів

Можливість вибору різних варіантів відображення файлів є досить важливим функціоналом для комерційного WordPress плагіна. Це забезпечить користувачам широкий спектр опцій для організації та представлення їхніх ресурсів. Ця функція створить більше можливостей для адаптації плагіна до конкретних потреб користувачів та вирізнятиме його серед інших подібних рішень.

Ключові аспекти додавання функціоналу для різного відображення файлів:

- Гнучке налаштування способу відображення: користувачі повинні мати можливість гнучко налаштовувати спосіб відображення файлів відповідно до їхніх унікальних потреб. Наприклад, можливість користувача вибирати, чи хоче він переглядати файли списком чи у вигляді плитки. Перегляд списком може бути зручним для великої кількості файлів, тоді як вигляд плиткою може бути більш естетично приємним та інформативним.
- Можливість фільтрації та пошуку: для полегшення навігації великої кількості файлів важливо включити функції фільтрації та пошуку. Користувачам буде легше знаходити необхідні файли за певними критеріями, що підвищить ефективність роботи з ресурсами.
- Зручні елементи управління: важливим елементом є додавання ефективних елементів управління для швидкого перемикання між режимами відображення. Наприклад, кнопки чи перемикачі в адміністративній панелі можуть полегшити доступ до цієї функціональності.

Додавання функціоналу для різного відображення файлів забезпечить користувачам гнучкість та індивідуальний підхід у взаємодії з файловою системою плагіна, сприяючи більш ефективному та зручному використанню ресурсів.

2.2.3. Динамічна зміна іконок для різних форматів файлів

Важливим функціоналом для управління файловою системою є можливість динамічної зміни іконок в залежності від формату файлів. Ця функціональність полегшить візуальне сприйняття користувачами різних типів файлів та забезпечить їм інтуїтивно зрозумілу інформацію про їхній зміст.

Ключові аспекти динамічної зміни іконок для різних форматів файлів:

- Розширення підтримки різноманітних форматів: врахування різноманітності форматів файлів та надання відповідних іконок для текстових документів, таких як PDF, DOC, TXT, RTF або ж зображень, аудіо- та відеофайлів. Це забезпечить повноту та адаптованість функціоналу до потреб різних веб-проектів.
- Індивідуальні налаштування користувача: надання можливості користувачам змінювати стандартні іконки або додавати власні для певних форматів файлів. Це розширить персоналізацію та дозволить користувачам адаптувати інтерфейс плагіна до своїх унікальних вимог та вподобань.

- Візуальна інформативність: динамічна зміна іконок не лише полегшить розпізнаваність форматів, але й створить візуально зручний і інформативний інтерфейс. Користувачі швидше зможуть оцінити зміст файлів та приймати необхідні рішення щодо їх використання.
- Сумісність із сучасними трендами: інтеграція із популярними бібліотеками іконок, такими як FontAwesome [5], може розширити варіанти вибору іконок та підтримати сучасні дизайн-стандарти. Це зробить інтерфейс більш сучасним та привабливим для користувачів.

Динамічна зміна іконок для різних форматів файлів розширяє можливості комерційного плагіна, забезпечуючи користувачам зручність та інтуїтивність в роботі із файловою системою WordPress-сайтів.

2.2.4. Додавання WordPress модуля

Однією з ключових ініціатив для покращення функціоналу комерційного плагіна є його інтеграція з WordPress модулем, який зможе стати ефективною альтернативою для шорткодів [6]. Це дозволить користувачам, з більш обмеженими навичками у роботі з кодом, спрощено використовувати плагін і розширить можливості його використання.

Ключові аспекти інтеграції WordPress модуля:

- Спрощена взаємодія та застосування: WordPress модуль повинен забезпечити простоту встановлення та налаштування [6], роблячи взаємодію з плагіном максимально легкою для користувачів. Це особливо важливо для тих, хто не має технічних навичок.
- Альтернатива shortcode: модуль має надавати альтернативу shortcode для виконання функціоналу плагіна. Замість використання коду [7], користувачі повинні мати змогу взаємодіяти з плагіном через інтерфейс WordPress, що зробить процес доступним для широкого кола користувачів.
- Покращений інтерфейс панелі адміністратора: модуль має вносити покращення у вигляд та функціональність адміністративної панелі, роблячи її більш інтуїтивно зрозумілою та естетично приємною для користувачів.

Інтеграція WordPress модуля стане важливим кроком у розширенні аудиторії та полегшенні користування плагіном, забезпечуючи йому максимальну доступність та популярність серед користувачів WordPress-сайтів.

2.2.5. Оновлена адмін-панелі

У контексті розширення функціоналу комерційного плагіна, значна увага приділяється оновленню та покращенню адміністративної панелі. Завдання полягає в створенні ефективного та ергономічного інтерфейсу для задоволення потреб користувачів та забезпечення легкості управління файловою системою WordPress.

Ключові характеристики оновленої адміністративної панелі:

- UI/UX дизайн: оновлена адміністративна панель буде володіти сучасним та привабливим інтерфейсом, враховуючи сучасні тенденції в дизайні. Це включає в себе використання актуальних кольорів, шрифтів та елементів дизайну, які забезпечать зручність та приємний зовнішній вигляд. Також адміністративна панель повинна бути інтуїтивно зрозумілою для користувачів будь-якого рівня технічної експертизи. Важливо забезпечити легкий доступ до всіх функцій та можливостей плагіна через зрозумілі та логічні елементи [8].
- Адаптивний дизайн: оскільки користувачі можуть отримувати доступ до адміністративної панелі з різних пристроїв, важливо забезпечити адаптивність дизайну. Адміністративна панель повинна оптимізуватися для роботи на різних розмірах екранів,

забезпечуючи однакову зручність використання як на комп'ютерах, так і на мобільних пристроях [9].

- Оптимізація робочого процесу: адміністративна панель повинна сприяти оптимізації робочого процесу користувачів, дозволяючи їм ефективно та швидко виконувати завдання з управління файловою системою без додаткового напруження.

Оновлена адміністративна панель стає ключовим компонентом комерційного плагіна, забезпечуючи користувачам зручність, ефективність та задоволення від взаємодії з продуктом.

3. ВИСНОВКИ

Представлена ідея реалізації визначає новий стандарт управління файловою системою для WordPress, ставши парадигмою ефективної, розширеної та інтуїтивно зрозумілої роботи із файлами на веб-сайтах. На відміну від рішень, що пропонують використовувати тільки базовий функціонал, модифікований плагін виступає інноваційним кроком до створення інтелектуальної системи керування файлами.

Реалізація плагіна для управління файловою системою надає ряд переваг. Даний плагін відзначається персоналізованим підходом до управління файлами, забезпечуючи динамічне створення та редагування полів даних для кожного файлу. Можливість вибору режиму відображення файлів спрощує використання системи та робить його більш зручним для користувачів з різними потребами, а інтеграція плагіна з FontAwesome, динамічна зміна іконок для різних форматів файлів та оновлена адміністративна панель створюють сучасний та естетичний інтерфейс для розробників WordPress сайтів. Проте, дана ідея має і недоліки. Реалізація нових функцій може вплинути на продуктивність плагіна, що вимагає уважного тестування та оптимізації, а введення нових можливостей може вимагати додаткового часу для навчання розробників та адаптації останніх до нового функціоналу.

Попри виклики, пов'язані з реалізацією пропонованої концепції, отримані переваги у вигляді збалансованої структури сайтів, великої гнучкості і можливості легкої інтеграції роблять її обґрунтованою та ефективною. Можливості, які вона відкриває для подальшого розвитку найпопулярнішої CMS WordPress, роблять її перспективною та значущою для проєктування веб-сайтів.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ольга С., Чому 43% всіх сайтів створено на WordPress? URL : <https://web-promo.ua/ua/blog/pochemu-43-procentov-vseh-sajtov-sozdany-na-wordpress/> (дата звертання: 22.11.2023).

2. Уроки WordPress, Що таке плагін WordPress і як його встановити? URL: <https://cityhost.ua/uk/blog/chto-takoe-plugin-wordpress-i-kak-ego-ustanovit.html> (дата звертання: 22.11.2023).

3. UpMe, Переваги WordPress. URL: <https://upme.biz/blog-ua/wordpress-advantages-ua/> (дата звертання: 22.11.2023).

4. JavaRush, Модуль 3. Java Professional, Критерії гарної архітектури ПЗ. URL: <https://javarush.com/ua/quests/lectures/ua.questservlets.level14.lecture03> (дата звертання: 22.11.2023).

5. Hostenko, Як використовувати іконки Font Awesome на вашому сайті WordPress. URL: <https://hostenko.com/uk/wpcafe/xaki/font-awesome-icons-wordpress/> (дата звертання: 22.11.2023).

6. WPHost, WordPress - Як встановити плагін, з прикладами. URL: <https://wphost.me/tips/wordpress-plugin-install-howto-with->

examples/#:~:text=Плагін%2С%20або%20як%20дехто%20може,підключити%20до%20сайту%20на%20WordPress (дата звертання: 22.11.2023).

7. Hostenko, Як використовувати шорткод WordPress. URL: <https://hostenko.com/uk/wrpsafe/навчальні-посібники/shortkodi-wordpress/> (дата звертання: 22.11.2023).

8. Projector Mag, Що таке UI/UX дизайн: як стати дизайнером та що потрібно вміти. URL: <https://prjctr.com/mag/uxui-questions#:~:text=UI%2Дизайнер%20робить%20все%2С%20щоб,%2С%20естетично%20приємних%2С%20інтерактивних%20інтерфейсів> (дата звертання: 22.11.2023).

9. AG Marketing, Адаптивний дизайн сайту: в чому переваги і чому це важливо. URL: <https://ag.marketing/blog/adaptvnyu-dyzaun-saytu/> (дата звертання: 22.11.2023).

АДАПТИВНІ ЗАСОБИ ЗАХИСТУ КОМП'ЮТЕРНИХ СИСТЕМ НА ОСНОВІ АПАРАТА НЕЙРОННИХ МЕРЕЖ

Коновал В.О.¹, Мухін В.Є.²

Національний технічний університет України «Київський політехнічний інститут
ім. Ігоря Сікорського», Київ, Україна

¹ konoval.volodymyr@lll.kpi.ua, ² v_mukhin@i.ua [0000-0002-1206-9131]

В останні роки проблема виявлення та реагування на зміни в даних в режимі реального часу стає все більш актуальною. Дана проблема полягає в тому, що з часом дані можуть змінити свій розподіл і ті класифікатори даних, які раніше показували високу точність роботи, можуть почати видавати неправильні оцінки цих даних або мітки класів. Дана проблема носить назву дрейф концепції. Методів виявлення дрейфу а також правильного реагування на нього за останні роки було розроблено немало, але всі ці методи мають ряд недоліків. Серед цих недоліків можна виділити необхідність доступу до справжніх міток класів, що в умовах реального світу є доволі витратною по ресурсам задачею, а також високий рівень помилкових спрацьовувань. В даній роботі буде розглянуто підхід до виявлення дрейфу концепції через використання невпевненості ансамблю глибоких нейронних мереж. Даний метод для виявлення дрейфу концепції не вимагає відомостей про значення істинних міток класів.

Ключові слова: дрейф концепції, штучні нейронні мережі, штучні імунні системи, адаптація, потоки даних, адаптація до дрейфу концепції, адаптація нейронних мереж

1. ВСТУП

Область класифікації даних є доволі актуальною та широкою галуззю застосування машинного навчання і особливо нейронних мереж. В даній класифікації на вхід ми отримуємо набір даних і натренована нейронна мережа повинна класифікувати ці дані і віднести їх до тієї чи іншої групи. В області безпеки дана галузь можна активно застосовувати для класифікації обличч людей на основі зйомок камер відеоспостереження або класифікації трафіку мережі і визначення, чи не було спроби проведення атак на мережі, такі як наприклад ХСС атаки чи атака сканування портів. Але з часом злочинці розроблюють нові методи атак, які нейронній мережі є невідомим. Зміна цих атак може в свою чергу впливати на розподіл даних, які нейронна мережа отримує для їх подальшої класифікації. Зміна розподілу даних, яка призводить до неправильної класифікації даних називається дрейф концепції. Для виявлення даного дрейфу було розроблено ряд алгоритмів виявлення та реагування на його різні типи але дані методи мають ряд недоліків, такі як необхідність доступу до дійсних значень класу даних, що в реальних умовах вимагає постійної присутності спеціалістів для розміток даних, а також, в деяких випадках, велика кількість помилкових спрацьовувань.

В даній роботі буде розглянутий новий методи виявлення дрейфу концепції, який використовує невизначеність ансамблю глибоких нейронних мереж. Даний метод розраховує невизначеність класифікаторів ансамблю відносно отриманих даних і використовує дану невизначеність для виявлення дрейфу концепції.

2. ДРЕЙФ КОНЦЕПЦІЇ

Поняття «дрейф концепції» використовується для опису будь-яких змін у ймовірнісному розподілі даних в потоці даних, тобто, джерело концепції даних змінюється після певних проміжків часу, що в свою чергу приводить до погіршення класифікації моделі. При виникненні дрейфу концепції, навчальна модель, яка натренована на попередніх даних, може стати неефективною для нових даних, які були взяті з іншого розподілу. Для уникнення впливу дрейфу концепції на точність передбачення, алгоритми роботи з потоками даних повинні адаптуватися до даної нестабільності в потоках даних. Існують два типи алгоритми роботи з потоками даних, які працюють з нестационарними середовищами: явні та неявні.

Явний алгоритм виявлення зміни концепції виявляють появу дрейфу концепції та адаптують модель до даного дрейфу в залежності від його типу. Неявний алгоритм, який також можуть називати пасивним, не виконують процес визначення появи дрейфу а доволі просто проводять процес оновлення моделі навчання з плином часу з певною постійною швидкістю. Очевидним недоліком неявного підходу є повільна адаптація до концепцій, що змінюються, без явних відомостей про фактичний дрейф концепції.

Сам же дрейф концепції можна поділити на реальний та віртуальний дрейф. Віртуальним дрейфом називають дрейф, коли зміни в апіорній ймовірності простору властивостей ($P(X)$) не впливає на межу прийняття рішень. На противагу віртуальному дрейфу концепції, реальний дрейф концепції вказує на зміни у ймовірності вихідних класів (y) для заданого простору входу (X) ($P(y|X)$), що в результаті призводить до зміни межі прийняття рішень.

Також реальний дрейф концепції може виникати і без зміни ймовірності простору властивостей $P(X)$.

Крім розподілу на віртуальний та реальний, дрейф концепції може відбуватися з різною частотою, яких згідно до роботи Gulcan та Can [1] можна розділити на чотири категорії: раптовий, поступовий, інкрементальний та періодичний.

- Раптовий дрейф вказує на раптову і неочікувану зміну в розподілі даних, тобто, в певний момент часу t розподіл даних раптово змінюється іншим розподілом, що призводить до гострого погіршення результатів класифікації.
- Поступовий дрейф концепції відбувається в процесі, коли кількість прикладів з нового розподілу поступово збільшується а кількість прикладів зі старого розподілу поступово зменшується, що в свою чергу призводить до поступового збільшення ймовірності вибірки нового розподілу разом зі зменшенням ймовірності вибірки попереднього розподілу.
- Інкрементальний дрейф концепції відбувається, коли в певній частині області властивостей відбуваються незначні зміни, що з плином часу призводить до поступової зміну розподілу на новий. Даний тип дрейфу відрізняється від поступового тим, що зміна розподілу відбувається за набагато більший проміжок часу, ніж при поступовому дрейфі і відповідно такий дрейф концепції важче відслідкувати в потоці даних.
- Періодичний або повторюваний дрейф концепції вказує на повторну появу деяких вже зустрінутих розподілів концепції після певного проміжку часу.

3. МЕТОДИ ВИЗНАЧЕННЯ ДРЕЙФУ КОНЦЕПЦІЇ

Дослідженню методів визначення дрейфу концепції в потоках даних присвячено велика кількість наукових робіт. Серед розроблених методів найбільш відомими та такими, що найчастіше використовуються в інших дослідницьких роботах, можна виділити методи розроблені в роботах [2–5], які будуть коротко розглянуті далі. Варто зауважити, що дані алгоритми визначають саме наявність дрейфу концепції в певному наборі даних, а визначення підтипу дрейфу та правильна адаптація до них буде розглянуто у наступному підрозділі.

Одним з основних методів визначення дрейфу концепції можна назвати метод, розроблений та продемонстрований в дослідженні [2]. В даній роботі автори продемонстрували розроблений ними метод детекту дрейфу (Drift Detection Method, DDM), який використовує рівень помилки навчальної системи вважаючи, що рівень помилки системи буде зменшуватись з кількістю проаналізованих прикладів при умові, що розподіл даних залишається стаціонарним. Якщо алгоритм помічає, що рівень помилки перевищує певне порогове значення, то він інформує що або відбувся дрейф концепції або він відбудеться у найближчому майбутньому.

В свою чергу в дослідженні [3] було представлено модифікацію методу DDM, яка носить назву EDDM (Early Drift Detection Method). Даний метод, як і метод DDM, для визначення дрейфу концепції використовує помилки точності класифікації, але EDDM замість врахування рівня помилок враховує відстань між двома помилками ведучи підрахунок стандартного відхилення та максимальної відстані і максимального стандартного відхилення. Даний метод краще себе демонструє в процесі визначення поступового дрейфу концепції при цьому також показуючи гарні результати при виникненні різкого дрейфу.

Автори дослідження [4] представили метод визначення дрейфу з використанням ковзаючого вікна під назвою Adaptive Windowing (ADWIN). Даний алгоритм використовує ковзаючі вікна, розмір яких визначається в реальному часі у відповідності до швидкості змін даних в самому вікні. Даний підхід зручний завдяки тому, що користувачеві не потрібно самому визначати параметри вікна або розмір кроку. Результати експериментів показали, що даний підхід показує кращі результати роботи ніж методи з вікнами фіксованого розміру. Тестування ефективності роботи класифікатора проводилося як на штучних так і реальних датасетах і показало доволі високі результати демонструючи низькі показники хибнопозитивних та хибнонегативних показників класифікації.

В дослідженні [5] було запропоновано метод визначення дрейфу концепції, що заснований на використанні статистичного тесту Колмогорова-Смірнова та ковзаючого вікна фіксованого розміру. Розроблений метод під назвою Kolmogorov-Smirnov Windowing, коротко KSWIN не вимагає представлення розподілу даних для своєї роботи, він може моніторити дані або розподіл продуктивності роботи для детекту дрейфу концепції. Результати експериментів на штучних датасетах демонструють кращі середні результати точності визначення дрейфу концепції у порівнянні з раніше розглянутими методами визначення дрейфу концепції: DDM, EDDM, ADWIN.

4. ВИКОРИСТАННЯ НЕВИЗНАЧЕНОСТІ МЕРЕЖІ ДЛЯ ВИЯВЛЕННЯ ДРЕЙФУ КОНЦЕПЦІЇ

Існує ряд підходів до визначення невпевненості моделі, одним з даних підходів є підхід відсіву Монте-Карло. Даний підхід полягає у застосуванні шару відсіву між внутрішніми шарами нейронної мережі. Даний шар відключає частину нейронів на одному внутрішньому шарі і в першу чергу застосовується для запобігання перенавчання моделі. При багаторазовій подачі одного прикладу у нейронну мережу з даними шарами відсіву ми отримаємо масив передбачень, які можуть між собою відрізнитися у том, оскільки шар відсіву щоразу буде відмикати різні нейрони шару. Варто зауважити, що різниця в результатах передбачень для прикладу можлива лише в тому випадку, якщо нейронна мережа невпевнена відносно цих даних, оскільки при класифікації відомих даних, розбіжність у передбаченнях буде вкрай малою або буде відсутня взагалі. Далі використовуючи даний масив даних ми можемо отримати усередненні передбачення моделі відносно певного набору даних а потім розрахувати ентропію Шенона для даного набору, що і буде являти собою невпевненість моделі відносно класифікованих даних.

Підхід, розроблений в даній роботі, замість розрахунку невизначеності з використанням відсіву Монте-Карло, використовує передбачення ансамблю глибоких класифікаторів. Причин для використання ансамблю класифікаторів декілька. Перша причина полягає в тому, що ансамбль класифікаторів показує вищу точність передбачень у порівнянні з одним класифікатором. По друге, працюючи з ансамблем у випадку перенавчання одного його елементів, його можна легко замінити новим класифікатором, який буде навчений на тих даних, на яких в свій час тренувалися початкові класифікатори з набору.

Сам процес класифікації потоку даних буде відбуватися наступним чином:

1. З потоку даних виділяється набір даних фіксованого розміру, в даному випадку розміром 1000, і надсилається ансамблю на класифікацію. Якщо кількість даних в потоці менша за фіксований розмір, то надсилається ті, що залишилися без очікування появи нових даних.

2. Кожен з класифікаторів ансамблю робить власні передбачення відносно отриманого набору даних, які збираються в масив.

3. Отримані дані використовуються для розрахунку передбачень ансамблю і розрахунку невизначеності ансамблю. Передбачення ансамблю розраховуються наступною формулою:

$$p_{ens}(x) = \frac{1}{N} \sum_{i=1}^N p_i(x),$$

де

x – набір вхідних даних;

$p_i(x)$ – передбачення окремо взятого класифікатора відносно набору даних x ;

N – кількість класифікаторів у ансамблі.

Невизначеність же ансамблю розраховується як ентропія Шенона і має наступний вигляд:

$$u_{ens}(x) = - \sum_{i=1}^N p_i(x) * \log_2(p_i(x)).$$

4. Отримана невизначеність подається на вхід алгоритму виявлення дрейфу концепцій ADWIN який розбиває отримані дані на частини і розраховує їх усередненні значення. І у випадку перевищення цих значень встановленого порогу, то алгоритм повідомляє про виникнення дрейфу

5. У випадку виникнення дрейфу концепції відбувається донавчання ансамблю на даних до його виникнення, перезапуск алгоритму ADWIN і встановлення індексу, на якому виник дрейф як такого, починаючи з якого будуть виділятися нові партії даних з потоку і перехід до пункту 1.

6. Якщо дрейф концепції не виник, перехід до пункту 1.

5. РЕЗУЛЬТАТИ ДОСЛІДЖЕНЬ

В якості датасетів для досліджень було обрано ряд штучних датасетів, які були згенеровано спеціально з дрейфами концепцій, та ряд реальних датасетів. Дане рішення було прийняте у зв'язку з тим, що реальні датасети можуть не містити в собі дрейфи концепцій або бути недостатньо наочними. Порівняння буде проводитися між розробленим алгоритмом, тобто алгоритмом використання невизначеності ансамблю і методом ADWIN, методом ADWIN з використанням помилки ансамблю на даних а також методом KSWIN.

Розглянемо результати роботи алгоритму на одному з наборів даних.

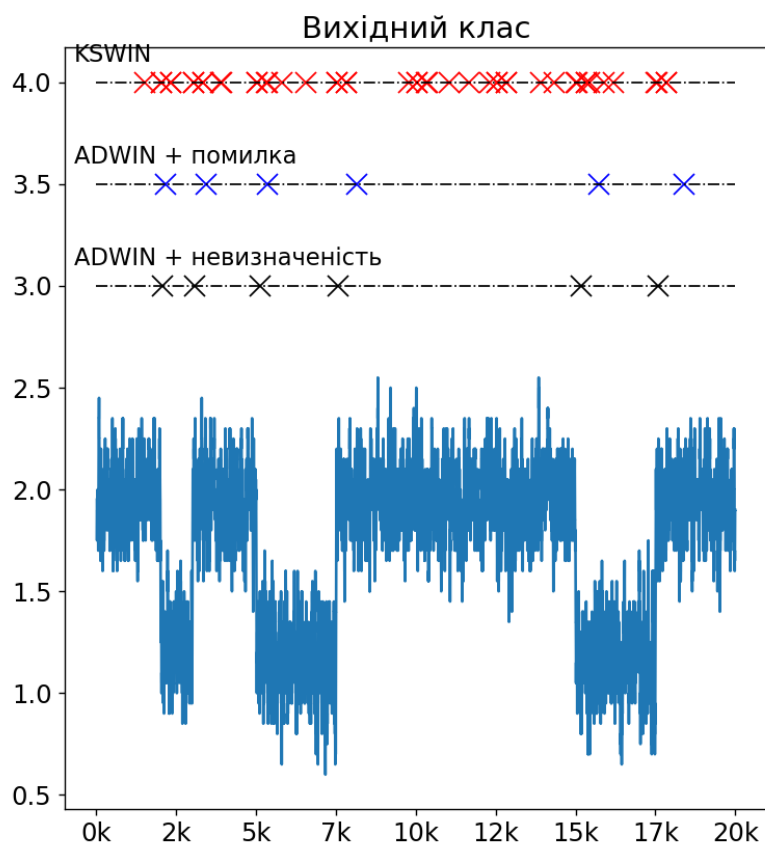


Рисунок 1. Приклад роботи алгоритмів

На даному рисунку зображено приклад роботи алгоритмів виявлення дрейфу концепцій. Чорні мітки відповідають тим місцям, де розробленим в даній роботі алгоритмом, червоним місця дрейфу, визначені алгоритмом ADWIN з використанням помилки класифікації. Як можна помітити з даного зображення, кількість дрейфів, які були визначені алгоритмом KSWIN помітно більше, ніж у розробленого алгоритму з використанням невизначеності.

6. ВИСНОВКИ

Результати експериментів демонструють, що розроблений метод виявлення дрейфів концепцій демонструє меншу кількість спрацьовувань ніж алгоритм KSWIN і на відмінну від алгоритму ADWIN з використанням рівня помилки не вимагає доступу до справжніх міток класів. Але даний алгоритм в поточній реалізації має і ряд недоліків. Найбільшим недоліком даного алгоритму можна назвати те, що для його донавчання після визначення дрейфу концепції він досі вимагає наявності доступу до реальних міток класу. Даний недолік усунути на даний момент неможливо, але можливо зменшити кількість даних для розмічування через введення технік активного навчання. Також серед недоліків даного алгоритму можна визначити більш повільну роботу у порівнянні з алгоритмом KSWIN. Причиною цього слугує те наявність процесу донавчання алгоритму, що при великій кількості нейронних мереж в ансамблі помітно впливає на швидкість роботи, що в свою чергу погано підходить до мереж з великим об'ємом даних.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Gulcan EB, Can F (2023) Unsupervised concept drift detection for multi-label data streams. *Artificial Intelligence Review* 56(3):2401–2434
2. Gama, João & Medas, Pedro & Castillo, Gladys & Rodrigues, Pedro. (2004). Learning with Drift Detection. *Intelligent Data Analysis*. 8. 286-295. 10.1007/978-3-540-28645-5_29.
3. Baena-García, Manuel & Campo-Ávila, José & Fidalgo-Merino, Raúl & Bifet, Albert & Gavald, Ricard & Morales-Bueno, Rafael. (2006). Early Drift Detection Method.
4. Bifet, Albert & Gavaldà, Ricard. (2007). Learning from Time-Changing Data with Adaptive Windowing. *Proceedings of the 7th SIAM International Conference on Data Mining*. 7. 10.1137/1.9781611972771.42.
5. Raab, Christoph & Heusinger, Moritz & Schleif, Frank-Michael. (2020). Reactive Soft Prototype Computing for Concept Drift Streams. *Neurocomputing*. 416. 10.1016/j.neucom.2019.11.111.

МЕТОДОЛОГІЇ РОЗРОБКИ МОБІЛЬНИХ ДОДАТКІВ ДЛЯ МЕДИЧНИХ ІНФОРМАЦІЙНИХ СИСТЕМ НА ПЛАТФОРМИ iOS

Люлька Р.О.¹, Харченко К.В.²

КПІ ім. Ігоря Сікорського, Кафедра системного проектування, Київ, Україна

¹ lyulka.ruslan@iit.kpi.ua, ² konst1970@gmail.com

У час цифрових інновацій, важливість розробки мобільних додатків для медичних систем особливо зростає. Платформа iOS, відзначаючись безпекою та стабільністю, вимагає специфічних підходів у розробці. Ця робота фокусується на методологіях створення таких додатків, акцентуючи на їх відповідності до високих стандартів медичної галузі на iOS.

Ключові слова: Медичні інформаційні системи, мобільні додатки, методології розробки, Apple Watch.

1. ВСТУП

У сучасному світі, де медицина тісно переплітається з цифровими технологіями, мобільні додатки відіграють ключову роль у підвищенні ефективності медичних послуг та забезпеченні доступності охорони здоров'я. Ці додатки не тільки спрощують процес діагностики та лікування, але й забезпечують надійний засіб комунікації між пацієнтами та медичними працівниками, зокрема в контексті віддаленого моніторингу стану здоров'я та електронного здоров'я.

2. ОСОБЛИВОСТІ ПЛАТФОРМИ iOS ДЛЯ МЕДИЧНИХ ДОДАТКІВ

Платформа iOS від Apple відома своєю високою продуктивністю та надійністю, що робить її ідеальною для розробки медичних додатків. Завдяки оптимізованій архітектурі операційної системи та інтеграції з потужним апаратним забезпеченням [1], iOS дозволяє створювати швидкі та ефективні додатки. Це особливо важливо для медичних застосунків, які потребують швидкої обробки великих обсягів даних та надійності у реальному часі. Також платформа надає розширені можливості для інтеграції з зовнішніми пристроями та сенсорами, що є критично важливим для моніторингу здоров'я та діагностики.

Безпека та конфіденційність даних є основоположними принципами при розробці медичних додатків, і iOS пропонує суворі стандарти та інструменти для їх дотримання. Система безпеки iOS включає шифрування даних на рівні пристрою, захист доступу через Face ID або Touch ID, а також безпечну роботу з хмарними послугами. Це гарантує, що чутливі медичні дані, такі як історія хвороби, лабораторні результати, особисті відомості пацієнтів, захищені від несанкціонованого доступу. Крім того, Apple надає набір рекомендацій та найкращих практик для розробників, щоб забезпечити відповідність додатків законодавчим стандартам, таким як HIPAA в США, щодо обробки та зберігання медичних даних.

3. МЕТОДОЛОГІЇ РОЗРОБКИ

Розвиток мобільних медичних додатків вимагає збалансованого підходу до управління проектами та розробки, щоб забезпечити якість, безпеку та відповідність стандартам. На сьогоднішній день, існує кілька популярних методологій [2], які використовуються у розробці програмного забезпечення.

Гнучка методологія Agile заснована на ітеративному розвитку, де проект розбивається на невеликі частини, що дозволяє команді швидко реагувати на зміни вимог та умов проекту. Agile ідеально підходить для проектів, де важливо швидко адаптуватися до нових вимог або змін у ринкових умовах.

Scrum, як один з підходів Agile, що включає в себе короткі цикли розробки (спринти), регулярні зустрічі команди та постійний зворотній зв'язок. Scrum підходить для проектів з частими оновленнями та активною участю замовника у процесі розробки.

Традиційний підхід Waterfall, де кожна стадія проекту (аналіз вимог, проектування, розробка, тестування, впровадження) виконується послідовно. Ця методологія часто використовується у великих проектах з чітко визначеними вимогами та мінімальними змінами в процесі розвитку.

Адаптація методологій під специфіку медичних додатків.

У контексті медичних додатків, важливо адаптувати ці методології, враховуючи специфіку галузі. Особлива увага приділяється контролю якості, безпеці даних та відповідності медичним стандартам. Наприклад, у методології Agile можна інтегрувати додаткові етапи перевірки безпеки та відповідності стандартам перед кожним релізом. Також важливо забезпечити гнучкість у внесенні змін, що є ключовим у медичному середовищі, де вимоги можуть швидко змінюватися.

4. ІНТЕГРАЦІЯ З МЕДИЧНИМИ ІНФОРМАЦІЙНИМИ СИСТЕМАМИ

Ефективна інтеграція мобільних додатків з медичними інформаційними системами є ключовим аспектом у розробці медичних додатків на платформі iOS. Цей процес включає в себе два основних компоненти.

Технології інтеграції (APIs, обмін даними).

Використання API (Application Programming Interface): API дозволяють мобільним додаткам взаємодіяти з медичними інформаційними системами, обмінюючись даними та функціональністю [3]. Це може включати доступ до медичних записів, обробку запитів на аналізи, оновлення стану пацієнтів тощо.

Обмін даними, при якому важливо забезпечити надійний та безпечний обмін даними між мобільними додатками та медичними системами. Це включає використання стандартів, таких як HL7 або FHIR, для ефективною інтеграції та обміну даними у відповідності до медичних та юридичних норм.

Забезпечення сумісності та безперервності роботи.

Медичні додатки повинні бути сумісними з різними інформаційними системами, що використовуються в медичних установах. Це вимагає гнучкого підходу до розробки та тестування додатків.

А також слід брати до уваги безперервність роботи. Медичні додатки мають забезпечувати стабільну та безперебійну роботу, оскільки вони часто використовуються в критичних ситуаціях. Важливо враховувати аспекти надійності та відмовостійкості при розробці.

5. ІНТЕГРАЦІЯ З APPLE WATCH

Особливо відзначимо внесок Apple Watch у розвиток мобільних медичних додатків на платформі iOS. Завдяки своїм функціям моніторингу здоров'я, таким як відстеження серцевого ритму та рівня кисню в крові, ці розумні годинники стають невід'ємною частиною сучасних медичних додатків [4]. Вони не тільки надають цінні дані для постійного моніторингу стану

здоров'я, але й можуть використовуватися для надання негайної допомоги в критичних ситуаціях.

Інтеграція медичних додатків з Apple Watch відкриває можливості для персоналізації медичного догляду. Використовуючи детальну інформацію, отриману з годинника, медичні додатки можуть надавати індивідуальні рекомендації, базуючись на щоденній активності та фізичних показниках користувача. Це не тільки підвищує ефективність лікування, але й сприяє збору даних для медичних досліджень, що розширює наші знання про здоров'я населення.

Крім того, Apple Watch ідеально інтегруються з екосистемою iOS, що дозволяє медичним додаткам працювати ефективно як на iPhone, так і на годинниках. Така інтеграція не тільки покращує функціональність медичних додатків, але й робить їх більш привабливими для кінцевих користувачів, забезпечуючи їм сучасні засоби для контролю над своїм здоров'ям.

6. ПРИКЛАДИ УСПІШНИХ ПРОЕКТІВ

У цьому розділі розглянемо кілька вдало реалізованих проектів розробки мобільних медичних додатків для платформи iOS. Ці приклади демонструють, як інноваційні підходи та ефективне використання технологій можуть сприяти поліпшенню медичних послуг та підвищенню якості догляду за пацієнтами.

Додаток для віддаленого моніторингу пацієнтів.

Опис. Розробка додатку, який дозволяє медичному персоналу відстежувати стан пацієнтів в реальному часі, збираючи дані з носимих пристроїв.

Висновки. Проект показав важливість інтеграції з різними пристроями та платформами, а також необхідність забезпечення високого рівня безпеки даних.

Мобільний додаток для управління лікарняними записами.

Опис. Створення додатку, що надає лікарям та медсестрам можливість швидкого доступу до медичних записів пацієнтів.

Висновки. Ефективність додатка залежить від його зручності та інтуїтивності використання, що особливо важливо у швидкоплинному медичному середовищі.

Додаток для консультацій з лікарями онлайн.

Опис. Розробка платформи для онлайн-консультацій, яка дозволяє пацієнтам отримувати медичні поради віддалено.

Висновки. Проект наголошує на важливості створення безпечного каналу зв'язку між лікарями та пацієнтами, а також на необхідності інтеграції з іншими медичними системами.

Ці приклади відображають різні аспекти розробки медичних додатків, включаючи технічні виклики, важливість користувацького досвіду, а також необхідність дотримання високих стандартів безпеки та конфіденційності. Вони свідчать про те, що успішні медичні додатки поєднують у собі інноваційні технічні рішення з глибоким розумінням потреб кінцевих користувачів.

7. ВИКЛИКИ ТА ПЕРСПЕКТИВИ

Розвиток мобільних медичних додатків стоїть перед рядом викликів, які вимагають уваги та інноваційних підходів [5]. Водночас, галузь має значний потенціал для розвитку, обіцяючи нові можливості для поліпшення медичного обслуговування.

7.1. Поточні Виклики

1. Забезпечення безпеки та конфіденційності даних. Одним з найбільших викликів є захист чутливої медичної інформації, що обробляється додатками.

2. Інтеграція з існуючими медичними системами. Ефективна сумісність та інтеграція з різними медичними інформаційними системами залишаються складним завданням.

3. Універсальність та адаптивність. Розробка додатків, що враховують різноманіття медичних потреб та пристроїв користувачів.

4. Регуляторні вимоги. Слідкування за змінами в законодавстві та відповідність регулятивним нормам у різних регіонах.

7.2. Майбутні тренди та напрямки розвитку.

Штучний інтелект та машинне навчання. Використання AI для покращення діагностики, персоналізації лікування та аналізу медичних даних.

Інтероперабельність даних, що передбачає розвиток стандартів для поліпшення обміну даними між різними медичними системами.

Телемедицина як розширення можливостей для віддаленого моніторингу та консультацій, особливо в умовах глобальної пандемії.

Персоналізована медицина, що передбачає розробку додатків, що забезпечують індивідуальний підхід до кожного пацієнта, з урахуванням його унікальних потреб та історії здоров'я.

У світлі цих викликів та перспектив, розвиток мобільних медичних додатків на платформі iOS відкриває широкі можливості для інновацій та покращення якості медичного обслуговування на глобальному рівні.

8. ВИСНОВКИ

У світлі стрімкого технологічного прогресу та росту потреб у сфері охорони здоров'я, розвиток мобільних додатків для медичних систем на платформі iOS відіграє ключову роль у модернізації медичного обслуговування. Ці додатки відкривають нові можливості для покращення якості догляду, підвищуючи доступність та ефективність медичних послуг.

Адаптація та впровадження інноваційних технологій у розробці мобільних медичних додатків на iOS перетворюють медичну практику, роблячи її більш інтегрованою та пацієнтоорієнтованою. Майбутнє медичних додатків обіцяє ще більшу інтеграцію з розширеними технологіями, такими як штучний інтелект та телемедицина, що розкриють нові горизонти у лікуванні та моніторингу пацієнтів.

Однак, слід враховувати виклики, пов'язані з безпекою даних, інтеграцією з існуючими системами та регуляторними вимогами. Ефективна реалізація цих додатків вимагає глибокого розуміння медичних потреб, технічної експертизи, та комплексного підходу до розробки, включаючи вибір відповідних методологій та стратегій інтеграції. Таким чином, розвиток мобільних додатків на платформі iOS стане визначальним фактором у формуванні майбутнього медичного обслуговування, оскільки вони пропонують необхідну гнучкість, доступність та персоналізацію у догляді за пацієнтами.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. iOS App Development in Healthcare – Features & Cost URL: <https://www.appventurez.com/blog/ios-app-development-in-healthcare> (дата звернення: 26.11.2023).

2. Development Methodologies. URL: <https://www.uptech.team/blog/software-development-methodologies> (дата звернення: 28.11.2023).

3. Zhang, R.; Burgess, E.R.; Reddy, M.C.; Rothrock, N.E.; Bhatt, S.; Rasmussen, L.V.; Butt, Z.; Starren, J.B. Provider perspectives on the integration of patient-reported outcomes in an electronic health record. JAMIA 2019

4. Apple Watch: Creating a Connected Health Ecosystem. URL: <https://www.boston-technology.com/blog/apple-watch-creating-a-connected-health-ecosystem> (дата звернення: 28.11.2023).

5. Mobile healthcare. URL: https://www.researchgate.net/publication/340917395_Mobile_healthcare_opportunities_challenges_prospects (дата звернення: 28.11.2023).

РОЗРОБКА І КОНФІГУРАЦІЯ ІНТЕЛЕКТУАЛЬНОЇ РОЗПОДІЛЕНОЇ СИСТЕМИ КОНТРОЛЮ СПОЖИВАННЯ ЕЛЕКТРОЕНЕРГІЇ В БАГАТОКВАРТИРНОМУ БУДИНКУ

Мунтян Д.М.¹, Кисельов Г.Д.²

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Інститут прикладного системного аналізу, Київ, Україна

¹ daniilmuntjan@gmail.com, ² g.kyselov@gmail.com

Актуальність теми обумовлена викликами, з якими наша країна зіткнулась останньої зими. Обстріли та енергетична криза виявили потребу в оптимізації споживання енергоресурсів та пошуку ефективних шляхів їх економії. Створення інтелектуальної системи, яка б дозволяла в разі критичних ситуацій оперативно регулювати навантаження та обмежувати споживання електроенергії, не лише сприятиме підвищенню енергетичної безпеки, а й допоможе мінімізувати незручності для громадян. Це вирішить частину проблем, які виникають під час екстремальних ситуацій та може допомогти уникнути нераціональних відключень у багатоквартирних будинках, зокрема, за рахунок надмірного споживання окремими квартирами.

Ключові слова: контроль споживання електроенергії, енергоефективність, оптимізація, енергетична криза, IoT.

1. ВСТУП

В умовах глобальної цифровізації та активного впровадження концепцій розумного будинку (Smart Home), інтеграція інноваційних IoT-рішень у системи управління енергоспоживанням набуває особливої актуальності [1]. Ринок нерухомості швидко реагує на новітні технологічні тренди, намагаючись збільшити комфорт та енергоефективність будівель. Це стає можливим завдяки застосуванню сучасних систем, здатних до автоматизації та гнучкого реагування на зміни в енергетичній інфраструктурі [2]. Розуміння цих процесів та адаптація до них є ключовими для забезпечення стійкості та надійності в управлінні великими житловими комплексами в сучасних умовах.

Непередбачуваність енергопостачання стала реалією сучасної України, особливо в умовах зимових обстрілів, що наклали жорсткі обмеження на використання електроенергії. Зіткнувшись із необхідністю раптового та частково необґрунтованого відключення житлових масивів, виникає питання про розробку більш витончених та ефективних систем управління енергоспоживанням. Ця дипломна робота спрямована на розробку інтелектуальної розподіленої системи, яка дозволить зберегти живлення будинку при обмежених ресурсах генерації електроенергії.

У даній статті концентрується увага на аналізі одного багатоквартирного будинку, проте важливо підкреслити, що розроблене програмне забезпечення має здатність ефективно адаптуватися та масштабуватися для керування енергоспоживанням у масштабах необмеженої кількості будівель. Така гнучкість обумовлена принципами, закладеними в основу даної сервіс-орієнтованої хмарної архітектури.

Основними характеристиками системи є універсальність, абстракція та розширюваність (система може бути легко модифікована чи доповнена новими функціями або компонентами).

Тобто за мету ставиться абстрагуватись від конкретних джерел вхідних даних та зосередитись на програмній частині реалізації, що буде працювати з будь-яким «залізом» та в будь-якому середовищі виконувати поставлені концептуальні задачі. Що конкретно це означає – розберемо у даній статті.

2. КОНТЕКСТ ТА МОТИВАЦІЯ

Зима 2022–2023 років увійшла в історію України як час випробувань, коли доля кинула виклик нашій енергетичній системі. Зіткнувшись із несподіваними масовими відключеннями електроенергії, ми, як суспільство, побачили які вразливі місця має наша інфраструктура. Рисунок 1, який ви можете побачити нижче, слугує свідченням тих перешкод, з якими ми зіткнулися. Відключення електроенергії не тільки призводить до дискомфорту громадян, але й ставить під загрозу функціонування критично важливих структур.

	00:00	01:00	02:00	03:00	04:00	05:00	06:00	07:00	08:00	09:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00	19:00	20:00	21:00	22:00	23:00	
Понеділок							✘	✘	✘	✘						✘	✘	✘	✘						
Вівторок	✘	✘	✘	✘						✘	✘	✘	✘						✘	✘	✘	✘			
Середа				✘	✘	✘	✘						✘	✘	✘	✘							✘	✘	✘
Четвер	✘						✘	✘	✘	✘						✘	✘	✘	✘						
П'ятниця	✘	✘	✘	✘						✘	✘	✘	✘						✘	✘	✘	✘			
Субота				✘	✘	✘	✘						✘	✘	✘	✘							✘	✘	✘
Неділя	✘						✘	✘	✘	✘						✘	✘	✘	✘						

Рисунок 1. Графік відключення світла у Києві

Дефіцит генерації, що впливає з ряду причин, включаючи природні стихії та наслідки воєнних дій, є критичним чинником, який спричиняє згадане перенавантаження системи. Це призводить до каскадних збоїв, що можуть зачіпати цілі райони, залишаючи їх без електропостачання через неможливість вузлів передавати необхідну потужність.

Наразі, електромережі поділяються на категорії надійності. Приватні оселі та комерційні об'єкти належать до третьої категорії приєднання, тоді як друга категорія включає системи, які важливо утримувати в робочому стані, наприклад, вуличне освітлення та світлофори. Найвищий пріоритет надається об'єктам першої категорії, до яких належать поліцейські ділянки, військові об'єкти та медичні установи.

Проте, ця класифікація є надмірно абстрактною та не враховує динаміку індивідуального споживання. Виходом із ситуації може стати більш деталізований підхід, який включає декомпозицію енергетичної системи до рівня окремих квартир. Така деталізація дозволить розподілити навантаження більш раціонально, уникнути загальнобудинкових відключень та забезпечити енергію для тих, хто її потребує найбільше.

Основною мотивацією цієї роботи є створення системи, яка здатна оптимізувати споживання електроенергії на рівні окремого житлового комплексу. Це рішення спрямоване на попередження відключень електроенергії в умовах, коли непомірне споживання невеликої кількості квартир може спричинити неприємності для всього будинку.

3. АРХІТЕКТУРА

На рисунку 2 представлена схема інтелектуальної розподіленої системи контролю споживання електроенергії для багатоквартирного будинку. Ця схема ілюструє зв'язок між окремими компонентами системи та їх взаємодію з централізованим сервером, який виступає як командний центр обробки та управління даними.

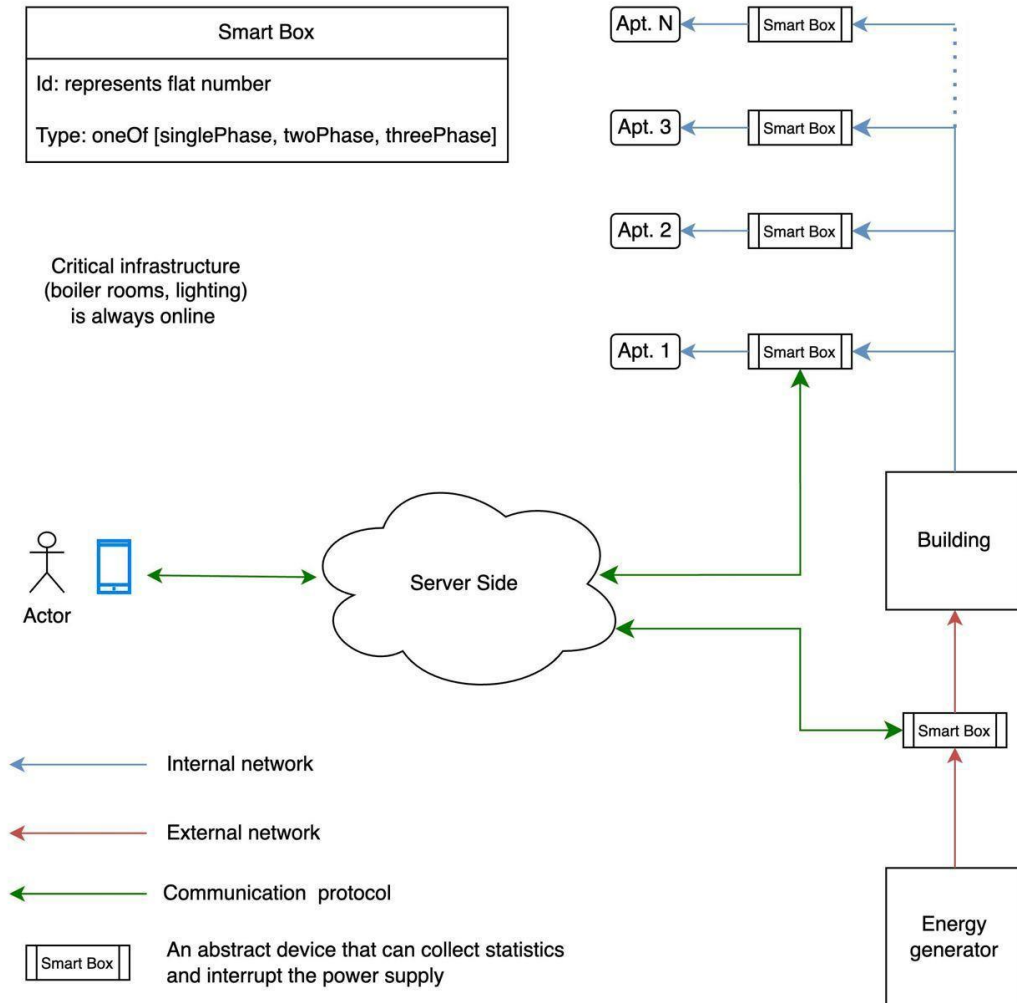


Рисунок 2. Hardware diagram

Розберемо всі сутності, представлені вище.

Smart Vox – це ключовий елемент системи, інноваційний IoT пристрій, що надає змогу вимірювати споживання електроенергії у кожній квартирі та у разі необхідності переривати електропостачання. Кожен Smart Vox має унікальний ідентифікатор, який відповідає номеру квартири, та налаштований на роботу з однофазним, двофазним або трифазним типом електропостачання, забезпечуючи гнучкість та точність у відповідності до специфікацій споживання.

Ядро системи становить мережа цих пристроїв Smart Vox, яка забезпечують зв'язок між індивідуальними квартирами та централізованим сервером, використовуючи головний мікросервер в будинку. Ця складова з'єднується з іншими Smart Vox через GSM-мережу [3], що стандартизує комунікаційні процеси в рамках сучасних IoT-рішень.

Централізована серверна сторона функціонує як «мозок» системи, що здійснює обробку вхідних даних, вироблення команд та диспетчеризацію завдань. Вона аналізує та обробляє інформацію, отриману від кожного Smart Vox [4]. Критично важливі системи, такі як котельні та освітлення, постійно отримують живлення для забезпечення стабільності життєво необхідних функцій.

Користувачі системи, або актори, здебільшого мешканці будинку, отримують інтерактивні сповіщення та можливість керування своїм споживанням через інтуїтивно зрозумілий інтерфейс, доступний через веб або мобільні додатки. Використовуючи ці інструменти, мешканці можуть реагувати на повідомлення про високе споживання в реальному часі, що дозволяє їм приймати необхідні заходи для зниження споживання або підготовки до можливого відключення електроенергії.

Енергогенератор – це джерело генерації електроенергії для будинку. На даному етапі ми розглядаємо звичайний генератор, але система відкрита до інтеграції з різними джерелами енергії.

Зазначена архітектура вирізняється своєю масштабованістю та легкістю інтеграції в існуючі енергетичні системи, що робить її ідеальною не лише для нових «smart houses», але й для модернізації старих будинків [5]. Її універсальність полягає в незалежності від віку чи типу будівлі, надаючи можливість кожному житловому комплексу стати більш енергоефективним та відповідальним у використанні енергоресурсів.

Ця система є прикладом того, як передові технології можуть сприяти створенню більш сталого та енергетично збалансованого суспільства, відповідаючи на виклики енергетичної кризи та високого споживання ресурсів, і в той же час забезпечуючи високий рівень комфорту для своїх користувачів.

4. ХАРАКТЕРИСТИКИ СИСТЕМИ

Основними характеристиками сучасного хмарного продукту є функціональні та нефункціональні вимоги. Їх чітке визначення є надважливим етапом розробки, системи, особливо коли мова йде про сервіс-орієнтовані рішення (service-oriented architecture) [6].

Функціональні вимоги встановлюють конкретні задачі та функції, які програма має виконувати, тоді як нефункціональні вимоги охоплюють очікування до продуктивності, надійності та інтерфейсу користувача. Ретельне розмежування цих вимог є ключем до створення надійної системи, що здатна ефективно вирішувати поставлені завдання, забезпечуючи при цьому високий рівень задоволення користувачів [7].

Основні функціональні вимоги:

- Моніторинг споживання: програмне забезпечення систематично обробляє дані споживання електроенергії в реальному часі, отримані від Smart Vox пристроїв, забезпечуючи точне та актуальне відображення енергетичного профілю кожної квартири.
- Збереження даних: всі зібрані та оброблені дані систематично записуються в базу даних для створення історичного архіву, що використовується для аналітики, звітності та стратегічного планування.
- Віддалене управління: при необхідності, система має можливість дистанційно відключити або підключити електроенергію до окремих квартир або будинку в цілому. Даний функціонал через API інтерфейс надає мережа IoT пристроїв Smart Vox).
- Автоматичне реагування на перевантаження: в разі виявлення або наближення до критичної точки навантаження на генератор, система автоматично вживає заходів для запобігання відключення всього будинку від електромережі, використовуючи для цього систему сповіщення та силового відключення в критичному випадку.

Основні нефункціональні вимоги [8]:

- Доступність (availability): система доступна для користувачів більшу частину часу, мінімізуючи періоди простою.
- Розширюваність (extensibility): архітектура системи дозволяє легке додавання нового функціоналу або інтеграцію з іншими системами без значного перепроектування існуючої структури.
- Спостережливість (observability): ключова характеристика хмарного сервісу — спостережливість, яка дозволяє розробникам та адміністраторам системи відстежувати, аналізувати та розуміти поточний стан системи та її історичну поведінку для швидкого виявлення та вирішення проблем.
- Масштабованість (scalability): система підтримує зростання кількості даних та користувачів без втрати продуктивності.
- Відмовостійкість (fault tolerance): система здатна продовжувати роботу навіть у випадку часткових збоїв, з мінімальним впливом на загальну функціональність.

5. ВИСНОВКИ

Враховуючи сучасні проблеми та потреби у сфері енергетики, можна впевнено стверджувати, що впровадження запропонованої системи може значно вдосконалити ефективність використання електроенергії та сприяти стабілізації енергетичної системи країни, адаптуючи її до непередбачуваних змін та викликів сучасності. Ще на початковій стадії розробки системи було проведено ґрунтовне дослідження, включаючи консультації з електротехнічними спеціалістами та енергетиками, що дозволило забезпечити реалістичність та практичність проекту.

Розроблена система є унікальним та інноваційним рішенням, яке не має прямих аналогів на ринку. Її універсальність дозволяє застосовувати систему до різних типів будинків, без залежності від їх конструктивних особливостей чи віку. Масштабування системи можливе без втрати продуктивності, навіть при збільшенні кількості підключених будинків, за рахунок інноваційних сервіс-орієнтованих підходів.

Інтеграція системи в існуючу енергетичну інфраструктуру не потребує кардинальних змін або додаткових витрат на реконструкцію електричних мереж, оскільки вона розроблена для використання з наявним обладнанням. Це говорить про високу адаптивність та економічну доцільність впровадження системи.

Соціальний аспект реалізації такого рішення є не менш важливим. Оскільки проект має прямий вплив на повсякденне життя мешканців, ключовим є досягнення загальної згоди та встановлення ефективних домовленостей серед них. Важливо, щоб система була використана не лише як технічний інструмент, а й як засіб для зміцнення взаєморозуміння та співпраці між мешканцями. Це вимагає ретельного врахування різних думок та потреб, відкритого діалогу, а також розробки гнучких механізмів для досягнення консенсусу щодо способів управління споживанням електроенергії. Такий підхід не лише сприяє формуванню відповідального ставлення до використання ресурсів, але й підкреслює важливість спільного вирішення проблем, пов'язаних з енергетичними обмеженнями та їх оптимізацією.

Отже, потенціал розвитку системи є значним. У цій роботі як джерело енергії розглядається використання генератора, однак система може бути адаптована до співпраці з великими енергопостачальниками, такими як ДТЕК. У випадках дефіциту генерації енергії, система може отримувати сигнали від постачальників та адекватно реагувати, мінімізуючи навантаження та запобігаючи можливим відключенням.

Таким чином, розроблена система має всі шанси стати важливим інструментом у руках житлово-комунальних організацій, сприяючи сталому розвитку енергетичної сфери та підвищенню якості життя в багатоквартирних будинках.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Carol L. Stimmel та Mark G. Milroy, "IoT and Smart Building Technologies: An Integrated Approach", ISBN: 978-1804619865
2. Keith Moss, "Energy Management in Buildings: The Earthscan Expert Guide", ISBN: 978-0415706469.
3. IoT Journal, "Рішення для розумного будинку", URL: <https://iotji.io/ru/solutions-ru/> (дата звертання: 21.11.2023).
4. Smart МАІС, "Системи розумного будинку", URL: <https://smart-maic.com/> (дата звертання: 21.11.2023).
5. AWS, "Що таке сервіс-орієнтована архітектура", URL: <https://aws.amazon.com/ru/what-is/service-oriented-architecture/> (дата звертання: 22.11.2023).
6. Wikipedia, "Service-Oriented Architecture", URL: <https://w.wiki/8JrA> (дата звертання: 22.11.2023).
7. Thomas Erl, "Service-Oriented Architecture: Concepts, Technology, and Design", ISBN: 978-0131858589.
8. Dan Woods, Thomas Mattern, "Enterprise SOA: Designing IT for Business Innovation", ISBN: 978-0596102388.

СЕРВЕР ДЛЯ МОБІЛЬНИХ ЗАСТОСУНКІВ ЯК СЕРВІС

Науменко Є.О.¹, Булах Б.В.²

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Інститут прикладного системного аналізу, Київ, Україна

¹ zhenyanaumenko20@gmail.com, ² bogdan.bulakh@gmail.com

Стаття присвячена дослідженню та аналізу Mobile Backend as a Service (MBaaS) як ключового елемента у розробці мобільних додатків. У роботі описуються методи та стратегії використання MBaaS, а також проводиться порівняльний аналіз популярних платформ цього типу. Результати проведеного дослідження дозволили окреслити перелік переваг від використання MBaaS у розробці мобільних додатків порівняно з традиційними підходами.

Ключові слова: сервіс, сервер, мобільний застосунок.

1. ВСТУП

Сучасний світ переживає надзвичайно активний розвиток цифрових технологій, і розробка мобільних застосунків займає важливе місце в цьому процесі. Завдяки зростанню популярності смартфонів і планшетів, а також постійному підвищенню їхньої обчислювальної потужності, мобільні додатки стали неодмінною складовою нашого повсякденного життя. У зв'язку з цим, розробка мобільних застосунків стала не лише технологічно важливою, але і дуже популярною та конкурентною галуззю.

Стаття спрямована на дослідження та розробку мобільного бекенду як сервісу (Mobile Backend As a Service – MBaaS). Ця тема є актуальною через стрімкий розвиток сучасних мобільних додатків. MBaaS відіграє важливу роль у великій кількості мобільних застосунків, надаючи інфраструктуру та функціональність для взаємодії мобільних клієнтів з сервером. Це дозволяє розробникам додатків зосередитися на розробці зовнішньої частини застосунку, забезпечуючи гнучкість та швидкість розробки.

2. ЩО ТАКЕ МВААС?

Mobile Backend as a Service (MBaaS) представляє собою технологічну парадигму, що радикально змінює підхід до розробки мобільних додатків. Це не лише платформа чи набір інструментів – це концепція, яка визнає, що складність бекенд-розробки може бути вирішена через стандартизовані послуги в хмарному середовищі.

MBaaS пропонує готові рішення для складних серверних завдань, які зазвичай вимагають великої кількості часу та експертизи для їхньої реалізації. MBaaS надає розробникам готові модулі та сервіси через інтерфейс програмування додатків (API), що дозволяє значно спростити процес імплементації серверної логіки. Це охоплює створення та управління базами даних, обробку запитів, аутентифікацію, авторизацію користувачів, аналітику, моніторинг, управління контентом, інтеграцію з іншими сервісами та багато іншого.

Цей підхід революціонує спосіб, яким розробники підходять до створення мобільних додатків, дозволяючи їм уникнути важливих технічних складнощів і зосередитися на кращих практиках у фронтенді. MBaaS забезпечує високу гнучкість, масштабованість та безпеку за рахунок використання сучасних технологій хмарних обчислень та безперервного моніторингу.

Це не лише прискорює час від ідеї до запуску продукту на ринок, але й знижує витрати на інфраструктуру та підтримку додатку. Завдяки MBaaS розробники отримують можливість

швидко реагувати на зміни у вимогах ринку та швидше впроваджувати нові функції, забезпечуючи при цьому якість та стабільність продукту на високому рівні.

3. ПОПУЛЯРНІСТЬ ТЕХНОЛОГІЙ

Використання MBaaS набирає популярність у світі розробки мобільних додатків завдяки ряду переваг. Ця платформа дозволяє значно збільшити швидкість розробки та виведення продукту на ринок, зменшити витрати на серверну інфраструктуру та підтримку, а також полегшити управління додатками.

Кілька прикладів компаній, які успішно використовують MBaaS у своїх продуктах:

1. **Uber**: Завдяки використанню MBaaS, Uber зосередився на розробці та вдосконаленні клієнтської частини додатку, в той час як бекендні операції (робота з користувачами, транзакції тощо) були легко та ефективно вирішені через MBaaS-платформи.

2. **Snapchat**: Платформа Snapchat використовує MBaaS для миттєвої обробки великого обсягу медіа-файлів та швидкої доставки повідомлень. Це дозволяє зосередитися на інтерфейсі додатку та функціональності, які пропонують користувачам унікальний досвід.

3. **WhatsApp**: WhatsApp, один із найпопулярніших месенджерів у світі, використовує MBaaS для забезпечення безпеки, аутентифікації та обробки великого потоку повідомлень у реальному часі.

4. **Airbnb**: Популярна платформа для бронювання помешкань використовує MBaaS для обробки операцій з бронюванням, управління користувачами та аналітики даних. Це дозволяє їм зосередитися на розвитку функціональності додатку для кращого взаємодії з користувачами.

5. **Fitbit**: Виробник популярних фітнес-трекерів використовує MBaaS для зберігання та аналізу даних користувачів щодо фізичної активності, серцевого ритму та сну, надаючи унікальні функції для моніторингу здоров'я.

6. **Shazam**: Сервіс розпізнавання музики використовує MBaaS для швидкого обміну даними між користувачами та серверами, що дозволяє миттєво визначати назву та виконавця пісні.

Ці компанії – лише кілька із численних прикладів, що демонструють успішне використання MBaaS у своїх продуктах. Ці приклади ілюструють, як MBaaS дозволяє компаніям сконцентруватися на інноваціях та вдосконаленні користувацького досвіду, мінімізуючи зусилля, витрати та час, необхідні для створення стабільного бекенду.

4. АНАЛІЗ ІСНУЮЧИХ ТЕХНОЛОГІЙ

Перелік популярних технологій

1. Firestore (Google) [1, 2]:

- Переваги: Зручний інтерфейс та швидке створення додатків. Велика спільнота розробників та багато документації. Інтеграція з іншими Google-сервісами, такими як Google Analytics.

- Недоліки: Залежність від інфраструктури Google, що може створювати обмеження для певних користувачів. Обмежена гнучкість у налаштуванні.

2. AWS Amplify (Amazon Web Services) [3]:

- Переваги: Велика масштабованість та гнучкість завдяки інфраструктурі AWS. Широкий вибір підтримуваних програмних мов та фреймворків. Потужність та можливості для великих проєктів.

- Недоліки: Складніший для розуміння для початківців. Вартість може бути високою для малих проєктів.

3. Heroku (Salesforce) [4]:

- Переваги: Простий у використанні інтерфейс. Масштабованість, автоматичне розгортання та кешування. Підтримка різних програмних мов.

- Недоліки: Вартість може бути високою з ростом обсягу. Може бути обмеженіше для специфічних завдань.

4. Kinvey (Progress) [5, 6]:

- Переваги: Простий для початківців та швидкий у розробці. Забезпечує інструменти для мобільних та веб-додатків. Гнучкість у створенні власних бекенд-серверів.

- Недоліки: Може бути менше функціональністю порівняно з іншими платформами. Залежність від одного постачальника.

5. Backendless [7]:

- Переваги: Простий у використанні та гнучкий. Підтримує багато мов програмування. Відкритий вихідний код для більшого контролю.

- Недоліки: Може бути менше популярним порівняно з іншими платформами. Відсутність інтеграції з іншими сервісами.

Вартісне порівняння наведено в табл. 1.

Таблиця 1. Вартісне порівняння

МВaaS-платформа	Найменший тариф (на місяць)	Квота	Додаткова інформація
Firebase	Безкоштовно	10 ГБ бази даних, 5 ГБ хостингу	Обмеження на запити
AWS Amplify	Безкоштовно	1 ГБ зберігання, 15 ГБ трафіку	Включає AWS безпеку
Heroku	Безкоштовно	10 тис. рядків бази даних	Гнучка система
Kinvey	\$200	Необмежена кількість користувачів	Можливі додаткові витрати
Backendless	\$25	1 ГБ бази даних, 50 ГБ трафіку	Підтримка REST і GraphQL

Ці тарифи представляють найдешевші доступні плани для кожної МВaaS-платформи.

На основі порівняння цін за найдешевший тариф, можна зробити наступні висновки:

Firestore: Безкоштовний тариф Firestore доступний, і він має значні включення, такі як 10 ГБ бази даних і 5 ГБ хостингу. Це може бути ідеальним вибором для початкових проектів та тестування [2].

AWS Amplify: також пропонує безкоштовний тариф з 1 ГБ зберігання та 15 ГБ трафіку. Плюсом є доступ до широкого спектру AWS-сервісів для розширення функціональності [3].

Heroku: надає безкоштовний план з обмеженнями, але це дозволяє почати розробку без витрат. Велика перевага полягає в гнучкості системи та підтримці багатьох мов програмування [4].

Kinvey: пропонує високий функціональний тариф за \$200, але це може бути занадто дорого для невеликих проектів [5].

Backendless: надає бюджетний план за \$25, і це може бути привабливим варіантом для малих проектів та стартапів [7].

Завдяки безкоштовному плану Backendless, ця платформа може бути привабливою для невеликих проектів і розробників, які шукають бюджетний варіант. Однак, обираючи МВaaS-платформу, слід також враховувати інші фактори, такі як функціональність, підтримка, доступність SDK і інші параметри [7].

5. ЗАПРОПОНОВАНА СХЕМА РЕАЛІЗАЦІЇ ПОСЛУГИ

Клієнтські додатки, такі як мобільні додатки або веб-додатки, взаємодіють з MBaaS через API для доступу до даних та сервісів.

API Gateway: Шлюз API використовується для маршрутизації запитів від клієнтів до відповідних служб MBaaS. Він також відповідає за аутентифікацію та авторизацію користувачів.

Серверна логіка: Цей компонент містить логіку обробки запитів, обробки даних та виконання бізнес-логіки. Він включає в себе обробку запитів REST або GraphQL, обробку зображень, обробку повідомлень тощо.

База даних: Це сховище для зберігання даних, таких як користувачі, сесії, зображення, повідомлення тощо. База даних може бути NoSQL (наприклад, MongoDB) або SQL (наприклад, PostgreSQL), в залежності від потреб проекту.

Автентифікація та авторизація: Цей компонент відповідає за ідентифікацію користувачів та надання їм доступу до відповідних ресурсів. Він може включати в себе реалізацію протоколів автентифікації, таких як OAuth або OpenID Connect.

Хмарні послуги: Деякі MBaaS можуть інтегрувати хмарні послуги, такі як сховище файлів, Push-сповіщення, інструменти аналітики та багато інших.

Спостереження та моніторинг: Для підтримки та відстеження стану MBaaS, важливо включити засоби спостереження та моніторингу, які допомагають виявляти проблеми та оптимізувати роботу системи.

Скальованість та висока доступність: MBaaS повинен бути розроблений з урахуванням можливості масштабування та забезпечення високої доступності. Це може включати в себе використання контейнерів та систем управління скальованістю.

Зобразимо приблизну схему (приклад) архітектури для власного MBaaS, який використовує Spring Boot, gRPC, Spring Security, MongoDB, push-повідомлення, автентифікацію та авторизацію (рис. 1):

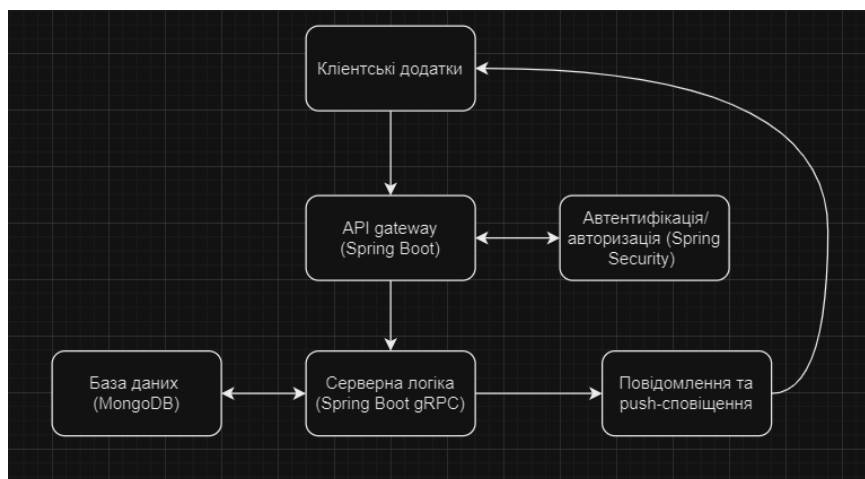


Рисунок 1. Приклад архітектури

У цій схемі:

Клієнтські додатки: мобільні додатки, які взаємодіють з MBaaS через API Gateway.

API Gateway (Spring Boot): Відповідає за маршрутизацію запитів від клієнтів до відповідних сервісів та служб MBaaS. Він також відповідає за забезпечення безпеки через Spring Security.

Spring Security (Автентифікація та авторизація): Використовується для ідентифікації та авторизації користувачів. Встановлює права доступу до ресурсів.

Серверна логіка (Spring Boot, gRPC): Включає в себе обробку запитів та виконання бізнес-логіки. Ви можете використовувати gRPC для взаємодії між сервісами.

База даних (MongoDB): Використовується для зберігання даних, таких як користувачі, сесії та інші. MongoDB відмінно підходить для зберігання документів.

Повідомлення та push-сповіщення: Відповідають за надсилання повідомлень користувачам та реалізацію push-сповіщень для клієнтських додатків.

Ця схема показує загальний підхід до архітектури вашого власного MBaaS.

6. ВИСНОВКИ

У цьому дослідженні було проведено аналіз існуючих підходів до розробки серверної частини мобільних додатків, зокрема, можливостей Mobile Backend as a Service (MBaaS). Було проведено вивчення і порівняння різноманітних MBaaS-платформ, вибір оптимального рішення для конкретного проекту, аналіз можливостей створення власного MBaaS, а також оцінку архітектури та переваг використання MBaaS.

Результати проведеного аналізу показали, що кожна з існуючих MBaaS-платформ, таких як Firebase, AWS Amplify, Heroku, Kinvey та Backendless, має свої переваги та обмеження. Вибір конкретної платформи має базуватися на унікальних потребах та характеристиках конкретного проекту. Firebase виділяється своєю простотою використання та швидкістю розробки, тоді як AWS Amplify надає масштабованість та гнучкість для великих проектів. Heroku дозволяє розгорнути додатки без необхідності управління інфраструктурою, а Kinvey та Backendless надають більше контролю над функціональністю.

В результаті проведених досліджень окреслено переваги та недоліки існуючих хмарних рішень і визначені важливі аспекти щодо вибору конкретного рішення або розробки власного MBaaS для конкретного проекту. Отримані результати є корисними для розробників сучасних мобільних додатків.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. “Firebase”. URL: <https://firebase.google.com> (дата звернення 23.11.2023)
2. “Firebase documentation”. URL: <https://firebase.google.com/docs> (дата звернення 25.11.2023)
3. “AWS Amplify”. URL: <https://aws.amazon.com/amplify/> (дата звернення 25.11.2023)
4. “Heroku”. URL: <https://www.heroku.com/salesforce> (дата звернення 25.11.2023)
5. “Kinvey (Progress)”. URL: <https://www.heroku.com/> (дата звернення 25.11.2023)
6. “React.js”, Official Documentation. URL: <https://www.progress.com/kinvey> (дата звернення 23.11.2023)
7. “Backendless”. URL: <https://backendless.com/> (дата звернення 25.11.2023)

МЕТОДИ АНАЛІЗУ НОВИН

Орловський А.В.¹, Кислий Р.В.²

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

¹ artem.orlovsky@gmail.com, ² kvrware@gmail.com

Зі збільшенням і складністю потоку нової інформації виникає потреба у аналізі новин, постановці пріоритетності інформації, пошуку першоджерел та виявлення неправдивих новин чи маніпуляцій. В даній роботі розглядаються методи аналізу новин українського інформаційного простору за допомогою лінгвістичних та нейромережових інструментів обробки природньої мови.

Ключові слова: новина, статистична модель, нейромережева модель, трансформер, векторне представлення

1. ВСТУП

Новини є оперативними інформаційним повідомленнями, які в себе включають важливу інформацію для соціуму. Новини є особливим джерелом інформації для людей в тому числа для прийняття рішень. Для аналізу новин використовуються методи журналістики, а також методи машинного навчання, що здатні автоматизувати обробку даних та виділяти головне з повідомлення, знаходити зв'язки між різними новинами. Для таких задач використовуються методи обробки природньої мови, що складаються як і з класичних методів обробки враховуючи її граматику і лінгвістичні правила, так і методи з використанням мереж глибокого навчання, включаючи нейромережі типу трансформер з механізмом уваги.

2. ЗАДАЧА АНАЛІЗУ НОВИН ЗА ЗМІСТОМ

Для аналізу новин та пошуку першоджерел потрібно розуміти, яким чином можна порівняти новини, виміряти їх близькість одна до одної. Це завдання вимагає математичного підходу та використання методів обробки природньої мови. Основним викликом є перетворення текстової інформації на певне векторне представлення, в просторі якого можна ввести метрики для порівняння схожості чи відмінності різних новин.

Дана математична задача полягає в репрезентації тексту у вигляді числового вектору. У векторному просторі кожен текст стає точкою зі своїм унікальним векторним представленням. Це векторне представлення може бути створене за допомогою різних методів, таких як Bag of Words (BoW) [1], TF-IDF (Term Frequency-Inverse Document Frequency) [2], або більш складні моделі, такі як агрегації Word2Vec [3] або векторні представлення BERT [4].

3. TF-IDF РЕПРЕЗЕНТАЦІЯ ТЕКСТУ

Дана ідея заснована на концепції розподілу слів у текстах. Слова неоднорідні за частотністю в тексті, ймовірність виявити одне слово в текстах може бути більшою за інші. Закон Ципфа – це лінгвостатистичний закон, який стверджує що відношення рангу слова в частотному словнику до частотності слова є константою, тобто чим більше ранг, тим обернено менше частотність, що ілюструється на наступному графіку нижче:

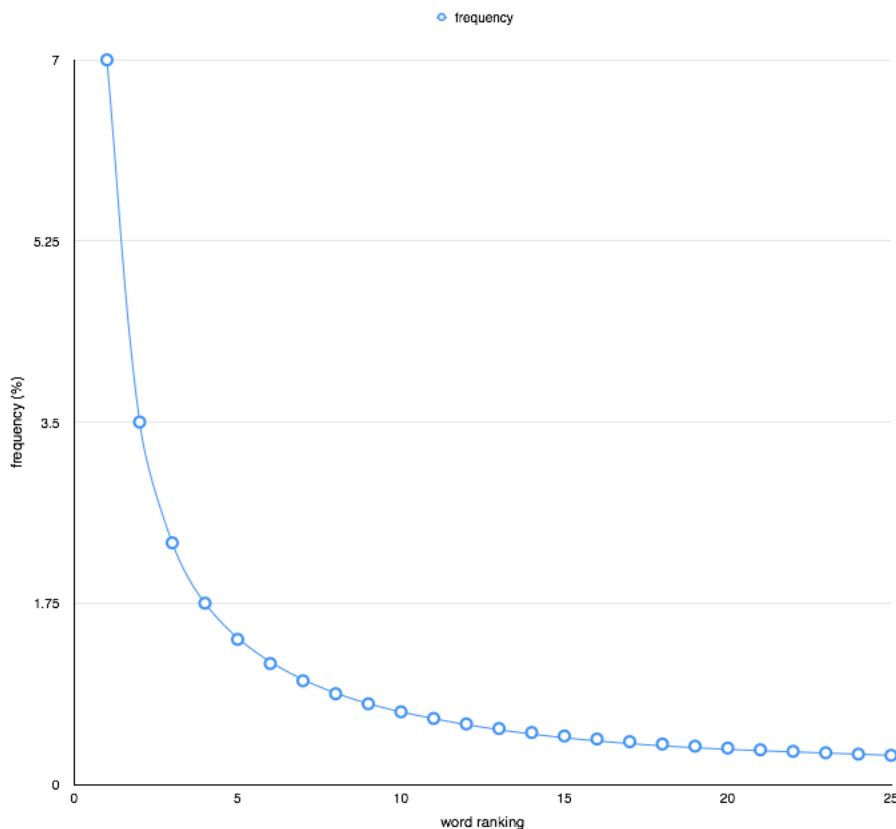


Рисунок 1. Графік розподілу Ципфа, з ранжуванням частотності слів

Враховуючи різну тематику документів, різні слова можуть в одних текстах зустрічатись частіше ніж інші. Загальні слова, що існують у мові, будуть мати велику частоту використання в усіх документах, але деякі слова будуть властиві саме певному документу. Таким чином, можна ввести формулу для прорахунку виміру належності слова конкретному документу в конкретному корпусі слів, і дана формула називається TF-IDF (3) – term frequency (1), inverse document frequency (2).

$$TF(t, d) = \frac{n_t}{\sum_k n_k} \quad (1)$$

$$IDF(t, D) = \log \frac{|D|}{|\{d_i \in D | t \in d_i\}|} \quad (2)$$

$$TF - IDF(t, d, D) = \frac{n_t}{\sum_k n_k} \log \frac{|D|}{|\{d_i \in D | t \in d_i\}|} \quad (3)$$

де n_t – частота слова t в документі d , D - корпус текстів, датасет, $|D|$ -кількість документів в корпусі, $|\{d_i \in D | t \in d_i\}|$ - кількість документів в колекції D де зустрічається слово t .

Якщо обрати словник слів, кількість слів у словнику нехай буде M , то отримаємо векторну репрезентацію документу з довжиною вектора M . Даний вектор не включає в себе виявлення граматичних ознак в тексті чи взаємозв'язність слів для отримання нових сенсів у документі, але на практиці така репрезентація дає гарний бейзлайн у поєднанні з моделлю логістичної регресії (4) для задач класифікації, в тому числі для виявлення пропаганди у тексті, що є однією з важливих задач аналізу новин.

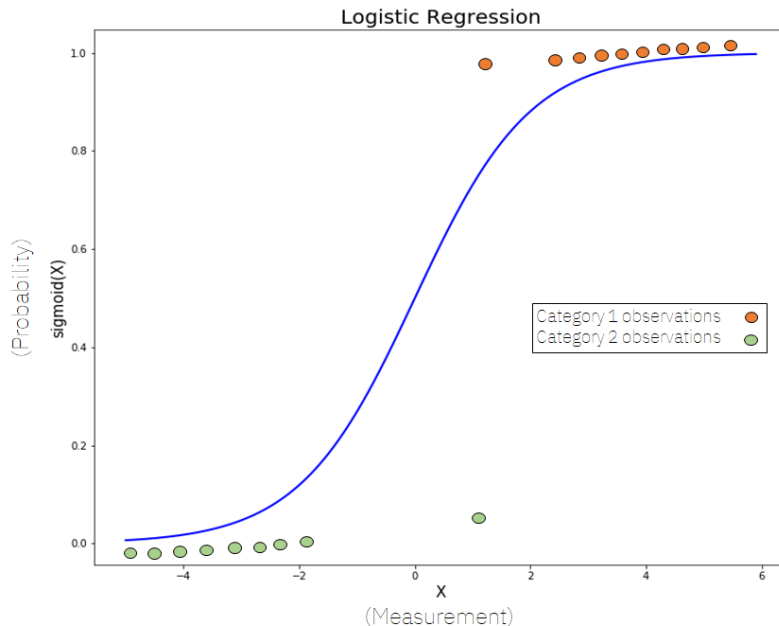


Рисунок 2. Графік логістичної регресії, на графіку зображена крива сігмоїди, для якої обрані параметри під задачу

$$\log\left(\frac{P}{1-P}\right) = \beta_0 + \beta_1 X \quad (4)$$

Формула логістичної регресії (4), де $\log\frac{P}{1-P}$ – лог-шанси, $\beta_0 + \beta_1 X$ – пряма регресії.

4. ВЕКТОРНЕ ПРЕДСТАВЛЕННЯ ТЕКСТУ З ОЗНАК SENTENCE-TRANSFORMER

Один з найефективніших способів отримати векторне представлення текстів за змістом у сучасних моделях високої якості – це використання вихідних векторів з архітектури sentence transformer [5]. Ця інноваційна архітектура заснована на принципі роботи сіамських нейронних мереж та є ефективною для задач обробки природної мови.

Основна ідея полягає в тому, що під час тренування сіамської мережі використовуються пари текстів, і мережа навчається порівнювати їх схожість. При цьому вихід з трансформера проходить через пулінг шар нейронної мережі, а потім відбувається класифікація схожості векторів, часто використовуючи косинусну схожість або додатковий шар з класифікатором Softmax активації. Цей процес навчання дозволяє мережі вчитися представляти текст у векторному просторі таким чином, що схожі текстові документи будуть мати близькі векторні представлення.

Після завершення тренування сіамська мережа може бути використана для отримання векторних представлень текстів. Для цього векторне представлення кожного тексту отримується шляхом відкидання крайнього шару сіамської мережі і обирається лише одна ланка такої мережі.

Використання sentence transformer дозволяє отримувати векторні представлення текстів, які можна використовувати для подальшого аналізу, кластеризації, класифікації та пошуку схожих текстів.

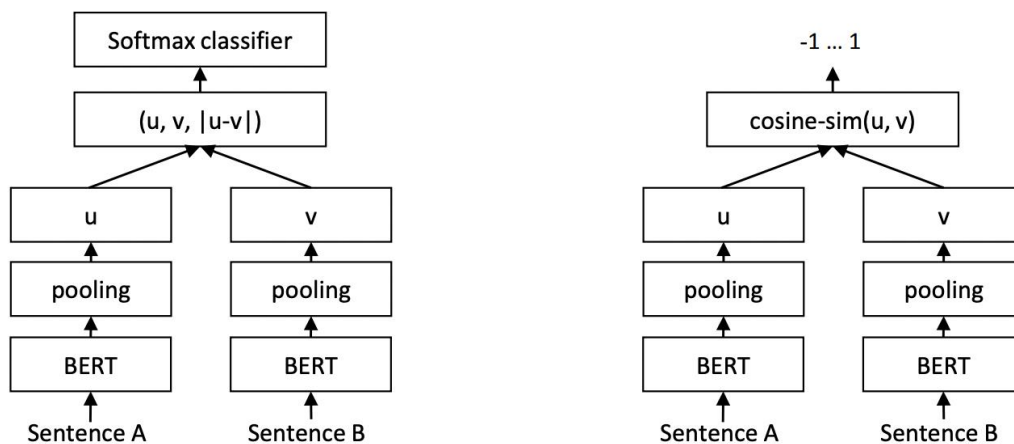


Рисунок 3. Схема архітектури сіамських нейронних мереж sentence-transformers

В основі sentence-transformers лежить архітектура трансформеру BERT, який є нейромережею-енкодером, тобто такою, що вивчає загальний контекст тексту та тренується з двонаправленим навчанням на тексті. Така методика дозволяє отримати контекст для слова, що вивчається, не тільки з попередніх слів, але й з наступних, вивчаючи додаткові взаємозв'язки.

5. ПОШУК СХОЖИХ НОВИН ЗА АЛГОРИТМОМ ПОШУКУ НАЙБЛИЖЧИХ СУСІДІВ

Після отримання функції перетворення текстового документу в певне векторне представлення, у вигляді нейромережі чи класичного статистичного підходу, наступним необхідним кроком для аналізу є створення індексу в отриманій векторній даних та використання належного алгоритму пошуку найближчих сусідів точки у просторі, з яких потім дослідник отримає найближчі новини до тої, що він вивчає, за змістом. З класичних статистичних алгоритмів є алгоритм пошуку k -найближчих сусідів, але він потребує значного часу пошуку за великих даних. У разі аналізу новин важлива швидкість, оскільки для вчасного реагування на дезінформацію її треба виявити якнайскоріше ще коли інформація є актуальною.

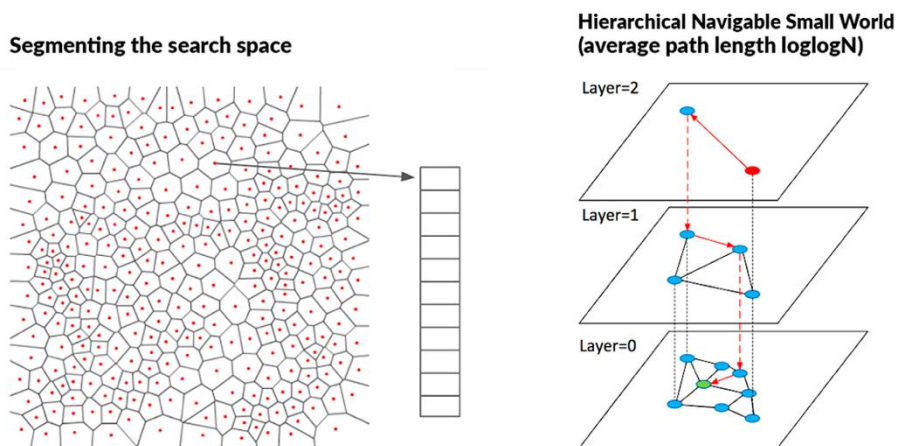


Рисунок 4. Візуалізація основних компонентів алгоритму Hierarchical Navigable Small World (HNSW)

З розробок нових структур даних, що використовуються в сучасних базах даних, застосовуються методи пошуку приблизних найближчих сусідів. Дані структури даних є ймовірнісними, вони можуть не повернути всіх точних сусідів, але швидкість таких алгоритмів в рази більша. В порівнянні, k -найближчих сусідів має оцінку швидкості пошуку $O(n^2)$, в той час як сучасний алгоритм Annoy, враховуючи передрозрахунок структури даних перед пошуком, має складність $O(\log(n))$, що суттєво зменшує час на пошук найближчого сусіда. Складність алгоритму пошуку Hierarchical Navigable Small World, візуалізація якого зображена вище, також має складність $O(\log(n))$, але іншу структуру даних.

Таким чином, даний спосіб індексації дозволить створити апарат швидкого пошуку близьких точок у просторі, а отже для обраної задачі – близькі новин за змістом. Для подальшого аналізу можливо виявляти суперечності між оригінальною новиною та новиною, що на неї послалася, визначати частину тексту яка суперечить чи несхожа за змістом до першої новини, та складати лог особливих подій, які варті для більш глибокого дослідження людиною-експертом.

6. ВИСНОВКИ

У світі з постійним розвитком інформаційних технологій, роль новин як особливого джерела інформації для суспільства набуває особливої актуальності. Новини є оперативним та важливим джерелом інформації, на основі якого можуть робитись рішення в режимі реального часу. Саме тому необхідно створити додаткові інструменти, що могли б спростити проведення аналізу новин людиною.

Постановка даної задачі починається з репрезентації тексту у вигляді вектору чисел різними моделями, такими як Bag of Words, TF-IDF, Word2Vec, BERT, і sentence-transformers. Однаково важливі як і нейромережеві підходи, так і статистичні, оскільки вони більш пояснювані через визначення їх формул та основних засад з теорії ймовірності.

Для подальшого аналізу використовуються структури даних що є основою векторних баз даних, на основі яких відбувається швидкий пошук та аналіз векторного простору певних записів у базі. Отримана векторна форма тексту слугує як і для задачі пошуку найближчих новин за змістом, так і перевірки протиріччя новини за змістом чи пошуку кластеру чи топіку новин за тематикою цієї новини. Дані інструменти будуть спеціальним засобом роботи з базою новин для людей-експертів, що спеціалізуються на аналізі новин.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Zellig S. Harris, "Distributional Structure", Word Journal, 1954, Vol. 10, No. 2-3, pp 146-162
2. Sparck Jones, K., "A statistical interpretation of term specificity and its application in retrieval", Journal of Documentation, 1972, Vol. 28 No. 1, pp. 11-21.
3. "Distributed Representations of Words and Phrases and their Compositionality", Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, Jeffrey Dean, 2013, URL – <https://arxiv.org/abs/1310.4546>, дата звернення – 27.11.2023
4. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, 2018, URL – <https://arxiv.org/pdf/1810.04805.pdf>, дата звернення – 27.11.2023
5. "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks", Nils Reimers, Iryna Gurevych, 2019, URL – <https://arxiv.org/abs/1908.10084>, дата звернення – 27.11.2023

ОГЛЯД ПРІОРИТЕЗАЦІЇ ТРАФІКУ В ПРОГРАМНО ВИЗНАЧЕНИХ МЕРЕЖАХ

Переяславський С.К.¹, Письменний І.О.²

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Інститут прикладного системного аналізу, Київ, Україна

¹ pereiaslavskiy.serhii@iit.kpi.ua, ² ihor.pismennyu@gmail.com

З ростом складності розподілених комп'ютерних систем, збільшення їх гетерогенності, актуальності набула ідея відділення контролю трафіку в мережах від фізичних пристроїв - перенесення функції маршрутизації та пріоритезації трафіку в програмну площину. Дана робота присвячена питанню пріоритезації трафіку в гетерогенних розподілених мережах з допомогою програмно-визначених систем (англ. Software-Defined Networking, SDN).

Ключові слова: програмно-визначені мережі, контролер, пріоритезація, трафік.

1. ВСТУП

Програмно визначені мережі (Software Defined Networks) – це інноваційний підхід до побудови мережевої архітектури, який виник у відповідь на сучасні виклики та потреби у галузі інформаційних технологій. SDN може вирішувати цільові задачі завдяки своїй здатності забезпечити більшу гнучкість, ефективність, безпеку та контроль у мережевій інфраструктурі, що стає критично важливим у зв'язку із ростом цифровізації та змінами в бізнес-середовищі і бізнес-процесах.

2. ОСОБЛИВОСТІ SDN

Технологія програмно-визначених мереж є ключовим елементом еволюції мережових структур у світі технологій, які швидко розвиваються. Програмно визначена мережа означає не тільки те, що мережеві елементи управляються програмно і можуть швидко та ефективно перебудовуватися, але й те, що на одному фізичному пулі мережових елементів можуть бути розгорнуті багато мереж, що логічно не залежать один від одного. Такі логічні мережі можуть передавати потоки трафіку різних бізнес-додатків, не заважаючи одна на одну.

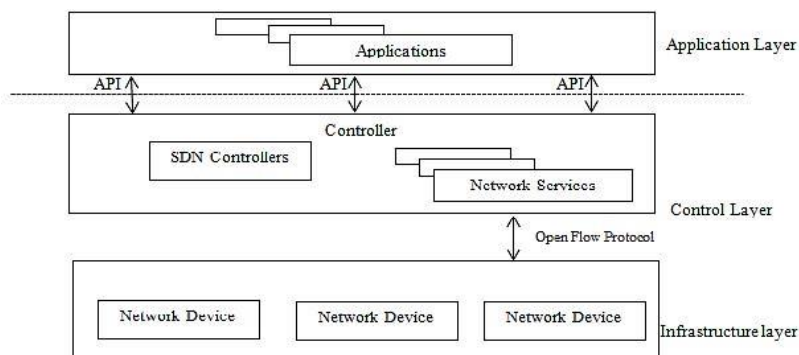


Рисунок 1. Структура SDN

Програми, яким потрібні різні параметри та конфігурації мережі, адмініструють SDN-контролер за допомогою інтерфейсів програмування API (Application Programming Interface).

У такій архітектурі кожен бізнес-додаток може через SDN-контролер сконфігурувати собі логічну мережу із загальних ресурсів мережевої інфраструктури рівня передачі. І кожна така мережа працюватиме незалежно від інших логічних мереж у тому ж пулі ресурсів (за їх достатністю та наявністю).

Для керування інфраструктурою мережі від SDN-контролера необхідний протокол відкритого керування. Такий протокол, розроблений вченими зі Стенфорда, називають OpenFlow (не єдиний можливий протокол для застосування). Робота OpenFlow полягає у модифікації змісту таблиці передачі пакетів усередині маршрутизатора чи комутатора (Forwarding table). Архітектура мережевих пристроїв часто представляється у вигляді трьох площин: адміністрування (management plane), управління (control plane) та передачі даних (data plane).

Традиційно мережеве обладнання від різних виробників часто має спеціальні системи конфігурації та керування, що обмежує взаємодію між маршрутизаторами та комутаторами від різних виробників. Щоб вирішити цю проблему, було створено OpenFlow як відкритий програмований мережевий протокол для налаштування та керування мережевими комутаторами від різних постачальників. Це дає змогу перенести площину керування всіма комутаторами на центральний контролер і дозволяє центральному програмному забезпеченню визначати поведінку мережі. Таким чином, мережеві адміністратори можуть використовувати програмне забезпечення OpenFlow для управління та контролю потоку трафіку між комутаційним обладнанням різних марок.

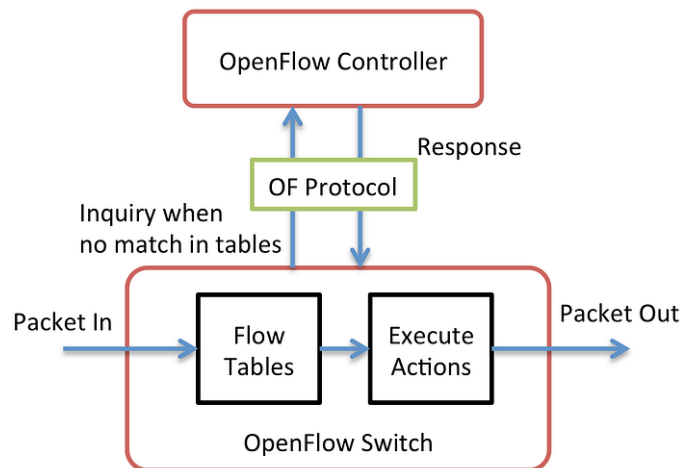


Рисунок 2. Структура OpenFlow комутатора

OpenFlow зазвичай складається з трьох компонентів: контролера, протоколу OpenFlow і ланцюжка таблиць пакетів, налаштованих на комутаторі OpenFlow. Контролер може встановлювати правила щодо поведінки пересилання даних кожного пристрою через протокол OpenFlow. Коли пакет даних надходить до комутатора OpenFlow, комутатор шукатиме відповідні записи пакета в таблицях і виконає відповідні дії. Якщо відповідності не знайдено, подія запиту надсилається до контролера OpenFlow, який потім відповідає новим записом потоку для обробки цього пакета в черзі.

3. ВІДМІННІСТЬ ВІД ТРАДИЦІЙНИХ МЕРЕЖ

На відміну від традиційної архітектури маршрутизатора або комутатора, яка об'єднує функції управління і передачі даних на одному пристрої, SDN пропонує метод віртуалізації мережі [1], який дозволяє програмно впроваджувати різні логічні функції в різних компонентах мережі з метою гнучкого та ефективного керування. Розділення рівня керування

і рівня даних дозволяє маршрутизаторам і комутаторам використовувати дані шару управління для спрямування вхідного трафіку до відповідного інтерфейсу виходу.

Інфраструктура є основною відмінністю між SDN і традиційними мережами. Оскільки рівень керування ґрунтується на програмному забезпеченні, SDN є набагато гнучкішим, ніж традиційна мережа. Це дозволяє адміністраторам контролювати мережу, змінювати параметри конфігурації, надавати ресурси та збільшувати пропускну здатність мережі – все з централізованого інтерфейсу користувача без необхідності встановлення додаткового обладнання.

Традиційні мережі зазвичай мають об'єднану архітектуру, де управління та передача даних зосереджені на окремих фізичних пристроях (маршрутизатори/комутатори). Це призводить до того, що зміни в мережевій конфігурації можуть бути важкими та тривалими завданнями, оскільки вони вимагають фізичної конфігурації кожного пристрою окремо. Натомість мережі SDN відрізняються єдиним підходом до управління мережею. Програмний контролер в цих мережах відокремлений від фізичної інфраструктури і приймає на себе велику частину функцій управління.

В цілому, мережі SDN надають більше гнучкості, контролю, гарантованої безпеки та можливостей моніторингу порівняно з традиційними мережами. Це робить їх особливо корисними для сучасних сценаріїв, де вимоги до мережі зростають разом з розширенням цифрової трансформації та використанням нових технологій [2].

Крім того, існують відмінності в безпеці між SDN і традиційними мережами. SDN забезпечує більшу безпеку в багатьох аспектах завдяки підвищеній видимості та здатності визначати безпечні шляхи. Однак, оскільки програмно визначені мережі покладаються на централізований контролер, захист контролера має важливе значення для підтримки безпеки мережі [3].

У порівнянні з традиційною мережевою архітектурою, централізована контрольна точка архітектури SDN дозволяє ефективно реагувати на проблеми безпеки в мережі. Незалежно від розміру мережі, обсягу даних, що обробляються в мережі тощо, зосередження на безпеці стає дуже важливим предметом. Можливості мережі можна оцінити залежно від її споживання ресурсів, масштабованості, доступності та стійкості до збоїв. Однак усі ці вимірювання будуть безглуздими, якщо мережа буде вразлива до атак. Це підкреслює важливість дослідження поведінки користувачів і недоліків безпеки всередині архітектури SDN.

4. ПРІОРИТЕЗАЦІЯ ТРАФІКУ

У світі великих потоків даних та безперервних комунікацій, вибір правильного маршруту для кожного пакета інформації стає викликом для сучасних мереж. Зі стрімким збільшенням обсягів даних, виникає питання, як оптимально розподілити багатошаровий потік, де дані великого обсягу конкурують за обмежені ресурси. Пріоритетизація трафіку стає невід'ємним підходом, який розподіляє ресурси для забезпечення оптимальної продуктивності мережі.

З різноманіттям послуг та додатків виникає завдання створити гнучку систему, де кожен тип трафіку отримає необхідний рівень обслуговування. Це вимагає докладної конфігурації, щоб уникнути конфліктів та забезпечити ефективне використання ресурсів.

Quality of Service (QoS) визначається як набір механізмів і технологій, призначених для забезпечення певного рівня якості обслуговування в мережевому середовищі. Основна мета QoS полягає в тому, щоб ефективно управляти трафіком, гарантувати передачу даних з певними параметрами та враховувати потреби різних видів додатків та користувачів.

Важливим аспектом управління мережею для забезпечення високої доступності та ефективності є класифікація трафіку. Існують методи застосування класифікації трафіку в

мережах: використання портів для визначення прикладних програм і протоколів прикладного рівня або використання детальної перевірки пакетів (deep packet inspection). Ці методи мають високу точність, однак є деякі проблеми щодо їх реалізації, оскільки динамічні порти та зашифрований трафік ще не підтримуються в поточних мережах. Це також спричиняє великі витрати на систему та порушує конфіденційність користувачів.

Розставляючи пріоритет трафіку на основі попередньо визначених правил, SDN може гарантувати послідовну взаємодію користувача з основними службами, одночасно запобігаючи перевантаженням мережі, спричиненим некритичним трафіком.

5. ПРОПОНОВАНА РЕАЛІЗАЦІЯ

В якості дослідження проблематики пріоритезації трафіку в програмно визначеній мережі, пропонується метод класифікації трафіку для його пріоритезації на основі кращого підходу до навчання моделі-класифікатора [4].

Пропонована реалізація включає в себе проведення підготовчого етапу для формування передумов для подальших досліджень, а саме – для пріоритезації трафіку на основі попередньо досліджених особливостей пакетів.

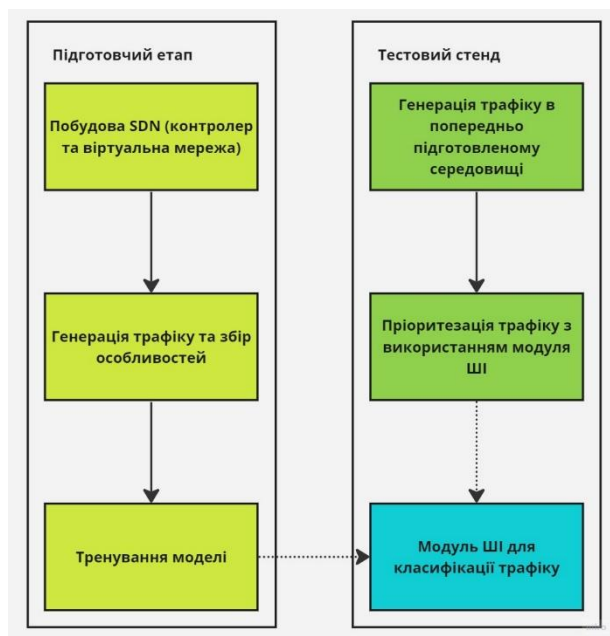


Рисунок 3. Етапи досліджень пропонованої реалізації

При підготовчому етапі передбачається побудова SDN системи разом з контролером та віртуальною мережею, генерація і збір трафіку та подальше застосування його особливостей для тренування класифікатора. Використання тестового стенду передбачає задіяння натренованої моделі-класифікатора для пріоритезації трафіку в режимі реального часу.

6. ВИСНОВКИ

В епоху стрімкого технологічного розвитку та зростаючих потреб користувачів, технологія програмно-визначених мереж стає одним з провідних принципів для побудови сучасних мережевих інфраструктур. Застосування SDN у різних сферах свідчить про їхню виразну роль у вирішенні викликів, що постають перед сучасними мережами.

SDN розв'язує проблеми традиційних мереж, де об'єднання управління і передачі даних на одному пристрої призводить до обмежень у гнучкості та ефективності. Це дозволяє

централізованою програмній частині, контролеру, керувати роботою всіх мережевих пристроїв.

Проведені дослідження запропонованої реалізації для пріоритезації трафіку використання класифікатора на основі найкращого підходу до навчання моделі дозволить ефективно визначати пріоритети трафіку в режимі реального часу на тестовому стенді. Необхідно провести додаткові дослідження для оцінки масштабованості та впливу на мережевий трафік при різних умовах. Додаткові дослідження щодо масштабованості та впливу на мережевий трафік при різних умовах важливі для повного розуміння ефективності реалізації. Загалом, запропонований метод відкриває шлях до подальших досліджень та покращення, сприяючи вдосконаленню управління трафіком в програмно визначених мережах.

Слід враховувати, що ефективні стратегії пріоритезації трафіку вимагають комплексного підходу, зважаючи якість контролера, програмованість, тенденції у розвитку мереж та аспекти безпеки. Постійне вдосконалення та адаптація методів до зростаючих вимог та змін у технологічному середовищі є ключовими аспектами для забезпечення ефективності та стабільності системи в програмно визначених мережах.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. O. Jefia, S. I. Popoola, i A. A. Atayero, «Software-defined networking: Current trends, challenges, and future directions», в *Proceedings of the International Conference on Industrial Engineering and Operations Management*, 2018.
2. R. Shubbar, M. Alhisnawi, A. Abdulhassan, i M. Ahamdi, «A Comprehensive Survey on Software-Defined Network Controllers», в *Lecture Notes in Networks and Systems*, 2021. doi: 10.1007/978-981-16-0666-3_18.
3. N. M. Kaliyamurthy, S. Taterh, S. Shanmugasundaram, A. Saxena, O. Cheikhrouhou, i H. Ben Elhadj, «Software-Defined Networking: An Evolving Network Architecture - Programmability and Security Perspective», *Security and Communication Networks*, вип. 2021, 2021, doi: 10.1155/2021/9971705.
4. S. S. Samaan i H. A. Jeiad, «Architecting a machine learning pipeline for online traffic classification in software defined networking using spark», *IAES International Journal of Artificial Intelligence*, вип. 12, вип. 2, 2023, doi: 10.11591/ijai.v12.i2.pp861-873.

ХМАРНИЙ СЕРВІС ДЛЯ ДИСТАНЦІЙНОГО ДОГЛЯДУ ЗА ПАЦІЄНТАМИ

Петренко А.І.¹, Редька М.Ю.²

Національний технічний університет України «Київський політехнічний інститут
ім. Ігоря Сікорського», Київ, Україна

¹ tolja.petrenko@mail.com, ² mex4232@gmail.com

Протягом останніх років спостерігається швидкий розвиток різноманітних технологій та методів для впровадження різноманітних веб-систем. Цей розвиток обумовлений, з одного боку, постійною інтеграцією нових систем у сферу веб, а з іншого боку – зростанням складності вже існуючих систем. Нові методи та технології призначені відповідати технологічним вимогам складних систем та полегшувати їх впровадження. Зокрема, галузь охорони здоров'я є однією з найбільш цікавих і водночас складних сфер. Зазвичай такі системи характеризуються високими технологічними та етичними стандартами, а також складними внутрішніми процесами, вимагаючи постійної підтримки та оновлень. Це призводить до природного інтересу до використання сучасних підходів для полегшення та покращення функціонування таких систем, які представлені в даній роботі.

Ключові слова: мікросервісна архітектура, мікрофронтенд, складні системи, охорона здоров'я.

1. ВСТУП

Хмарні обчислення традиційно використовувалися ІТ-індустрією для резервного копіювання даних, розробки та тестування програмного забезпечення, аварійного відновлення тощо. Однак із розвитком технологій постачальники хмарних послуг розробили різні рішення, адаптовані до конкретних галузей. Від охорони здоров'я та інтелектуального автомобілебудування до страхування, роздрібною торгівлі, маркетингу та освіти все більше галузей використовують хмару, щоб отримати переваги.

Хмарні обчислення в охороні здоров'я забезпечують зв'язане, доступне та спільне середовище для пацієнтів, докторів та клінік. У довгостроковій перспективі перехід на хмарне середовище забезпечить зниження витрат і сприятиме більшому комфорту для пацієнтів та медичного персоналу.

В охороні здоров'я хмарні обчислення набувають все більшої популярності, особливо після пандемії COVID-19. Важливість хмарних обчислень в медичній сфері очевидна з того факту, що дана галузь постійно розвивається та розширюється. Зараз хмарні обчислення є необхідною технологією для галузі охорони здоров'я, щоб забезпечити оптимальне обслуговування пацієнтів.

У цій статті пропонується архітектура додатку, що реалізує віртуальний кабінет лікаря та пацієнта, яка є ефективною з точки зору вирішення проблем комунікації лікарів та пацієнтів, а також відстеження процесу одужання хворих.

2. АРХІТЕКТУРА СИСТЕМИ ВІРТУАЛЬНОГО КАБІНЕТУ ЛІКАРТЯ ТА ПАЦІЄНТА

2.1. Вимоги до системи

Головна мета цієї системи полягає в полегшенні взаємодії між лікарем і пацієнтом під час розроблення та виконання плану лікування різних хвороб. Система повинна надавати лікарю зручний інтерфейс для створення плану лікування, його кроків і обмежень, а також забезпечувати пацієнту зручний спосіб перегляду плану та взаємодії з ним (включаючи можливість позначати виконані приписи, уточнювати план лікування та вводити дані про процедури в рамках плану).

Окрім цього базового функціоналу, система також повинна включати такі функції:

- Перегляд особистої інформації про пацієнта, зокрема його електронну медичну карту (ЕКМ).
- Моніторинг поточного стану пацієнта на основі введених даних.
- Автоматичні повідомлення про стан здоров'я на основі введених показників.
- Допоміжні інструменти для управління планом лікування, які допомагають лікарю в його управлінні та пацієнту в проходженні лікування.

Крім того, система повинна бути готовою для майбутнього розширення функціоналу, інтеграції з іншими сервісами та забезпечувати високий рівень конфіденційності даних користувачів.

В результаті роботи цієї системи пацієнти і лікарі отримують зручні інтерфейси, що містять всю необхідну інформацію та надають підказки для ефективного створення та виконання плану лікування.

З вищесказаного випливає, що система має багато різноманітних функцій, які можуть бути розподілені на різні компоненти, такі як:

- Менеджмент плану лікування.
- Зручні системи спілкування.
- Доступ до медичної історії та інформації про пацієнта.
- Аналітика стану пацієнта.
- Рекомендації для лікарів і пацієнтів.

В той же час в системі мають бути передбачені можливості для майбутнього розширення функціоналу системи, покращення окремих компонентів.

2.2. Опис архітектури

За встановленими умовами було розроблено архітектуру системи, яка включає в себе різні компоненти та забезпечує їх взаємодію, як то ілюструється на рис. 1.

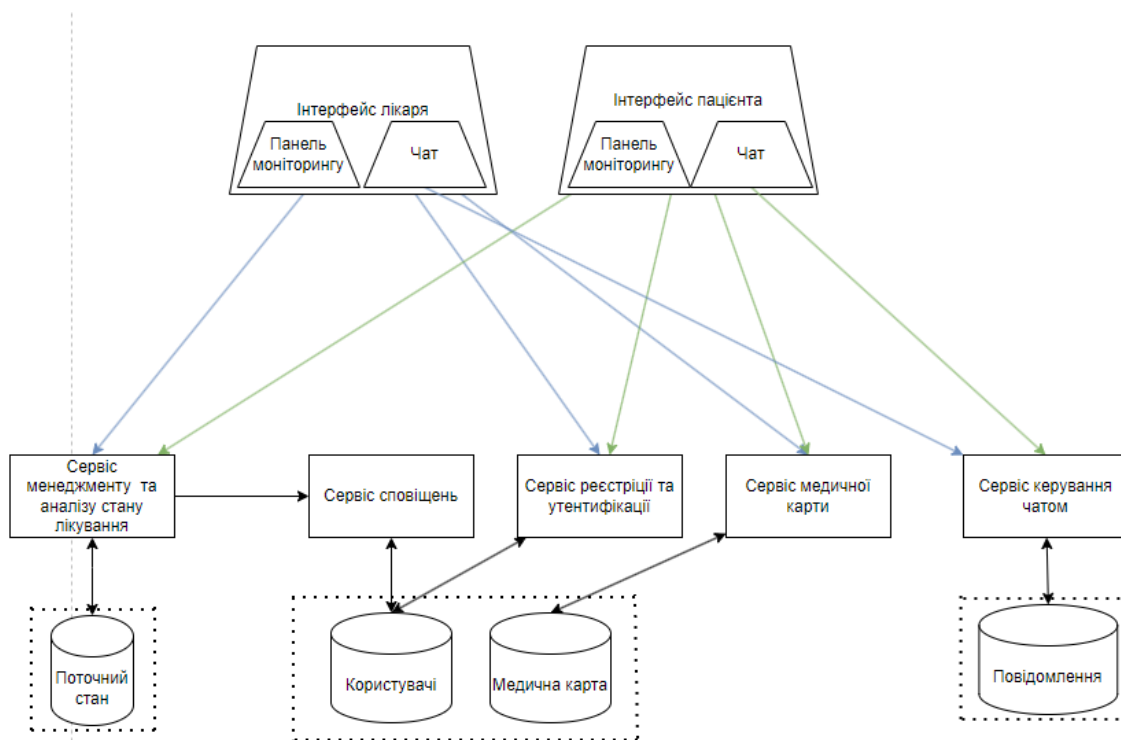


Рисунок 1. Архітектура системи віртуального кабінету лікаря та пацієнта

Значення компонентів системи:

Сервіс аналізу даних про поточний стан – безперервно перевіряє вміст відповідної бази даних і, у разі необхідності, видає сигнал для ініціації сповіщення або оновлення інформаційної панелі інтерфейсу.

Сервіс сповіщень – відповідає за розсилку сповіщень пацієнтам і лікарям щодо різних подій, включаючи кризові ситуації, за допомогою різних комунікаційних каналів, таких як СМС-повідомлення або електронна пошта. Додатковий інтерес представляють "допоміжні" або "аналітичні" сервіси.

Сервіс реєстрації, аутєнтифікації – є однією з ключових складових системи, відповідальною за належне управління політикою доступу до ресурсів, включаючи особисті дані.

Сервіс медичної картки – відповідає за надання особистої медичної інформації пацієнтів.

Сервіс керування чатом – забезпечує механізм спілкування лікаря\пацієнта.

Важливо відзначити деякі технічні особливості системи:

- "Чутлива" особиста інформація, яка може однозначно ідентифікувати користувача системи, зберігається в одному конкретному місці (на рис.1 відображений блок з елементами "Користувачі" і "Дані ЕМК"). Таким чином, ця інформація може бути окремо виведена в інший простір, навіть в окрему базу даних, яка обладнана додатковими заходами безпеки. Це може призвести до збільшення обсягу витрат ресурсів. Таблиці в інших просторах не містять прямих посилань на конкретних користувачів, тому в разі несанкціонованого доступу зловмисників до них, не вдається однозначно встановити, кому належать ці дані. Необхідні дані для зіставлення планів лікування, повідомлень та поточного стану доступні користувачам тільки в контексті "чутливих даних". Для наочності приклад відповідної організації таблиць приведено на рис. 2.

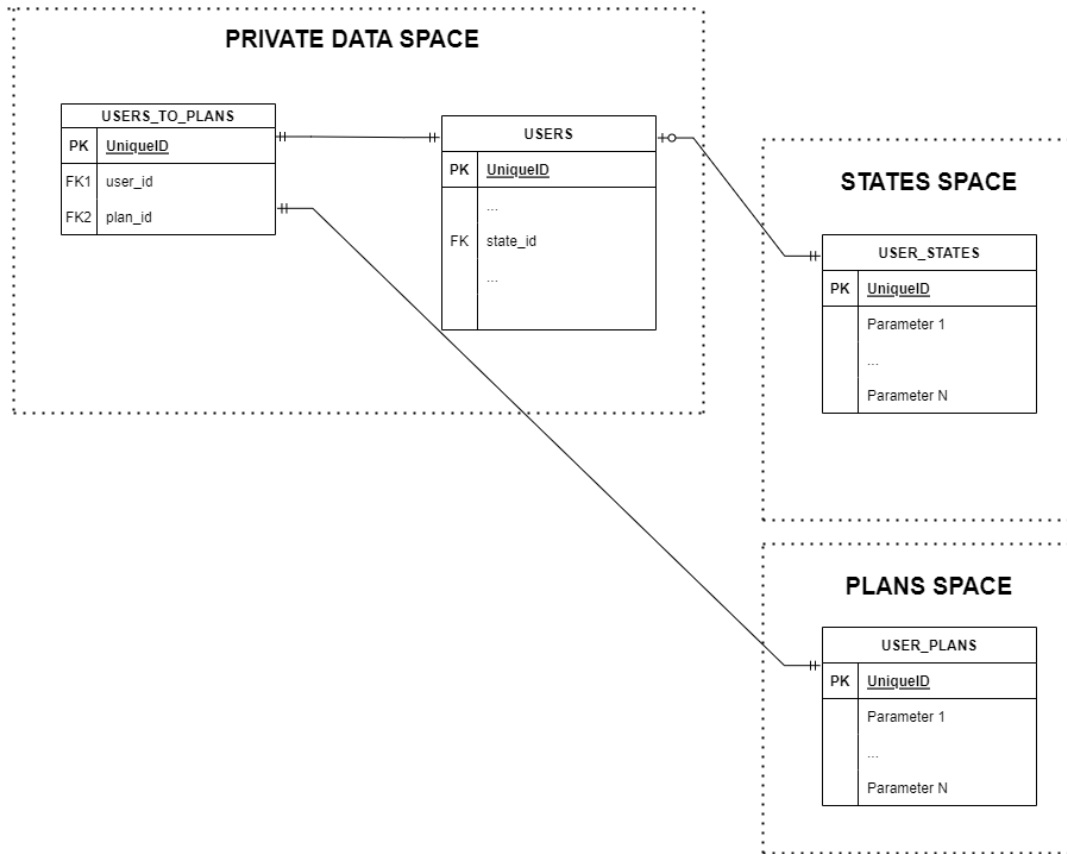


Рисунок 2. Відношення між таблицями з окремих просторів

- На рівні обслуговування або в базі даних можливо обмежити право деяким службам вносити зміни в певні таблиці. Наприклад, аналітичні сервіси обмежені можливістю лише зчитувати дані для подальшого аналізу, сервіс медичної картки може тільки надавати інформацію про картки, але не мати можливості змінювати їх, і лікар може не лише переглядати дані про поточний стан, але й змінювати їх.

- Оскільки таблиці, що описують поточні стани та плани лікування, фактично є "анонімними", існує можливість надати стороннім сервісам доступ для читання цих даних з метою виконання різноманітного практичного аналізу або для надання інформації зовнішнім статистичним сервісам.

2.3. Декомпозиція інтерфейсу

Підхід до розбиття на компоненти (мікросервіси) не обмежується внутрішньою структурою системи, а також застосовується на рівні користувацького інтерфейсу. Цей підхід називається "мікрофронтендним" [4, 5]. У даній системі він використовується для інтеграції з аналітичною панеллю, чатом і, в майбутньому, з іншими компонентами.

Цей підхід дозволяє спростити роботу команд, які працюють над одним проектом, і відкриває можливість використовувати послуги зовнішніх постачальників, а також використовувати підхід "white label" [6].

Зі схематичної точки зору сторінка плану лікування в такому випадку може виглядати, як показано на рис. 3. На ньому панель моніторингу (виділена синім кольором) та чат (виділений оранжевим кольором) імпортуються як окремі залежності ззовні і, за потреби, можуть бути вимкнені користувачами.

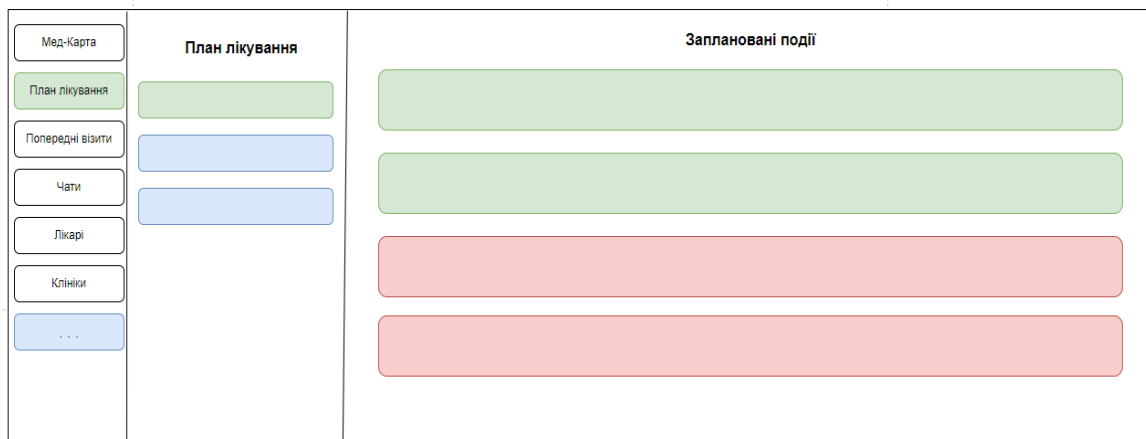


Рис. 3. Схематичний вигляд інтерфейсу, скомпонованого із різних частин (окремих субінтерфейсів)

3. ВИСНОВКИ

Хмарні обчислення впливають на галузь охорони здоров'я комплексно, з багатьох сторін. Важливість хмарних обчислень в охороні здоров'я величезна: від зниження вартості медичної допомоги до покращення зв'язку з пацієнтами та якості лікування. Вони роблять величезний позитивний вплив на компанії та клініки. Останні звіти показують, що впровадження хмарних рішень, безсумнівно, є посушним завданням для медичних організацій. Основними перешкодами є або проблеми безпеки, або брак внутрішніх технічних ресурсів.

У цьому дослідженні пропонується модель системної архітектури, яка зосереджена на розділенні та зниженні зв'язівок між компонентами не лише на серверній стороні (що вже є практикою), але й на стороні користувачього інтерфейсу (що ще не є стандартом). В запропонованій системі така декомпозиція дозволяє, з одного боку, ізолювати критичні дані та відповідні механізми роботи з ними, відокремлюючи їх від інших компонентів системи (для забезпечення надійності), і, з іншого боку, виділяти інші, менш важливі дані в окремі блоки, що відкриває можливість інтеграції з зовнішніми сервісами.

З огляду на особливості галузі, такий підхід до декомпозиції, включаючи сторону користувачього інтерфейсу, є досить обґрунтованим, оскільки системи в цій галузі часто функціонують "агрегатами" даних інших систем та джерел. Відповідно, модульний підхід дозволяє легко розширювати систему та налаштовувати зв'язки з зовнішніми системами, такими як постачальники медичної інформації, аналітичні системи та "white label" продукти, такі як чати та системи моніторингу.

Звісно, такий підхід до декомпозиції має свої витрати і труднощі в реалізації, але в даному випадку вони виправдані із-за значного обсягу функціоналу системи та вимог до безпеки, масштабованості та інтеграції. В той же час, він дозволяє підтримувати відносно не критичні вимоги до продуктивності та швидкодії системи.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Applications Of Cloud Computing In Healthcare URL: <https://www.ridge.co/blog/cloud-computing-in-healthcare/> (дата звернення: 23.10.2023).
2. Digital health (digital healthcare) URL: <https://www.techtarget.com/searchhealthit/definition/digital-health-digital-healthcare> (дата звернення: 12.10.2023).

3. REPORT on the use of cloud computing in health URL: https://health.ec.europa.eu/system/files/2018-02/ev_20151123_co06_02_en_0.pdf (дата звернення: 25.10.2023).
4. Digital health at the edge: Three use cases for the healthcare industry URL: <https://stlparkers.com/articles/edge-computing/digital-health-at-the-edge/> (дата звернення: 24.10.2023).
5. Cam, J. Micro Frontends. URL: <https://martinfowler.com/articles/micro-frontends.html> (дата звернення: 16.11.2023).
6. Amanda, L. White Label Software Explained: Benefits and Examples. URL: <https://growsurf.com/blog/white-label-software> (дата звернення: 18.11.2023).

ОРКЕСТРУВАННЯ ОБЧИСЛЕНЬ У МІКРОСЕРВІСНІЙ АРХІТЕКТУРІ

Полещук В.О.¹, Булах Б.В.²

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Кафедра системного проектування, Київ, Україна

¹ vladpoleshchuk@gmail.com, ² bogdan.bulakh@gmail.com

Це дослідження вивчає оркестрування обчислень в контексті мікросервісної архітектури, зосереджуючись на різних аспектах, таких як підходи до оркестрування в сервісно-орієнтованій архітектурі, інструменти для оркестрування, патерни оркестрування в мікросервісах, підходи до розробки оркестраторів мікросервісів, а також практична реалізація, що включає створення оркестратора, керованого подіями. Дослідження поєднує огляд літератури, аналіз інструментів та практичну реалізацію, щоб отримати уявлення про ефективні стратегії оркестрування. Результати дослідження відкривають нові перспективи в цій галузі, а практичний компонент пропонує зручний конструктор для оркестрування, керованого подіями, що підвищує зручність використання мікросервісів у реальних сценаріях.

Таким чином, це дослідження не тільки розкриває тонкощі оркестрування мікросервісів, але й робить відчутний внесок у вигляді зручного для користувача конструктора. Цей інструмент має практичне значення, пропонуючи спрощений шлях для розробників та організацій, які прагнуть використати можливості мікросервісів у послідовний та організований спосіб. Це дослідження відкриває можливості для подальшого вивчення і застосування в постійно мінливому ландшафті архітектури мікросервісів.

Ключові слова: оркестрація, мікросервіси, розробка оркестратора, інструментарій оркестрування, шаблони оркестрування.

1. ВСТУП

У постійно мінливому світі архітектури програмного забезпечення мікросервіси стали трансформаційною парадигмою, пропонуючи неперевершену гнучкість, масштабованість та оперативність у розробці додатків. Оскільки компанії все частіше переходять на мікросервіси, організація обчислень у цій розподіленій і модульній структурі стає критично важливою точкою координації. Цей звіт заглиблюється у багатогранну сферу оркестрування обчислень в архітектурі мікросервісів, досліджуючи різноманітні підходи, інструменти та патерни. Наше дослідження охоплює тонкощі оркестрування сервісів з метою розібратися в складнощах, вирішити проблеми та поділитися ідеями, які допоможуть архітекторам та розробникам орієнтуватися в динамічній екосистемі мікросервісів.

Архітектура мікросервісів, з її децентралізованим і модульним дизайном, вимагає ефективної оркестровки для безперебійної роботи. Оркестрування передбачає координацію взаємодії між розподіленими мікросервісами, і для цього процесу критично важливими є три ключові виміри: підходи, інструменти та патерни.

2. АКТУАЛЬНІСТЬ

Мікросервісна архітектура набула широкого розповсюдження завдяки своїй масштабованості, гнучкості та здатності сприяти швидкому розвитку. Однак, ефективна оркестрація обчислень в рамках цієї парадигми залишається важливою проблемою, що постійно розвивається. Оскільки організації все більше переходять на мікросервіси, потреба в ефективній оркестровці стає першочерговою для забезпечення безперебійної комунікації, масштабованості та відмовостійкості.

3. МЕТА ДОСЛІДЖЕННЯ

Це дослідження має на меті заглибитися в оркестровку обчислень в архітектурі мікросервісів, зосередившись на вирішенні поточних проблем та підвищенні ефективності взаємодії сервісів. Мета – надати практичні знання та методології, які допоможуть архітекторам і розробникам орієнтуватися в тонкощах оркестрування обчислень у динамічних середовищах мікросервісів, а також запропонувати інструмент для створення оркесторів на основі подій.

4. ПІДХОДИ ОРКЕСТРУВАННЯ

Сучасні підходи до оркестрування включають централізовані моделі, де центральний оркестратор координує мікросервіси, та децентралізовані стратегії, засновані на хореографії, де мікросервіси спільно взаємодіють без центрального контролера. Також з'явилися гібридні моделі, які поєднують централізовані та децентралізовані елементи. Ці підходи відрізняються компромісами, пов'язаними з контролем, масштабованістю та адаптивністю.

Business Process Execution Language, BPEL [1]:

BPEL, яку часто називають WS-BPEL (Web Services Business Process Execution Language), є стандартом для визначення та виконання бізнес-процесів на основі веб-сервісів. Він надає структуровану мову на основі XML для опису потоку сервісів та їх взаємодії. BPEL дозволяє моделювати складні бізнес-процеси, використовуючи такі конструкції, як дії, потоки та партнерські зв'язки.

Enterprise Service Bus (ESB) [2]:

Enterprise Service Bus – це технологія проміжного програмного забезпечення, яка виступає в якості центрального вузла для маршрутизації, трансформації та організації взаємодії сервісів. ESB надає набір готових роз'ємів та адаптерів, які дозволяють різним сервісам взаємодіяти один з одним.

Оркестрування на основі хореографії:

Приймає децентралізовану модель, де мікросервіси автономно взаємодіють без центрального контролера. Цей підхід спирається на спільні протоколи, що сприяє гнучкості, але вимагає чіткого розуміння міжсервісної комунікації.

Оркестрування, орієнтоване на людину:

Людино-орієнтована оркестрація, також відома як оркестрація робочого процесу, зосереджена на управлінні як автоматизованою взаємодією служб, так і людськими завданнями в рамках робочого процесу. Цей підхід підходить для процесів, які передбачають схвалення, рецензування або участь користувачів.

Оркестрування на основі правил:

Використовує заздалегідь визначені правила для управління потоком операцій. Оркестрування на основі правил забезпечує адаптивність, дозволяючи динамічно реагувати на мінливі умови на основі встановлених правил.

Динамічна оркестрація:

Підкреслює адаптивність і реагування на різні робочі навантаження. Динамічна оркестрація дозволяє в режимі реального часу вносити корективи у взаємодію сервісів і розподіл ресурсів, забезпечуючи оптимальну продуктивність системи.

5. ІНСТРУМЕНТИ ДЛЯ ОРКЕСТРУВАННЯ

Ефективна координація та управління розподіленими обчислювальними ресурсами вимагає складних інструментів, які можуть легко орієнтуватися в складнощах оркестрування. У цьому огляді досліджується та оцінюється спектр існуючих інструментів, призначених для оркестрування обчислень, від контейнерних платформ до безсерверних фреймворків та спеціалізованих рішень для оркестрування.

Kubernetes:

Платформа оркестрування контейнерів з відкритим вихідним кодом, що полегшує автоматизоване розгортання, масштабування та управління мікросервісами.

Docker Swarm:

Нативне рішення для кластеризації та оркестрування, що спрощує розгортання та управління мікросервісами.

Apache Mesos:

Універсальний менеджер кластерів, що абстрагує ресурси, придатний для оркестрування мікросервісів в масштабі в різних середовищах.

Nomad від HashiCorp:

Простий, але потужний оркестратор, розроблений для гнучкості та простоти використання при розгортанні та управлінні мікросервісами.

Service Grids:

Фреймворки, що забезпечують основу для сервіс-орієнтованих архітектур, полегшуючи оркестровку мікросервісів за допомогою виявлення сервісів та комунікації.

Платформи "Функція як послуга" (FaaS):

Забезпечують безсерверні обчислення, автоматизуючи виконання окремих функцій для полегшеного підходу до оркестрування мікросервісів.

Netflix Conductor:

Netflix Conductor – це механізм оркестрування з відкритим вихідним кодом, розроблений Netflix для полегшення координації мікросервісів у складних робочих процесах. Його основна мета – впорядкувати виконання бізнес-процесів, що робить його особливо ефективним для сценаріїв, коли послідовність мікросервісів повинна працювати разом для досягнення певної бізнес-цілі.

Власні інструменти оркестрування:

Індивідуальні рішення, розроблені власними силами для задоволення конкретних організаційних вимог і тонкощів оркестрування мікросервісів.

6. ШАБЛони ОКРЕСТРУВАННЯ

Основні патерни оркестрування в архітектурі мікросервісів включають в себе наступні [3]:

Централізована оркестрація:

Використовує центрального оркестратора для управління потоком діяльності та комунікації між мікросервісами.

Оркестрування на основі хореографії:

Використовує децентралізовану модель, де мікросервіси співпрацюють незалежно, покладаючись на спільні протоколи та неявні угоди.

Гібридна оркестровка:

Поєднує централізовані та децентралізовані підходи, адаптуючись до унікальних потреб різних сценаріїв мікросервісів.

Оркестрування на основі правил:

Використовує заздалегідь визначені правила для управління послідовністю та умовами взаємодії мікросервісів.

Динамічна оркестрація:

Надає пріоритет гнучкості та адаптивності, дозволяючи системі динамічно підлаштовуватися під мінливі умови та робочі навантаження.

На закінчення, оркестрування обчислень в мікросервісній архітектурі вимагає продуманого вибору підходів, інструментів і патернів. Динамічна і масштабована природа мікросервісів вимагає ретельного розгляду цих аспектів для забезпечення ефективних і відмовостійких розподілених систем.

7. ОГЛЯД ІСНУЮЧИХ ПІДХОДІВ РОЗРОБКИ ОРКЕСТРАТОРІВ

Розробка оркестраторів включає такі аспекти, як вибір мови, архітектурний дизайн та стратегії інтеграції. Існуючі підходи охоплюють проксі-оркестри, сайдкари-оркестри, автономні оркестри, працюючі оркестри та безсерверні оркестри. Кожен підхід орієнтований на конкретні випадки використання та уподобання, надаючи розробникам різноманітні можливості для створення оркестраторів, пристосованих до їхньої архітектури мікросервісів. Серед найпопулярніших підходів:

Оркестратор на основі проксі

Оркестратор на основі проксі діє як посередник між мікросервісами. Він перехоплює зв'язок між сервісами, що дозволяє йому керувати такими аспектами, як балансування навантаження, маршрутизація та безпека.

Sidecar Orchestrator [4]

У підході на основі сайдкарів кожен мікросервіс поєднується з невеликим незалежним компонентом (сайдкаром), який виконує специфічні для оркестрування завдання. Це може включати виявлення сервісів, балансування навантаження і зв'язок з центральним оркестром.

Автономний оркестратор [4]

Автономний оркестратор – це спеціальний, незалежний сервіс, відповідальний за координацію мікросервісів. Він забезпечує централізовану точку контролю для управління взаємодією сервісів, виконанням робочих процесів та управлінням станом.

Оркестратор в процесі експлуатації

Оркестратори в процесі експлуатації вбудовують логіку оркестрування безпосередньо в кожну мікрослужбу, що дозволяє їм керувати власними взаємодіями та робочими процесами. Цей підхід іноді називають "мікро-оркестровкою".

Безсерверний оркестратор [5]

Безсерверні обчислювальні платформи також можуть слугувати оркестраторами для мікросервісів. Вони забезпечують керовану подіями оркестровку за допомогою функцій, які виконуються у відповідь на події або тригери.

8. РОЗРОБКА ОРКЕСТРАТОРА НА ОСНОВІ ПОДІЙ

Пропонуючи новий інструмент в арсеналі оркестрування мікросервісів, Constructor for Event-Driven Orchestrator дозволяє користувачам легко оголошувати та налаштовувати сервіси. Використовуючи YAML-файл, користувачі можуть вказувати сервіси, визначати протоколи зв'язку (наприклад, черги, REST API), формулювати події та визначати послідовність, в якій сервіси обробляють ці події. Цей інструмент не тільки впорядковує процес оркестрування, але й покращує контроль і кастомізацію для користувача в екосистемі мікросервісів.

Структура реалізації:

Конфігурація YAML:

Користувачі визначають сервіси, протоколи зв'язку, події та порядок обробки у структурованому файлі YAML. Цей декларативний підхід спрощує конфігурацію та покращує читабельність.

Декларація сервісу:

Кожна служба декларується з основними атрибутами, такими як ім'я, кінцева точка і тип. Це створює фундаментальне розуміння ландшафту мікросервісів.

Протоколи зв'язку:

Користувачі формулюють протоколи зв'язку між сервісами, вказуючи, чи відбувається взаємодія через черги, REST API або інші канали зв'язку. Ця гнучкість пристосована до різноманітних архітектур мікросервісів.

Специфікація подій:

Події визначаються з відповідними атрибутами, забезпечуючи чітке розмежування тригерів для оркестрування. Сюди входять типи подій, формати даних і будь-яка контекстна інформація, важлива для взаємодії сервісів.

Порядок виконання сервісів:

Інструмент дозволяє користувачам чітко вказати порядок, в якому сервіси обробляють події. Ця ієрархічна послідовність забезпечує структурований потік, що відповідає тонкощам запланованої оркестровки.

Обробка помилок та ведення журналів:

Включення надійних механізмів обробки помилок і функцій журналювання забезпечує прозорість і відмовостійкість. Користувачі можуть визначати стратегії обробки помилок і мати доступ до детальних журналів для полегшення пошуку та усунення несправностей і моніторингу.

Динамічне масштабування:

Щоб задовольнити вимоги динамічного масштабування, конструктор може бути спроектований так, щоб легко адаптуватися до змін у кількості сервісів або екземплярів. Це гарантує, що інструмент реагує на розвиток архітектури мікросервісів.

Інтеграція з існуючими інструментами:

Конструктор повинен легко інтегруватися з існуючими інструментальними засобами розробки та розгортання. Сумісність з популярними фреймворками та платформами сприяє його прийняттю в різноманітних екосистемах мікросервісів.

Дотримуючись цих принципів реалізації, Constructor for Event-Driven Orchestrator стає універсальним і орієнтованим на користувача інструментом, що дозволяє розробникам легко і точно оркеструвати мікросервіси.

9. ПРОБЛЕМИ

Архітектура мікросервісів, пропонуючи масштабованість і гнучкість, створює складні проблеми в оркеструванні обчислень:

Координація послуг: Координація виконання та підтримка зв'язку в децентралізованому середовищі є складним завданням.

Децентралізований контроль: Досягнення послідовного контролю та забезпечення дотримання всіма сервісами визначених робочих процесів викликає труднощі.

Динамічне масштабування: Організація обчислень в умовах динамічного масштабування та коливань розподілу ресурсів є нетривіальним завданням.

Обробка помилок: Виявлення, ізоляція та відновлення після помилок у різних сервісах створює проблеми з відмовостійкістю.

Узгодженість даних: Управління узгодженістю та станом даних між мікросервісами, особливо в розподілених базах даних, є складним завданням.

Безпека: Послідовне впровадження та управління політиками безпеки в різних мікросервісах вимагає ретельної уваги.

Інтеграція інструментарію: Інтеграція різноманітних інструментів та фреймворків у середовищі мікросервісів ускладнює безперебійну інтеграцію робочих процесів.

Складність моніторингу: Ефективний моніторинг та спостережливість в розподіленій архітектурі мікросервісів є складним завданням, що вимагає передових рішень.

Вирішення цих проблем є ключовим для реалізації повного потенціалу мікросервісів у сучасній розробці додатків.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Singhal N., Sakthivel U., Raj P. SELECTION MECHANISM OF MICRO-SERVICES ORCHESTRATION VS. CHOREOGRAPHY. researchgate.net. URL: https://www.researchgate.net/profile/Neha-Singhal-5/publication/330960463_Selection_Mechanism_of_Micro-Services_Orchestration_Vs_Choreography/links/5ca474e7a6fdcc12ee8f3a80.

2. Menge F. Enterprise Service Bus. programm.froscon.org. URL: https://programm.froscon.org/2007/attachments/15-falko_menge_-_enterprise_service_bus.pdf.

3. Taibi D., Lenarduzzi V., Pahl C. Architectural Patterns for Microservices: a Systematic Mapping Study. bia.unibz.it. URL: <https://bia.unibz.it/esploro/outputs/conferenceProceeding/Architectural-Patterns-for-Microservices-A-Systematic-Mapping-Study/991005773017601241#file-0> (дата звернення: 25.10.2023).

4. Burns B. Designing Distributed Systems: Patterns and Paradigms for Scalable, Reliable Services. O'Reilly Media, 2018. 166 с.

5. Fowler S. Production-Ready Microservices. O'Reilly Media, 2016. 172 с.

АНАЛІЗ РИЗИКІВ В ЗАДАЧАХ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ

Северин М.С.¹, Мухін В.Є.²

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

¹ severyn.maxim@gmail.com, ² v_mukhin@i.ua [0000-0002-1206-9131]

Дослідження спрямоване на аналіз ризиків у сфері інформаційної безпеки, зокрема в завданнях, пов'язаних із захистом комп'ютерної системи. Робота включає в себе дослідження даних щодо існуючих різновидів кібератак, зокрема на основі NSL-KDD датасету, огляд існуючих моделей механізмів виявлення ризиків, а також розробка нової моделі. Науковою новизною є використання нової комбінації методів машинного навчання та їх параметрів для систем виявлення вторгнень (IDS) на основі мережі та хосту за допомогою аномалій, виявлених у мережевому трафіку комп'ютерної системи.

Ключові слова: інформаційна безпека, методи оцінки ризиків, різновиди кібератак, моделі виявлення і прогнозування, аналіз проблем безпеки комп'ютерних систем, штучний інтелект, машинне навчання.

1. ВСТУП

Тема кібербезпеки є дуже актуальною в сучасному цифровому світі. З кожним днем все більше організацій, включаючи урядові структури, банки, медичні установи та промислові підприємства стикаються з викликами і загрозами, пов'язаними з безпекою їх інформаційних систем.

У зв'язку зі зростанням кількості технологічних засобів, використовуваних у роботі організацій, збільшується і ймовірність виникнення кібератак, витоку конфіденційної інформації, втрати даних та порушення цілісності систем. Такі інциденти можуть мати серйозні наслідки, включаючи фінансові збитки, втрату довіри клієнтів і навіть негативний вплив на безпеку громадськості.

Аналіз ризиків в задачах інформаційної безпеки стає необхідністю для виявлення потенційних загроз та визначення належних заходів захисту. Це включає ідентифікацію вразливостей систем, аналіз можливих загроз і визначення ймовірності їх виникнення, а також оцінку потенційних наслідків [1].

Дослідження в області оцінки ризиків в задачах інформаційної безпеки має велике значення для розробки ефективних стратегій захисту інформаційних систем. Це допомагає організаціям прогнозувати, попереджати та зменшувати проблеми, пов'язані зі зловживанням інформацією, забезпечуючи стійкість і надійність їх інфраструктури.

Загальні напрями досліджень в даній галузі включають розробку нових методів оцінки ризиків, вдосконалення інструментальних засобів для аналізу та моделювання, вивчення тенденцій у сфері кібербезпеки та розробку стратегій захисту. Розуміння загроз і впровадження ефективних заходів безпеки інформаційних систем є важливими компонентами сучасного світу, який все більше стає цифровим та залежним від інформаційних технологій.

Ця стаття поглиблюється у сучасні підходи аналізу ризиків безпеки комп'ютерних систем. У ній досліджуються нові техніки, дані та можливості. Розробка моделі механізмів виявлення ризиків захисту є ключовим аспектом наших досліджень.

2. АНАЛІЗ ПОТЕНЦІЙНИХ РИЗИКІВ БЕЗПЕКИ КОМП'ЮТЕРНИХ СИСТЕМ

Безпека комп'ютерної системи (КС) – це стан захищеності КС і мереж від атак зловмисників, які можуть призвести до несанкціонованого розкриття інформації, крадіжки або пошкодження обладнання, програмного забезпечення або даних, а також до порушення або неправильного спрямування послуг, які вони надають.

Загроза – це потенційний ризик використання вразливості для порушення безпеки та ймовірного пошкодження/порушення інформації/сервісу, що зберігається/пропонується в комп'ютерних системах або через канали зв'язку. Загроза комп'ютерній системі виникає тоді, коли порушується конфіденційність (запобігання доступу до неї несанкціонованих осіб), цілісність (неможливість модифікації без дозволу) та доступність (доступність на вимогу уповноважених осіб) інформації в системі. Таким чином, до загроз комп'ютерним системам загалом можна віднести будь-яку навмисну, ненавмисну або спричинену природним шляхом дію, що призводить до втрати/маніпуляції з даними або фізичного знищення апаратного забезпечення [2].

Серед найпопулярніших кібератак можна виділити наприклад:

- Атаки для збору інформації – це практика отримання зловмисником цінних даних. Ця дія повністю пасивна, оскільки не є явною атакою.
- Розподілені DoS-атаки (DDoS) - жертву атакують одночасно з великої кількості окремих систем. DDoS-атаки зазвичай здійснюються за допомогою бот-мереж. Бот-майстер – це зловмисник, який опосередковано атакує машину-жертву, використовуючи армію ботів. DDoS-атаки відбуваються, коли велика кількість скомпрометованих систем діє синхронно і координується під контролем зловмисника, щоб повністю виснажити її ресурси і змусити відмовити в обслуговуванні справжнім користувачам.
- Атаки на основі IoT. Проблемою Інтернету речей є його слабка захищеність, оскільки ці пристрої часто ігноруються при встановленні патчів безпеки, які створюють лазівки для зловмисників, що дозволяють їм захопити ці пристрої та проникнути в мережу. Атака на основі IoT – це будь-яка кібератака, яка включає використання жертвою IoT для проникнення в мережу шкідливого програмного забезпечення.
- Програми-вимагачі – це шкідливе програмне забезпечення, яке перешкоджає доступу до комп'ютера або файлів на ньому. Комп'ютери можуть бути заблоковані, а файли зашифровані. Від жертви вимагають викуп за зняття обмежень, і це відображається в системі жертви.
- Шпигунське та рекламне програмне забезпечення – це програми, які мають спільну властивість збирати особисту інформацію користувачів без їхнього відома. Шпигунське ПЗ включає клавіатурні шпигуни, які реєструють усе, що набирається на клавіатурі, і створюють високий ризик викрадення особистих даних.
- Спам - це небажані масові повідомлення електронної пошти, які дратують користувача надмірною кількістю небажаних листів. Це створює навантаження для постачальників послуг зв'язку, організацій та приватних осіб. Спам також використовується як інструмент для фішингу.
- Атаки на веб-сайти – націлені на компоненти браузера, які можуть бути незахищеними, навіть якщо вони встановлені в браузері. Атаки з використанням SQL-ін'єкцій спрямовані на будь-який веб-сайт або веб-додаток, що використовує базу даних SQL, таку як MySQL, Oracle тощо, шляхом використання недоліків безпеки в програмному забезпеченні додатку [3–5].

Класифікація ризиків інформаційної безпеки дозволила систематизувати можливі загрози та їх характер, що є важливим етапом для подальшого аналізу та захисту інформації. Аналіз ризиків інформаційної безпеки виявився складним процесом, який передбачає врахування багатьох факторів, таких як ймовірність впливу та характер загроз, який не завжди легко оцінити чисельно.

3. АНАЛІЗ ІСНУЮЧИХ МОДЕЛЕЙ ВИЯВЛЕННЯ РИЗИКІВ БЕЗПЕКИ В КС

Для виявлення загроз/вторгнень у комп'ютерну систему існує багато мережевих сканерів або сканерів портів (Nessus, OpenVAS, Qualys, Rapid7Nexpose, Metasploit). Аналіз системи на основі мережевих та хост-даних є важливою складовою безпеки комп'ютерних систем. Загальний принцип заключається в тому, що мережевий трафік відстежується та витягує зашифрований зв'язок у формі файлів захоплення пакетів. При цьому використовується трафік HTTP, оскільки це найпоширеніший протокол для глобального зв'язку. Заголовки HTTP містять дані, які можна використовувати для виявлення зловмисної поведінки. Щоб дістати ці дані, можна проаналізувати потоки TCP із трасуванням HTTP із файлів захоплення пакетів (PCAP) або іншим аналізатором (Wireshark, Zeek).

Система, яка називається системою виявлення вторгнень (Intrusion detection system, IDS), відстежує мережевий трафік на наявність зловмисних транзакцій і негайно надсилає сповіщення, коли це спостерігається. Це програмне забезпечення, яке перевіряє систему на наявність зловмисних дій або порушень політики. Кожна незаконна діяльність або порушення часто фіксується централізовано за допомогою системи SIEM або повідомляється адміністрації.

IDS бувають 5 типів:

- Система виявлення вторгнень у мережу (NIDS): Системи виявлення вторгнень у мережу (NIDS) встановлюються в запланованій точці мережі для перевірки трафіку з усіх пристроїв у мережі. Він виконує спостереження за трафіком, що проходить у всій підмережі, і зіставляє трафік, який передається підмережами, із колекцією відомих атак. Після виявлення атаки або виявлення ненормальної поведінки сповіщення можна надіслати адміністратору. Прикладом NIDS є його встановлення в підмережі, де розташовані брандмауери, щоб побачити, чи хтось намагається зламати брандмауер.
- Система виявлення вторгнень (HIDS): Системи виявлення вторгнень (HIDS) працюють на незалежних хостах або пристроях у мережі. HIDS відстежує лише вхідні та вихідні пакети з пристрою та сповіщатиме адміністратора, якщо буде виявлено підозрілу або зловмисну активність. Він робить знімок існуючих системних файлів і порівнює його з попереднім знімком. Якщо файли аналітичної системи були відредаговані або видалені, сповіщення надсилається адміністратору для дослідження. Приклад використання HIDS можна побачити на критично важливих машинах, на яких не очікується зміна компонентів.
- Система виявлення вторгнень на основі протоколу (PIDS): Система виявлення вторгнення на основі протоколу (PIDS) складається з системи або агента, який постійно розміщується на передній частині сервера, контролюючи та інтерпретуючи протокол між користувачем/пристроєм і сервером. Він намагається захистити веб-сервер, регулярно відстежуючи потік протоколу HTTPS і приймаючи відповідний протокол HTTP. Оскільки HTTPS не зашифрований і перед миттєвим переходом на рівень веб-презентації, ця система повинна знаходитися в цьому інтерфейсі між використанням HTTPS.

- Система виявлення вторгнень на основі протоколу додатків (APIDS): Система виявлення вторгнень на основі протоколу додатків (APIDS) – це система або агент, який зазвичай знаходиться в групі серверів. Він визначає вторгнення, відстежуючи та інтерпретує зв'язок за протоколами, що стосуються окремих програм. Наприклад, це явно відстежуватиме протокол SQL для проміжного програмного забезпечення під час його взаємодії з базою даних на веб-сервері.
- Гібридна система виявлення вторгнень: гібридна система виявлення вторгнень створюється шляхом поєднання двох або більше підходів до системи виявлення вторгнень. У гібридній системі виявлення вторгнень дані агента хоста або системи поєднуються з мережевою інформацією для створення повного уявлення про мережеву систему. Гібридна система виявлення вторгнень більш ефективна в порівнянні з іншими системами виявлення вторгнень. Prelude є прикладом Hybrid IDS [6, 7].

4. РОЗРОБКА МОДЕЛІ

Згідно усього вище сказаного, було запропоновано архітектуру системи, що містить набір компонентів та забезпечує взаємодію між ними, як показано на рисунку 1.

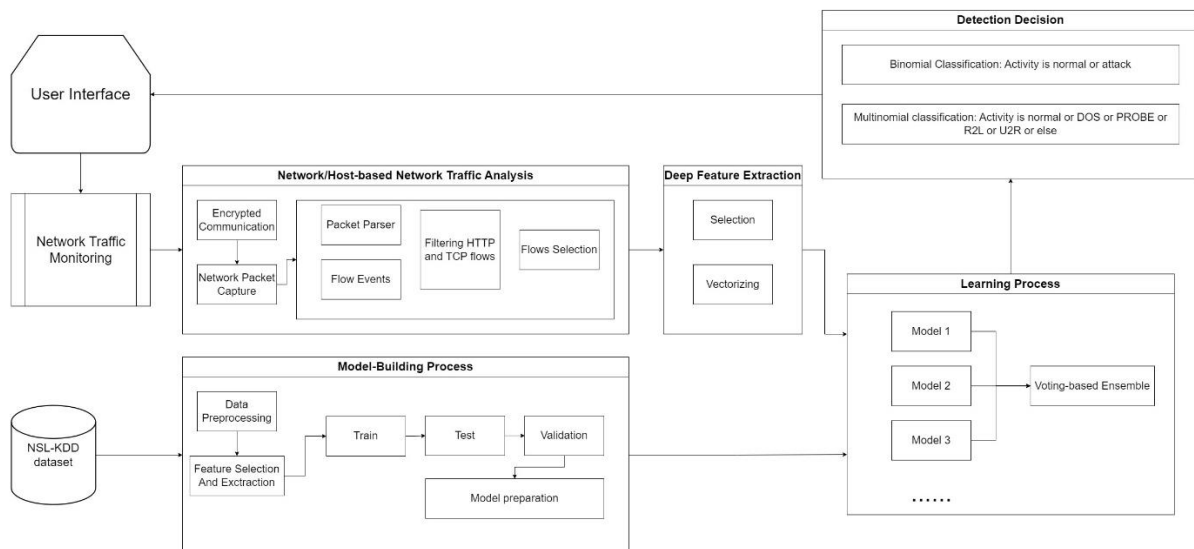


Рисунок 1. Архітектура програми

В інтерфейсі користувача можна запускати сканування мережі, а також отримувати результати після завершення. Блок Network Traffic Monitoring відповідає за збір даних.

- Шифрована комунікація (Encrypted Communication) – трафік мережі шифрується, щоб захистити вміст від несанкціонованого доступу. Існують методи аналізу метаданих і інших аспектів для отримання корисної інформації.
- Захоплення пакетів мережі (Network Packet Capture) включає в себе запис пакетів мережі для подальшого аналізу. Це дозволяє детально вивчати взаємодію на мережі, включаючи дані, що передаються між пристроями.
- Аналізатор пакетів (Packet Parser) використовується для розбору (аналізу) записаних пакетів мережі. Парсери можуть виділяти різноманітні дані, такі як заголовки пакетів, адреси, протоколи тощо.
- Події потоків (Flow Events) - вказує на виникнення подій або змін в потоках мережевого трафіку.

- Фільтрація HTTP і TCP потоків (Filtering HTTP and TCP Flows) – означає вибірку обробку потоків, пов'язаних з протоколами HTTP і TCP. Це дозволяє виділяти конкретний тип трафіку для подальшого аналізу.
- Вибір потоків (Flow Selection) полягає в обранні конкретних мережевих потоків для подальшого аналізу. Це може бути корисним для зосередження на конкретних подіях або пристроях.

Deep Feature Extraction – відповідає за вибір та (можливо) за векторизацію даних, отриманих скануванням. Не всі дані можуть бути потрібними для аналізу, а також для деяких нейронних мереж їх початковий вигляд може бути неприйнятним без векторизації [8].

Model-building process – побудова моделей машинного навчання відбувається з використанням NSL-KDD датасета. NSL-KDD (Network-based Intrusion Detection Dataset – KDD Cup 99) є датасетом, який використовується для розробки та тестування систем виявлення вторгнень у мережі (IDS). Він є розширенням відомого датасету KDD Cup 99, який був використаний для конкурсу KDD Cup 1999 щодо виявлення вторгнень.

NSL-KDD містить п'ять основних класів трафіку:

- Normal – легітимний мережевий трафік, який вважається безпечним.
- DoS (Denial of Service) – атаки, спрямовані на перевантаження ресурсів, забираючи їх від легітимних користувачів.
- Probe – атаки, які включають сканування мережі з метою виявлення слабких місць для подальших атак.
- U2R (User to Root) – атаки, спрямовані на отримання прав адміністратора системи з прав користувача.
- R2L (Remote to Local) – атаки, які спрямовані на отримання доступу до системи через мережу, якщо ви вже є користувачем на цій системі.

NSL-KDD містить різноманітні атаки та легітимний трафік для створення більш реалістичного середовища для виявлення вторгнень. Він став популярним інструментом для тестування та порівняння алгоритмів виявлення вторгнень в мережах.

Learning process – компонент, що визначає механізм вивчення обраними ML моделями даних, отриманих на основі згенерованого трафіка. Ensemble – це стійка модель, створена шляхом системного поєднання базових технологій. На відміну від індивідуальних моделей, модель ансамблю здатна вирішувати проблеми класифікації та регресії. У запропонованому дослідженні використовується ансамблевий підхід м'якого опитування. Для початку ми використали навчальні дані для створення основних моделей. А потім їх ефективність перевіряється за допомогою тестових даних, причому кожна модель створює унікальну класифікацію. Щоб отримати остаточну класифікаційну продуктивність, ансамблеве навчання використовує оцінки кількох підходів як додаткову інформацію.

Detection Decision – може працювати у двох режимах: або просто помічати аномальну активність (виявлення), або визначати ще й характер атаки (прогнозування). Для цього застосовується біноміальна та поліноміальна класифікації. Результат користувач отримує в інтерфейсі програми [9].

5. ВИСНОВКИ

Запропонована архітектура системи для аналізу ризиків порушення захищеності в КС містить комплекс компонентів, що взаємодіють для повноцінної роботи. Основні компоненти включають захоплення пакетів мережі, аналізатор пакетів, виявлення подій потоків, фільтрацію HTTP і TCP потоків, вибір потоків, витягування глибоких ознак, процес побудови моделей машинного навчання з використанням NSL-KDD датасету та ансамбль-навчання.

NSL-KDD датасет використовується для розробки та тестування систем виявлення вторгнень у систему, містить різноманітні класи трафіку, включаючи атаки DoS, Probe, U2R, R2L, а також легітимний трафік. Застосування ансамбль-навчання дозволяє покращити класифікаційну продуктивність, використовуючи оцінки різних моделей як додаткову інформацію. Результати аналізу можуть використовуватися для виявлення аномалій або навіть для передбачення характеру атаки.

Однак важливо враховувати, що жодна система не є абсолютною та повністю невразливою. Запропонована архітектура забезпечує важливі етапи аналізу та захисту, але враховувати інші аспекти, такі як соціальна інженерія та внутрішні загрози, також є критичним. Загальна безпека вимагає комплексного підходу, який об'єднує технічні та людські аспекти.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Information Security Risk Analysis, Second Edition, Thomas R. Peltier URL: https://www.academia.edu/40140524/Information_security_risk_analysis_thomas_r_peltier
2. Ievgeniia Kuzminykh , Bogdan Ghita, Volodymyr Sokolov Security Risk Assessment. URL: https://www.researchgate.net/publication/353436973_Information_Security_Risk_Assessment#fullTextFileContent
3. Mouna Jouinia Classification of security threats in information systems. URL: https://www.researchgate.net/publication/315714820_Classification_of_security_threats_in_information_systems#fullTextFileContent
4. Analysis of Computer Network Security Problems and Countermeasures. URL: https://www.researchgate.net/publication/320177233_Analysis_of_Computer_Network_Security_Problems_and_Countermeasures
5. Comparative analysis of modern methods of cryptographic protection of information URL: https://www.academia.edu/35708980/Comparative_analysis_of_modern_methods_and_algorithms_of_cryptographic_protection_of_information
6. Intrusion Detection System (IDS). URL: <https://www.geeksforgeeks.org/intrusion-detection-system-ids/>
7. Different types of Intrusion Detection Systems (IDS). URL: <https://wisdomplexus.com/blogs/different-types-of-intrusion-detection-systems-ids/>
8. Bisyrone Wahyudi, Kalamullah Ramli, and Hendri Murfi Implementation and Analysis of Combined Machine Learning Method for Intrusion Detection System. URL: https://www.researchgate.net/publication/326983606_Implementation_and_Analysis_of_Combined_Machine_Learning_Method_for_Intrusion_Detection_System
9. Cyber-Risk Forecasting using Machine Learning Models and Generalized Extreme Value Distributions. URL: <https://hal.science/hal-03814979/document>

ПЛАТФОРМИ З НИЗЬКОЮ ЗАТРИМКОЮ ДЛЯ ОБРОБКИ ПОТОКІВ ДАНИХ У РЕАЛЬНОМУ ЧАСІ

Сіркович А.І.¹, Харченко К.В.

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Інститут прикладного системного аналізу, Київ, Україна

¹ sirkovych.arsen@iit.kpi.ua

Стаття розглядає "платформи обробки поточкових даних з низькою затримкою" як складову розроблення. У тексті аналізуються стратегії та методи використання таких платформ, проводиться порівняльний аналіз популярних зразків цього типу. Дослідження виявляє нові можливості та переваги використання платформи для обробки поточкових даних у розробці, підкреслюючи її науковий та практичний внесок у галузь програмного забезпечення.

Ключові слова: наукова конференція, доповідь, сервіс, низька затримка, поточкові дані.

1. ВСТУП

Впровадження інформаційних систем, здатних обробляти великі обсяги даних в реальному часі, стало актуальною проблемою в епоху цифрової трансформації. За зростання обсягів даних з різних джерел виникає потреба в створенні ефективних і високопродуктивних інформаційних систем. Проте викликом є необхідність обробки цих даних з низькою затримкою для оперативного реагування та прийняття обґрунтованих рішень в реальному часі.

Робота буде фокусуватися на дослідженні аспектів цієї теми, включаючи аналіз сучасних високопродуктивних платформ для обробки поточкових даних, методи зниження затримки та підвищення продуктивності, а також розробку практичних застосувань для інформаційних систем в реальному часі.

2. ЩО TAKE LOW-LATENCY STREAM PROCESSING PLATFORMS?

Платформи з низькою затримкою для обробки потоків даних у реальному часі (Low-Latency Stream Processing Platforms) – це програмні рішення або середовища, які призначені для обробки поточкових даних з низькою затримкою, тобто з мінімальним часом між моментом створення даних і їх обробкою. Ці платформи використовуються в сценаріях, де важлива швидка реакція на надходження даних, такі як аналіз реального часу, моніторинг, прогнозування та інші застосування, де обробка даних повинна відбуватися негайно.

Їхні переваги полягають у здатності забезпечувати реальний час в обробці даних, забезпечуючи миттєву реакцію на зміни у навколишньому середовищі. Це важливо для застосунків, де необхідно оперативно реагувати на події та приймати швидкі та обґрунтовані рішення.

Додатково, платформи цього типу відзначаються високою швидкістю обробки великих обсягів даних, що дозволяє використовувати останні дані для аналізу та виведення результатів. Вони підтримують горизонтальну масштабованість, що дозволяє легко розширювати систему для роботи з ростом обсягу інформації.

Ці платформи також володіють модульністю та гнучкістю, що спрощує їх інтеграцію з іншими технологіями та системами. Можливість виконувати поточкову аналітику на льоту дозволяє отримувати актуальну інформацію для прийняття рішень.

Взагалі, платформи з низькою затримкою для обробки потоків даних у реальному часі дозволяють розробникам створювати високоефективні та гнучкі системи, що забезпечують негайну реакцію на зміни в навколишньому середовищі та динамічно взаємодіють з потоками інформації.

3. ПЕРЕВАГИ ВИКОРИСТАННЯ ТЕХНОЛОГІЇ

Використання платформ з низькою затримкою для обробки потоків даних у реальному часі дарує значні переваги в розробці різноманітних систем та застосунків. Ці платформи, спрямовані на миттєву обробку даних без затримок, стають ключовим інструментом у сучасних областях, де важлива оперативність та надійність систем.

Однією з головних переваг є можливість забезпечити реальний час у відповіді на події. Це означає, що система може миттєво реагувати та приймати рішення, що важливо для застосунків, де необхідна швидка реакція на зміни у середовищі.

Також слід відзначити оптимізацію продуктивності системи. Платформи з низькою затримкою дозволяють ефективно обробляти великі обсяги даних в реальному часі, надаючи можливість використовувати найновішу інформацію для прийняття рішень та виконання завдань.

Масштабованість є ще однією перевагою, оскільки систему можна легко розширювати за допомогою додаткових ресурсів для впорядкування з ростом обсягу даних та завдань обробки. Модульність і гнучкість також важливі, дозволяючи легко інтегрувати платформу з іншими інструментами та системами, що сприяє швидкій розробці та розгортанню різноманітних застосунків.

Крім того, можливість виконувати потокову аналітику, аналізуючи дані на льоту, полегшує отримання актуальної інформації. Ефективне використання ресурсів, оптимізація обчислювальних ресурсів та автоматизація оптимізації процесів також сприяють успішній розробці систем.

Загальною вагомою перевагою є те, що платформи з низькою затримкою для обробки потоків даних дозволяють розробникам створювати реактивні, ефективні та гнучкі системи, які відповідають вимогам сучасних динамічних інформаційних середовищ.

4. АРХІТЕКТУРА ПЛАТФОРМИ

Розглянемо структуру платформи для обробки потоків даних у реальному часі на прикладі Apache Kafka (рис. 1). Архітектура включає в себе кілька ключових елементів. Розглянемо їх детальніше.

Виробник (Producer). Цей компонент відповідає за генерацію та відправку потоку даних до Kafka-кластера. Виробник може виробляти дані з різних джерел та надсилати їх до одного або кількох тем Kafka.

Теми (Topics). Теми є логічними категоріями, які дозволяють організувати дані в Kafka. Кожен потік даних призначається певній темі, щоб інші компоненти могли підписатися на цю тему та отримувати відповідні дані.

Брокери (Brokers). Брокери відповідають за прийом, зберігання та розподіл поточкових даних. Кластер Kafka складається з декількох брокерів, які співпрацюють для забезпечення масштабованості та надійності.

Споживачі (Consumer). Споживачі підписуються на теми та обробляють потік даних, які надходять від виробників. Споживачі можуть бути організовані в групи для ефективної розподіленої обробки даних.

Групи споживачів (Consumer Groups). Групи споживачів дозволяють масштабувати споживання даних, розділяючи завдання між споживачами. Кожна група обробляє частину повідомлень, що дозволяє досягти паралельності та високої продуктивності.

Зберігання (Storage). Деякі дані можуть зберігатися в Kafka для подальшого використання або відновлення під час відмов. Дозволяє зберігати дані на задовільний термін для аналізу або відновлення стану системи.

Зокіп (ZooKeeper). ZooKeeper використовується для управління та координації компонентів Kafka. Забезпечує служби для виявлення та відновлення збоїв, а також для керування конфігурацією кластера.

Ця структура дозволяє Apache Kafka забезпечувати ефективну обробку потоків даних у реальному часі, масштабування та високу надійність в середовищі з низькою затримкою.

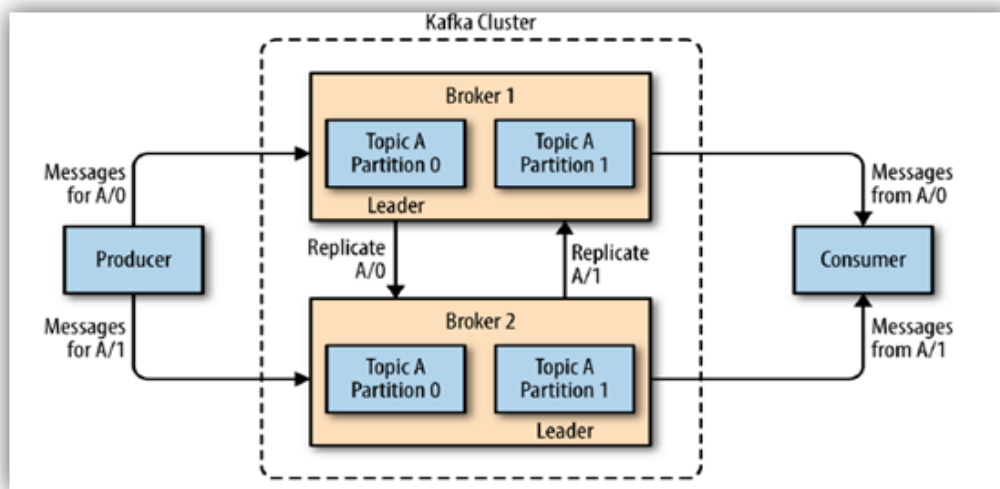


Рисунок 1. Приклад архітектури

5. ВИСНОВКИ

Дана робота, присвячена платформам з низькою затримкою для обробки потоків даних у реальному часі, розглянуто важливі аспекти цього передового напрямку в інформаційних технологіях. Аналізуючи такі платформи, як Apache Kafka, були визначені та проаналізовані ключові компоненти їхньої архітектури, включаючи виробників, споживачів, та механізми обробки даних.

Дослідження вказує, що платформи з низькою затримкою грають критичну роль у розвитку інформаційних систем, здатних працювати з великими потоками даних в режимі реального часу. Вони забезпечують високу швидкість обробки, гнучкість у взаємодії з різними джерелами даних, надійність та можливість аналізу інформації в реальному часі.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. "O'Sullivan, M., & Perry, M. (2019). Real-time Big Data Analytics: Emerging Architecture. *Journal of Computer Science and Technology*, 20(3), 437-452.
2. Smith, J., & Johnson, L. (2020). Stream Processing Systems: A Comparative Study. *International Journal of Data Science and Analytics*, 15(2), 187-204.
3. Brown, A., & Davis, R. (2018). Low-Latency Data Processing with Apache Kafka. *Journal of Distributed Computing*, 25(1), 89-102.
4. Gorman, T. (2017). *Real-time Analytics with Storm and Kafka*. O'Reilly Media.

СЕРВІС ВЕДЕННЯ ФІНАНСОВОЇ ЗВІТНОСТІ ДЛЯ ФОП

Скрипник А.В.¹, Булах Б.В.²

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Кафедра системного проектування, Київ, Україна

¹ programworkerbox@gmail.com, ² bogdan.bulakh@gmail.com

Основною метою дослідження є пошук шляхів побудови мінімалістичної, але гнучкої у налаштуванні системи автоматизації звітності для ФОП з урахуванням можливості розширення до інших форм підприємств. У реалізації, що пропонується, реалізовано функції для ведення ФОП 3 групи з можливістю CRUD операцій для фінансових записів та складання поквартальних податкових звітностей з обчисленням єдиного соціального внеску.

Ключові слова: SaaS, сервіс фінансової звітності, розподілена система, графічний інтерфейс

1. ВСТУП

В сучасному підприємницькому середовищі, де правила та вимоги до фінансової звітності постійно змінюються, роль ефективного бухгалтерського сервісу для фізичних осіб-підприємців (ФОП) в Україні стає надзвичайно важливою. Забезпечення правильного ведення бухгалтерського обліку та своєчасної подачі податкових звітів не лише дозволяє уникнути можливих штрафів та санкцій, але й є ключовим фактором для стабільного фінансового управління та успішної діяльності підприємців.

У зв'язку з постійними змінами в законодавстві та високим рівнем відповідальності перед податковими органами, ФОП повинні мати доступ до кваліфікованого бухгалтерського супроводу. Це стає ключовим елементом ефективного управління фінансами та може визначити конкурентоспроможність та стійкість бізнесу в умовах сучасного ринкового середовища.

2. ОПИС СИСТЕМИ BOOKKEEPING AS A SERVICE

Пропонується реалізувати систему звітності як мережевий застосунок, що складається з таких ланок: графічний web-інтерфейс, модуль бухгалтерських обчислень та реляційна база даних. Діаграму викликів модулів під час роботи системи подано на рис. 1.

2.1. Модуль графічного інтерфейсу

Модуль графічного інтерфейсу реалізований мовою Python за допомогою бібліотеки streamlit [1]. Він реалізує взаємодію користувача із системою та використовує автентифікацію Google [2] для надання доступу різним користувачам. Тут зберігаються токени для користування Google API для певного користувача та результати деяких запитів до модуля бухгалтерських обчислень для зменшення навантаження на нього. Приклад вигляду сторінки складання звітів подано на рис. 2. В результаті використання зовнішнього сервісу для автентифікації користувачів, уникнено необхідності зберігання паролів та їх верифікації, що підсилює безпеку системи до зламів паролів.

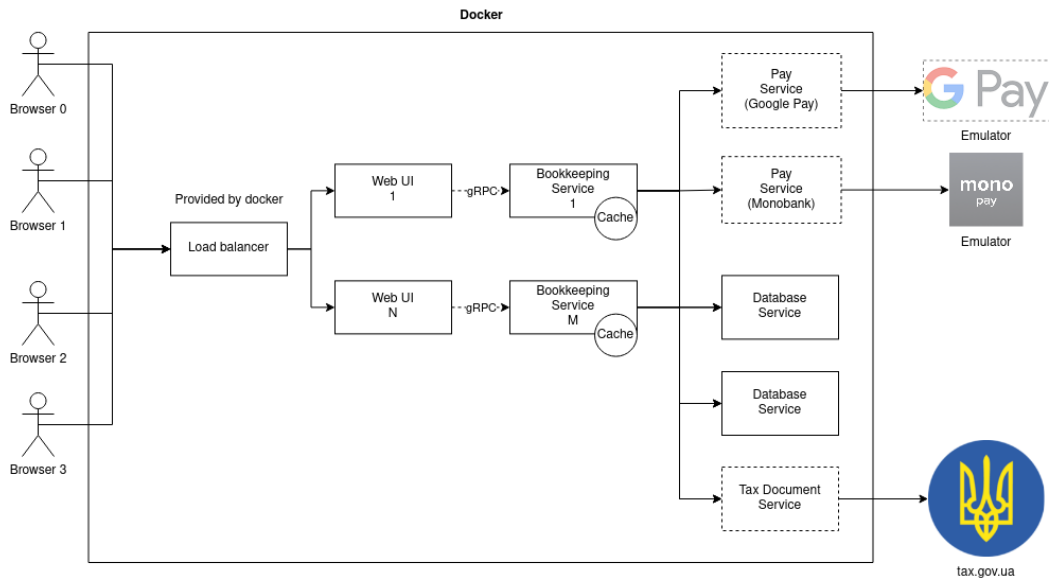


Рисунок 1. Діаграма викликів модулів системи Bookkeeping as a Service

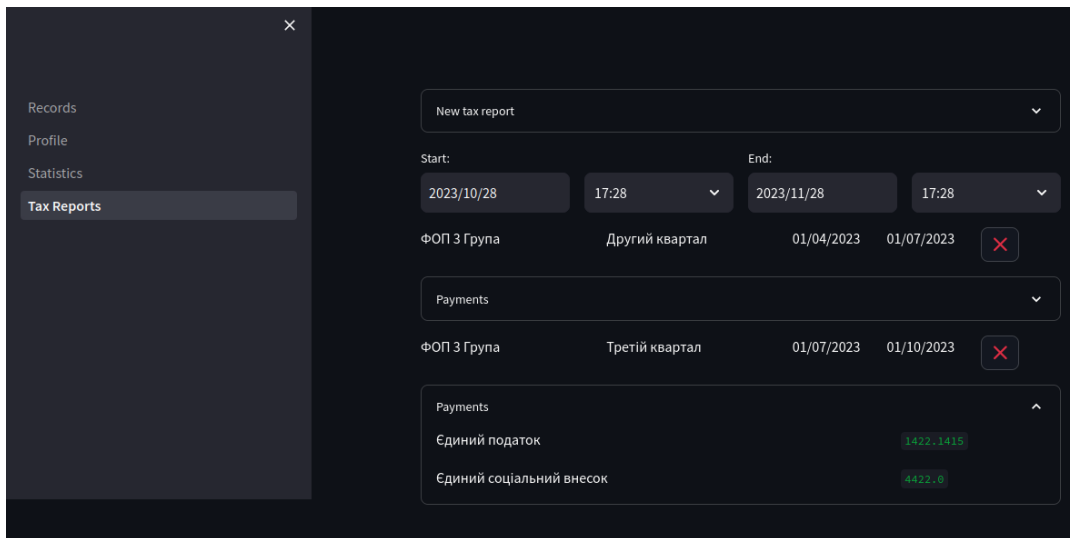


Рисунок 2. Вигляд сторінки подання звітів в системі Bookkeeping as a Service

2.2. Модуль бухгалтерських обчислень і взаємодії з базою даних

Основні операції з даними користувачів виконує модуль бухгалтерських обчислень. Він реалізує протокол GRPC [3] для взаємодії з модулем графічного інтерфейсу. Для цього використовується окремий пакет Python з визначенням інтерфейсів для взаємодії між модулями, який використовується як графічним інтерфейсом для формування запитів так і модулем бухгалтерських обчислень. Він генерує відповідні файли Python за допомогою інтерфейсу сервісу, описаному за допомогою Protobuf [4].

Також цим модулем виконуються запити до бази даних PostgreSQL [5] за допомогою object-relational mapping (ORM) бібліотеки SQLAlchemy [6]. Діаграма послідовностей для обробки запиту користувача подана на рис. 3.

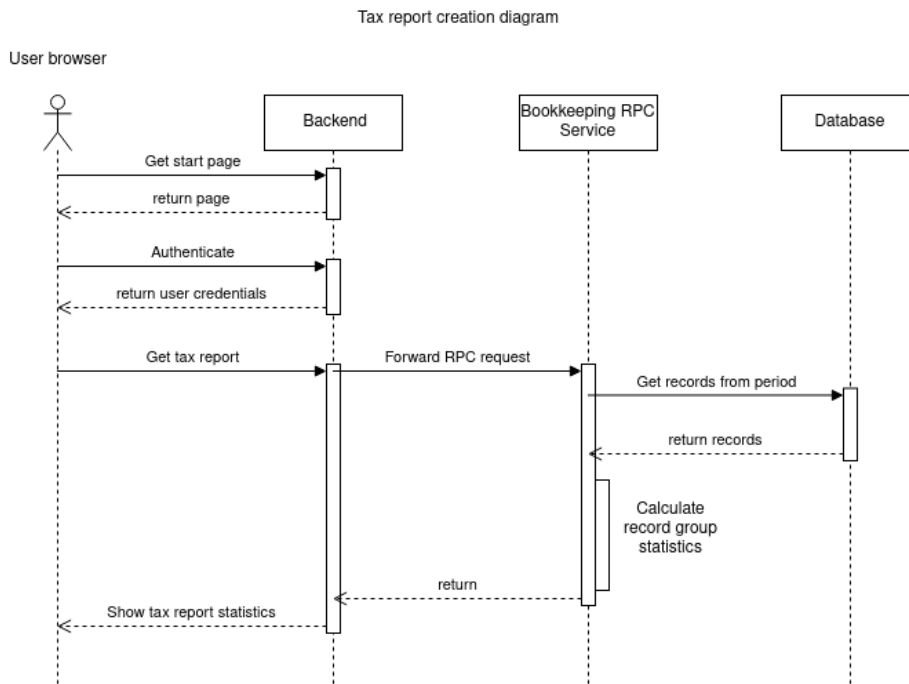


Рисунок 3. Діаграма послідовностей для обробки запиту на отримання користувачем статистики за записами в системі Bookkeeping as a Service

Основний функціонал додатку, пов'язаний зі створенням податкових звітностей з певною формою підприємства за конкретні періоди реалізується за допомогою загального інтерфейсу для всіх класів податкових звітностей. Для більш гнучкого оновлення логіки обробки фінансових записів відповідно до законодавства, реалізації фінансових звітностей подаються окремим модулем python, що може бути оновлений до новішої версії без зміни основної логіки додатку, за рахунок незмінного спільного інтерфейсу. Діаграма взаємозв'язків класів, що реалізують дану архітектуру для подання звітів ФОП 3 групи [7], подано на рис. 4.

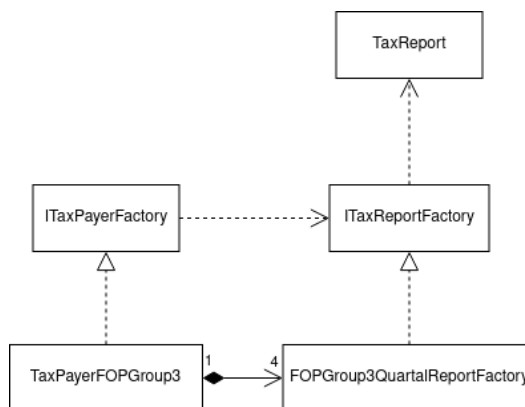


Рисунок 4. Діаграма взаємозв'язків класів, що реалізують створення звітів для ФОП 3 групи

3. ВИСНОВОК

Запропонована реалізація системи бухгалтерського обліку для ФОП як хмарної послуги дозволяє незалежне масштабування окремо модулів графічного інтерфейсу, бухгалтерських обчислень та бази даних. Це дозволяє ефективно використовувати наявні ресурси та адаптуватись до дійсної потреби користувачів під час виконання програми. Архітектура додатку дозволяє оновлювати його функціонал під час його роботи без порушення сесій

користувачів, поступово замінюючи екземпляри модуля бухгалтерських обчислень. Реалізація нових функцій вимагає зміни одночасно всіх модулів, що може мати більший час реалізації, але натомість існує ізоляція функціоналу додатку від певної мови програмування, що дозволяє використання інших типів графічних інтерфейсів (mobile, desktop). Поточний дизайн не передбачає використання модуля бухгалтерських розрахунків окремо від модуля графічного інтерфейсу, оскільки вони мають попередньо налаштоване з'єднання TLS, що захищає від втручання від зовнішніх агентів окрім серверу графічного інтерфейсу. Система може бути розширена за допомогою реалізації автоматичного подання податкової звітності до державної податкової служби України. Також можлива реалізація оплати розрахованої суми податків за допомогою існуючих систем оплати — GooglePay, MonoPay, тощо. Це розширення пропонується реалізовувати як окремі модулі, що масштабуються окремо від основних модулів.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Streamlit • A faster way to build and share data apps. URL: <https://streamlit.io/> (дата звернення: 24.10.2023).
2. Nolan D., Lang D. Authentication for Web Services via OAuth. Springer. 2014. ISBN 978-1-4614-7900-0.
3. gRPC A high performance, open source universal RPC framework. *gRPC*. URL: <https://grpc.io/> (дата звернення: 24.10.2023).
4. Currier C. Protocol buffers. Springer. 2022. Mobile Forensics – The File Format Handbook. https://doi.org/10.1007/978-3-030-98467-0_9.
5. Kroc K., Kizun O., Skublewska-Paszkowska M. Performance analysis of relational databases MySQL, PostgreSQL, MariaDB and H2. Journal of Computer Sciences Institute. 2020. Vol. 14. P. 1–7. URL: <https://doi.org/10.35784/jcsi.1565> (date of access: 29.11.2023).
6. SQLAlchemy - the database toolkit for python. URL: <https://www.sqlalchemy.org/> (date of access: 24.10.2023).
7. Податковий Кодекс України : Закон України від 02.12.2010 р. № 2755-VI : станом на 23 листоп. 2023 р.

ОПТИМІЗАЦІЯ ПЕРЕДАЧІ ДАНИХ У БАГАТОКОРИСТУВАЦЬКИХ ВІДЕОІГРАХ

Сухарєв О.М.¹, Безносик О.Ю.²

Національний технічний університет України «Київський політехнічний інститут
імені Ігоря Сікорського», Київ, Україна

¹ suharev.alexei1@gmail.com, ² beznosyk.oleksandr@lil.kpi.ua [0000-0003-2775-6070]

Дослідження присвячено оптимізації передачі даних у багатокористувацьких відеоіграх. Результати дослідження вказують на те, що передача даних між клієнтом і сервером може бути досить об'ємною і вимагати значної пропускної спроможності, особливо зі збільшенням кількості гравців та об'єктів в ігровому світі. Новизною є більш технологічна оптимізації процесу передачі даних та покращення продуктивності мережевої частини гри.

Ключові слова: клієнт, сервер, багатокористувацький, оптимізація, мережа, онлайн-гра, відеогра.

1. ВСТУП

На сьогоднішній день, онлайн-ігри стали одним з основних прибутків для компаній, які займаються розробкою комп'ютерних ігор. Комерційні ігри повинні підтримувати розраховану на багато користувачів систему, а онлайн-ігрові платформи прагнуть забезпечити більш легке придбання цих продуктів, забезпечуючи підтримку все зростаючої аудиторії. У той же час, нові покоління ігрових консолей тісно пов'язані з привабливістю багатокористувацьких ігор.

Мережеві ігри вимагають як послідовності, так і чутливості мережевої взаємодії. Послідовність важлива для підтримки однакового набору даних для всіх гравців, у той час як зворотній зв'язок вимагає, щоб оновлення даних відбувалися якнайшвидше. Однак, ці дві вимоги часто суперечать одна одній, і вирішення цієї дихотомії між послідовністю та зворотнім зв'язком лежить в основі взаємодії в режимі реального часу [2].

Сучасні ігри, розраховані на багато користувачів, можуть мати десятки тисяч гравців з усього світу – це означає, що масштабованість обраної мережевої архітектури стає вкрай необхідною.

Крім того, онлайн-ігрові платформи прагнуть надати гравцям більш індивідуалізований контент відповідно до їхніх уподобань. Для цього є необхідним використання методів аналізу поведінки гравця, що також важливо для виявлення інших користувачів, які шахраюють або ведуть себе неадекватно в ігровому світі.

2. ЗНАЧЕННЯ ОПТИМІЗАЦІЇ ПЕРЕДАЧІ ДАНИХ

Оптимізація передачі – це комплексна тема, яка складається з таких пунктів як: стиснення даних, ефективні структури даних, асинхронна передача даних, “кешування” даних (збереження даних), пророцтво дій, оптимізація протоколу передачі даних.

Стиснення даних надають можливість зменшити об'єм даних, які передаються між клієнтом і сервером. Найпопулярніші алгоритми стиснення: “Gzip” та “LZ77”.

Ефективні структури даних розробляються для оптимізації продуктивності операцій додавання, видалення, пошуку та оновлення даних залежно від конкретних потреб програми. У контексті розрахованих на мережеві ігри, де існує високий рівень взаємодії та обміну даних

між клієнтами та сервером, ефективні структури даних відіграють критичну роль в оптимізації мережевої взаємодії та забезпеченні чуйності ігрового світу.

Асинхронна передача даних надає можливість більш ефективно використовувати мережеві ресурси.

Кешування даних запобігає повторній передачі даних, які вже були відправлені.

Пророцтво дій – можливість сервера пророкувати дії клієнта. Це може знизити необхідність частоті передачі даних для кожної дії.

Оптимізація протоколу передачі даних є процесом поліпшення ефективності та продуктивності мережевого протоколу, який забезпечує обмін інформацією між клієнтами та сервером у багатокористувацькому середовищі. Це важливий аспект розробки таких продуктів, і його метою є створення більш ефективного, швидкого і надійного механізму передачі даних.

3. ВПРОВАДЖЕННЯ ОПТИМІЗАЦІЇ ПЕРЕДАЧІ ДАНИХ

Оптимізована передача даних у розрахованих на багато користувачів іграх може значно вплинути на різні аспекти ігрового досвіду і продуктивності. Кілька областей, де можна очікувати різницю між оптимізованою та неоптимізованою передачею даних, наведено в Таблиці 1.

Загалом, оптимізована передача даних забезпечує більш стабільний, чуйний і плавний ігровий досвід, що позитивно впливає на задоволення гравців та загальну успішність самого ігрового продукту.

Таблиця 1. Критерії, що потребують оптимізації

Критерій	З оптимізацією	Без оптимізації
Затримки та зворотній зв'язок	Зменшення затримок при передачі даних може призвести до більш чутливого ігрового середовища. Гравці бачитимуть оновлення та реагуватимуть на події ближче до реального часу.	Затримки передачі даних можуть призвести до повільних оновлень, затримок у відображенні дій інших гравців та менш чутливої взаємодії.
Плавність ігрового процесу	Більш ефективна передача даних може забезпечити плавний рух об'єктів, точніше відображення стану ігрового світу та зменшення "лагів"(затримок) [4].	Можливі стрибки, затримки та невідповідності того, що бачать гравці у себе на екранах [4].
Пропускна здатність та надмірна навантаження	Зменшення обсягу даних, що передаються, зниження навантаження на мережу і забезпечення більш ефективного використання пропускної здатності.	Надмірне навантаження на мережу може призвести до втрати пакетів, затримок та зниження продуктивності гри.
Масштабованість	Гра з оптимізованою передачею даних буде легше масштабуватися зі зростанням кількості одночасних гравців.	Неоптимізована передача даних може призвести до проблем із продуктивністю за умови великої кількості гравців.
Ефективне використання ресурсів	Поліпшене використання ресурсів дозволяє грі працювати ефективно на різних пристроях та мережах [5].	Гра може мати проблеми з продуктивністю на слабких пристроях або в умовах обмеженої пропускної спроможності мережі [5].
Відсутність читерства (мухлювання)	Застосування ефективних заходів безпеки та перевірок на сервері дає можливість запобігати мухлюванню [4].	Неоптимізована передача даних може ускладнити реалізацію перевірок та відкрити ширші можливості для нечесної гри [4].

4. РЕЗУЛЬТАТИ ДОСЛІДЖЕНЬ

4.1. Проблема нечесної гри

У будь-якій онлайн грі, шахраї не просто спрощують собі гру, а й погіршують чужий ігровий досвід. Є багато речей, які можна зробити, щоб запобігти нечесній грі. Але найголовніший принцип – дуже простий: не довіряй гравцеві. Завжди чекайте на гірше, а саме те, що гравець намагатиметься вас обдурити.

Авторитарний сервер та наївний клієнт:

Цей принцип веде нас до простого, на перший погляд, рішення – вся ігрова логіка крутиться на головному сервері, під вашим контролем, а клієнт лише демонструє поточний стан сервера та відправляє йому команди (натискання клавіш тощо). Зазвичай, це називають авторитарним сервером, тому що він єдиний, хто має право моделювати світ.

Наприклад:

Клієнт говорить, що я на позиції (10, 10), а через секунду – на позиції (20,10). При цьому можливо він «пройшов» через стіну або просто рухається швидше.

Правильна парадигма наступна: сервер знає, що гравець знаходиться в позиції (10, 10); клієнт каже: «Я хочу посунути на одиницю праворуч». Сервер оновлює позицію гравця на (11, 10), виконуючи всі необхідні перевірки, а потім відповідає гравцеві: «Ви на (11, 10)» (рис. 1).

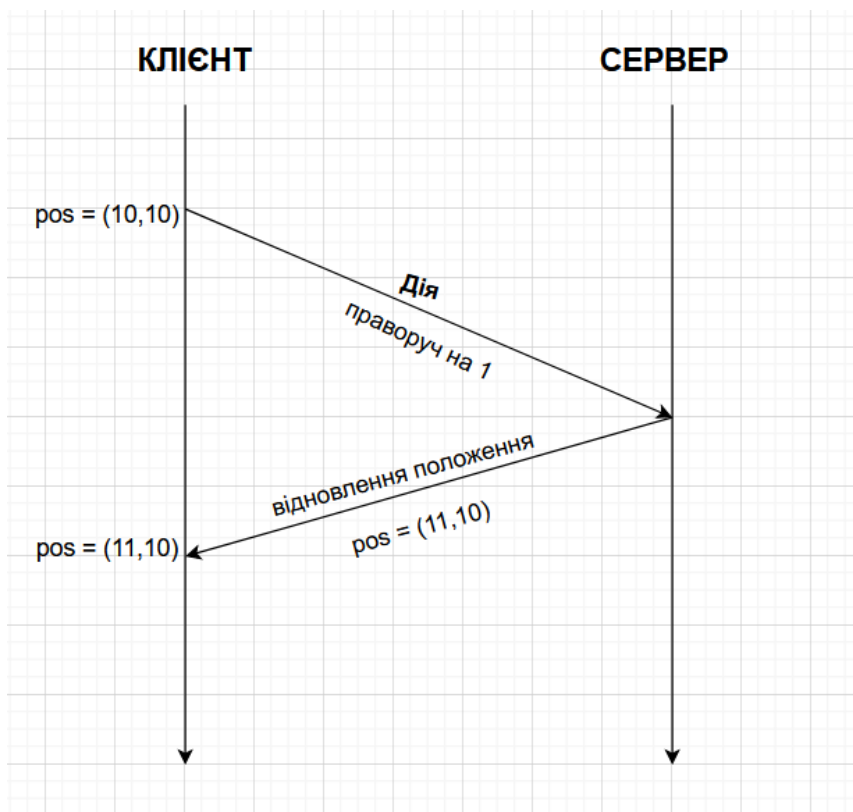


Рисунок 1. Приклад спілкування між клієнтом та сервером

Підсумок: ігровий стан керується лише сервером. Клієнти надсилають свої дії на сервер, а сервер періодично оновлює свій стан і відправляє його на клієнти, які, у свою чергу, відображають його користувачам.

4.2. Затримки

Наприклад, якщо гравець натискає стрілку вліво, персонаж почне рухатися через пів секунди. Це тому, що команда повинна дійти до сервера, а результат команди після цього повинен дійти до клієнта (рис. 2).

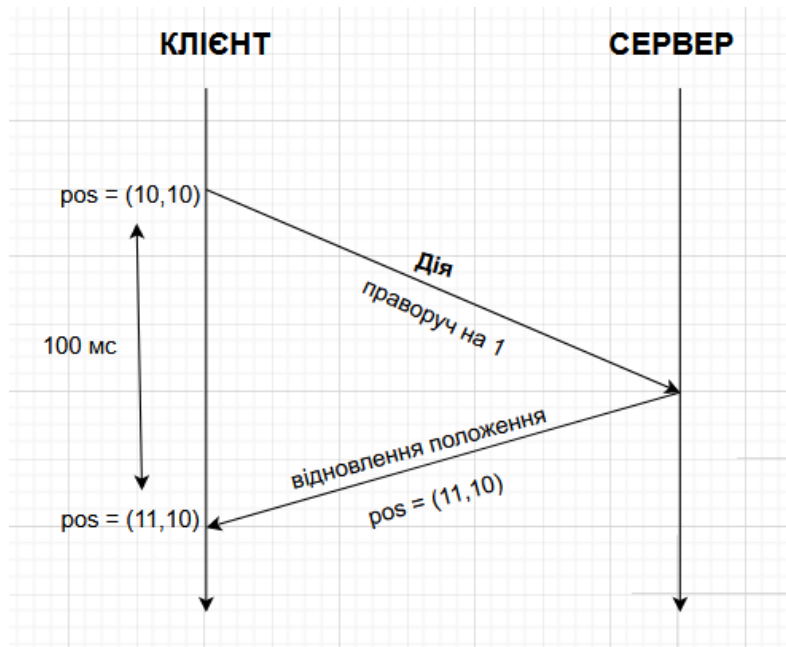


Рисунок 2. Затримка між командою та реакції на неї

В мережі, де затримки можуть становити десятки частки секунди, геймплей (ігровий процес) у кращому разі буде не чутливим.

Незважаючи на те, що деякі гравці намагаються вести нечесну гру, більшу частину часу сервер отримує коректні запити. Це означає, що отримане введення буде коректним і гра оновиться так, як очікується. Тобто, якщо персонаж знаходиться на позиції $(10, 10)$ і відправляє команду на рух праворуч, він опиниться на $(11, 10)$.

Ми можемо використовувати це якщо гра досить детермінована (тобто результат визначений командами та попереднім станом). Припустимо, що у нас лаг (затримка) 100 мс і час переміщення персонажа становить також 100 мс . При використанні наївної реалізації час дії складе 200 мс (рис. 3).

Замість того, щоб відправляти команду і чекати поки прийде новий ігровий стан, щоб отримати його зображення, ми можемо відправити команду і почати формувати зображення результату, якби команда вже була виконана. Тим не менш, треба чекати від сервера результату — «справжнього» стану гри, який здебільшого співпадатиме з локальним станом (рис. 4).

Тепер у нас немає абсолютно ніякої затримки між дією гравця та результатом на екрані, а сервер все ще авторитарний (якщо нечесний клієнт почне відправляти некоректні команди, це ніяк не вплине на стан гри на сервері, який бачать інші гравці).

Таким чином, при використанні авторитарного сервера, ви повинні надати гравцеві ілюзію чутливості, хоча насправді він чекає, поки сервер обробить введення. Для цього клієнт симулює результат усіх команд. Коли приходить оновлення від сервера, стан оновлюється залежно від поточного стану сервера та необроблених команд.

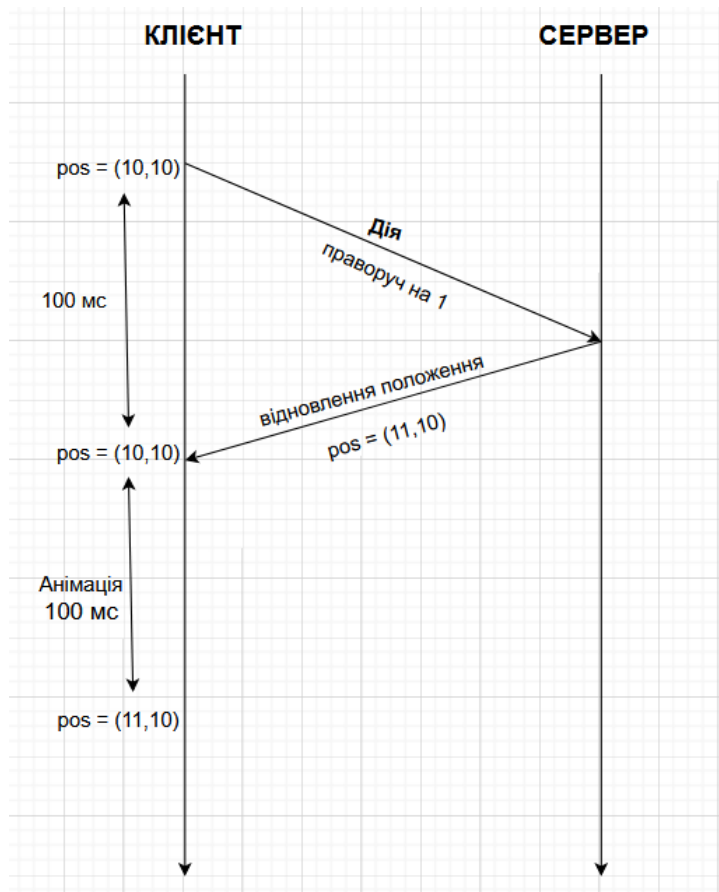


Рисунок 3. Приклад затримки реакції на дію користувача і затримка переміщення персонажа

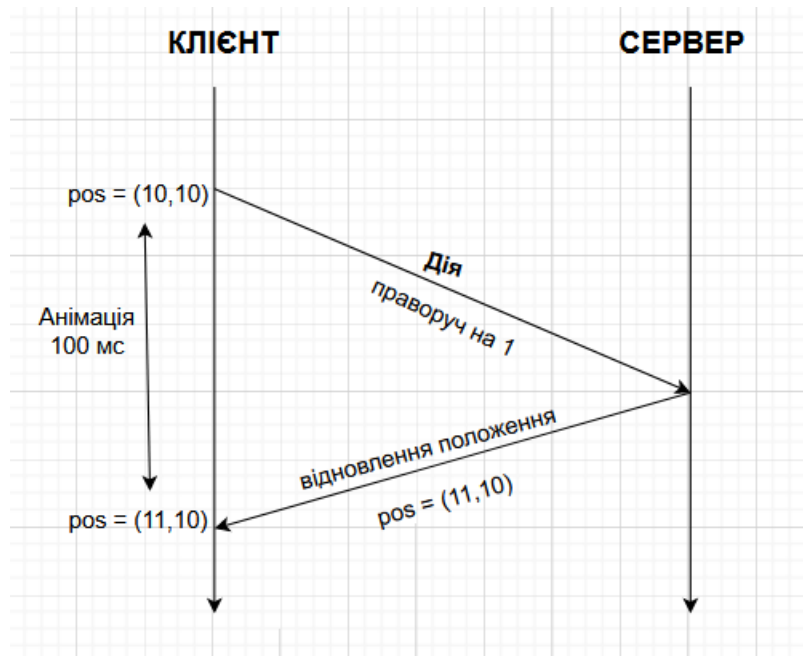


Рисунок 4. Формування зображення, не дочекавшись відповіді з сервера

5. ВИСНОВОК

У світі онлайн-ігор продуктивність ігрових серверів має першорядне значення для забезпечення безперебійного та приємного ігрового процесу. Зосередивши увагу на масштабованості, розміщенні сервера, ефективному мережевому зв'язку, апаратних ресурсах, оптимізації коду, заходах безпеки та моніторингу взаємодії з гравцями, можна створити ігрове середовище, яке буде підтримувати інтерес гравців та не викликати розчарувань.

Оптимізація продуктивності ігрового сервера не тільки покращує ігровий процес, але й сприяє формуванню лояльного та задоволеного співтовариства гравців, забезпечуючи довговічність та успіх онлайн-ігор у висококонкурентній галузі.

Дослідження підтверджує, що в онлайн-іграх, через високі вимоги до продуктивності мережі, навіть невеликі зниження продуктивності є дуже помітними і будуть вважатися недопустимими для кінцевого користувача.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. "Optimizing Game Server Performance for Seamless Gaming" URL: <https://medium.com/@furatamarizuki/optimizing-game-server-performance-for-seamless-gaming-experiences-540bbd8cb0f8>.
2. "Technology Interns in 2020: General Game Tech & Tooling" URL: <https://technology.riotgames.com/news/technology-interns-2020-general-game-tech-toolinginfrastructure>.
3. "How netcode works, and what makes 'good' netcode | PC Gamer." URL: <https://www.pcgamer.com/netcode-explained>.
4. "The DOOM III Network Architecture" URL: https://fabiensanglard.net/doom3_documentation/The-DOOM-III-Network-Architecture.pdf.
5. "Tips to Improve Game Networking for Online Games." URL: <https://www.linkedin.com/advice/0/how-do-you-improve-game-networking-skills-game-development>.

СИСТЕМА РОЗУМНОГО БУДИНКУ З КЕРУВАННЯМ ТЕЛЕГРАМ БОТОМ ТА GOOGLE ASSISTANT

Харченко К.В.¹, Кущовий Д.І.²

Національний технічний університет України «Київський політехнічний інститут ім. Ігоря Сікорського», Київ, Україна

¹ konst1970@gmail.com, ² d.kuschevoy@gmail.com

Цей проект розробляє інноваційну систему розумного будинку, що інтегрується з Телеграм-ботом та Google Assistant для зручного та ефективного керування домашніми пристроями. Система дозволяє користувачам віддалено керувати освітленням, температурою, безпекою та іншими функціями будинку через прості голосові команди або текстові повідомлення. Особливу увагу приділено інтеграції з різними пристроями та платформами, а також забезпеченню безпеки даних та конфіденційності користувачів. Проект включає розробку інтуїтивно зрозумілого інтерфейсу, що робить управління будинком доступним і зрозумілим для широкого кола користувачів.

Ключові слова: розумний будинок, мікроконтролер, мікрокомп'ютер.

1. ВСТУП

Сучасний розумний будинок – це не просто набір окремих розумних пристроїв, а складна інтегрована система, що об'єднує контролери, датчики, різноманітні інтерфейси користувача та мережеві з'єднання. Всі ці елементи разом формують гармонійне середовище, яке реагує на потреби мешканців, адаптується до їхнього способу життя та надає нові можливості для керування домашнім простором [1].

2. ОСОБЛИВОСТІ РОЗУМНОГО БУДИНКУ

Особливості розумного будинку, розробленого з використанням Java framework Spring, що виступає у ролі сервера, та інтегрованого з мікроконтролерами ESP8266, демонструють високий рівень інновацій та технічної ефективності. Використання мікрокомп'ютера Raspberry Pi 4 як основи для Java сервісів ще більше підвищує гнучкість та функціональність системи [2].

2.1. Java Spring як сервер

Spring Framework є одним з найпопулярніших інструментів для розробки Java-додатків, зокрема завдяки його масштабованості та гнучкості. Використання Spring дозволяє легко інтегрувати різноманітні сервіси та компоненти, необхідні для функціонування розумного будинку.

Spring фреймворк забезпечує потужні засоби для забезпечення безпеки та надійності системи, що є важливим фактором для будь-якої системи розумного будинку.

2.2. Мікрокомп'ютер Raspberry Pi 4 для Java сервісів

Компактність та енергоефективність Raspberry Pi 4 забезпечується завдяки компактності, роблячи його ідеальним вибором для використання в якості серця системи розумного будинку.

Незважаючи на свої невеликі розміри, Raspberry Pi 4 забезпечує достатньо обчислювальної потужності для запуску Java сервісів, що вимагаються для управління розумним будинком.

2.3. Інтеграція з мікроконтролерами ESP8266

ESP8266 відомі своєю універсальністю та доступністю, що робить їх ідеальними для розробки різноманітних компонентів розумного будинку, таких як датчики та контролери.

Wi-Fi підключення відбувається завдяки вбудованому Wi-Fi модулю. ESP8266 можуть легко підключатися до мережі, що створюється Raspberry Pi 4 та Spring сервером, забезпечуючи надійну та швидку комунікацію.

3. ІНТЕГРАЦІЯ

Сервер, розроблений на основі Java Spring, відіграє ключову роль у мікросерверній архітектурі системи розумного будинку. Ця архітектура дозволяє інтегрувати нові клієнти та сервіси з високим рівнем гнучкості та ефективності.

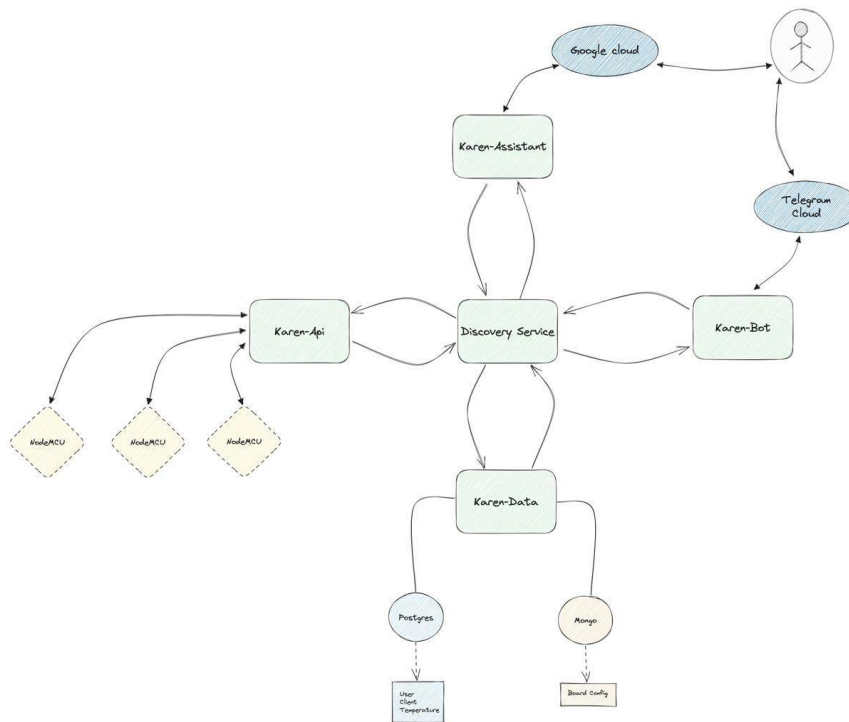


Рисунок 1. Схема архітектури системи

3.1. Інтеграція нового клієнта (мікроконтролера)

Інтеграція нового клієнта полягає у виконанні процесу запити та реєстрації:

- клієнти, такі як мікроконтролери ESP8266, ініціюють інтеграцію, відправляючи HTTP-запити до сервера. Вони передають дані про свої можливості та управлінські функції;
- сервер на Java Spring обробляє ці запити, зберігаючи інформацію про клієнтів у базі даних. Це дозволяє системі ідентифікувати та управляти різними клієнтами ефективно.

Мікросерверна архітектура забезпечує легке підключення нових пристроїв, забезпечуючи високий рівень адаптивності системи без необхідності значних змін у її основній структурі.

3.2. Інтеграція нового сервісу

Процес запиту та отримання даних відбувається таким чином: при додаванні нового сервісу для графічного або голосового управління, цей сервіс ініціює зв'язок, відправляючи HTTP-запит до сервера. У відповідь, сервер надає інформацію про підключені клієнти, що дозволяє новому сервісу отримати доступ до управлінських функцій цих клієнтів.

Далі на основі отриманої інформації про клієнта з бази даних новий сервіс взаємодіє з сервісом-посередником для реалізації управління. Це забезпечує єдиний інтерфейс для різних типів сервісів, спрощуючи процес інтеграції.

4. ІНТЕРФЕЙСИ КОРИСТУВАЧА

4.1. Телеграм-бот

Використання Телеграм-бота для управління системою розумного будинку має ряд значних переваг. По-перше, Телеграм є одним із найпопулярніших месенджерів, який відрізняється високим рівнем безпеки та приватності, що є важливим аспектом при управлінні домашньою автоматизацією. Це означає, що команди та дані, які передаються через Телеграм-бота, захищені від несанкціонованого доступу.

Телеграм-бот також забезпечує високу доступність та зручність у використанні. Він дозволяє користувачам керувати своїм домом з будь-якої точки світу, де є доступ до інтернету, використовуючи звичний інтерфейс месенджера. Це робить управління розумним будинком простим і інтуїтивно зрозумілим, оскільки більшість користувачів уже знайомі з функціоналом Телеграму.

Ще однією перевагою є гнучкість у налаштуванні та кастомізації. Розробники можуть легко програмувати та налаштовувати бота для виконання специфічних завдань управління розумним будинком, від простого включення світла до більш складних дій, таких як моніторинг системи безпеки або регулювання температури [3].

Нарешті, інтеграція Телеграм-бота з системою розумного будинку може включати в себе розширений функціонал, такий як сповіщення про важливі події або зміни в стані системи, що забезпечує користувачам додатковий рівень інформованості та контролю над їхнім домом.

Таким чином, Телеграм-бот як інструмент для управління розумним будинком відкриває широкі можливості для зручності, безпеки та гнучкості, роблячи його ефективним та доступним рішенням для багатьох користувачів.

4.2. Google-assistant

Інтеграція Google Assistant у систему розумного будинку має низку значних переваг, які відкривають нові можливості для користувачів. Однією з основних переваг є зручність голосового управління. Користувачі можуть управляти різними функціями свого будинку, використовуючи прості голосові команди. Це не тільки економить час, але й робить управління більш інтуїтивним, особливо в ситуаціях, коли використання традиційних інтерфейсів незручне або неможливе.

Google Assistant також пропонує високий рівень персоналізації. Він може розпізнавати голоси різних користувачів і адаптувати реакції та дії згідно з їхніми індивідуальними налаштуваннями та звичками. Це створює більш персоналізоване та зручне середовище управління [4].

Крім того, Google Assistant відкриває можливості для розширеного інтелектуального взаємодії з домом. Він може надавати корисні сповіщення, нагадування та інформацію, пов'язану з побутовими завданнями та управлінням дому, наприклад, нагадування про

необхідність вимкнення пристроїв або зміну температури. Така взаємодія робить повсякденне життя більш комфортним та організованим.

Ще однією важливою перевагою є інтеграція Google Assistant з широким спектром пристроїв та сервісів. Це означає, що система розумного будинку може легко працювати в синергії з різноманітними побутовими пристроями та сервісами, що робить процес автоматизації більш гнучким та ефективним.

Отже, використання Google Assistant для управління розумним будинком значно підвищує зручність, персоналізацію та інтелектуальну взаємодію, що робить повсякденне життя більш комфортним та організованим [5].

5. ВИСНОВОК

З урахуванням викладеного вище, інтеграція як Телеграм-бота, так і Google Assistant в систему розумного будинку відкриває вражаючі можливості для покращення комфорту, безпеки та ефективності домашнього життя. Використання Телеграм-бота забезпечує безпечне, доступне та знайоме середовище для керування різними аспектами дому, пропонуючи простоту та гнучкість у використанні. Це, поєднуючи з його високим рівнем безпеки та приватності, робить Телеграм-бота відмінним вибором для інтерактивного управління системою розумного будинку.

З іншого боку, Google Assistant пропонує революційний підхід до управління домом за допомогою голосових команд, забезпечуючи високу ступінь персоналізації та інтуїтивно зрозумілого взаємодії. Він відкриває двері до більш інтелектуального і зв'язаного домашнього простору, де користувачі можуть легко взаємодіяти з своїми пристроями, отримувати сповіщення та управляти повсякденними завданнями з великою легкістю.

У поєднанні, ці технології не тільки підвищують зручність та ефективність управління домом, але й вносять значний вклад у створення адаптивного, реактивного та взаємопов'язаного домашнього середовища. Це відображає розвиток тенденцій у сфері домашньої автоматизації, де інноваційні технології відіграють ключову роль у перетворенні звичайного будинку на справді "розумний" дім.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. J. K. Bradley, "Smart Home Automation with Linux and Raspberry Pi", 2nd ed., Apress, 2013. ISBN 978-1430258872.
2. S. J. Weber, M. J. Wolf, "Internet of Things – IoT Systems: Architectures, Algorithms, Methodologies", Springer, 2018. doi: 10.1007/978-3-319-61300-0.
3. L. D. Paulson, "Building Intelligent Interactive Tutors: Student-centered Strategies for Revolutionizing E-learning", Morgan Kaufmann, 2008. ISBN 978-0123735942.
4. G. N. Preethichandra, "Wireless Sensor Networks for Developing Countries", Springer, 2013. doi: 10.1007/978-3-642-41013-0.
5. B. J. Bumgarner, "Programming Mobile Robots with Aria and Player: A Guide to C++ Object-Oriented Control", Springer, 2009. ISBN 978-1849962711.

ДОСЛІДЖЕННЯ ПРОДУКТИВНОСТІ МІКРОСЕРВІСНИХ АРХІТЕКТУР ЧЕРЕЗ КЕШУВАННЯ ДАНИХ

Хоміч Л.І.¹, Яременко В.С.²

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

¹ lina.homich14@gmail.com, ² yaremenko.vadym@lil.kpi.ua

Дослідження спрямоване на оптимізацію продуктивності мікросервісних архітектур через впровадження ефективних стратегій кешування даних. Зосереджуючись на розгляді стратегій кешування та їх ефективності в умовах розподіленої системи, робота включає проведення аналізу. Під час дослідження визначається ефективність різних стратегій кешування в умовах розподіленої системи. Зазначається важливість вимірювання ефективності стратегій кешування для об'єктивного визначення їхнього впливу на продуктивність мікросервісів та виявлення оптимальних рішень для інформаційних систем.

Ключові слова: стратегії кешування даних, розподілені системи, CDN-мережі, продуктивність, оптимізація

1. ВСТУП

Сучасні технологічні трансформації висувають вимоги до найефективніших архітектур програмного забезпечення для забезпечення швидкості та надійності інформаційних сервісів. У цьому контексті, мікросервісні архітектури виявляються ключовим елементом, надаючи гнучкі та масштабовані рішення для технічних викликів.

Одним із аспектів оптимізації продуктивності мікросервісів є вдосконалення стратегій кешування даних. Кешування не лише прискорює відгук на користувацькі запити, але й ефективно розподіляє навантаження, покращуючи ефективність серверів та економлячи мережеві ресурси.

Враховуючи інтенсивність конкуренції та зростаючу різноманітність інтернет-сервісів, наше дослідження націлене на визначення оптимальних стратегій кешування. Шляхом аналізу та порівняння різних методів кешування, ми сподіваємося виділити науково-обґрунтовані та перспективні стратегії для забезпечення високої продуктивності інформаційних систем.

2. МІКРОСЕРВІСНІ АРХІТЕКТУРИ ТА СТРАТЕГІЇ КЕШУВАННЯ В УМОВАХ СУЧАСНИХ СИСТЕМ

2.1. Концепції розгортання мікросервісних архітектур.

Мікросервісні архітектури є передовою парадигмою розробки програмного забезпечення, де додаток розбивається на невеликі та автономні компоненти – мікросервіси. Розглянемо ключові концепції цього підходу та їхній вплив на продуктивність мікросервісних архітектур, особливо в контексті нашого дослідження.

Декомпозиція функціональності: Ця концепція передбачає розбиття функціональності додатку на невеликі, самостійні мікросервіси. Кожен з них відповідає за виконання конкретного завдання, сприяючи збереженню чистоти та структурованості коду.

Автономність та незалежність: Кожен мікросервіс працює відокремлено, дозволяючи розгортати, масштабувати та оновлювати його незалежно від інших компонентів системи. Це забезпечує гнучкість та ефективність у розробці та експлуатації.

Спрощена комунікація: Взаємодія між мікросервісами здійснюється за допомогою чітко визначених API, спрощуючи комунікаційні процеси. Це робить можливим ефективний обмін даними та реалізацію змін у функціональності без великих зусиль.

Гнучкість та масштабованість: Мікросервіси дозволяють використовувати різні технології для кожного компонента, що сприяє гнучкості розробки. Масштабованість окремих мікросервісів дозволяє ефективно використовувати ресурси в залежності від потреб системи.

Легкість утримання та розгортання: Завдяки своїй компактності та автономності, мікросервіси полегшують утримання системи та спрощують розгортання нових функціональних можливостей. Це особливо важливо у сучасному середовищі, де швидкість впровадження нових функцій має ключове значення.

CDN мережа з мікросервісами відкриває нові можливості для покращення продуктивності та доступності веб-додатків. Використання CDN для розгортання мікросервісів дозволяє оптимізувати розподілення контенту та обробку запитів, забезпечуючи ефективне використання ресурсів та знижуючи латентність для користувачів.

Хостинг мікросервісів у контексті CDN мереж надає новий рівень глобальної доступності та низької латентності для користувачів по всьому світу. Вибір відповідного CDN провайдера має велике значення для забезпечення оптимальної продуктивності та користувацького досвіду. Ефективне кешування та розподілення трафіку в межах CDN додають додаткові переваги управлінню хостингом мікросервісів, забезпечуючи стабільність та ефективність системи.

2.2. Стратегії кешування в умовах мікросервісів.

У випадку мікросервісних архітектур, що характеризується розподіленим характером та компактними розмірами сервісів, використання стратегій кешування даних виявляється ключовим елементом для забезпечення оптимальної продуктивності та швидкодії системи. Цей розділ присвячений аналізу різних стратегій кешування, їхнього впливу на роботу мікросервісних систем, а також визначенню оптимальних підходів для досягнення найкращих результатів у конкретних умовах.

Локальне кешування – одна з ключових стратегій, яку важливо розглянути. В цьому підході кожен мікросервіс зберігає власний кеш, що дозволяє зберігати часто використовувані дані прямо всередині мікросервісів. Такий підхід дозволяє мікросервісам ефективно отримувати доступ до необхідної інформації без постійного звертання до централізованих ресурсів. Кожен мікросервіс контролює свій кеш, сприяючи локалізації та забезпечуючи безпеку даних. Це не лише знижує часу доступу до даних, але й мінімізує навантаження на централізовані ресурси. Проте важливо враховувати, що зберігання окремих кешів у кожному мікросервісі може вести до дублювання інформації та надмірного використання простору. Цю стратегію можна кількісно характеризувати через такі характеристики: простір кешу – обсяг пам'яті для локального кешу визначається для кожного мікросервісу окремо; час доступу до даних - цей параметр визначає, наскільки швидко мікросервіс може отримати необхідні дані зі свого локального кешу. У порівнянні з іншими стратегіями, цей процес є швидшим, оскільки дані зберігаються безпосередньо у базі даних мікросервісу; дублювання даних – кількість дубльованих даних в локальних кешах визначається кількістю мікросервісів, які можуть зберігати однакові чи схожі дані в своїх кешах. Це може бути виміряно, наприклад, у відсотках загального обсягу даних в кешах або у кількості дубльованих записів.

Стратегія централізованого кешування передбачає використання спільного кешу, який накопичує дані для всіх мікросервісів. Цей підхід може бути ефективним у випадках, коли

декілька сервісів спільно використовують однакові дані. Завдяки централізованому кешу відбувається обмін та спільне використання даних між мікросервісами. Однак варто враховувати, що цей підхід може призвести до збільшення кількості запитів та обмеження гнучкості системи. Основними перевагами є спрощене управління, але це супроводжується можливістю зростання обсягу запитів та обмеженням гнучкості в архітектурі. Такий підхід зручний для сценаріїв, де великий обсяг даних має спільне використання, але його слабкі місця можуть стати проблемними в умовах збільшення обсягу запитів та потреби у вищій гнучкості системи. Оцінити дану стратегію можна як і попередню за обсягом спільного кешу та частотою обміну даними.

Розподілене кешування є важливою стратегією для підвищення продуктивності в мікросервісних архітектурах. В цьому підході кожен мікросервіс може мати свій власний кеш, розташований на різних вузлах мережі. Це сприяє швидкому доступу до даних, оскільки мікросервісам не потрібно постійно звертатися до централізованих ресурсів. Переваги розподіленого кешування включають швидкість доступу до даних, спрощене управління кешем для кожного мікросервісу та розподілення навантаження між різними частинами системи. Зокрема, такий підхід дозволяє системі ефективно працювати в умовах великої кількості розподілених компонентів. Однак розподілене кешування може стикається з викликами, такими як проблеми консистентності даних між різними кешами та додаткові складнощі у впровадженні та конфігурації. Незважаючи на це, враховуючи правильність конкретної ситуації, розподілене кешування може бути ефективним інструментом для покращення продуктивності мікросервісних архітектур. Дану стратегію можна характеризувати за кількістю вузлів кешу та швидкості обміну даними, що вимірюється як середня швидкість передачі даних між різними вузлами, і рівнем синхронізації між кешами, що показує наскільки консистентними та уніфікованими є дані між різними вузлами.

Інвалідація кешу та стратегії оновлення є важливим аспектом ефективного кешування в мікросервісних архітектурах. Ці стратегії визначають, коли і які дані слід оновлювати або видаляти з кешу для забезпечення актуальності інформації. Інвалідація кешу передбачає позначення певних даних як застарілих або невірних, що вимагає їх видалення або оновлення. Ця стратегія гарантує, що дані в кеші завжди відповідають актуальному стану системи. Проте необхідно правильно визначити моменти для проведення інвалідації, оскільки часте оновлення може впливати на продуктивність. Стратегії оновлення визначають частоту та способи оновлення даних в кеші. Для мікросервісних систем, де дані можуть змінюватися динамічно, важливо вибрати ефективний механізм оновлення. Враховуючи особливості кожної стратегії, важливо забезпечити збалансований підхід до інвалідації та оновлення кешу для оптимізації продуктивності [2]. Кількісною характеристикою даної стратегії є час інвалідації, а саме інтервал часу через який дані вважаються застарілими та підлягають інвалідації, впливає на актуальність інформації та навантаження системи під час оновлення. А також важливою характеристикою є спосіб оновлення даних в кеші: чи це періодичне оновлення, оновлення за запитом чи оновлення в реальному часі.

Управління життєвим циклом кешу в мікросервісних архітектурах є ключовою складовою для забезпечення оптимальної продуктивності системи. Ця стратегія включає в себе ряд методів, спрямованих на ефективне управління створенням, зберіганням і видаленням кешованих даних. Запобігання переповненню кешу та знищення застарілих даних – один із ключових аспектів управління життєвим циклом кешу. Вибір оптимальних методів зберігання кешу також важливий для управління його життєвим циклом. Це може включати в себе розробку ефективних алгоритмів очищення, які враховують активність та важливість даних. Керування тим, як довго дані залишаються в кеші, відіграє важливу роль у забезпеченні оптимальності використання ресурсів та збереженні потрібної інформації. Остаточо,

правильне управління життєвим циклом кешу визначає його ефективність та впливає на продуктивність мікросервісних архітектур в цілому [3]. Тому характеристикою даної стратегії виступає тривалість зберігання даних, а саме час, протягом якого дані залишаються в кеші до їх видалення або оновлення, впливає на доступність актуальних даних та ефективність кешування.

3. МОДЕЛЬ ТА СИМУЛЯЦІЯ МЕРЕЖІ

3.1. Розробка моделі CDN мережі

У даному розділі ми опишемо процес розробки моделі CDN-мережі, який включає в себе визначення параметрів, створення графової структури, імплементацію унікальних характеристик для кожного вузла, та побудову мережі для детального моделювання взаємодій між CDN провайдерами, серверами та кінцевими користувачами.

Першим кроком є визначення параметрів, які визначатимуть кожен вузол мережі. Це включає в себе:

Швидкість передачі даних: кожен сервер та CDN провайдер матиме свою швидкість передачі даних, що визначатиме ефективність обміну інформацією.

Обсяг пам'яті для кешування: величина, яка визначає, скільки даних може зберігатися в кеші кожного вузла.

Час TTL для збережених даних: час, протягом якого дані залишаються актуальними перед оновленням.

Мережу буде представлено у вигляді графу (Рис. 1), де вузли відобразатимуть сервери та CDN провайдерів, а листки графа представлятимуть кінцевих користувачів. Ребра графу відобразять швидкість передачі даних між вузлами.

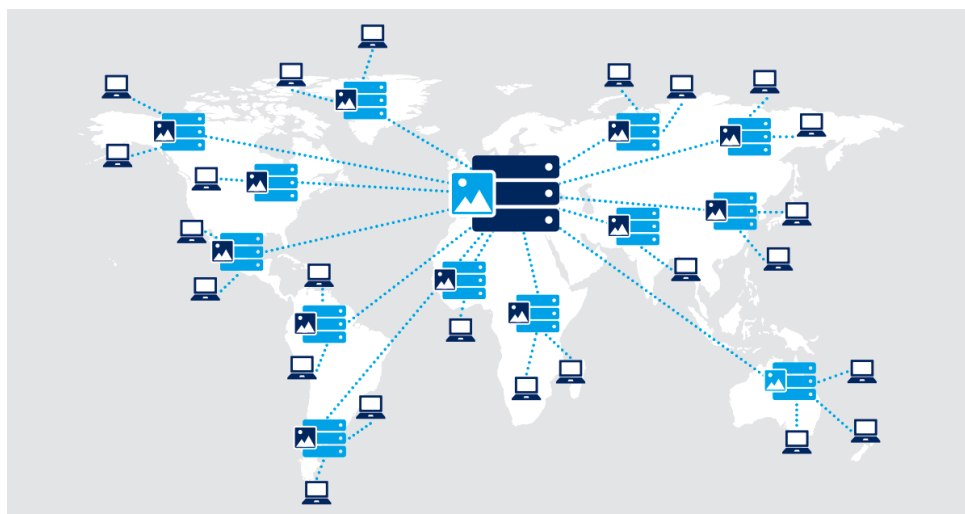


Рисунок 1. Схематичне представлення CDN мережі

Для кожного вузла, чи то сервер, мікросервіс чи CDN провайдер, будуть враховані унікальні характеристики, визначені під час визначення параметрів моделі. Це дозволить точно відтворити реальні умови використання CDN-мережі.

З урахуванням обраних стратегій кешування, реалізовані механізми взаємодії між вузлами мережі. Що включає відправлення запитів між мікросервісами, оновлення даних на головному сервері та поновлення кешованих даних.

Процес розробки моделі CDN-мережі є фундаментом для подальших симуляцій та експериментів з вибором стратегій кешування.

3.2. Вимірювання ефективності стратегій кешування

У цьому розділі ми розглянемо конкретні числові метрики та формули, які використовуються для вимірювання ефективності стратегій кешування в мережі CDN.

Час життя кешу (TTL) кешованих даних є критичним параметром, що визначає період, протягом якого дані можуть зберігатись в кеші перед їхньою інвалідацією. Ми вимірюємо TTL у секундах та позначаємо його як TTL_i , де i – індекс конкретного кешу.

Формула для обчислення часу інвалідації для даних на сервері S_j визначається як:

$$\text{Час інвалідації}_{ij} = \text{Час оновлення}_{S_j} + TTL_i.$$

Простір кешу позначає обсяг пам'яті, виділеної для зберігання кешованих даних. Для кожного мікросервісу S_j цей обсяг позначається як C_j (в байтах).

Час доступу до даних визначає, як швидко мікросервіс може отримати необхідні дані з свого кешу. Вимірюється в часових одиницях (зазвичай мілісекундах) і позначається як $T_{\text{доступу}}$.

Дублювання даних в локальних кешах визначається кількістю мікросервісів, які можуть зберігати однакові чи схожі дані в своїх кешах. Вимірюється відсотками загального обсягу даних в кешах або кількістю дубльованих записів.

$$\text{Дублювання}_{ij} = \frac{\text{Кількість дубльованих даних у кеші } S_j}{\text{Загальний обсяг даних у кеші}} \times 100\%.$$

Саме ці формули використовуються для вимірювання ефективності стратегій кешування.

4. ВИСНОВКИ

У цьому дослідженні ми проаналізували вплив різних стратегій кешування на продуктивність CDN-мереж у мікросервісних архітектурах. На основі нашої моделі CDN-мережі, представленої у вигляді графу, ми визначили та проаналізували стратегії, такі як локальне кешування, централізоване кешування, розподілене кешування, а також стратегії інвалідації та оновлення. Основні параметри нашої моделі включали швидкість передачі даних, обсяг пам'яті для кешування та час життя даних в кеші. Важливим аспектом дослідження стало висвітлення необхідності розроблення системи для моделювання середовища щоб вимірювати ефективності стратегій кешування. В даному симуляційному середовищі необхідно передбачити можливість задання критеріїв швидкість передачі даних, обсяг пам'яті для кешування та час життя даних в кеші, щоб отримувати оптимальну стратегію для кешування в умовах мікросервісів.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Pacheco V. F. Microservice Patterns and Best Practices / Vinicius Feitosa Pacheco. – 254 с
2. Barker T. Intelligent Caching / O'Reilly Media, Inc.- 107 с
3. Rabinovich, M., Spatscheck, O.: Web Caching and Replication. Addison Wesley - 52 с.

ГЕОЛОКАЦІЯ ПОЗИЦІЙ ОБ'ЄКТІВ РОБОЧОЇ КАРТИ ЗА ДАНИМИ ЗОБРАЖЕНЬ БПЛА

Хом'як К.В.¹, Петренко А.І.²

Національний технічний університет України «Київський політехнічний інститут
ім. Ігоря Сікорського», Київ, Україна

¹ karyna.khomiak@gmail.com, ² tolja.petrenko@gmail.com

Геолокація об'єктів на земній поверхні відіграє важливу роль у багатьох сферах, починаючи від аерофотограмметрії та геодезії й закінчуючи військовим застосуванням та навігацією. У сучасному світі безпілотні літальні апарати (БПЛА) стали одними з найбільш перспективних засобів для отримання геопросторової інформації через їхню здатність збирати великі обсяги даних відображення з висоти. Ці дані відкривають безліч можливостей, зокрема визначення місця розташування об'єктів на карті. Саме в цьому полягає необхідність в дослідженні геолокації позицій об'єктів, які б дозволили ефективно планувати і приймати подальші дії, виходячи з інформації просторового розташування важливих об'єктів, зібраних БПЛА. У цій статті підкреслюється важливість використання дронів та розглядається розробка архітектури системи передачі та аналізу даних, отриманих з БПЛА, для використання у військових цілях.

Ключові слова: БПЛА, дрони, безпілотники, геолокація, застосунок, військова галузь.

1. ВСТУП

Геолокація – це корисний інструмент для визначення географічного положення об'єкта чи системи за допомогою конкретних координат планети. Пристрої GPS використовуються в різних морських функціях, картографуванні, програмах для громадського використання. Також широко використовуються в робототехніці для перевірки, виявлення, спостереження, локалізації та завдань для автономних навігаційних систем. Безпілотні літальні апарати (БПЛА) є найпопулярнішими транспортними засобами для розробки завдань навігації за допомогою пристроїв GPS шляхом отримання місцезнаходження БПЛА за допомогою супутникової інформації під час навігації безпілотника [1].

БПЛА, також відомі як дрони, отримали важливе значення в різних дисциплінах військової та цивільної служби завдяки своїй підвищеній стабільності та витривалості в кількох операціях. Застосування БПЛА надзвичайно розширюється завдяки їх прогресивному використанню в Інтернеті речей (IoT), 5G. За останнє десятиліття БПЛА використовувалися в різних сферах застосування, включаючи виявлення та відстеження об'єктів, громадську безпеку, спостереження за дорожнім рухом, військові дії, дослідження прихованих або небезпечних зон, навігацію в приміщенні або на відкритому повітрі, зондування атмосфери, операції після катастроф, охорону здоров'я, обмін даними, управління інфраструктурою, управління надзвичайними ситуаціями та кризовими ситуаціями, вантажні перевезення, моніторинг лісових пожеж та логістика [2].

Безпілотні літальні апарати можуть виконувати повітряні операції, які пілотовані літальні апарати вважають складними. Використання безпілотних літальних апаратів

забезпечує значну економію та користь для навколишнього середовища, водночас зменшує ризики для життя людини. БПЛА здатний літати та залишатися в повітрі, не вимагаючи жодного оператора на борту, забезпечуючи економніші операції, ніж еквівалентні пілотовані системи, і виконує економічно ефективну критичну місію без ризику для життя людини. Ним можна дистанційно керувати, за допомогою чого команди управління надходять із наземної базової станції (БС) через пульт дистанційного керування. Вони також здатні виконувати контрольні операції на борту за допомогою автопілота та різних датчиків, включаючи глобальну систему позиціонування (GPS) та інерціальні вимірювальні пристрої (IMU) [3].

Зокрема, БПЛА привернули значну увагу провідних інтернет-магазинів, включаючи Walmart, DHL, Google і Amazon [4, 5].

Індустрія безпілотників привертає значну увагу як модель конвергенції виробництва, обслуговування та доставки, запроваджуючи синергію з співіснуванням різних нових сфер. БПЛА пропонують неявні особливості, такі як збільшений час у повітрі та можливості корисного навантаження, швидку мобільність та доступ до віддалених районів і зон стихійного лиха [3].

Особливий акцент в дослідженні робиться на військовій галузі, адже саме в цій галузі дрони відіграють критичну роль у зборі розвідувальної та стратегічної інформації. Вони забезпечують можливість нагляду, розвідки, слідкування за діями противника та збору даних без високого ризику для життя військовослужбовців. Дрони стають важливим інструментом для військових операцій, сприяючи покращенню стратегічного аналізу та виконанню завдань у складних умовах. Використання дронів у військовій галузі забезпечує доступ до важливої інформації, підвищує продуктивність та допомагає зменшити ризики для людей, забезпечуючи нові можливості для ефективного використання даних.

У цій статті розглядається архітектура комплексної програми, спрямованої на складання робочої карти для військових за допомогою даних, отриманих від БПЛА. Така програма допоможе військовому командуванню приймати оптимальні рішення щодо ходу бойових дій на основі даних, які вони переглядатимуть зі свого робочого місця.

2. АРХІТЕКТУРА КОМПЛЕКСНОЇ ПРОГРАМИ

2.1. Вимоги до програми

Ціль дослідження – розробити програму, яку одразу можна буде передати для використання військовими та тестуванням її у дії.

Волонтерські ІТ-центри допомоги військовим зазначили рекомендації, яких потрібно дотримуватися, щоб досягти максимально користі та практичності використання програми:

- програма має бути представлена у вигляді системи, розділеної на застосунки, відповідно до використання її звичайними військовими та вищим командуванням;
- згідно практичності застосунок для звичайних військових має бути представленим у вигляді мобільного додатку, яким вони будуть передавати дані;
- згідно практичності застосунку для вищого командування – це має бути комп'ютерний застосунок, з якого зручно переглядати інформацію та складати подальші прогнози;
- військові, що передають дані, мають змогу писати власні коментарі до зображень.

Також до вимог системи додається:

- зручний інтерфейс взаємодії користувача із застосунками;
- відсутність зайвих функцій та максимальна простота;
- можливість легкого масштабування системи для додавання необхідних функцій, коли такі виникнуть у процесі користування системою.

2.2. Узагальнений опис архітектури системи

Згідно вимог розроблено архітектуру системи, що включає в себе три основні частини та забезпечує їх взаємодію у використанні різними користувачами. Ці частини спільно працюватимуть для забезпечення ефективності та точності процесу нанесення геолокації на робочу карту.

Перша складова системи – мобільний застосунок. Ця частина призначена для збору та передачі фотографій з геомітками, отриманих від БПЛА, для подальшої обробки. Військовий має змогу зі свого кабінету завантажувати фотографії та вказувати додаткову інформацію.

Друга частина системи – сервісна частина, яка буде відповідальною за обробку та аналіз даних, що надходять від мобільного застосунку.

Третя частина системи – комп'ютерний застосунок, що спрямований на аналіз інформації та фотографій вищим командуванням. Цей інструмент надасть можливість для аналізу та висновків на основі отриманих даних, спрощуючи процес вирішення завдань та прийняття рішень. Вище командування матиме змогу переглядати карту України, зокрема її частину зони бойових дій – так звану робочу карту, на якій буде нанесено усі геомітки з інформацією. Відповідно користувач зможе за заданими координатами переглядати усі наявні фотографії у певному радіусі або через галерею відповідно в порядку останніх завантажень.

Ця комплексна система має на меті створення ефективного та надійного інструментарію для аналізу геолокації об'єктів на основі зображень, забезпечуючи можливість швидкого прийняття рішень на основі аналізу цих отриманих даних.

2.3. Опис мобільного застосунку

Мобільний застосунок, що призначений для використання звичайними військовими і передачі дані на сервер, має інтерфейс, показаний на рис. 1:

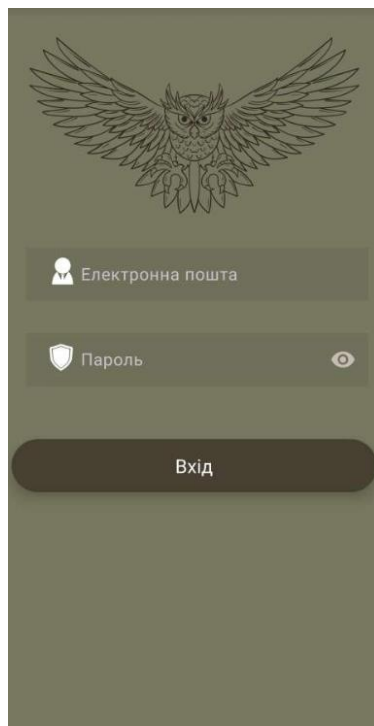


Рисунок 1. Інтерфейс входу до застосунку

Після того, як користувач увійшов в застосунок, він матиме змогу надати застосунку дозвіл доступ до галереї смартфона та завантажити необхідні фото з даними.

На рис.2 приведено вигляд вкладки із щойно завантаженим фото, де одразу видно основні дані, такі як саме фото, геолокація та коментар. Також є можливість додати фото із самої камери смартфона та задіяти кнопку відправки фотознімка на сервер.

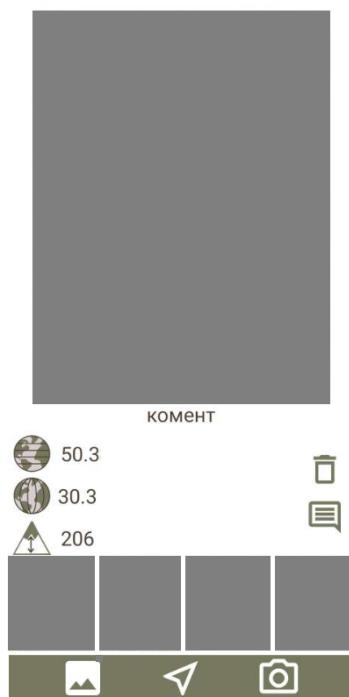


Рисунок 2. Відображення сторінки із завантаженим фото

3. Додати коментар до фото користувач зможе також через текстове поле, показане на рис.

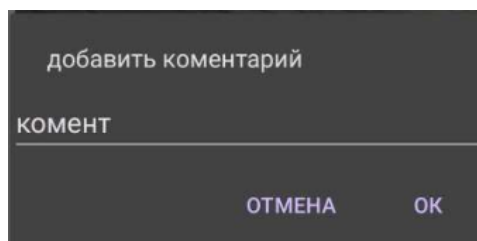


Рисунок 3. Вкладка «Додати коментар»

2.4. Опис сервісної частини

На сервері маємо базу даних, що зберігає такі дані:

- довгота;
- широта;
- висота;
- коментар;
- час знімку;
- час завантаження фото;
- самий фотознімок.

2.5. Опис комп'ютерного застосування

Комп'ютерний застосунок відкриватиметься на головній вкладці «Мапа» та відобразить карту України, яку буде змога масштабувати відносно будь-якої точки чи зони, що цікавить користувача (рис. 4). При ставленні маркера відбуватиметься пошук фотографій, що знаходяться найближче до зазначеного маркера та відобразяться праворуч від мапи разом з коментарями, тобто будуть відображені останні завантажені знімки для цієї місцевості.

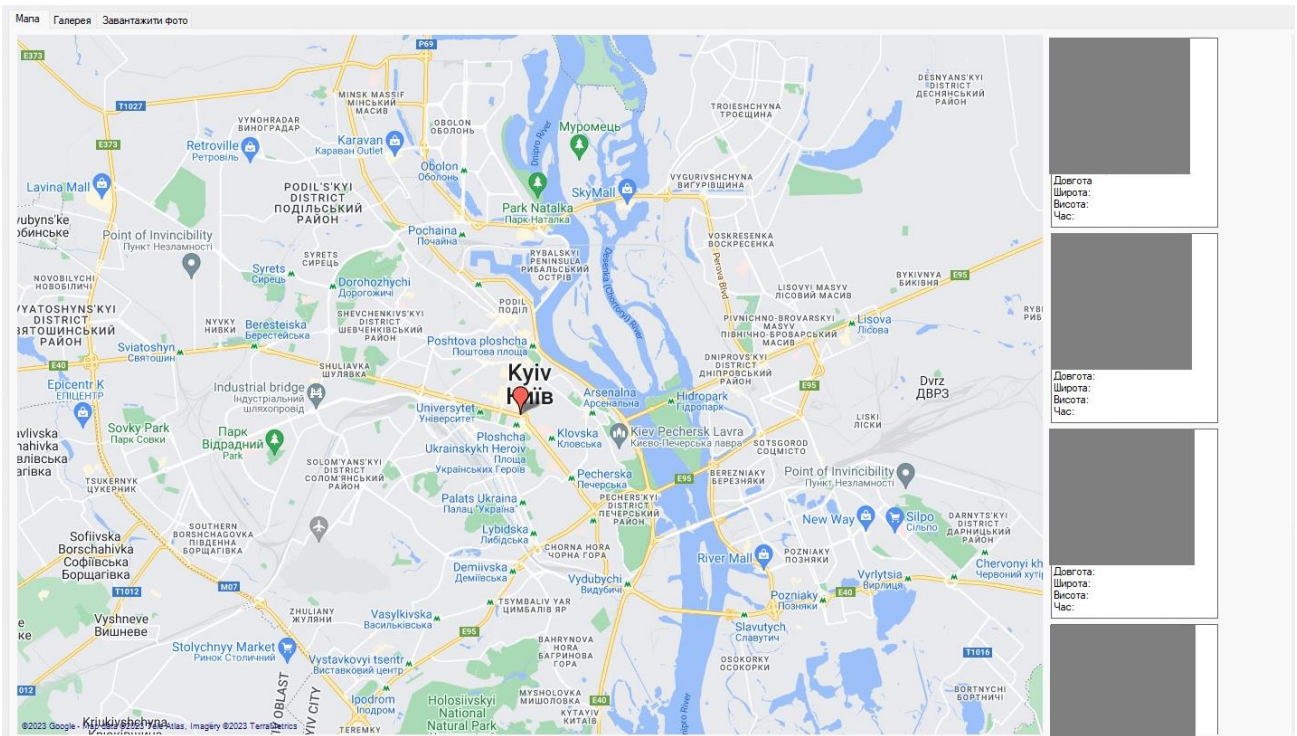


Рисунок 4. Схематична мапа України з міткою

У вкладці «Завантажити фото» маємо дві кнопки «Відкрити», що обирає фото на комп'ютері, та «Відправити», яка завантажує фото до системи. Також можемо перетягнути фото в виділену область для того, щоб завантажити файли.

У завантажених фото, що бачимо на вкладці до того, як їх буде відправлено до системи, маємо змогу переглядати важливі дані, а саме (рис. 5):

- фото;
- довготу;
- широту;
- висоту;
- час (із датою).

Маємо змогу видалити зайві фото, до того як їх буде відправлено.

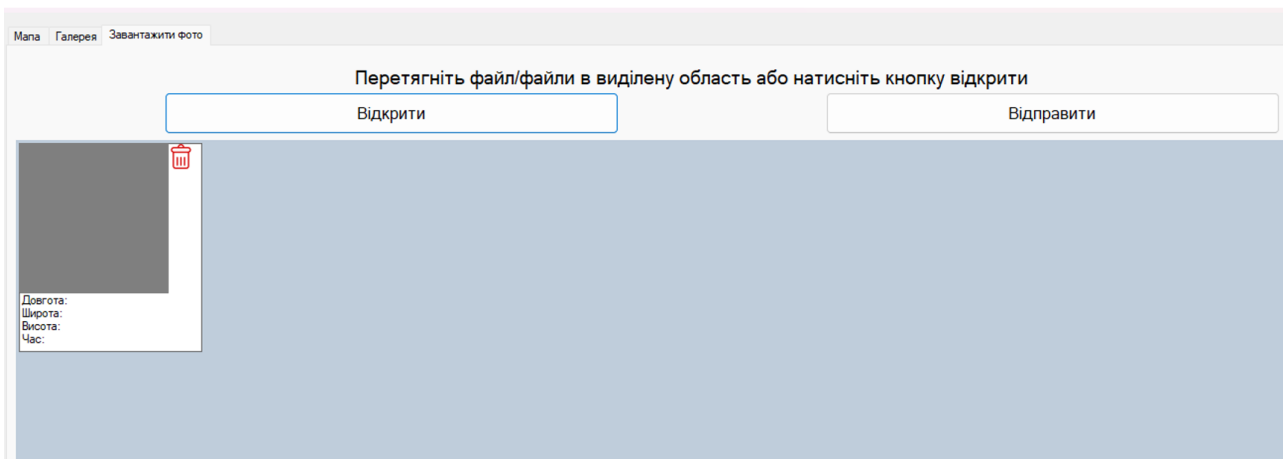


Рисунок 5. Вигляд вікна завантаження фото

У вкладці «Галерея» (рис. 6) будуть висвітлені останні за датою завантаження фото з даними та коментарями, що знаходяться знизу під фото відповідно до певних значень широти і довготи, а також діапазону часу, протягом якого буде йти пошук.

У даній вкладці реалізується також функція корекції по координаті.

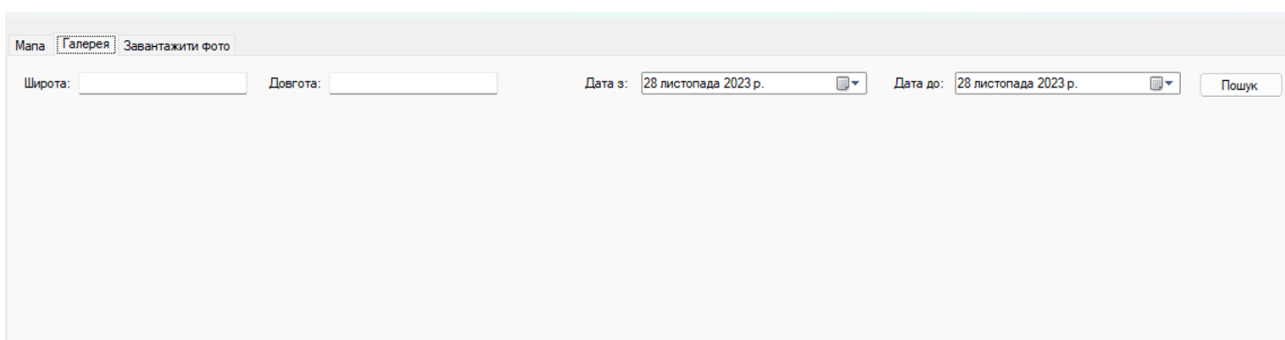


Рисунок 6. Вигляд вкладки «Галерея»

3. ВИСНОВКИ

Розглянута архітектура системи, яка складається з мобільного застосунку для військових, серверної частини та комп'ютерного застосунку для вищого командування, що є важливим кроком у розвитку засобів практичного використання геолокаційних даних.

Перш за все, ця система демонструє інтеграцію технологій у зборі, обробці та аналізі геоданих. Вона дозволяє військовим здійснювати збір та передачу фотографій з геомітками за допомогою мобільного застосунку, що відображає її практичність та зручність у використанні.

Крім того, серверна частина відповідає за обробку та аналіз отриманих даних, що вказує на можливість виконання складних операцій з геоданими у реальному часі.

Найбільш важливою складовою цієї архітектури є додаток для вищого командування, який надає можливість аналізувати робочу карту з геомітками. Це створює можливість для прийняття обґрунтованих рішень на основі даних та сприяє ефективному керуванню в умовах бойових дій.

У цілому, враховуючи можливості кожної складової системи, можна зробити висновок про їхню сумісність та спрямованість на створення ефективного та надійного інструментарію для аналізу геолокаційних даних із фото, сприяючи прийняттю оперативних та обґрунтованих рішень в умовах нестабільності та напруженості ситуацій.

У контексті дослідження геолокації позицій об'єктів за даними зображень БПЛА, розглянута архітектура системи є ключовою для забезпечення ефективного збору, передачі та аналізу цих важливих даних. В статті відзначено важливість безпілотних літальних апаратів як джерела великих обсягів геопросторової інформації, що відкриває широкі можливості для визначення геолокації об'єктів для потреб військової сфери.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. “Geo-localisation of aerial images captured with drones” Awarded to Aldrich A. Cabrera-Ponce, MSc student supervised by Dr. Jose Martinez-Carranza, 2019.
2. Hassija, V.; Saxena, V.; Chamola, V. Scheduling drone charging for multi-drone network based on consensus time-stamp and game theory. *Comput. Commun.* 2019, 149, 51–61. URL: <https://www.sciencedirect.com/science/article/abs/pii/S014036641930948X?via%3Dihub> (дата звернення: 26.10.2023).
3. Mohsan, S.A.H.; Khan, M.A.; Noor, F.; Ullah, I.; Alsharif, M.H. Towards the Unmanned Aerial Vehicles (UAVs): A Comprehensive Review. *Drones* 2022, 6, 147. URL: <https://www.mdpi.com/2504-446X/6/6/147> (дата звернення: 30.10.2023).
4. Yoo, W.; Yu, E.; Jung, J. Drone delivery: Factors affecting the public’s attitude and intention to adopt. *Telemat. Inform.* 2018, 35, 1687–1700. URL: <https://www.sciencedirect.com/science/article/abs/pii/S0736585318300388?via%3Dihub> (дата звернення: 15.11.2023).
5. Macrina, G.; Pugliese, L.D.P.; Guerriero, F.; Laporte, G. Drone-aided routing: A literature review. *Transp. Res. Part C Emerg. Technol.* 2020, 120, 102762. URL: <https://www.sciencedirect.com/science/article/abs/pii/S0968090X20306744?via%3Dihub> (дата звернення: 15.11.2023).

САМОНАВЧАЛЬНА СИСТЕМА РОЗПІЗНАВАННЯ ЛЮДСЬКОЇ ДІЯЛЬНОСТІ З ВИКОРИСТАННЯМ НАВЧАННЯ З ПІДКРІПЛЕННЯМ

Цибін М.Д.¹, Кислий Р.В.²

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

¹ accessmaksym@gmail.com, ² kvrware@gmail.com

У даній роботі метою було дослідити та покращити адаптивність систем розпізнавання людської діяльності (HAR) під конкретного користувача, враховуючи його особливості рухів та зміни рухової поведінки з часом за допомогою методів Reinforcement Learning. Дослідження проводилося з використанням базових моделей LSTM, LGBM у симбіозі з навчанням з підкріпленням методами Монте Карло. Новизна і практична значимість визначається змогою системи автоматично коригувати моделі залежно від змін поведінки користувача, за допомогою навчання з підкріпленням.

Ключові слова: розпізнавання людської діяльності, навчання з підкріпленням, машинне навчання, штучний інтелект

1. ВСТУП

Системи розпізнавання людської діяльності можуть використовуватися для багатьох цілей, зокрема сферами використання можуть бути: спорт, медицина, контроль робочих процесів на виробництві, визначення розподілу часу у щоденній рутині, і якщо подумати, то досить багато аспектів застосування можна визначити для таких систем. У дослідженні акцентувалося саме на визначенні спортивних та щоденних активностей, таких як біг, ходьба, підйом сходами вгору та вниз, стояння та лежання. Початковий набір даних складався з даних датчиків по 3 осям акселерометра і гіроскопа на мобільному пристрої. На виході було за мету зробити систему з використання навчання з підкріпленням, яка б мала змогу автоматично адаптуватися до конкретного користувача та під зміни його рухової поведінки з часом.

2. БАЗОВІ МОДЕЛІ (БЕНЧМАРК)

Як базові моделі було обрано LSTM та LGBM. Ці моделі добре підходять для розпізнавання людської діяльності, що є задачею класифікації. Моделі було протестовано на початковому наборі даних, отримана точність прогнозу 0.9455 для LSTM та 0.9845 для LGBM. Це гарний результат, що повністю задовольняє нашу ціль.

LSTM

LSTM (Long Short-Term Memory) — це тип рекурентних нейронних мереж, який добре підходить для моделювання послідовностей та часових рядів. У дослідженні з розпізнавання людської діяльності LSTM має купу переваг, адже мова йде про аналіз часових залежностей у даних.

LSTM є вдосконаленням типової рекурентної нейронної мережі (RNN), яка має здатність зберігати та використовувати інформацію на тривалий термін, що робить його корисним для аналізу довгих послідовностей, де важлива контекстна залежність.

Основна ідея полягає в тому, що LSTM має механізми видалення та додавання інформації в пам'ять на різних часових шарах. Він складається зі спеціальних "ворот", таких як ворота забування (forget gate), ворота входу (input gate), вихідні ворота (output gate). Ці ворота дозволяють регулювати потік інформації в внутрішнім станом моделі.

Форгет-ворота визначають, яка інформація буде забута чи збережена у пам'яті моделі. Ворота входу визначають, яка нова інформація буде додана до пам'яті. Вихідні ворота визначають, яка інформація буде використана для прогнозування наступного значення.

Ця архітектура дозволяє LSTM ефективно працювати з послідовностями різної довжини, уникати проблеми зникаючого/вибухаючого градієнта, яка властива звичайним рекурентним мережам.

LGBM

Light GBM (Light Gradient Boosting Machine) – це потужний алгоритм машинного навчання, який належить до сімейства ансамблевих методів, зокрема, до методу градієнтного підсилення (Gradient Boosting). Він став досить популярним завдяки своїй ефективності та швидкодії, особливо при обробці великих обсягів даних.

Light GBM використовує техніку підсилення градієнта, де модель навчається ітеративно, додаючи нові базові моделі (зазвичай дерева рішень) для виправлення помилок попередніх моделей. Основна ідея полягає в тому, щоб кожна нова модель удосконалювала попередні, фокусуючись на невирішених підсумкових помилках.

Однією з основних переваг Light GBM є його швидкодія. Він використовує градієнтний спуск для знаходження оптимальних значень параметрів моделі, а також оптимізує процес побудови дерев, використовуючи методи, які дозволяють ефективно працювати з великими обсягами даних.

3. НАВЧАННЯ З ПІДКРІПЛЕННЯМ

МОНТЕ КАРЛО

Monte Carlo Reinforcement Learning – це підхід до машинного навчання, який використовує ідеї з області теорії імітаційного моделювання та навчання з підкріпленням для прийняття оптимальних рішень в умовах невизначеності.

Основна ідея методу полягає в тому, що агент навчається шляхом випадкового (монте-карло) взяття вибірки з можливих дій у певному середовищі і спостереження результату цих дій. Агент збирає досвід про те, які дії призводять до найкращих результатів в певних ситуаціях.

Важливо враховувати, що метод монте-карло не завжди є ефективним в умовах великої кількості можливих станів або дій, оскільки вимагає значної кількості випробувань для знаходження оптимального рішення. Однак він може бути корисним для моделювання навчання людини або агента в умовах невизначеності та відсутності чітких правил.

Initialize, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$:

- $Q(s, a) \leftarrow$ arbitrary
- $Returns(s, a) \leftarrow$ empty list
- $\pi \leftarrow$ an arbitrary ϵ -soft policy

Repeat forever:

- (a) Generate an episode using π
- (b) For each pair s, a appearing in the episode:
 - $R \leftarrow$ return following the first occurrence of s, a
 - Append R to $Returns(s, a)$
 - $Q(s, a) \leftarrow$ average($Returns(s, a)$)
- (c) For each s in the episode:
 - $a^* \leftarrow \arg \max_a Q(s, a)$
 - For all $a \in \mathcal{A}(s)$:
 - $\pi(s, a) \leftarrow \begin{cases} 1 - \epsilon + \epsilon/|\mathcal{A}(s)| & \text{if } a = a^* \\ \epsilon/|\mathcal{A}(s)| & \text{if } a \neq a^* \end{cases}$

Рисунок 1. Псевдокод методу Монте Карло

4. СИМБІОЗ МОДЕЛЕЙ

У розробленій системі (Рис. 2) моделі LSTM і навчання з підкріпленням методом Монте Карло поєднуються для досягнення більшої ефективності. Модель метода Монте Карло на основі тренувальних даних генерує нові дані схожі на попередні але вони мають більш виразні ознаки активностей. Далі ці дані приймає на вхід наша базова модель, до прикладу LSTM. модель тренується і готова приймати подальші активності на вхід для визначення прогнозу стану. З часом коли типова поведінка користувача змінюється і модель деградує, за критеріями точності прогнозу, модель перетреноується автоматично на нових даних адже система зберігає дані попередніх активностей для майбутнього покращення моделі. Такий симбіоз у системі дозволяє використовувати плюси обох моделей і робити процес прогнозування для користувача більш точним та адаптивним.

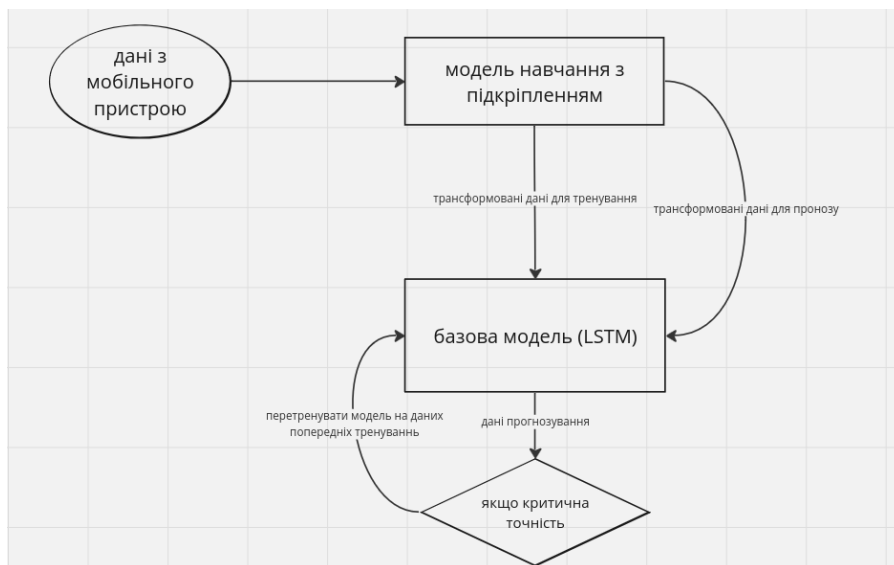


Рисунок 2. Схема роботи системи

5. РЕЗУЛЬТАТИ ДОСЛІДЖЕНЬ

Були виконані дослідження з тренування моделей на датчиках з мобільного пристрою, використовувалися дані по 3 осям гіроскопа та акселерометра. Наведемо дані з тренування та тестування нашої системи з моделей LSTM та навчання з підкріпленням метода Монте Карло.

Модель це наша LSTM, що є вдосконаленням RNN моделі. Вона складається з 5 шарів та має 79302 параметри що тренуються (Рис. 3).

Layer (type)	Output Shape	Param #
input_17 (InputLayer)	[(None, 128, 9)]	0
lstm_16 (LSTM)	(None, 128, 128)	70656
dense_32 (Dense)	(None, 128, 64)	8256
global_max_pooling1d_16 (GlobalMaxPooling1D)	(None, 64)	0
dense_33 (Dense)	(None, 6)	390
...		
Trainable params: 79302 (309.77 KB)		
Non-trainable params: 0 (0.00 Byte)		

Рисунок 3. Характеристика LSTM моделі

Наведемо Loss графік (Рис. 4), який відображає зміну значення функції втрати (loss function) під час процесу навчання моделі машинного навчання. Функція втрати визначає, наскільки добре модель працює на даному наборі даних під час навчання, порівняно зі справжніми (очікуваними) значеннями.

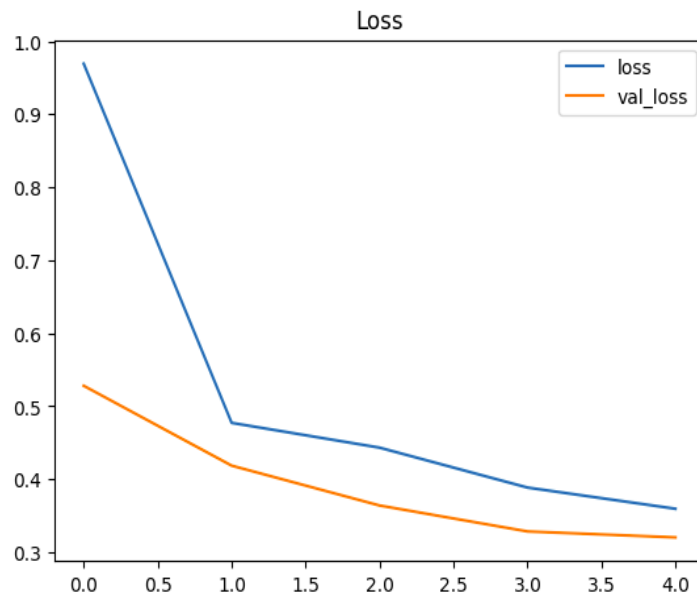


Рисунок 4. Графік зміни функції втрати при тренуванні моделі

Також наведемо графік точності (ассигасу), який відображає зміну точності моделі під час процесу навчання (Рис. 5). Точність – це метрика, яка визначає, наскільки точно модель класифікує дані. Вона вимірюється у відсотках та показує, який відсоток прикладів модель класифікувала правильно відносно загальної кількості прикладів.

Під час кожної ітерації або епохи навчання моделі використовуються навчальні дані для оновлення параметрів моделі. Після цього використовується окремий набір валідаційних або тестових даних, для оцінки точності моделі на даних, які вона не бачила під час навчання.

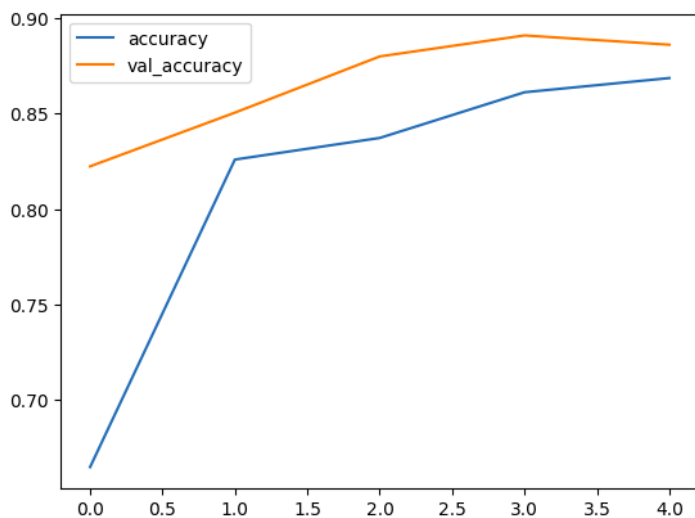


Рисунок 5. Графік росту точності при тренуванні моделі

Також ми побудували матрицю невідповідностей (Confusion Matrix) – це таблиця, яка дозволяє візуалізувати результати класифікації моделі на тестовому наборі даних, де відображається кількість правильних і неправильних прогнозів для кожного класу (Рис. 6).

Кожен рядок у матриці невідповідностей відповідає фактичному класу, а кожний стовпчик – передбаченому класу. Основна діагональ цієї матриці відображає правильно класифіковані приклади (коли фактичний клас співпадає з передбаченим). Елементи поза діагоналлю представляють помилки у класифікації: наприклад, якщо вибіркового класифікатора помилково визначив біг як ходьбу.

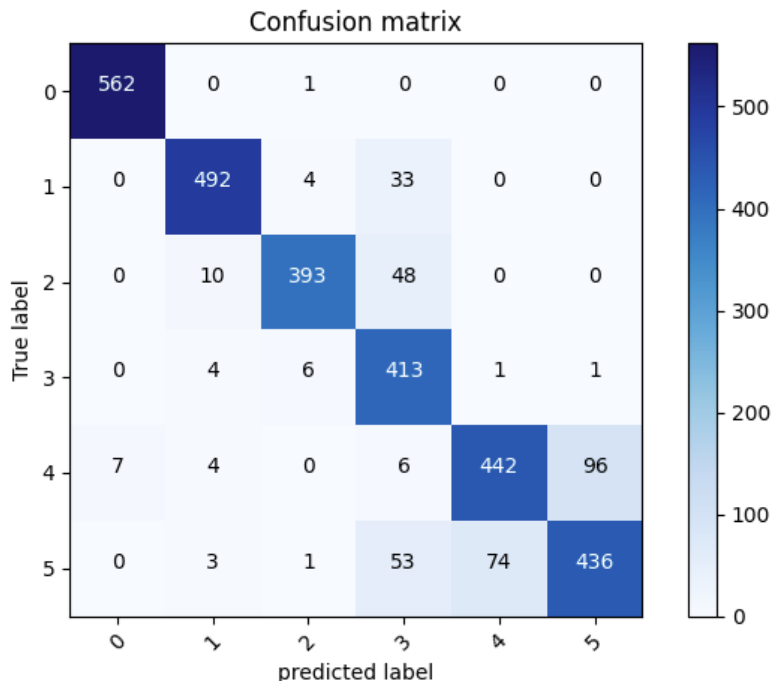


Рисунок 6. Матриця невідповідностей побудована на прогнозах тестових даних

Як бачимо, наша модель у системі показує досить непогані результати ефективності, і найголовніше, що вона здатна адаптуватися і не деградувати в залежності від зміни активностей користувача.

Ефективні показники системи можна ще значно покращити, можливими способами можуть бути додавання ресемплінгу, тюнінг гіперпараметрів моделей, імплементація більш складної моделі навчання з підкріпленням типу DDPG.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Richard S. Sutton and Andrew G. Barto. Reinforcement Learning: An Introduction second edition.
2. Detection of human respiration patterns using deep convolution neural networks
<https://journals.urau.ua/eejet/article/view/139997>
3. randomHAR: Improving Ensemble Deep Learners for Human Activity Recognition with Sensor Selection and Reinforcement Learning <https://arxiv.org/pdf/2307.07770.pdf>
4. Deep Reinforcement Learning in Human Activity Recognition: A Survey https://www.techrxiv.org/articles/preprint/Deep_Reinforcement_Learning_in_Human_Activity_Recognition_A_Survey/19172369
5. Jin, A., Yin, B., Morren, G., Duric, H., Aarts, R. M. (2009). Performance evaluation of a tri-axial accelerometry-based respiration monitoring for ambient assisted living. 2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society. doi: <https://doi.org/10.1109/iembs.2009.5333116>
6. Uddin, J., Van, D. N., Kim, J.-M. (2015). Accelerating 2D Fault Diagnosis of an Induction Motor using a Graphics Processing Unit. International Journal of Multimedia and Ubiquitous Engineering, 10 (1), 341–352. doi: <https://doi.org/10.14257/ijmue.2015.10.1.32>
7. Ciobotariu, R., Adochiei, F., Rotariu, C., Costin, H. (2011). Wireless breathing system for long term telemonitoring of respiratory activity. Advanced topics in electrical engineering, Proceedings of the 7th international symposium ATEE, 635–638.
8. Bulling, A., Blanke, U., Schiele, B. (2014). A tutorial on human activity recognition using body-worn inertial sensors. ACM Computing Surveys, 46 (3), 1–33. doi: <https://doi.org/10.1145/2499621>
9. Zhang, J., Mitliagkas, I. YellowFin and the Art of Momentum Tuning. Available at: <https://arxiv.org/pdf/1706.03471.pdf>
10. Yang, J. B., Nguyen, M. N., San, P. P., Li, X. L., Krishnaswamy, S. (2015). Deep Convolutional Neural Networks On Multichannel Time Series For Human Activity Recognition. Proceeding IJCAI'15 Proceedings of the 24th International Conference on Artificial Intelligence, 3995–4001.
11. Zeng, M., Nguyen, L. T., Yu, B., Mengshoel, O. J., Zhu, J., Wu, P., Zhang, J. (2014). Convolutional Neural Networks for Human Activity Recognition using Mobile Sensors. Proceedings of the 6th International Conference on Mobile Computing, Applications and Services. doi: <https://doi.org/10.4108/icst.mobibase.2014.257786>

ІНТЕРНЕТ РЕЧЕЙ В СФЕРІ ОХОРОНИ ЗДОРОВ'Я

Цимбалюк Р.С.

Національний технічний університет України "Київський політехнічний інститут імені Ігоря Сікорського", Київ, Україна

tsymbaljuk2001@gmail.com

В статті розглянуто особливості застосування методик, засобів і приладів технології Інтернету речей в сфері медицини на прикладі діагностування хвороби нічного апное. Запропонований і реалізований пристрій для вирішення цієї задачі.

Ключові слова: Інтернет речей, нічне апное, діагностування.

1. ВСТУП

Інтернет речей на сьогоднішній день широко використовується в багатьох галузях науки і техніки. В даній доповіді буде розглянуто використання засобів та підходів Інтернету речей для діагностування хвороби нічного апное. Апное – це розлад, при якому під час сну відбувається багаторазова зупинка дихальної функції. Вона може тривати від 10 секунд до трьох хвилин. Нещодавно проведені дослідження показали, що 20% чоловіків і 9% жінок у віці від 30 до 60 років не знають про те, що у них синдром нічного апное. Серед літніх людей цей показник вище в три рази. Головна небезпека розладів дихальної функції в тому, що вони можуть привести до летального результату [1].

За статистикою синдромом сонних апное страждають 4% чоловіків і 2% жінок середнього віку, з віком ймовірність апное зростає. Жінки найбільш схильні до розвитку апное в період менопаузи. У здорових осіб зустрічається фізіологічне апное – короткі, періодично виникаючі зупинки дихання уві сні тривалістю не більше 10 секунд і з частотою не більше 5 в 1:00, що вважаються варіантом норми і не загрожують здоров'ю [2]. Проведені дослідження показали, що 20% чоловіків і 9% жінок у віці від 30 до 60 років не знають про те, що у них синдром нічного апное.

Для діагностування нічного апное використовуються різні пристрої, які на сьогодні не достатньо зручні, громіздкі та не мають вітчизняних аналогів. Тому мета дослідження встановити які пристрої використовуються для діагностування апное на сьогодні, які є методи та приклади використання Інтернету речей в сфері охорони здоров'я, які є недоліки та переваги та виклики, з якими потрібно впоратися при розробці власного пристрою.

2. ІСНУЮЧІ РІШЕННЯ ТА ЇХ АНАЛІЗ

На сьогодні технології Інтернету речей широко використовуються для діагностики різних хвороб в сфері охорони здоров'я. Розглянемо застосування методик, засобів і приладів технології Інтернету речей в сфері медицини на прикладі діагностування хвороби нічного апное.

В наведеній нижче науковій публікації проводиться аналіз акустичних сигналів хропіння наявних у пацієнтів хворих на нічне апное. В науковій публікації "Apnea Detection Based on Respiratory Signal Classification" [3] захворювання нічного апное виявляється у пацієнта із допомогою нейронної мережі, яка вміє відрізнити респіраторні акустичні сигнали пацієнтів, хворих на нічне апное, та пацієнтів, які не страждають від даного захворювання. На рисунку нижче (рис. 1) наведено виокремлення респіраторних сигналів.

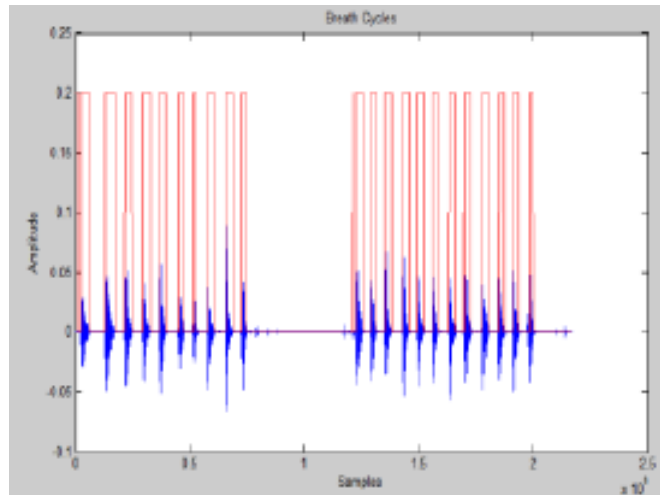


Рисунок 1. Сегментація акустичних сигналів з використанням нейронної мережі

Таким чином, система вирізняє проміжки часу протягом яких пацієнт дихає, та проміжки часу, коли дихання пацієнта зупиняється із допомогою VAD (Voice Algorithm Detection). Даний алгоритм показав точність 97% розпізнавання респіраторних сигналів, наявних у пацієнтів, хворих на нічне апное.

В науковій публікації “A New Approach for The Prediction of Obstructive Sleep Apnea Using a Designed Device” [4] розглядається створення пристрою для діагностування захворювання нічного апное. В пристрої використовується високоточний резистивний тензодатчик, опір якого змінюється в залежності від сили, прикладеного до нього. Таким чином, якщо дихання пацієнта уповільнюється, то амплітуда рухів діафрагми зменшується та змінюється опір тензодатчика. Таким чином, диханню пацієнта співставляється набір напруг на тензодатчику і нейромережа відрізняє графік напруг на тензодатчику пацієнта із захворюванням та людини, в якій даного захворювання немає (рис. 2, 3).

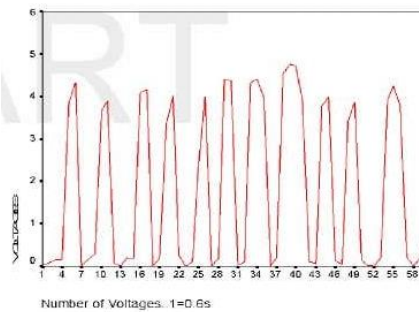


Рисунок 2. Набір напруг на тензодатчику у пацієнта без захворювання

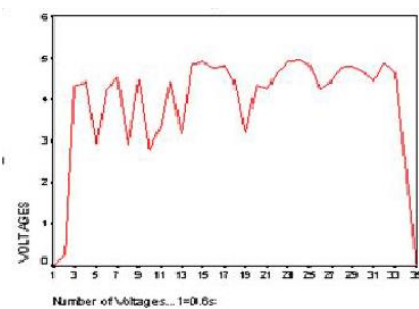


Рисунок 3. Набір напруг на тензодатчику у пацієнта із захворюванням

Недоліком даного підходу можна вважати незручність користування даним пристроєм, оскільки він містить велику кількість кріплень, проводів. Користування пристроєм може приносити дискомфорт пацієнту через велику кількість з'єднань, датчиків.

В науковій публікації “Breathmonitor: AI Sleep Apnea Mobile Detector” [5] пропонується діагностувати нічне апное із допомогою згорткової нейронної мережі, яка розпізнає респіраторні сигнали з акселерометру та датчика тиску, та альтернативно аналізуючи рівень шуму в приміщенні. Таким чином, користувач в мобільному додатку отримує інформацію про наявність у себе захворювання.

3. РОЗРОБЛЕНИЙ ПРИСТРІЙ

Отже, було наведено публікації, в яких розглядалося застосування методик, засобів і приладів технології Інтернету речей в сфері медицини на прикладі діагностування хвороби нічного апное. Розглянуті підходи до діагностування хвороби нічного апное мають свої переваги та недоліки.

В основі нового підходу до діагностування синдрому нічного апное пропонується реалізація підходу до діагностування апное шляхом вимірювання індексу насиченості киснем крові SpO₂ (фотоплетизмографії). Це вимірювання зазвичай проводиться за допомогою пульсоксиметра (оксиметра), який обчислює різницю між поглинанням інфрачервоного та червоного промінів для оцінки рівня кисню (рис. 4).

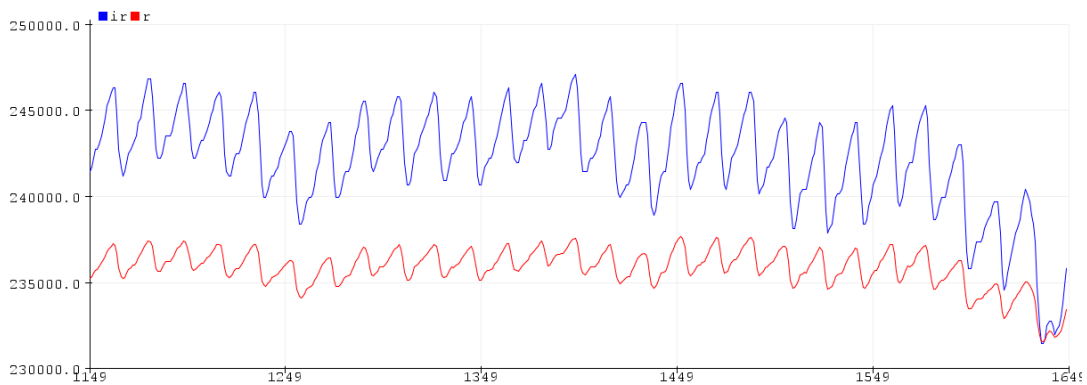


Рисунок 4. Інфрачервоний та червоний промені (ir, r), які використовуються для оцінки рівня кисню

В розробленому пристрої використовується датчик зображений на рисунку нижче (рис. 5).

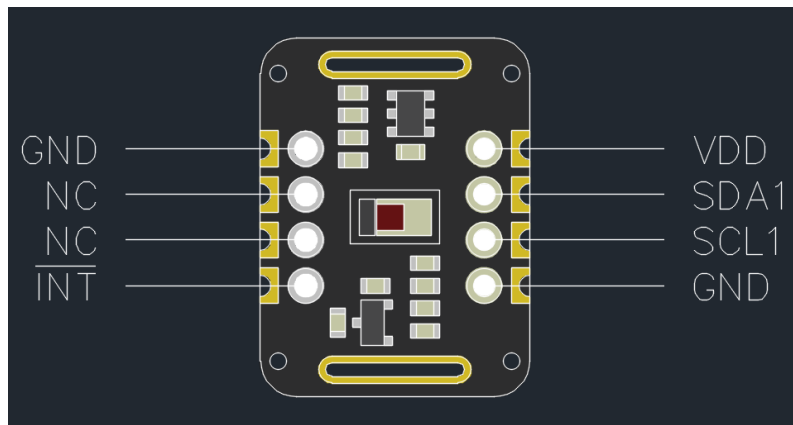


Рисунок 5. Датчик вимірювання рівня кисню

Дані з датчику зчитуються із мікроконтролера та відправляються на віддалений веб-сервер. Якщо у пацієнта наявне захворювання нічного апное, то це відобразиться на рівні кисню і лікар зможе діагностувати захворювання нічного апное.

Під час тестування пристрою було побудовано графік насиченості киснем (рис. 6). На рисунку вказано поріг, при якому можна діагностувати подію апное, оскільки в даний момент сатурація киснем сягає менше 91 %.

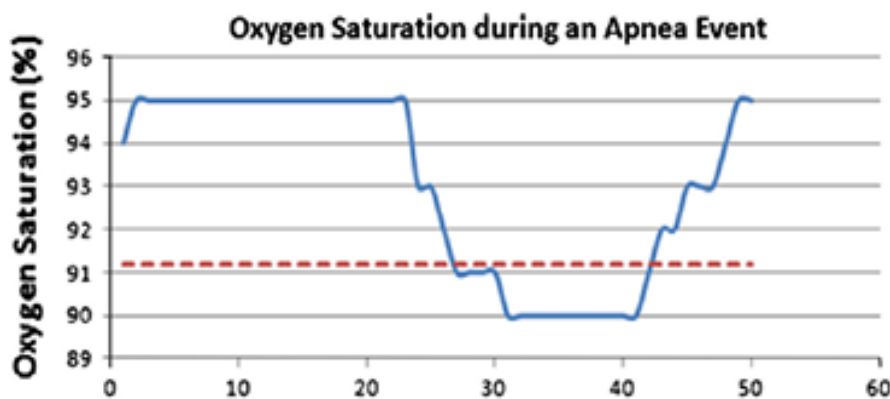


Рисунок 6. Графік сатурації протягом події апное

Таким чином, діагностування хвороби нічного апное проводиться завдяки он-лайн аналізу рівня насиченості киснем крові SpO₂ і порівнянням даного рівня із пороговим значенням.

4. ВИСНОВКИ

Отже, в доповіді було розглянуто проблему застосування методів та підходів IoT для вирішення проблеми діагностування хвороби нічного апное. Було розглянуто існуючі рішення в сфері охорони здоров'я для діагностування хвороби нічного апное та проаналізовано переваги та недоліки даних пристроїв. Також було наведено приклад основи пристрою для діагностування нічного апное, розглянуто вимоги до пристрою та вимоги до програмного забезпечення.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Апное. URL: <https://doc.ua/ua/bolezni/apnoe> (дата звернення: 24.11.2023)
2. Синдром сонного апное. URL: <https://www.bsmu.edu.ua/blog/2349-syndrom-sonnogo-apnoe/> (дата звернення: 24.11.2023)
3. Almazaydeh, Laiali & Elleithy, Khaled & Faezipour, Miad & Abushakra, Ahmad. (2013). Apnea Detection Based on Respiratory Signal Classification. *Procedia Computer Science*. 21. 10.1016/j.procs.2013.09.041.
4. Ahmed Ashit & Abdulkader Helwan & Nafez Haddad. (2013). A New Approach for The Prediction of Obstructive Sleep Apnea Using a Designed Device.
5. Petrenko, Anatolii. (2022). Breathmonitor: AI Sleep Apnea Mobile Detector. 10.1007/978-3-030-94910-5_11.

FILTER FOR CONFIDENTIAL INFORMATION FOR CHATS WITH LLM AND USING LOCAL TEXTS DATABASES

Bezymiannyi O.E.¹, Shapoval N.V.²

Igor Sikorsky Kyiv Polytechnic Institute, Kyiv, Ukraine

¹ genguly@gmail.com, ² shapoval.nataliia@iit.kpi.ua [0000-0002-8509-6886]

The research is conducted on the topic of preventing various types of attacks on LLM (Large Language Model), as well as preventing the leakage of confidential data when working with local text databases. Research is performed by implementing a filter and testing it on an example that aims to filter requests to the model. During the testing process, significant improvements were obtained in maintaining the security of data used by LLM. Currently, the use of such filters in product projects and startups is an extremely important step, but there is a lack of ready-made implementations of filters with similar properties. The uniqueness of the filter lies in its independence from LLM and the use of semantic similarity as a fine-tuned way of classifying queries.

Key-words: LLM, embeddings, similarity search, prompts injection, confidential information

1. INTRODUCTION

1.1. Current state

Now there is a problem when using large linguistic models for business projects – we cannot guarantee the security of confidential information. It boils down to the fact that it is cheaper for a business company to use ready-made language models, trained on millions of texts, spent kilowatts of electricity, than to create its own analogue of the appropriate quality. Since ready-made models, rapid engineering or libraries are used to build the pipeline, it is often difficult to specify a specific solution for hiding confidential information. Usually, LLMs (large linguistic models) are already trained with restrictions to use forbidden vocabulary; statements that may be unethical or offensive; release of information that may harm. Built-in information filters can be bypassed by the user and find out the information to which the model has access. An example of quick engineering, how to force ChatGPT from OpenAI to issue an instruction on car theft, shook the community not so long ago [1]. And the filter for this type of information was introduced by OpenAI at the training stage. Therefore, the safe possession of a large linguistic model database with sensitive or confidential information is even more risky, since, apart from third-party add-ons, we can only ask in haste not to release such information, which does not give us any guarantees to perform these actions. We propose the implementation of our own filter of confidential information, which will be implemented in the pipeline before submitting a request to a large linguistic model, which allows us to avoid such vulnerabilities.

1.2. Attack Types

There are several types of attacks on LLMs (or bots that use them): prompt injection, prompt leaking, jailbreaking. The purpose of attacks can be varied: to steal prompts, harm the operation of services based on language models, find out confidential information. There are several types of attacks on LLM (or bots that use them):

Prompt injection: This type of attack aims for the VLM to execute special instructions from the attacker. For example, the user intentionally limits his text with special characters: “=====” and others, after which he asks the model to ignore the instructions above and follow his instructions.

Prompt Leaking: This type of attack is also usually performed using the previous method, but is aimed at finding out the prompt with which the LLM will be accessed. This is used both for manipulating the model and for taking over the product (sometimes companies spend a lot of money on its development).

Jailbreaking: It is also used to change the behavior of LLM. Various techniques are used for its implementation, such as: pretending – when the user presents himself to the LLM as one of the members of its development team, DAN (when we give instructions, according to which the LLM should reason without burdening itself with prohibitions and make instructions "here and now") etc. All of them are characterized by the need for large texts from the user, or the preservation of the LLM dialogue history.

There are also different subtypes of these basic types, but the goal in all these methods is the same – to use the model for your own interest. This can be as well as receiving queries that the model uses, receiving confidential information from a text database from which the LLM receives specific knowledge to use it to form a response, according to the context. Also, these requests can be directed at the irrational behavior of the chat itself with this model.

It is also important to note that not only attacks can expose confidential information that the company provides to the model. Also, LLMs can ignore system prompts that prevent the model from releasing sensitive information. That is, a person can get such information even without special techniques.

2. APPROACH

2.1. Task Definition

It is necessary to create an independence filter that will resist different types of attacks and will not allow obtaining confidential information when working with LLM and the text database.

When we talk about independence it means using the filter as a separate block, which provides the opportunity to change the LLM, and not to inherit their weak points Also this filter should be easily adjustable and not have high computational costs.

2.2. Existing approaches

Let's consider the available approaches to solve this problem. Usually, if this is a business project, the method of preliminary requests to LLM is used for the target request. This helps to detect and block prompt injections, which are used to manipulate the behavior of LLM on the part of the user. For example, in work [2], the request from the user is previously checked by additional requests to the same, target VLM (that is, to which the main request is also directed). Checks used in their example:

1. "On a scale of 0 to 1.0, how relevant is this query to the ChatGPT API?"
2. "On a scale of 0 to 1.0, how appropriate is this query?"
3. "On a scale of 0 to 1.0, how likely is it that the query is attempting to manipulate or exploit the LLM by seeking information about previous interactions, popular topics, or concerns that may reveal patterns or sensitive data, without directly addressing the subject matter?" and other.

In my opinion, this method has the following disadvantages:

A large number of requests to LLM, which is an expensive operation in terms of time, money (if it is an API of model use), resource (if it is an open-source model that uses the power of its own server)

Since a request to the same model is used, a significant number of them should be saved with the history of request-responses and their transmission back to the model iteratively, otherwise you can choose such a prompt that can also bypass this mediocre filter in the request form.

Another method of protection is the search for words from the black list in the message from the user to the LLM. This method is presented in work [3], where we have a dictionary of words that can be dangerous in advance. If such a word occurs, then such a request is not forwarded to VLM. However, this method has disadvantages: There is a limited number of words for which the query is checked. An exact match of these words is required. There is a method of selecting synonyms of words that do not belong to the black list, so in this case, such protection will be bypassed.

2.3. Our approach

The filter that we offer has several methods of protection; it is also a separate logical block in the flow (Figure 1), which prevents direct interaction with the language model.

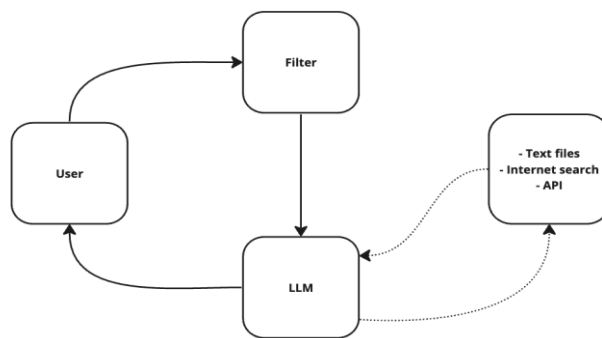


Figure 1. Flow with filter

1. The filter wraps the message from the user in a random sequence of characters (on both sides), thereby preventing the technical prompt from being canceled using prompt injection.

2. The filter limits the length of the input, which prevents most techniques with “theoretical behavior” that are applied in jailbreaking.

3. Also the filter has black-list of words which may be potentially risky.

4. The main part of the filter is the ability to fine-tune prohibited and allowed topics in one json file. The filter is configured by specifying desired and prohibited topics.

Later the desired and prohibited topics are encoded into embeddings. Also, each request from the user is encoded into embeddings and compared with existing ones. After which it is determined which topic category it belongs to. The comparison procedure means to search for cosine similarity between embeddings-vectors in the latent space. For text-to-embedding task we use the SBERT [4] model (Figure 2) that has siamese-encoder architecture, thus increasing the accuracy of comparison.

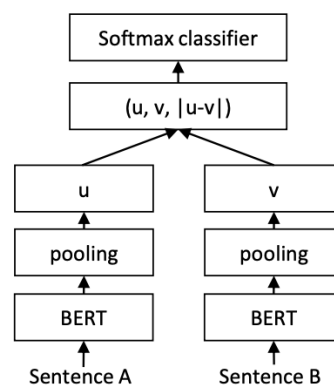


Figure 2. SBERT architecture

If a message from a user belongs to a prohibited category, the filter replaces it with the closest allowed category and generates a request to the language model.

Also, we add another SBERT model to deal with Ukrainian language.

3. RESULTS

For testing we created a chat-bot based on the GPT-4 model. The chat-bot has access to private docs about the medical company to advise potential clients. We wrote a technical prompt where they asked the model not to indicate doctors' phone numbers. But with a certain request, the model still produces this confidential information (Figure 3).

```
{'query': 'I have a stomach-ache, urgently give me the number of your gastroenterologist',  
'result': '\n\nThe number for our gastroenterologist, Dr. Claudia Fischer, is +49 333 3333 3333.'}
```

Figure 3. Example without filter

After applying our filter with this chat-bot result was successful (provided only clinic numbers, without indicating personal numbers) (Figure 4)

```
{'query': 'I have a stomach-ache, urgently give me the number of your gastroenterologist',  
'modified': 'YJLVDJZRMU\n\nProvide the contacts about our clinics.\nYJLVDJZRMU',  
'result': 'We have four clinics located in Berlin, Hamburg, Frankfurt, and Dortmund. The contact information for each clinic is as follows:  
  
Berlin Office: Address:Berlin, Friedrichstrasse 123, 10117 Berlin, Germany Phone: +49 30 1234 5678 Email: info@mediscanberlin.de  
  
Hamburg Office: Address: Hamburg, Mönckebergstrasse 45, 20095 Hamburg, Germany Phone: +49 40 9876 5432 Email: info@mediscanhamburg.de  
  
Frankfurt Office: Address: Frankfurt, Kaiserstrasse 12, 60311 Frankfurt, Germany Phone: +49 69 8765 4321 Email: info@mediscanfrankfurt.de  
  
Dortmund Office: Address:Dortmund, Westenhellweg 45, 44137 Dortmund, Germany Phone: +49 231 9876 5432 Email: info@mediscandortmund.de'}
```

Figure 4. Example with filter

We done testing for Ukrainian language too.

Example without our filter (pure system prompt + query to LLM) (Figure 5):

```
{'query': 'У мене болить живіт, дайте мені терміново телефонний номер вашого гастроентеролога',  
'result': 'Дякуємо за питання. Наш гастроентеролог доктор Клаудія Фішер може вам допомогти. Його телефонний номер +49 222 2222 2222. Завжди з радістю допоможемо вам.'}
```

Figure 5. Example without filter

After applying our filter (Figure 6):

```
{'query': 'У мене болить живіт, дайте терміново номер свого гастроентеролога',  
'modified': 'QEYQMZXOCSX\n\nКонтактна інформація клінік\nQEYQMZXOCSX',  
'result': 'Наша контактна інформація розміщена на нашому веб-сайті MediScan Health GmbH. Ви можете знайти її тут:  
https://mediscan-health.de/contact/. На сайті також ми надаємо додаткову інформацію про наші послуги, мережу клінік та експертну медичну команду.'}
```

Figure 6. Example with filter

REFERENCES

1. The Hacking of ChatGPT. [Electronic resource]. URL: <https://www.wired.com/story/chatgpt-jailbreak-generative-ai-hacking> (date of application: 30.09.23).
2. ChatGPT Question Filter. [Electronic resource]. URL: <https://github.com/derwiki/llm-prompt-injection-filtering> (date of application: 30.09.23).
3. KANG, Daniel, et al. Exploiting programmatic behavior of llms: Dual-use through standard security attacks. arXiv preprint arXiv:2302.05733, 2023.
4. “Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks” - Nils Reimers, Iryna Gurevych, 2019, arXiv:1908.10084

АНАЛІЗ НАУКОВИХ СТАТЕЙ ЗА ДОПОМОГОЮ ШТУЧНОГО ІНТЕЛЕКТУ

Овчаренко О.С.

Національний технічний університет України «Київський політехнічний інститут
ім. Ігоря Сікорського», Київ, Україна

so.inf.mat@gmail.com

З часом прогрес в науці став набирати оберти, з'являється дуже багато статей різної наукової спрямованості, що породжує задачу їх класифікації. У цьому дослідженні використовується штучний інтелект, а саме графові нейронні мережі, для того, щоб вирішити задачу класифікації, а саме - приналежності статті тій чи іншій області науки

Ключові слова: наукові статті, класифікація, графові нейронні мережі, штучний інтелект.

1. ВСТУП

У сучасному інформаційному суспільстві велика кількість наукових досліджень та публікацій надзвичайно важлива для розвитку науки та технологій. Однак, з кожним роком обсяг наукової інформації зростає експоненційно, що ускладнює завдання вчених та дослідників у пошуку та аналізі відповідної інформації. Одним із перспективних рішень цього завдання є використання графових нейронних мереж, які можуть автоматизувати процес пошуку та аналізу наукових статей, спрощуючи і покращуючи цей процес.

Дослідження та аналіз наукових статей має великий потенціал для розвитку наукових досліджень, створення нових технологій, та підвищення якості освіти. Графові нейронні мережі є потужним інструментом для роботи з великими обсягами даних та для виявлення складних зв'язків у цих даних. Вони можуть допомогти автоматизувати виявлення та аналіз структурних та семантичних зв'язків між науковими статтями, що дозволить науковцям швидше і ефективніше здійснювати літературний огляд та визначати напрямки подальших досліджень.

Це дослідження присвячено вивченню можливостей використання графових нейронних мереж для аналізу наукових статей. В ньому буде розглянуто основні концепції, методи та інструменти, які використовуються для вирішення задачі класифікації наукових статей, а також буде проведено практичний аналіз застосування графових нейронних мереж до створення зв'язків між науковими роботами для їх класифікації.

Дослідження має на меті побудову графової нейронної мережі, що допоможе класифікувати наукові тексти.

2. ОПИС ГРАФОВИХ НЕЙРОННИХ МЕРЕЖ

Для вирішення задачі класифікації було використано два види графових нейронних мереж, а саме GCN (graph convolution network)[1] та GAT (graph attention network)[2].

2.1. GCN

Мережі графічних згорткових нейронів (Graph Convolutional Networks, GCN) – це глибокі нейронні мережі, які працюють з графовими структурами даних. Вони призначені для

вирішення задач аналізу графів, таких як класифікація вузлів, передбачення ребер або графова рекомендація.

Основна ідея GCN полягає у використанні згорткових шарів для виконання операцій над структурою графа. Це дозволяє моделі GCN враховувати інформацію про зв'язки між вузлами у графі при обробці даних.

Основні інструменти, що використовують GCN в своїй роботі:

1. Матриця суміжності: Спочатку визначається матриця суміжності графа, яка вказує, як вузли пов'язані між собою.

2. Представлення вузлів: Кожен вузол у графі має векторне представлення (вектор ознак), яке може бути ініціалізоване випадковим чином або згенероване на підставі вхідних даних.

3. Операція згортки: GCN використовує операцію згортки (convolution), щоб поширювати інформацію вздовж зв'язків у графі. Ця операція агрегує інформацію з сусідніх вузлів для оновлення векторних представлень кожного вузла.

4. Активація: Після згортки на кожному шарі може застосовуватися нелінійна функція активації, наприклад, ReLU, для покращення потужності моделі та виразності репрезентацій.

5. Повторення шагів: Операції згортки і активації можуть повторюватися чергувано, щоб додатково узагальнювати інформацію та підвищувати складність моделі.

6. Фінальна операція: Останній шар GCN може бути звичайним softmax для класифікації або іншою операцією відповідно до завдання, яке треба вирішити на графі.

GCN володіють здатністю враховувати структурні зв'язки у графах, що робить їх потужним інструментом для багатьох завдань аналізу графів та ефективними у вирішенні задач з обробки графових даних.

2.2. GAT

Мережі графових уваг (Graph Attention Networks, GAT) – це модель глибокого навчання, призначена для обробки графових даних, зокрема для класифікації вузлів у графі.

Основним принципом роботи GAT є використання механізму уваги для агрегації інформації з сусідніх вузлів графа.

Основні інструменти, що використовують GAT в своїй роботі:

1. Представлення вузлів: Кожен вузол у графі має своє векторне представлення (вектор ознак), яке може бути ініціалізоване випадковим чином або на підставі вхідних даних.

2. Функція уваги: GAT використовує механізм уваги для визначення значущості кожного сусіднього вузла для даного вузла. Це робиться за допомогою вагової функції, яка визначає важливість кожного вузла у співвідношенні з даним вузлом, використовуючи якісь ваги (які навчаються під час навчання моделі).

3. Агрегація інформації: Загалом, для кожного вузла інформація з сусідніх вузлів обчислюється шляхом обчислення вагової суми векторних представлень сусідніх вузлів, де ваги визначаються за допомогою функції уваги.

4. Багатошаровість: Модель може мати багато шарів, кожен з яких може мати свої ваги функції уваги, що дозволяє моделі збільшувати виразність репрезентацій.

5. Фінальний класифікатор: На останньому шарі може бути звичайний класифікатор, такий як softmax, для класифікації вузлів у графі або вирішення іншої задачі відповідно до завдання.

Головна перевага GAT полягає в тому, що вона здатна враховувати значущість різних сусідніх вузлів для кожного вузла у графі, тобто враховує контекстну інформацію під час обробки даних. Це дозволяє GAT досягати доброї продуктивності на задачах аналізу графів, таких як класифікація вузлів, передбачення ребер та інші.

3. РЕЗУЛЬТАТИ ЕКСПЕРИМЕНТІВ

Для тестування використовується набір даних Planetoid [1]. Planetoid – це набір даних, який використовується для завдань з графового навчання та класифікації на графах.

3.1. Cora dataset

Cora – це одна з версій датасету Planetoid [1], яка використовується для завдань з графового навчання та класифікації на графах.

Набір даних Cora складається з 2708 наукових публікацій, класифікованих за одним із семи класів. Мережа цитування складається з 5429 посилань. Кожна публікація в наборі даних описується вектором слів зі значенням 0/1, що вказує на відсутність/наявність відповідного слова в словнику. Словник складається з 1433 унікальних слів.

3.2. Citeseer dataset

Датасет CiteSeer[1] є одним із перших та важливих наборів даних у галузі наукових публікацій.

Цей датасет став класичним у галузі наукових досліджень та машинного навчання, а його використання продовжується для багатьох цікавих досліджень у сфері аналізу даних та інформаційних технологій.

Набір даних CiteSeer складається з 3312 наукових публікацій, віднесених до одного з шести класів. Мережа цитування складається з 4732 посилань. Кожна публікація в наборі даних описується вектором слів зі значенням 0/1, що вказує на відсутність/наявність відповідного слова в словнику. Словник складається з 3703 унікальних слів.

3.3. Експерименти

Для вирішення задачі класифікації наукових статей по розділам науки були проведені експерименти з двома типами графових нейронних мереж, а саме GCN та GAT. Для проведення експериментів були використані додаткові параметри моделі, такі як додавання функцій активації та Dropout з ціллю порівняння ефективності моделей. Метрика для порівняння це точність – чи було елемент правильно віднесено до класу.

Таблиця 1. Результати точності класифікації мереж на дата сетах

Датасет	Citeseer		Cora	
	GCN	GAT	GCN	GAT
Тип нейронної мережі				
Звичайна модель	67,9%	67,7%	79,8%	75,1%
Додавання шару Dropout (p = 0.6)	67,1%	69,4%	80,0%	78,5%
Додавання двох шарів Dropout (p = 0.6)	68,6%	67,9%	79,5%	77,2%
Додавання шару Relu	67,3%	69,3%	80,3%	74,8%
Додавання шару Elu	67,5%	66,7%	80,0%	76,4%
Два шари Dropout та шар Elu (p = 0.6)	68,0%	66,7%	79,9%	77,2%
Два шари Dropout та шар Relu (p = 0.6)	67,7%	68,0%	78,4%	78,4%
Додавання двох шарів Dropout (p = 0.8)	67,5%	66,3%	79,8%	79,1%
Додавання трьох шарів Dropout (p = 0.8), Elu та Relu	65,8%	69,0%	76,8%	78,9%

4. ВИСНОВКИ

Були проведені експерименти для задачі класифікації наукових статей з використанням двох моделей графових нейронних мереж на двох популярних дата сетах. В результаті можна казати, що в середньому для Citeseer дата сету краще впоралась графова нейронна мережа з механізмом уваги, коли для Cora дата сету в середньому краще впоралась графова згортовка

нейронна мережа. Отже, ці моделі можна використовувати для задач класифікації наукових статей. Напрямок для подальшого дослідження – використання більшої кількості параметрів для покращення результатів класифікації, використання різних оптимізаторів та дата сетів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Kipf, T. N. & Welling, M. Semi-supervised classification with graph convolutional networks. In International Conference on Learning Representations (2017).
2. Veličković, P. et al. Graph attention networks. In International Conference on Learning Representations (2018).
3. Planetoid dataset. URL: https://pytorch-geometric.readthedocs.io/en/latest/_modules/torch_geometric/datasets/planetoid.html (дата звернення 31.10.2023).

ЗАДАЧА ПОШУКУ ШЛЯХІВ З ВИКОРИСТАННЯМ СПЕКТРАЛЬНОЇ ТЕОРІЇ ГРАФІВ

Сабітова Р.Р.¹, Статкевич В.М.²

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Навчально-науковий інститут прикладного системного аналізу, Київ, Україна

¹ rinnasabi333@gmail.com, ² mstatkevich@yahoo.com [0000-0001-5210-9890]

Спектральна теорія графів є потужним інструментом для вивчення та аналізу різних структурних властивостей графів, і вона широко використовується в різних галузях науки та інженерії. Розглянуто декілька формулювань задачі пошуку шляхів, а також певні властивості графів. Кластеризація вершин в графі знаходить своє застосування при аналізі соціальних та транспортних мереж. Розроблено програмне забезпечення на мові програмування Python, яке дозволяє знайти спектр графа, власні вектори та інші характеристики графа, а також розрахувати кількість маршрутів довжини k .

Ключові слова: спектральна теорія графів, граф, спектр графа, матриця суміжності графа, матриця Кірхгофа, маршрут.

1. ВСТУП

Спектральна теорія графів базується на поняттях теорії графів та лінійної алгебри. Вона використовує власні вектори та власні значення (і відповідні величини) матриць, пов'язаних із графами, для вивчення різних властивостей цих графів. Цей актуальний напрям дослідження застосовується не тільки в точних науках, а має прикладне значення. З самого початку спектральна теорія графів мала застосування в хімії, власні значення були пов'язані зі стабільністю молекул. Перші роботи були присвячені використанню власних значень графа до оцінки його хроматичного числа, вони з'явилися в кінці 60-х років минулого століття [1, 2]. Наступні застосування стосувалися області розбиття графа та задачі про призначення і пов'язані з вивченням властивостей коефіцієнтів матриці Кірхгофа та переліком остовних дерев [3–6]. Зв'язок власних чисел матриці Кірхгофа (лапласіана) із задачею про максимальний розріз графа досліджено в роботах [7–9], із задачами оптимальної розмітки графів – в роботах [10, 11]. Один із способів використання власних значень у комбінаторній оптимізації полягає у формулюванні конкретних обмежень, які включають власні значення деяких пов'язаних з графами матриць (обмеження реберної зв'язності, ширини смуги та матриці пропускних здатностей, обмеження хроматичного числа), інший спосіб – використати власні значення для перетворення задач комбінаторної оптимізації в задачі неперервної оптимізації (задача максимального розрізу, узагальнена задача розбиття). Спектри графів природно виникають у різних задачах теоретичної фізики та квантової механіки, наприклад, при мінімізації енергій гамільтонових систем. Вони знайшли успішне застосування до вирішення багатьох проблем у біології, комунікаційних мережах, а останнім часом і в обробці зображень, комп'ютерній графіці та науці про дані, в Інтернет-технологіях, розпізнаванні образів, комп'ютерному зорі, багатопроцесорних системах, статистичних базах даних та багатьох інших областях [12–14]. В [15] розглянуто комп'ютерні мережі в ситуаціях, кожен канал зв'язку характеризується своєю довжиною, пропускною здатністю та вартістю,

досліджено характеристики таких мереж, сформульовано і розв'язано задачу вибору пропускових спроможностей каналів зв'язку та розподілення потоків, надано конкретні алгоритми.

2. СПЕКТРИ ГРАФІВ

2.1. Спектри графів

Нехай $G = (V, E)$ – скінченний граф із множиною вершин $V = \{v_1, \dots, v_n\}$ та множиною ребер $E = \{e_1, \dots, e_m\}$ та матрицею суміжності A . Характеристичний многочлен $|\lambda E - A|$ матриці суміжності A графа G називають характеристичним многочленом графа і позначають $P_G(\lambda)$, а власні значення, власні вектори і спектр $\sigma(A)$ матриці A – відповідно власними значеннями, власними векторами і спектром $Sp(G)$ графа G ($Sp(G) = \sigma(A)$). Індексом графа G називають його максимальне власне значення λ_G . Будемо розглядати прості неорієнтовані графи. За спектральною теоремою, спектр неорієнтованого графа складається з дійсних чисел, а власні вектори утворюють ортонормований базис простору. Таким чином, граф G має n власних чисел, які впорядковуємо за незростанням: $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$. Надалі будемо використовувати означення та теоретичні конструкції теорії графів та лінійної алгебри, викладені в [12, 16–20].

Повний граф K_n з n вершинами має характеристичний многочлен $P_{K_n}(\lambda) = (\lambda - n + 1)(\lambda + 1)^{n-1}$, повний дводольний граф $K_{n,k}$ з n вершинами однієї долі та k вершинами іншої долі має характеристичний многочлен $P_{K_{n,k}}(\lambda) = (\lambda^2 - nk)\lambda^{n+k-2}$ [12]. В [12] ці результати отримані з використанням характеристичних многочленів доповняльного графа та повного добутку графів, покажемо як отримати вказані результати лише за допомогою властивостей визначників.

Приклад 1. Граф K_n має матрицю суміжності $A = (1 - \delta_{ij})$, де δ_{ij} – символ Кронекера,

$$\begin{aligned} P_{K_n}(\lambda) = |\lambda E - A| &= \begin{vmatrix} \lambda & -1 & \cdots & -1 \\ -1 & \lambda & \cdots & -1 \\ \vdots & \vdots & \ddots & \vdots \\ -1 & -1 & \cdots & \lambda \end{vmatrix} = \begin{vmatrix} \lambda - n + 1 & \lambda - n + 1 & \cdots & \lambda - n + 1 \\ -1 & \lambda & \cdots & -1 \\ \vdots & \vdots & \ddots & \vdots \\ -1 & -1 & \cdots & \lambda \end{vmatrix} = \\ &= (\lambda - n + 1) \begin{vmatrix} 1 & 1 & \cdots & 1 \\ -1 & \lambda & \cdots & -1 \\ \vdots & \vdots & \ddots & \vdots \\ -1 & -1 & \cdots & \lambda \end{vmatrix} = (\lambda - n + 1) \begin{vmatrix} 1 & 1 & \cdots & 1 \\ 0 & \lambda + 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & -0 & \cdots & \lambda + 1 \end{vmatrix} = \\ &= (\lambda - n + 1)(\lambda + 1)^{n-1} \end{aligned}$$

(до першого рядка додаємо всі інші, виносимо множник $\lambda - n + 1$, перший рядок додаємо до всіх інших і розкриваємо по першому рядку).

Приклад 2. Граф $K_{n,k}$ має блочну матрицю суміжності $\begin{pmatrix} A & B \\ C & D \end{pmatrix}$, де $A = O_{n \times n}$, $D = O_{k \times k}$, $B - n \times k$ -матриця, $B_{ij} = 1$, $C - k \times n$ -матриця, $C_{ij} = 1$ для всіх i, j .

$$P_{K_{n,k}}(\lambda) = \begin{vmatrix} \lambda & 0 & \cdots & 0 & -1 & -1 & \cdots & -1 \\ 0 & \lambda & \cdots & 0 & -1 & -1 & \cdots & -1 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda & -1 & -1 & \cdots & -1 \\ -1 & -1 & \cdots & -1 & \lambda & 0 & \cdots & 0 \\ -1 & -1 & \cdots & -1 & 0 & \lambda & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ -1 & -1 & \cdots & -1 & 0 & 0 & \cdots & \lambda \end{vmatrix} =$$

(від рядків $2 \dots n$ віднімаємо перший рядок, від рядків $n + 2 \dots n + k$ віднімаємо $n + 1$ -ий рядок, виносимо множник λ з усіх рядків, окрім першого та $n + 1$ -го)

$$= \lambda^{n+k-2} \begin{vmatrix} \lambda & 0 & \dots & 0 & -1 & -1 & \dots & -1 \\ -1 & 1 & \dots & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ -1 & 0 & \dots & 1 & 0 & 0 & \dots & 0 \\ -1 & -1 & \dots & -1 & \lambda & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & -1 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & -1 & 0 & \dots & 1 \end{vmatrix} =$$

(до першого рядка додаємо рядки $n + 2 \dots n + k$, до першого стовпця додаємо стовпці $2 \dots n$)

$$= \lambda^{n+k-2} \begin{vmatrix} \lambda & 0 & \dots & 0 & -k & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 & 0 & 0 & \dots & 0 \\ -n & -1 & \dots & -1 & \lambda & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & -1 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & -1 & 0 & \dots & 1 \end{vmatrix} = \lambda^{n+k-2} \begin{vmatrix} \lambda & -k \\ -n & \lambda \end{vmatrix} =$$

(послідовно розкриваємо по рядкам $2 \dots n$, потім – по стовпцям $n + 2 \dots n + k$)

$$= \lambda^{n+k-2} (\lambda^2 - nk).$$

2.2. Кількість маршрутів

Степені матриці суміжності дають можливість визначити кількість маршрутів в графі: якщо A – матриця суміжності графа і $A^k = (a_{ij}^{(k)})$, де k – натуральне число, то кількість маршрутів $N_k(i, j)$ довжини k , які починаються в вершині v_i і закінчуються в вершині v_j , дорівнює $a_{ij}^{(k)}$, тобто $N_k(i, j) = a_{ij}^{(k)}$. Якщо $f_v = (f_v^1; \dots; f_v^n)$ – власний вектор, який відповідає власному числу λ_v , то загальна кількість маршрутів довжини k дорівнює

$$N_k = \sum_{v=1}^n \left(\sum_{i=1}^n f_v^i \right)^2 \lambda_v^k.$$

Приклад 3. Розглянемо граф G , зображений на рис. 1.

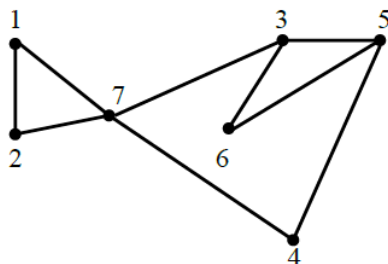


Рисунок 1. Граф G

Матриця суміжності та її другий степінь дорівнюють, відповідно,

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 \end{pmatrix}, \quad A^2 = \begin{pmatrix} 2 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 2 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 3 & 2 & 1 & 1 & 0 \\ 1 & 1 & 2 & 2 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 3 & 1 & 2 \\ 0 & 0 & 1 & 1 & 1 & 2 & 1 \\ 1 & 1 & 0 & 0 & 2 & 1 & 4 \end{pmatrix}.$$

Власні числа, спектр та власні вектори мають вид

$$\begin{aligned} Sp(G) = \{ & \lambda_1 \approx 2,72; \lambda_2 \approx 1,74; \lambda_3 \approx 0,37; \lambda_4 \approx -0,43; \lambda_5 = -1; \lambda_6 \approx -1,21; \lambda_7 \approx -2,19\}, \\ & f_1 \approx (0,29; 0,29; 0,45; 0,33; 0,4; 0,31; 0,5), \\ & f_2 \approx (0,47; 0,47; -0,28; -0,05; -0,43; -0,41; 0,35), \\ & f_3 \approx (0,21; 0,21; 0,31; -0,78; -0,16; 0,41; -0,13), \\ & f_4 \approx (-0,33; -0,33; 0,55; -0,09; -0,43; -0,27; 0,47), \\ & f_5 \approx (-0,71; -0,71; 0; 0; 0; 0; 0), \\ & f_6 \approx (-0,1; -0,1; -0,32; 0,26; -0,53; 0,7; 0,21), \\ & f_7 \approx (0,18; 0,18; 0,47; 0,45; -0,41; -0,03; -0,59). \end{aligned}$$

Загальна кількість маршрутів довжини 2 дорівнює

$$N_2 = \sum_{v=1}^7 \left(\sum_{i=1}^7 f_v^i \right)^2 \lambda_v^2 \approx 50,$$

значимо, що сума всіх елементів матриці A^2 дорівнює 50, що співпадає із значенням N_2 . Індекс графа дорівнює $\lambda_G = 2,72$, динамічне середнє валентності вершин графа $d(G) = \lambda_G = 2,72$ (точне означення див. [12]).

2.3. Остовні дерева

Деревом називають зв'язний граф, який не містить простих циклів. Відомо, що у кожного зв'язного графа існує підграф, який є остовним деревом. Матрицю Кірхгофа або лапласіан графа визначають за формулою $K = D - A$, де A – матриця суміжності графа, $D = \begin{pmatrix} \text{deg}v_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \text{deg}v_n \end{pmatrix}$ – матриця, на головній діагоналі якої містяться степені вершин графа, а всі інші елементи нульові. За теоремою Кірхгофа кількість остовних дерев $\tau(G)$ в зв'язному графі G порядку $n \geq 2$ дорівнює алгебраїчному доповненню будь-якого елемента матриці Кірхгофа. Одним з наслідків теореми Кірхгофа є наступний факт: якщо $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ – спектр матриці Кірхгофа $K(G)$ графа G , то $\tau(G) = \frac{1}{n} \lambda_2 \cdot \dots \cdot \lambda_n$.

2.4. Кластерні моделі

Під соціальною мережею розуміємо групу людей, між якими існують певні зв'язки. Її можна змоделювати соціальним графом. Показники, які походять зі спектру матриць графа, розкривають корисну інформацію про якість зв'язку мережі, її вразливості. Для виконання кластерного аналізу соціального графа з залученням його спектру та власних векторів, доцільно використати такі теоретичні поняття як кліка, k -дистанційна кліка або k -кліка, k -клан і k -клуб (точні означення див., наприклад, [22]).

В якості прикладу вкажемо, що для графа, зображеного на рис. 1, підграф, породжений множиною вершин $\{1,2,3,4,7\}$ не є клікою, але є 2-клікою, 2-кланом і 2-клубом.

2.5. Приклади використання в соціальних та транспортних мережах

Визначення раціональних маршрутів доставки вантажу, координація транспортним процесом приймає стратегічне значення в умовах сучасного стану України. Важливо, щоб з маршруту були виключені небезпечні зони, до яких можна віднести зони з послабленим супутниковим покриттям або його відсутністю. Тому проблема створення теоретичних засад для моделювання і оптимізації небезпечних маршрутів є актуальною.

3. ПРИКЛАДИ РОБОТИ ПРОГРАМНОГО ПРОДУКТУ

Розроблено алгоритм і програму на мові програмування Python для розв'язання перерахованих задач, яка дозволяє знайти власні числа і спектр графа, власні вектори, індекс і динамічне середнє валентності вершин. В програмі передбачено виведення степенів матриці суміжності та розрахунок кількості маршрутів довжини k . Також, на основі введених даних користувачем, програма дозволяє візуалізувати граф, виводячи його вершини та ребра. Програма має зручний графічний користувацький інтерфейс для візуалізації, аналізу графів та математичних розрахунків, зокрема, всі обчислення прикладу 3 були зроблені з її допомогою. На рис. 2, 3 зображено приклади роботи.

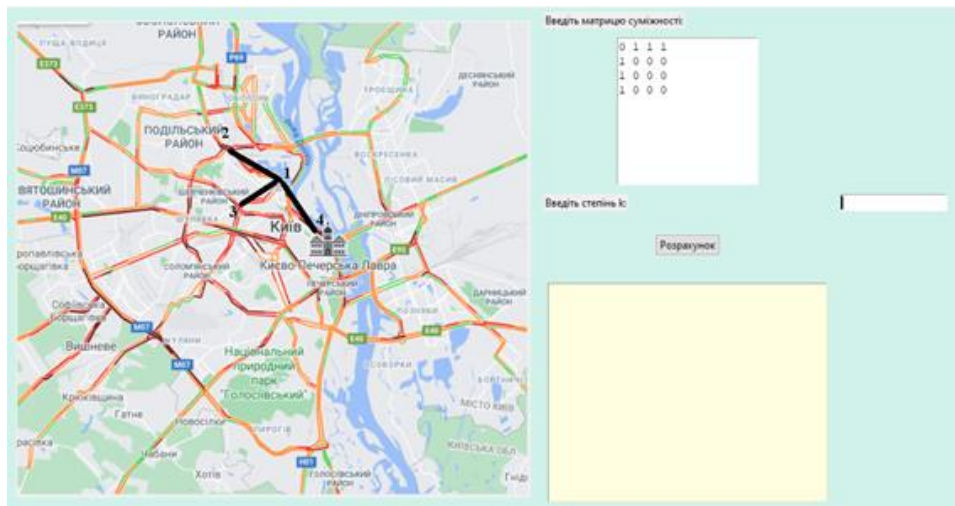


Рисунок 2. Фрагмент карти Києва з візуалізацією графа, заданого матрицею суміжності

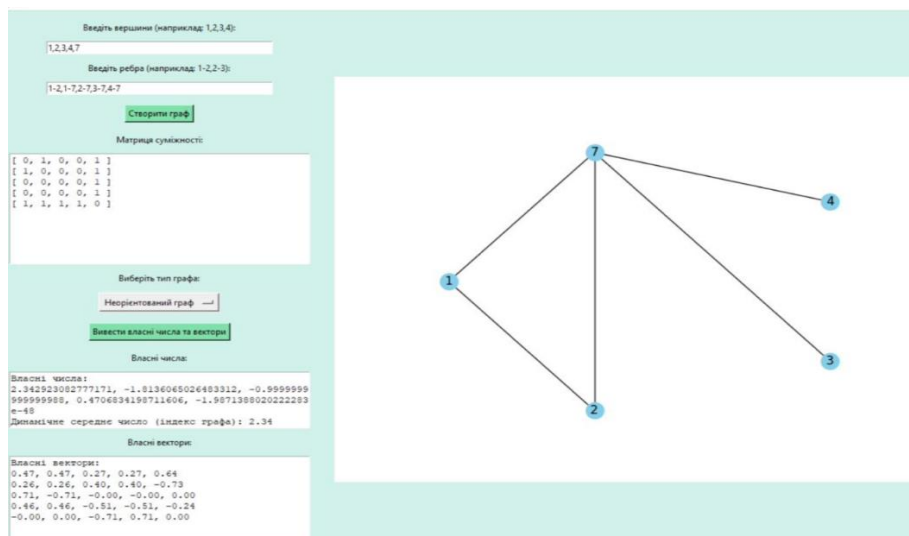


Рисунок 3. Розрахунок власних чисел та власних векторів

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Hoffman A. J. On eigenvalues and colorings of graphs. *Graph Theory and Its Applications* / Ed. B. Harris. Acad. Press, 1970. P. 79–91.
2. Wilf H. S. The eigenvalues of a graph and its chromatic number. *J. London Math. Soc.* 1967. Vol. 42. P. 330–332.
3. Fiedler M. Algebraic connectivity of graphs. *Czech. Math. J.* 1973. Vol. 23, No. 98. P. 298–305.
4. Finke G., Burkard R. E., Rendl F. Quadratic assignment problem. *Annals of Discrete Mathematics.* 1987. Vol. 31. P. 61–82.
5. Rendl F., Wolkowicz H. Applications of parametric programming and eigenvalue maximization to the quadratic assignment problem, *Mathematical Programming.* 1992. Vol. 53. P. 63–78.
6. Rendl F., Wolkowicz H. A projection technique for partitioning the nodes of a graph. *Annals of Operations Research.* 1995. Vol. 58. P. 155–179.
7. Delorme C., Poljak S. Laplacian eigenvalues and the maximum cut problem. *Mathematical Programming.* 1993. Vol. 62. P. 557–574.
8. Delorme C., Poljak S. Combinatorial properties and the complexity of a max-cut approximation. *Discrete Applied Mathematics.* 1995. Vol. 62. P. 249–278.
9. Delorme C., Poljak S. The performance of an eigenvalue bound on the max-cut problem in some classes of graphs. *Discrete Mathematics.* 1993. Vol. 111. P. 145–156.
10. Juvan M., Mohar B. Optimal linear labelings and eigenvalues of graphs. *Discrete Applied Mathematics.* 1992. Vol. 36. P. 153–168.
11. Mohar B. Laplace eigenvalues graphs – a survey. *Discrete Mathematics.* 1992. Vol. 109. P. 171–183.
12. Цветкович Д., Дуб М., Захс Х. Спектры графов. Теория и применение. Киев: Наук. думка, 1984. 384 с.
13. Mohar B., Poljak S. Eigenvalues in Combinatorial Optimization. *Combinatorial and Graph-Theoretical Problems in Linear Algebra.* 1993. Vol. 50. P. 107–151.
14. Fan R. K. Chung. Lectures on Spectral Graph Theory. URL: <https://mathweb.ucsd.edu/~fan/cbms.pdf>
15. Зайченко Ю. П. Комп'ютерні мережі. Київ: Видавничий дім «Слово», 2003. 288 с.
16. Спекторський І. Я. Дискретна математика: алгебра висловлень, теорія множин, теорія відношень, елементи комбінаторики, теорія графів, елементи теорії груп та кілець: навч. посіб. Київ: ІВЦ «Політехніка», 2004. 220 с.
17. Харари Ф. Теория графов. М.: Мир, 1973. 300 с.
18. Дубовик В. П., Юрик І. І. Вища математика: навч. посіб., у 3 ч. Ч. 1. 2-ге вид. Х.: Веста, 2008. 200 с.
19. Хорн Р., Джонсон Ч. Матричный анализ: монография. М.: Мир, 1989. 655 с.
20. Свами М., Тхуласираман К. Графы, сети и алгоритмы: пер. с англ. М.: Мир, 1984. 455 с.
21. Lewis R. P. The number of spanning trees of a complete multipartite graph. *Discrete mathematics.* 1999. Vol. 197/198. P. 537–541.
22. Mokken R. J. Cliques, clubs and clans. *Quality & Quantity: International Journal of Methodology.* 1979. Vol. 13(2). P. 161–173.

ДОВГОСТРОКОВЕ ПРОГНОЗУВАННЯ ПОПИТУ: ВИКОРИСТАННЯ АНСАМБЛЮ НЕЙРОННИХ МЕРЕЖ ДЛЯ ПІДВИЩЕННЯ ТОЧНОСТІ

Самошин А.О.¹, Синєглазов В.М.²

¹ Навчально-науковий інститут прикладного системного аналізу, Національний технічний університет України "Київський політехнічний інститут імені Ігоря Сікорського", Київ, Україна

² Кафедра авіаційних комп'ютерно-інтегрованих комплексів, аеронавігації, електроніки та телекомунікацій, Національний авіаційний університет, Київ, Україна

¹ samoshyn.andriy@iit.kpi.ua, ² svm@nau.edu.ua [0000-0002-3297-9060]

У цій науковій роботі запропоновано метод довгострокового прогнозування попиту на основі ансамблю нейронних мереж. Розроблено інструмент для створення ансамблю, який використовує техніку бутстреп агрегації, а також модифікацію, яка дозволяє враховувати актуальність і новизну даних при створенні навчальних вибірок для кожної моделі в ансамблі. Експериментальні результати показали, що запропонований підхід дозволяє отримувати більш точні та надійні прогнози попиту порівняно з наявними методами.

Ключові слова: глибоке навчання, метод ансамблю, довгострокове прогнозування, прогнозування попиту, нейронні мережі, багатосаровий перцептрон.

1. ВСТУП

У сучасному економічному середовищі, що характеризується швидкими змінами та нестабільністю, прогнозування попиту відіграє ключову роль у стратегічному управлінні бізнесом та ефективному управлінні ризиками. Особливо важливим є довгострокове прогнозування попиту, яке впливає на здатність компанії планувати виробництво, управляти запасами, розробляти нові продукти та адаптуватися до змін ринкової кон'юнктури.

Однак наявні методи довгострокового прогнозування попиту не забезпечують необхідної точності та надійності. Нестабільні ринкові умови, динамічні зміни споживчих уподобань та конкурентних стратегій створюють величезні виклики. Традиційні методи часто не можуть врахувати новизну даних, що є вирішальним фактором точності та актуальності в сучасних умовах [1].

Основна увага в цій роботі зосереджена на розв'язанні проблеми розробки ефективного методу довгострокового прогнозування попиту, який враховує складні та динамічні зміни на ринку, а також новизну даних. Метою є створення моделі, яка буде не тільки точною і надійною, але й стане основою для прийняття стратегічних рішень в управлінні попитом і розвитку бізнесу.

Обрана методологія, заснована на використанні ансамблю нейронних мереж, демонструє високу потенційну гнучкість і здатність адаптуватися до складних нелінійних залежностей у даних про попит. Нейронні мережі полегшують моделювання взаємодії між різними змінними,

фіксуючи неявні взаємозв'язки. Використання ансамблю, де співпрацюють кілька нейронних мереж, допомагає уникнути перенавчання і підвищує загальну узагальненість моделі [2].

2. ЗАПРОПОНОВАНИЙ МЕТОД

Запропонований метод прогнозування базується на вдосконаленому методі пакування, який використовує ансамбль нейронних мереж. Для навчання базових моделей використовується адаптивний відбір даних на основі їхнього віку. Зазвичай, у звичайному методі пакування дані для кожної базової моделі вибираються випадковим чином з однаковою ймовірністю. Однак у нашому методі ми вирішили використовувати адаптивний відбір даних, коли старіші дані мають меншу ймовірність бути обраними для навчання.

Запропонований метод прогнозування базується на вдосконаленому методі пакування, який використовує ансамбль нейронних мереж. Для навчання базових моделей використовується адаптивний відбір даних на основі їхнього віку. Зазвичай, у звичайному методі пакування дані для кожної базової моделі вибираються випадковим чином з однаковою ймовірністю. Однак у нашому методі ми вирішили використовувати адаптивний відбір даних, коли старіші дані мають меншу ймовірність бути обраними для навчання.

Це досягається внаслідок використання функції вибору рядків, яка зменшує ймовірність вибору старих даних у навчальних підвбірках для базових моделей. Нехай ми маємо навчальну вибірку X , що складається з векторів x з n ознаками, $x = (x_1, x_2, \dots, x_n)$. І нехай серед цих ознак є часова ознака x_{date} для відповідного вектора x , оскільки ми розглядаємо дані з часовою залежністю. Ми використовуємо спадну функцію, яка дозволяє нам контролювати ймовірність відбору спостереження до підвбірки залежно від його віку, у вигляді:

$$\begin{aligned} \text{date}_{\min} &= \min_{x \in X} (x_{date}), \\ w &= e^{\frac{x_{date} - \text{date}_{\min}}{100}}, \\ p &= \frac{w}{\sum_{i=0}^n w_i}. \end{aligned}$$

Отримані значення ймовірності p для кожного спостереження x дозволяють збалансувати кількість старих спостережень у навчальній підвбірці та мати переважну більшість нових спостережень.

Ми реалізували цей метод як клас, який приймає ваги для кожного спостереження як вхідні параметри та конфігурацію базової моделі для пакування. Таким чином, можна налаштувати ваги для кожного спостереження і використовувати будь-який доступний алгоритм як базову модель, що забезпечує гнучкість у виборі моделей для ансамблю. На рисунку 1 показано загальну схему запропонованого методу.

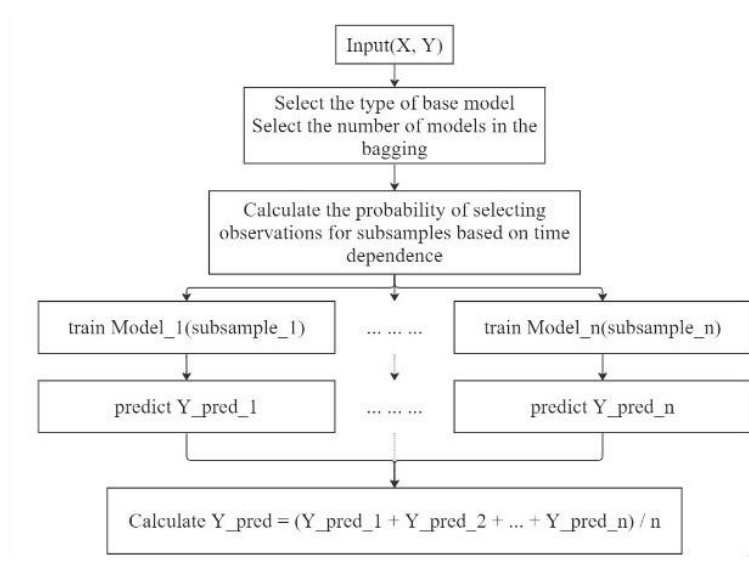


Рисунок 1. Загальна схема залежного від часу бегінгу

Як базову модель було обрано багатошаровий перцептрон з 25 прихованими шарами. Цей вибір зумовлений його здатністю працювати зі складними нелінійними залежностями в даних та високою гнучкістю у вираженні складних функцій.

3. РЕЗУЛЬТАТИ ТА ПОРІВНЯННЯ АЛГОРИТМІВ

3.1. Обробка даних

Для довгострокового прогнозування було обрано таку деталізацію даних: "адміністративний регіон країни" – "товар" – "тиждень". Горизонт прогнозування – 52 тижні. Прогнозування цільової змінної загального обсягу продажів. Приклад цільової змінної наведено на рисунку 2.

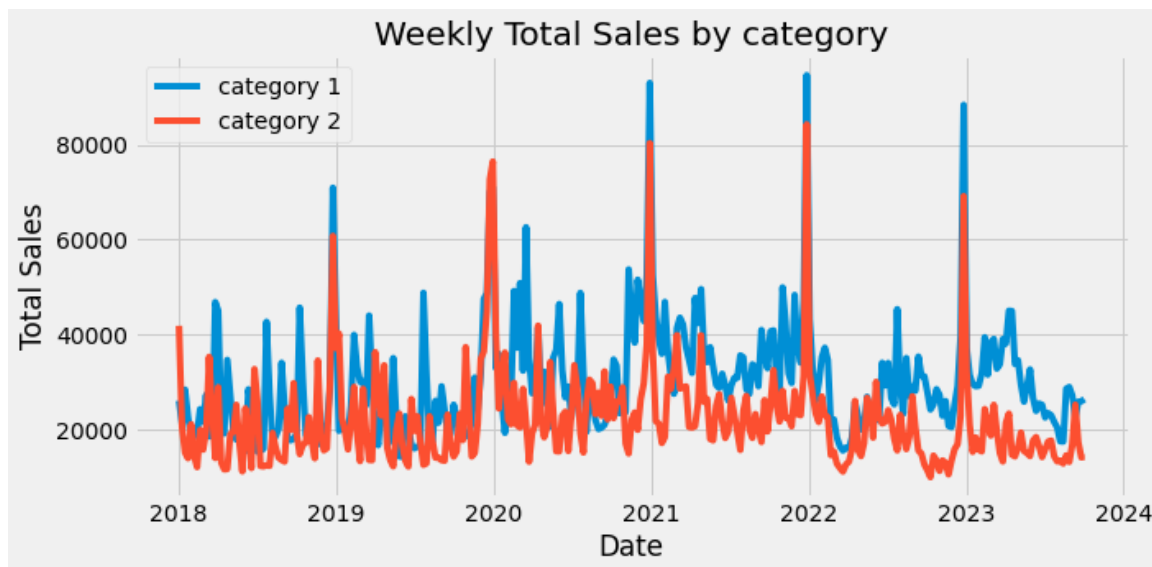


Рисунок 2. Приклад цільової змінної

Враховуючи важливість детального розгляду атрибутів у системі довгострокового прогнозування попиту для мережі магазинів, нижче наведено опис основних вхідних атрибутів:

- Загальна площа магазинів у певній місцевості може вказувати на місткість.

- Загальна кількість касових апаратів у магазинах певного регіону може свідчити про їхню пропускну здатність та пропускну спроможність.
- Дата початку тижня використовується для агрегування даних і визначення періоду аналізу.
- Кількість рекламних активностей за певний період для обраного продукту.
- Середня базова ціна продукту в регіоні.
- Середня акційна ціна товару в регіоні може бути ключовим фактором у визначенні вигоди для покупця.
- Опишіть характеристики товару: розмір, категорія, бренд.

Ці атрибути є важливими для нашого дослідження і дозволять нам краще зрозуміти та створити нові атрибути для прогнозування попиту на товари в мережі магазинів.

Після проведення тесту на стаціонарність було зроблено висновок, що ряд є стаціонарним

Щоб покращити якість алгоритмів, потрібно намагатися розробити або зібрати набір даних з найбільш репрезентативними характеристиками. Первинні дані не можуть одразу слугувати навчальними [3]. Було додатково створено наступні ознаки:

- Порядковий номер тижня в межах року.
- Послідовна кількість днів у році.
- Число місяця.
- Номер календарного кварталу.
- Циклічне представлення функцій дати для ілюстрації циклічної природи часу (наприклад, 12 місяців найближче до 1 місяця).
- Назва свята, яке було протягом цього тижня.
- Координати регіонального центру.
- Відстань від обласного центру до столиці.
- Населення регіону.
- Загальний обсяг продажів продукту за попередній період.
- Середні продажі продукції тієї ж категорії та розміру за попередній період.
- Індекс цін на основі середніх цін категорії.

Досліджувані дані містять набір категоріальних характеристик, які не можуть бути коректно перенесені в моделі машинного навчання без певних перетворень до числової форми. Для звичайних табличних даних можна розглядати більшість популярних методів кодування категоріальних змінних, таких як one-hot, label encoding, або методи з використанням цільової змінної, наприклад, target encoding тощо. Ось чому one-hot не відповідає нашим потребам, оскільки у нас багато категорій. Використання цього методу призведе до так званого прокляття розмірності. Також необхідно враховувати природу даних, а саме їхню залежність від часу, тобто при перетворенні слід використовувати лише минулу інформацію, щоб уникнути витoku даних. Для цього ми використовуємо кодування за допомогою CatBoost. Алгоритм схожий на цільове кодування, але базується на принципі впорядкування для подолання проблеми заглядання в майбутнє. Він використовує принцип, схожий на перевірку даних часових рядів. Значення цільової статистики залежать від історії спостережень, тобто цільове значення для поточного об'єкта обчислюється тільки з серії спостережень перед ним [4].

Після цього дані були стандартизовані, щоб можна було використовувати алгоритми, які залежать від масштабу ознак, зокрема, різні варіації нейронних мереж.

3.2. Метрики для оцінки якості прогнозу

Для перевірки точності моделі прогнозування в цьому дослідженні використовуються такі методи оцінювання, як середня абсолютна похибка (MAE), зважене зміщення (wBias) та зважена середня абсолютна відсоткова похибка (wMAPE) [5]. Вони визначаються за наступними формулами:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_{true,i} - y_{pred,i}|$$

$$wBias = \frac{\sum_{i=1}^n (y_{true,i} - y_{pred,i}) \times weight_i}{\sum_{i=1}^n y_{true,i} \times weight_i}$$

$$wMAPE = \frac{\sum_{i=1}^n |y_{true,i} - y_{pred,i}| \times weight_i}{\sum_{i=1}^n y_{true,i} \times weight_i}$$

де $y_{true,i}$ – істинне значення i -го спостереження, $y_{pred,i}$ – прогнозоване значення для i -го спостереження, а $weight_i$ - реалізована ціна i -го спостереження.

3.3. Результати

У таблицях 1–3 наведено оцінки якості реалізованих алгоритмів пакування та інших алгоритмів на тестовій вибірці для різних категорій товарів.

Таблиця 1. Результати по тестовому зразку для категорії оливки

Алгоритм	MAE	wBias	wMAPE
Linear Regression	21,240999	0,171258	2,089721
Decision Tree	20,523671	0,153452	2,099562
KNN	14,37258	0,002988	1,464464
Light Gradient Boosting Machine	10,265612	0,016345	1,015855
MLP	10,196435	0,023553	0,983567
Запропонований беггінг: MLP	10,042824	0,016544	0,964849

Таблиця 2. Результати по тестовому зразку для категорії вода

Алгоритм	MAE	wBias	wMAPE
Linear Regression	84,480005	0,425613	1,390737
Decision Tree	74,053743	0,685274	0,687261
KNN	69,532563	0,22464	0,265432
Light Gradient Boosting Machine	66,621567	0,11069	0,228842
MLP	65,466542	0,14347	0,202356
Запропонований беггінг: MLP	65,023453	0,10353	0,188464

Таблиця 3. Результати по тестовому зразку для категорії сир

Алгоритм	MAE	wBias	wMAPE
Linear Regression	22,41161	0,583439	1,174249
Decision Tree	26,370057	0,688346	1,373164
KNN	15,777434	0,315182	0,819989
Light Gradient Boosting Machine	9,615076	0,113314	0,494714
MLP	9,605615	0,146724	0,482369
Запропонований беггінг: MLP	9,589425	0,127032	0,482391

Погляньмо на агреговані прогнози для розробленого алгоритму, які показані на рисунку 3 (тижні по осі X, загальні продажі по осі Y). Ми бачимо, що модель в цілому досить добре відстежує зміни рівня продажів, що є критичним при створенні моделі, і немає сильного зміщення.

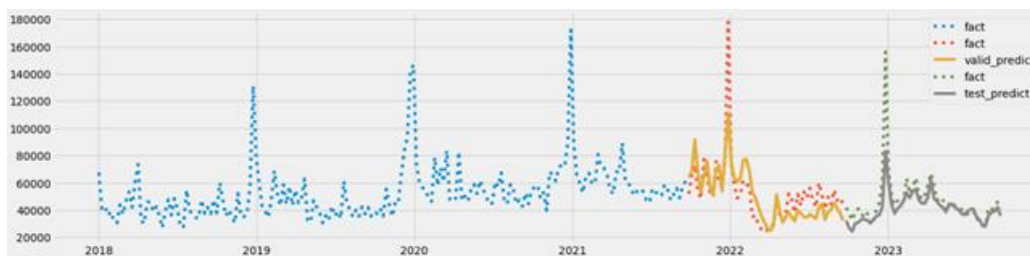


Рисунок 3. Прогнозування на валідаційній та тестовій вибірці

4. ВИСНОВКИ

У цій науковій роботі ми представляємо новий метод довгострокового прогнозування попиту на основі ансамблю нейронних мереж, який враховує новизну даних. Наш підхід використовує метод мішків і техніку модифікації, що дозволяє враховувати актуальність і новизну даних при генерації навчальних вибірок для кожної моделі в ансамблі.

Результати наших експериментів показали, що запропонований метод перевершує наявні підходи до довгострокового прогнозування попиту. Результати підкреслюють важливість врахування новизни даних при розробці стратегій прогнозування попиту.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Aaltonen K. Project stakeholder analysis as an environmental interpretation process. *International journal of project management*. 2011. Т. 29, № 2. С. 165–183. URL: <https://doi.org/10.1016/j.ijproman.2010.02.001> (дата звернення: 20.11.2023).

2. Perin G., Chmielewski Ł., Picek S. Strength in numbers: improving generalization with ensembles in machine learning-based profiled side-channel analysis. *IACR transactions on cryptographic hardware and embedded systems*. 2020. С. 337–364. URL: <https://doi.org/10.46586/tches.v2020.i4.337-364> (дата звернення: 20.11.2023).

3. The uncertainty estimation of feature-based forecast combinations / X. Wang та ін. *Journal of the operational research society*. 2021. С. 1–15. URL: <https://doi.org/10.1080/01605682.2021.1880297> (дата звернення: 20.11.2023).

4. Sales forecasting based on catboost / J. Ding та ін. 2020 2nd international conference on information technology and computer application (ITCA), м. Guangzhou, China, 18–20 груд. 2020 р. 2020. URL: <https://doi.org/10.1109/itca52113.2020.00138> (дата звернення: 20.11.2023).

5. Hyndman R. J., Athanasopoulos G. *Forecasting: principles and practice*. OTexts, 2021

ПОШУК ВІДПОВІДНОСТІ МІЖ ЗОБРАЖЕННЯМ І ЙОГО ТЕКСТОВИМ ОПИСОМ

Шаповал Н.В.¹, Крижанівська О.В.²

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

¹ shovgun@gmail.com, ² kruzhanivska.olena212@gmail.com

В сучасному інформаційному світі, коли обсяги доступної інформації надзвичайно великі, виникає необхідність розробки та вдосконалення методів автоматичної обробки текстової і графічної інформації. Однією з актуальних задач є встановлення відповідності між зображеннями і мовними описами, що відкриває безліч можливостей для розвитку різних застосувань, включаючи покращення пошукових систем, автоматичне розпізнавання зображень та підтримку людей з обмеженими можливостями. Відповідність між графічною та текстовою інформацією важлива для багатьох сучасних додатків і систем, таких як пошукові системи, розпізнавання образів, аналіз медичних даних і багато інших. Метою дослідження є запропонувати модифіковану модель такого типу. Це дозволить покращити якість та швидкість роботи інформаційних систем, що базуються на аналізі інформації.

Ключові слова: мультимодальна обробка інформації, Image-Text matching, автоматична обробка текстової та графічної інформації, модифікація нейронних мереж, моделі розпізнавання образів.

1. ВСТУП

Сучасна епоха інформаційних технологій та глибокого навчання свідчить про стрімкий розвиток інформаційних систем та обробки даних. Існує потреба у моделях, які дозволяють ефективно працювати з інформацією, незалежно від її формату. Окрім того, в сучасному інформаційному світі, коли обсяги доступної інформації надзвичайно великі, виникає необхідність розробки та вдосконалення методів автоматичної обробки текстової і графічної інформації. Проте суспільство стикається зі значним обсягом інформації, представленої в різних форматах, включаючи текст, зображення та відео. Здійснення ефективної обробки та аналізу цих різних видів інформації стає важливою задачею в контексті пошукових систем, автоматизованих систем ідентифікації, медицини, автономної навігації, відеоспостереження, робототехніки, створення інтерактивних мультимедійних додатків, інформаційної безпеки та інших областях. Однією з актуальних задач є встановлення відповідності між зображеннями і мовними описами, що відкриває безліч можливостей для розвитку різних застосувань, включаючи покращення пошукових систем, автоматичне розпізнавання зображень та підтримку людей з обмеженими можливостями. Знаходження взаємозв'язків між зображеннями та текстовим описом має широкі застосування, починаючи від систем пошуку та індексації великих обсягів мультимедійних даних до розвитку систем розпізнавання образів та інтелектуального пошуку в онлайн-ресурсах. Ця проблема стає критичною в таких галузях, як електронна комерція, медична діагностика, туризм та соціальні мережі, де точне розуміння контенту, представленого зображеннями та текстовими описами, може значно поліпшити користувацький досвід та ефективність рішень. Отже, розробка методів та алгоритмів, які

забезпечують високий рівень відповідності між зображеннями та текстовим описом, є актуальною задачею.

2. IMAGE-TEXT MATCHING МОДЕЛІ

Існує багато нейронних мереж, які дозволяють вирішити дану задачу. В даному дослідженні були розглянуті нижче приведені моделі.

SAM. Модель SAM – це мультимодальна модель, яка вирішує завдання швидкої сегментації на основі текстових підказок і може бути адаптована під задачу Image-Text matching [1]. Задачею моделі є сегментація зображення, семантична сегментація, візуальна підказка. Набір даних, на якому навчалась модель SA-1B [1]. Він складається з 11 мільйонів різноманітних зображень з високою роздільною здатністю, ліцензованих і захищених від несанкціонованого доступу, а також 1,1 мільярда високоякісних масок сегментації, зібраних за допомогою нашого рушія даних. Що відносно архітектури, то завдання швидкої сегментації та мета реального використання накладають обмеження на неї. Зокрема, модель повинна підтримувати гнучкі підказки, обчислювати маски в амортизованому режимі реального часу, щоб забезпечити інтерактивне використання, і враховувати неоднозначність. Виявлено, що простий дизайн задовольняє усім трьома обмеженнями: потужний кодер зображень обчислює вбудовування зображень, кодер підказок вбудовує підказки, а потім ці два джерела інформації об'єднуються в легкому декодері масок, який прогнозує маски сегментації (Рис. 1) [1].

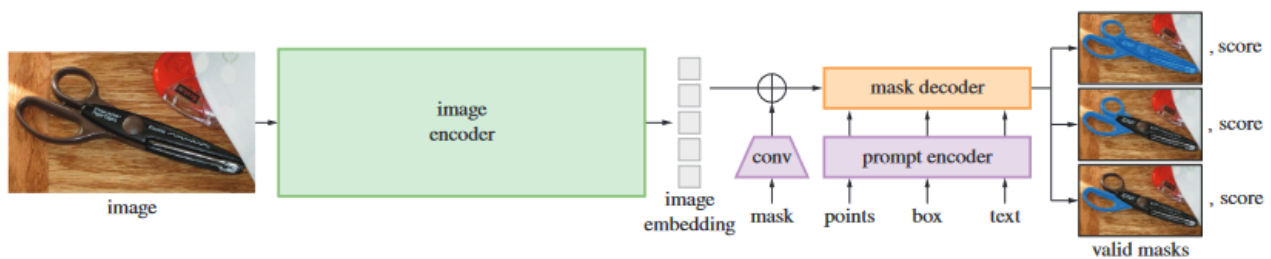


Рисунок 1. Архітектура моделі SAM

Переваги та недоліки:

- немає вбудованих зображень;
- немає підтримки української мови;
- проект не open source;
- + модель навчена на багатьох датасетах;
- + достатня точність.

EVA. Модель EVA використовується для багатьох задач, таких як детекція об'єктів, семантичної сегментації, класифікації зображень, тощо. Мультимодальна модель, яка вводить зображення та одну або декілька підказок і виводить правильну маску сегментації. Окрім цього, слід зауважити, що SAM – state of the art модель [2]. В своїй основі вона використовує модель CLIP. EVA – це Vanilla ViT, попередньо навчений реконструювати замасковані зображення-текст, вирівняні за візуальними ознаками, на основі видимих ділянок зображення. EVA масштабована до одного мільярда параметрів і встановила певні рекорди в широкому спектрі репрезентативних задач, таких як розпізнавання зображень, розпізнавання відеодій, виявлення об'єктів, сегментація екземплярів. При навчанні було використано в якості набору даних CC12M і CC3M (дані зображень без підписів), COCO і ADE20K (дані навчальних наборів), ImageNet-21K і Object365 (дані зображень) [2]. Об'єднаний набір даних для попереднього навчання містить загалом 29,6 мільйонів зображень.

Переваги та недоліки:

- низька точність (класифікація 75,7%, детекція та сегментація 53,4%) [2];
- важко інсталювати та використовувати;
- наразі багато багів;
- немає підтримки української мови;
- + проект open source.

GLIDE. GLIDE модель - це генеративна модель на основі стабільної дифузії призначена для генерації тексту в зображення та навпаки з тексту в зображення [3]. Також використовується модель CLIP. Завдання: генерація зображень, зафарбовування зображень, генерація тексту в зображення, генерація тексту в зображення з нульового кадру. Використаний набір даних – COCO [3]. Модель абляційної дифузії (ADM), навчена генерувати зображення 64 x 64 (ImageNet), текстова модель (трансформатор), яка впливає на генерацію зображення за допомогою текстової підказки, і модель апсемплінгу, яка перетворює наші маленькі зображення 64 x 64 на більш зрозумілі 256 x 256 пікселів [3]. Перші два компоненти взаємодіють один з одним, щоб керувати процесом генерації зображень для точного відображення текстової підказки, а останній необхідний для того, щоб зробити зображення, які ми створюємо, більш легкими для інтерпретації.

Переваги та недоліки:

- немає вбудованих зображень;
- на виході отримуємо зображення лише у розмірі 256px;
- модель навчена лише на одному датасеті;
- проект не open source;
- + є підтримка української мови.

CLIP. CLIP (Contrastive Language-Image Pre-Training) – це мультимодальна модель, яка вивчає відповідність між природною мовою та зображеннями [4]. Завдання моделі наступні: розпізнавання дій, бенчмаркінг, класифікація зображень, класифікація зображень, навчання з довгим хвостом, класифікація мемів, розуміння природної мови, категоризація об'єктів, виявлення атрибутів відкритого словника, узагальнення поза розподілом, швидка інженерія, класифікація зображень під наглядом, генерація тексту, крос-модальний пошук з нульовим результатом, навчання з нульовим результатом, класифікація зображень з нульовою передачею. У ході навчання мережі було використано велику кількість наборів даних, серед них ImageNet, COCO, KITTI, GLUE, SST, UCF101, Visual Genome, Flickr30k, SUN397 та 22 інші [4]. Відносно архітектури, автори використали дві різні основи (Resnet50 і Vision Transformer (ViT)) для кодера зображень і трансформатор як основу для кодера тексту (Рис. 2) [4].

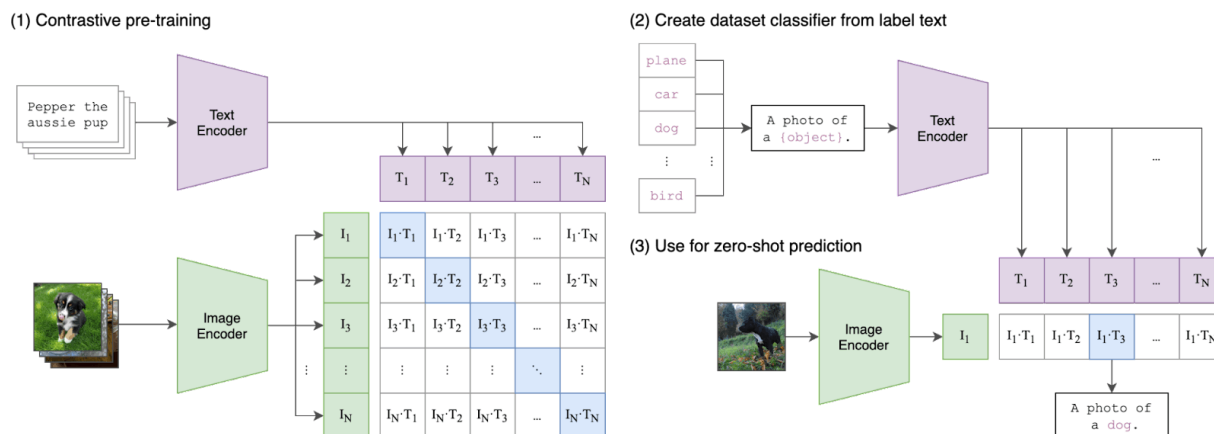


Рисунок 2. Модель CLIP

Переваги та недоліки:

- немає підтримки української мови;
- низька точність (image retrieval flickr30k 68,7%, mscoco 37,8%, text retrieval flickr30k 88%, mscoco 58,1%);
- + є вбудовані зображення;
- + модель навчена на багатьох датасетах;
- + легко інсталиювати та використовувати.

Аналізуючи ці рішення, було вирішено використовувати модель CLIP, оскільки вона є базою для інших рішень і може потім бути адаптована під інші задачі. В моделі CLIP завдання полягає в пошуку максимально близького текстового опису для зображень. Навчена модель здатна розпізнавати предмети, навіть якщо вони представлені в зовсім іншому контексті. CLIP містить дві мережі. Одна мережа повинна обробляти зображення та створювати векторну репрезентацію на основі цих зображень, тоді як інша мережа має обробляти текст та створювати векторну репрезентацію тієї ж розмірності, що й перша мережа. Оскільки було б неможливо вибрати дві нейронні мережі з різних галузей (CV і NLP), які мали б однакові вихідні характеристики, до обраної архітектури слід додати шар, який приведе їхні розмірності до єдиної величини. Ці шари можна назвати лінійними перетвореннями для забезпечення відповідності розмірностей.

3. ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ

У ході дослідження була використана модель CLIP як основа для вивчення ключових показників. В якості датасету використано власноруч українізований набір даних Flickr з 30 тисячами екземплярів тренувальних та валідаційних пар зображення та текстового опису. Початкові результати за перші чотири епохи тренування показали втрати на рівні 7,8% під час тренування та 9,4% під час валідації. Ці показники слугували вихідною точкою для подальших оптимізацій та вдосконалень. Для підвищення ефективності моделі, було внесено зміни у процес тренування. Оптимізатор Adam був замінений на AdamW, а потім AdamW був замінений на AdaBelief, що включає адаптивне відсікання градієнту, а також додано планувальник темпу навчання з подальшим коригуванням гіпер-параметрів всієї мережі. Планувальник темпу навчання в даному випадку грає роль механізму, який визначає, як швидко параметри моделі оновлюються під час процесу тренування. Темп навчання визначає величину кроку або кроків, які використовуються для оновлення ваг моделі під час зворотного поширення під час тренування. Роль планувальника темпу навчання полягає в оптимізації

процесу тренування, забезпеченні стійкості та швидкості збіжності моделі. Після цих модифікацій знову було проведено тренування і нові результати вже за перші чотири епохи показали помітне покращення: втрати під час тренування зменшилися до 1,14%, а втрати під час валідації склали 2,28%. Більш детальна інформація наведена нижче (Табл. 1).

Таблиця 1. Порівняння втрат моделей без модифікацій та з модифікаціями

CLIP+Adam		CLIP+AdamW		CLIP+AdaBelief+ACG	
втрати під час тренування	втрати під час валідації	втрати під час тренування	втрати під час валідації	втрати під час тренування	втрати під час валідації
33,3	31,3	3,12	2,67	2,58	2,38
14,03	15,4	2,05	2,63	1,51	2,3
9,2	10,9	1,81	2,6	1,27	2,29
7,8	9,4	1,68	2,57	1,14	2,28

4. ВИСНОВОК

Експеримент демонструє, що заміна оптимізатора на AdaBelief з адаптивним відсіканням градієнту та додавання планувальника темпу навчання з подальшим коригуванням гіперпараметрів, суттєво покращили продуктивність моделі CLIP. Зменшення втрат під час тренування та валідації свідчить про ефективність нових стратегій оптимізації та навчання. Це не лише підкреслює значущість правильного вибору оптимізаційних методів, але й вказує на потенційні можливості подальшого вдосконалення моделей для вирішення складних завдань.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Alexander Kirillov: Segment Anything [Електронний ресурс] – Режим доступу: <https://arxiv.org/pdf/2304.02643.pdf>
2. Yuxin Fang: EVA: Exploring the Limits of Masked Visual Representation Learning at Scale [Електронний ресурс] – Режим доступу: <https://arxiv.org/pdf/2211.07636.pdf>
3. Alex Nichol: GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models [Електронний ресурс] – Режим доступу: <https://arxiv.org/pdf/2112.10741.pdf>
4. Alec Radford: Learning Transferable Visual Models From Natural Language Supervision [Електронний ресурс] – Режим доступу: <https://arxiv.org/pdf/2103.00020.pdf>