КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА
Факультет комп'ютерних наук та кібернетики
Кафедра обчислювальної математики

Кваліфікаційна робота
на здобуття ступеня бакалавра
за спеціальністю 113 Прикладна математика на тему:

Прогнозування вартості акцій методами машинного навчання

Виконала студентка 4-ого курсу:
Буря Діана Тарасівна _____

Науковий керівник:
Асистент кафедри обчислювальної математики
Денисов Сергій Вікторович _____

Засвідчую, що в цій роботі немає запозичень з праць інших авторів без відповідних посилань
Студентка _____

Роботу розглянуто й допущено до захисту на засіданні кафедри обчислювальної математики
29 травня 2023 р.,
протокол № 8
Завідувач кафедри
проф. Ляшко С.І. _____

Київ - 2023

ЗМІСТ

АНОТАЦІЯ

Дипломна робота складається зі вступу, 2 розділів, висновку, списку використаних джерел (18 найменувань). Загальний обсяг роботи становить 31 сторінок, основний текст роботи викладено на 27 сторінках.

Ключові слова: нейронна мережа, рекурентна нейронна мережа, прогнозування, машинне навчання, ціна акції

Реферат. В даній роботі проведене прогнозування цін акцій за допомогою нейронних мереж. Для початку були визначені вхідні дані та параметри, що потрібні для навчання мережі. Далі було побудовано прогнози двома методами, а саме за допомогою нейронних мереж та рекурентних нейронних мереж. Також було проведено порівняння точності цих методів за допомогою оцінки MSE(середньої квадратичної помилки) та зроблено висновки.

Key Words: neural network, recurrent neural network, forecasting, machine learning, stock price

Abstract. In this work, stock price forecasting was carried out using neural networks. To begin with, the input data and parameters required for network training were determined. Next, predictions were made using two methods, namely using neural networks and recurrent neural networks. A comparison of the accuracy of these methods was also carried out using the MSE (mean squared error) estimate, and conclusions were drawn.

ВСТУП

У сучасному світі фінансові ринки мають значний вплив на економічну стабільність країн та підприємства. Отримання точних та надійних прогнозів щодо зміни цін акцій є важливим завданням для інвесторів та фінансових аналітиків. На сьогоднішній день, у зв'язку зі зростанням доступності великих обсягів фінансових даних та розвитком штучного інтелекту, нейронні мережі стають потужним інструментом для прогнозування ціни акцій.

Мета даного дослідження полягає в розробці, застосуванні та порівнянні двох типів побудови нейронної мережі на основі історичних фінансових даних та фінансових показників компанії.

У порівнянні з класичними методами для прогнозування ціни, нейронні мережі можуть здійснювати глибше та складніше аналітичне моделювання, використовуючи велику кількість вхідних змінних та автоматично виявляючи нелінійні залежності між ними. Нейронна мережа може "навчитися" наявним фінансовим даним та здатна робити прогнози на основі цих знань, а також автоматично коригувати свої ваги та параметри при зміні умов ринку.

В цій роботі буде розглянута побудова нейронної мережі та рекурентної нейронної мережі. Будуть порівняні результати прогнозування цими двома методами.

Однак, варто відзначити, що прогнозування цін акцій за допомогою нейронних мереж також має свої обмеження. Фінансові ринки піддаються впливу багатьох непередбачуваних факторів, таких як геополітичні події, економічні зміни, новини та інші зовнішні фактори, які можуть значно вплинути на ціну акцій. Прогнози не завжди можуть бути точними і надійними в умовах непередбачуваних подій.

Все ж, це залишається доволі актуальною та перспективною темою для досліджень.

РОЗДІЛ 1
ПОПЕРЕДНЯ ОБРОБКА ДАНИХ

1.1 Визначення параметрів

Вхідні дані для навчання нейронної мережі, яка прогнозує ціни акцій, можуть включати різноманітні фактори, які можуть впливати на ціни акцій. Основні типи вхідних даних, які можна використовувати, включають:

1. Історичні ціни акцій: Це дані про ціни акцій у попередній період. Можна використовувати дані за певний період, такі як ціни закриття, мінімальні та максимальні ціни, обсяг торгів тощо. В цій роботі була взята ціна акцій, а саме ціна закриття за кожен день впродовж року компанії Google[1], починаючи з 2022.05.31 по 2023.05.24.

2. Фінансові показники: Це дані про фінансовий стан компанії, які можуть включати такі показники, як прибуток, чистий прибуток, відсоток рентабельності, співвідношення P/E (Price-to-Earnings) та інші показники, які відображають фінансову продуктивність компанії.

3. Макроекономічні показники: Це дані про загальний економічний стан, які можуть включати показники такі як ВВП, безробіття, індекси споживчої ціни, ставки процентів, політичні фактори тощо. Ці показники можуть впливати на ринок загалом та ціни акцій зокрема. В даній роботі були взяті декілька таких показників для розуміння економічного стану, а саме ефективна ставка федеральних фондів США(EFFR)[7], рівень безробіття[8] та індекс споживчих цін(CPI)[9], що характеризує зміни загального рівня цін на товари та послуги, які купує населення для невиробничого сподивання. В цій роботі я взяла саме ринок США, адже саме ця країна є лідером світової економіки[10], то ж багато в чому економіка світу залежить від економіки США.

4. Технічний аналіз: Це дані, які використовуються для вивчення патернів та трендів на ринку. Вони можуть включати індикатори, такі як середня рухома, індекс сили ринку, індикатори перекупленості/перепроданості тощо.

Для навчання нейронної мережі я буду використовувати такі індикатори як

1. RSI(Relative Strength Index, відносний індекс сили)[11]: технічний аналітичний інструмент, що використовується в інвестиційному аналізі для оцінки перевищення або падіння цін активу. Він допомагає трейдерам та інвесторам визначити ступінь перекуплення або перепроданості акцій, ф'ючерсів, валют або інших фінансових інструментів. RSI розраховується за формулою, яка враховує середню прибутковість і середню збитковість активу за певний період часу. Цей період часу, зазвичай, беруть в 14 днів.

   RSI = 100 - (100 / (1 + (avg_gains / avg_losses))), (1.1.1)

   де avg_gains – середній приріст ціни за певний період (зазвичай 14 днів); avg_losses - середнє зниження ціни за певний період (зазвичай 14 днів).

   RSI має діапазон від 0 до 100. Зазвичай, значення вище 70 вказують на перекупленість активу, що може свідчити про можливе послаблення ціни. Значення нижче 30 вказують на перепроданість активу, що може означати можливе зростання ціни.

2. MACD (Moving Average Convergence Divergence, збіжність-розбіжність середніх рухів)[12]: технічний аналітичний індикатор, що використовується в інвестиційному аналізі для виявлення змін тренду та генерації сигналів купівлі або продажу фінансових інструментів, таких як акції, ф'ючерси, валюти тощо. MACD розраховується шляхом віднімання швидкої

експоненціальної середньої (EMA) від повільної EMA. Зазвичай використовуються значення EMA за 12 та 26 періодів.

3. Stochastic Oscillator (стохастичний осцилятор)[13]: технічний аналітичний індикатор, який використовується в інвестиційному аналізі для оцінки перекуплення або перепроданості активу та визначення можливих змін тренду ціни. Він базується на порівнянні поточної ціни активу з діапазоном цін за певний період часу. Стохастичний осцилятор складається з двох ліній - %K та %D. Лінія %K відображає відношення поточної ціни активу до діапазону цін за вибраний період. Лінія %D представляє собою просту рухому середню (зазвичай за 3 періоди) лінії %K і допомагає згладити коливання. В даній роботі буде використовуватись тільки дані з лінії %K. Стандартний період, що використовується для стохастичного осцилятора, - це 14 періодів.

$$S\%K = 100 * (C - L) / (H - L), \quad (1.1.2)$$

де C – поточна ціна, L - мінімальне значення ціни попередньому періоді даних (за 14 днів), H – максимальне значення ціни в попередньому періоді даних.

Стохастичний осцилятор має діапазон від 0 до 100. Значення понад 80 вказують на перекупленість активу, що може свідчити про можливе зниження ціни. Значення менше 20 вказують на перепроданість активу, що може означати можливе підвищення ціни. Трейдери використовують стохастичний осцилятор для виявлення перекупленості або перепроданості активу і використовують ці дані як сигнал для входу або виходу з ринку.

4. ADI технічний аналіз (Accumulation/Distribution Index)[14] - це індикатор, що використовується в технічному аналізі фінансових

ринків. Він допомагає визначати сильність та напрямок цінової тенденції активу шляхом оцінки акумуляції або дистрибуції. ADI базується на принципі, що великий обсяг торгівлі вказує на активну участь учасників ринку і сигналізує про силу тенденції. Індикатор обчислюється шляхом акумуляції (додавання) або дистрибуції (віднімання) обсягу за певний період часу. Позитивне значення ADI вказує на позитивну акумуляцію (покупку), тобто переважну активність покупців. Негативне значення ADI означає негативну дистрибуцію (продаж), тобто переважну активність продавців. Коли ціна активу зростає, а ADI також підтверджує позитивну акумуляцію, це може вказувати на зміцнення позиції купців і підтримувати тенденцію зростання. Зворотність ситуації може вказувати на можливе змінення тенденції.

Також була проведена попередня обробка даних та сформований датасет для зручності навчання нейронної мережі.

Варто зазначити, що розраховуючи індикатор MACD, було видалено перші 26 значень, адже вони потрібні були потрібні тільки для розрахунку. Тому тепер наші дані будуть з 2022.07.11 по 2023.05.24:

Date	Unemployment Rate	EFFR	CPI	Close price	Macd	S%K	RSI	ADX
11/07/2022	3.5	1.58	296.276	116.5225	1.006581	67.116122	62.415742	113.980229
12/07/2022	3.5	1.58	296.276	114.8495	1.517395	52.027057	54.376052	114.042320
13/07/2022	3.5	1.58	296.276	112.1870	0.955223	28.013529	50.218384	113.909797
14/07/2022	3.5	1.58	296.276	111.4400	0.743217	21.276212	48.228545	113.733383
15/07/2022	3.5	1.58	296.276	112.7670	1.124159	33.244645	40.570261	113.664356
...	...	...	...	...	...	...	...	...
18/05/2023	3.7	5.08	309.350	123.5200	3.013370	100.000000	81.716418	110.978640
19/05/2023	3.7	5.08	309.350	123.2500	4.011908	98.525396	82.537688	111.855165
22/05/2023	3.7	5.08	309.350	125.8700	5.403053	100.000000	90.149374	112.856225
23/05/2023	3.7	5.08	309.350	123.2900	6.542981	87.512101	81.550900	113.601495
24/05/2023	3.7	5.08	309.350	121.6400	7.403536	78.478759	79.391771	114.175674

221 rows × 8 columns

(Мал. 1) Отриманий датасет

1.2 Кореляція

Визначимо, чи є залежність у ціни акції Google до фінансових показників та ціни акції, але взятої на один день раніше.

Був застосований коефіцієнт кореляції Пірсона[15]. Коефіцієнт кореляції Пірсона (або просто кореляція Пірсона) - це статистичний показник, який використовується для вимірювання ступеня лінійної залежності між двома змінними. Він вказує на те, наскільки сильно та у якому напрямку змінюються дві змінні разом. Кореляція Пірсона вимірюється від -1 до 1, де значення -1 вказує на повну негативну кореляцію (зміна однієї змінної супроводжується зворотною зміною іншої змінної), значення 1 вказує на повну позитивну кореляцію (зміна однієї змінної супроводжується однаковою зміною іншої змінної), а значення 0 вказує на відсутність лінійної залежності між змінними.

Кореляція Пірсона обчислюється за формулою, яка враховує коваріацію між двома змінними та їх стандартні відхилення. Формула виглядає наступним чином:

$$r = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}}$$

(Мал.2) Формула обчислення коефіцієнту Пірсона

де r - коефіцієнт кореляції,

x_i - значення змінної x у вибірці,

$\bar{x}$ - середнє значення змінної x,

y_i = значення змінної y у вибірці,

$\bar{y}$ = середнє значення y-змінної.

Існують різні ступені кореляції, в залежності від яких визначається чи значущий взаємозв'язок між вибраними даними чи ні.

1. якщо 1 > |r| > 0.5 зв'язок значущий
2. якщо 0.5 > |r| > 0.3 - зв'язок середньої значущості
3. якщо 0.3 > |r| > 0 - зв'язок слабкий

```
Unemployment Rate 0.3425731559519778
EFFR -0.32817365037686436
CPI 0.21873248737753515
Close price 0.9682779683956235
Macd 0.38856404770640884
S%K 0.386085847721254
RSI 0.47114564726993213
ADX 0.8157730290775635
```

(Мал. 3) Кореляція між ціною та фінансовими показниками

Як бачимо на мал.3 очікувано сильну кореляцію має ціна акції Google та зміщена ціна акції Google на 1 день назад. Також сильний зв’язок між ціною акції та ADX. Зв’язок середньої значущості має ціна акції з такими фінансовими показниками як індекс безробіття, EFFR, MACD, стохастичний осцилятор та RSI. Найслабкіший зв’язок вона має з CPI. Тому в подальшому ми можемо тренувати нейронну мережу без цього показника.

РОЗДІЛ 2
ПОБУДОВА НЕЙРОННОЇ МЕРЕЖІ

2.1 Теоретична частина. Нейронна мережа

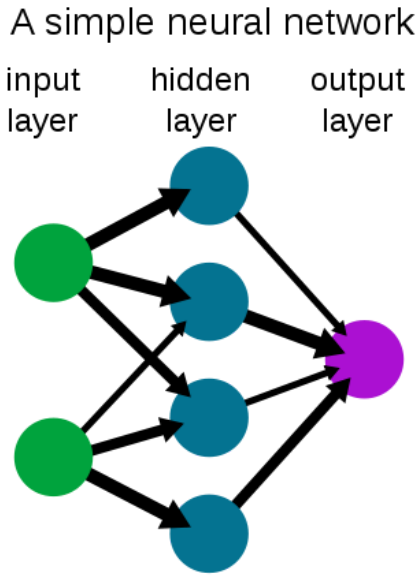
Нейронні мережі, також відомі як штучні нейронні мережі або глибинні нейронні мережі, є моделлю обчислення, натхненною роботою людського мозку. Вони здатні вчитися на основі великих обсягів даних і застосовувати ці знання для вирішення різноманітних задач.

Теоретична основа для побудови сучасної нейронної мережі була покладена Олександром Бейном ще у 1873[16]. В його дослідженнях було встановлено, що активність тіла і думки виникають внаслідок взаємодії нейронів у мозку. За теорією Бейна будь-яка діяльність спричиняє активацію конкретного набору нейронів, а повторення цих дій зміцнює зв'язки між цими нейронами. Таке повторення і призводить до формування пам'яті.

В 1943 році МакКалох та Піттс вперше представили модель нейрона, яка була базовим блоком для розуміння нейронних мереж[16]. Пізніше, в 1958 році Франк Розенблатт запропонував персептрон - перший штучний нейронний шар, який міг вчитися і розпізнавати прості образи. Основною проблемою в подальшому вивченні нейронних мереж були обмеження обчислювальними можливостями тогочасних комп’ютерів, відсутність великих наборів даних та обмеженим розумінням того, як працюють глибокі нейронні мережі.

Перейдемо до теоретичного пояснення як працює нейронна мережа. Вона складається зі з'єднаних штучних нейронів, які працюють разом, обробляючи вхідні дані і генеруючи вихідні результати. Основна структурна одиниця нейронної мережі - це нейрон. Кожен нейрон отримує вхідні сигнали,

обробляє їх і генерує вихідний сигнал. Нейрони згруповані у шари - вхідний шар, приховані шари і вихідний шар.



(Мал. 4) Спрощений вид на штучну нейронну мережу

Процес роботи нейронної мережі можна узагальнити в такі кроки:

1. Вхідний шар: Кожен нейрон у вхідному шарі отримує вхідні дані або ознаки. Позначимо їх як x_i.

2. Ваги: Кожному вхідному значенню призначаються ваги, які відображають його важливість для роботи нейрона. Ваги - це параметри, які навчаються під час процесу навчання моделі. Позначимо їх як w_i.

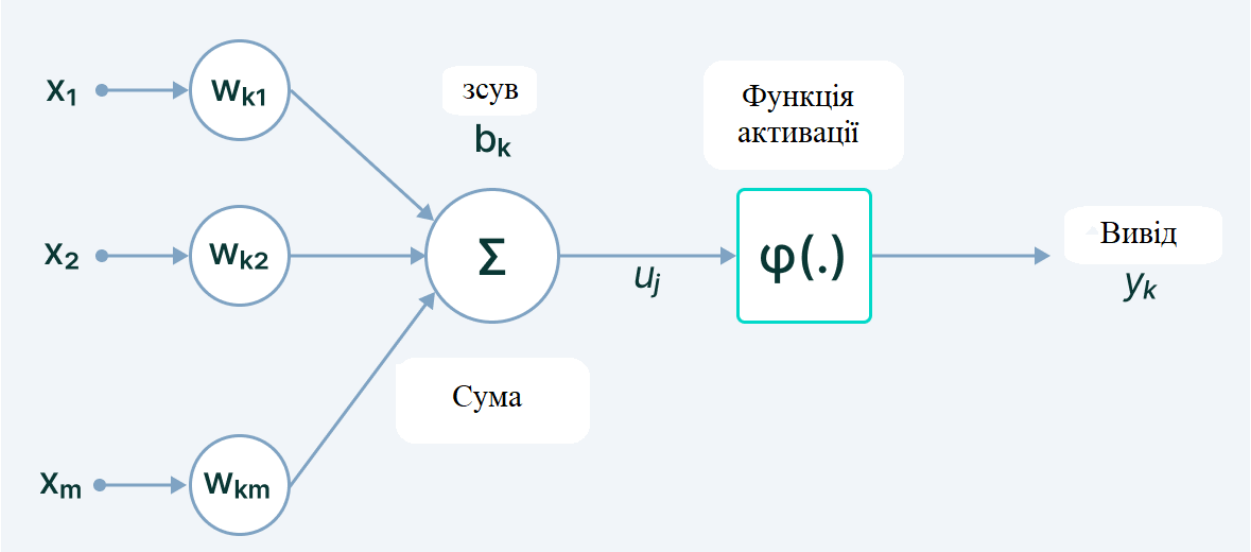
3. Взважена сума: Нейрон обчислює взважену суму вхідних значень, перемножаючи кожне вхідне значення на відповідну вагу та додаючи їх разом.

4. Функція активації: Отримане значення зваженої суми піддається функції активації. Я використовувала функцію сигмоїда. Вона приводить значення до діапазону між 0 і 1, використовуючи формулу:

$$\sigma(x) = 1 / (1 + e^{-x}), \qquad (2.1.1)$$

де x - зважена сума вхідних значень.

5. Вихід: Отримане значення після застосування функції активації стає вихідним значенням нейрона. Воно може використовуватися для подальших обчислень, передачі інформації до наступного шару нейронної мережі або як вихідна інформація моделі.



(Мал.5) Побудова нейрону

2.2 Реалізація нейронної мережі

Для побудови нейронної мережі було взято 7 відібраних за допомогою кореляції вхідних параметрів(x_1, x_2, x_3, x_4, x_5, x_6, x_7), які були описані вище у розділі 1.1. Кількість схованих нейронів може змінюватись. Позначимо нейрон на схованому шарі як h_i та розпишемо як рахується його значення. Кожен схований нейрон має свої ваги та зсув, які будуть використовуватись для обчислення його значення.

Для початку перемножуються вхідні значення на ваги:

$$x_i \cdot w_i \tag{2.2.1}$$

Потім ці значення додаються і утворюється сума

$$z_i = x_1 w_1 + x_2 w_2 + ... + x_n w_n + b_i, \quad (2.2.2)$$

де b_i - є зсувом для схованого нейрону h_i.

Далі застосуємо функцію активації:

$$h_i = \sigma(z_i), \quad (2.2.3)$$

де $\sigma(z_i)$ - функція активації сигмоїда.

Алгоритм, описаний вище, має назву пряме поширення помилки (forward propagation).

Тепер опишемо алгоритм, який буде оновлювати ваги застосовуючи значення помилки вихідних результатів. Для цього нам потрібно визначити, як ми будемо рахувати помилку, адже за допомогою неї ми будемо коригувати ваги.

Для того, щоб оцінити точність прогнозування побудованої нейронної мережі, застосуємо MSE. MSE є скороченням від "Mean Squared Error" (середньоквадратична похибка). Це один з найпоширеніших методів оцінити ефективність моделі машинного навчання. Використовується для вимірювання відповідності прогнозованих значень моделі до фактичних даних. MSE обчислюється шляхом взяття різниці між прогнозованими значеннями моделі та відповідними фактичними значеннями, підносить цю різницю до квадрата, а потім обчислює середнє значення цих квадратів. Формульно, MSE можна виразити таким чином:

$$MSE = 1/n \sum_{i=1}^{n} (y_i - \hat{y}_i)^2, \quad (2.2.4)$$

де MSE - середньоквадратична похибка,

n - кількість прикладів у наборі даних,

y_i - фактичне значення для і-го прикладу,

$\hat{y}_i$ - прогнозоване значення для і-го прикладу.

Чим менше значення MSE, тим краще модель будує прогнози.

Тепер перейдемо до коригування ваг. Зворотне поширення помилки (Backpropagation) - це алгоритм, що використовується для оновлення ваг нейронної мережі на основі помилки вихідних результатів. Він працює у зворотному напрямку, починаючи з вихідного шару і поширюючи помилку назад до початкових шарів, щоб мінімізувати помилку передбачення мережі.

Припустимо, ми маємо нейронну мережу з вихідним значенням y і очікуваним виходом t. Цільова функція для вимірювання помилки позначимо як E. Завдання полягає в тому, щоб оновити ваги нейронної мережі, щоб мінімізувати помилку E.

Процес зворотного розповсюдження можна розділити на такі кроки:

1. Обчислюємо помилку вихідного шару, порівнюючи вихідне значення y з очікуваним значенням t:

$$\delta = \partial E / \partial y, \tag{2.2.5}$$

де δ - помилка вихідного шару,

$\partial E/\partial y$ - похідна цільової функції за вихідним значенням.

2. Розповсюджуємо помилку назад до попередніх шарів мережі. Для кожного нейрона в попередньому шарі обчислюємо його помилку δ_i, використовуючи помилку поточного шару та ваги, пов'язані з нейронами в поточному шарі:

$$\delta_i = \partial E / \partial z_i = \sum \delta_j w_{ji}, \tag{2.2.6}$$

де δ_i - помилка нейрона в попередньому шарі,

$\partial E/\partial z_i$ - похідна цільової функції за виваженою сумою нейрона,

δ_j - помилка поточного шару,

w_{ji} - вага зв'язку між нейронами в поточному та попередньому шарах.

3. Оновлюємо ваги нейронів, щоб зменшити помилку. Використовуємо метод градієнтного спуску для зміни ваги:

$$w_{ji}\_\text{new} = w_{ji}\_\text{old} - \alpha \cdot \partial E / \partial z_i, \quad (2.2.7)$$

де w_{ji}_new – нове значення ваги,

w_{ji}_old – попереднє значення ваги,

α – швидкість навчання (learning rate),

$\partial E / \partial z_i$ – похідна цільової функції за вагою.

Отже, у процесі навчання ваги моделі піддаються коригуванню з використанням алгоритму зворотного розповсюдження помилок. Мета в тому, щоб модель мінімізувала похибки між прогнозованими та фактичними значеннями цін на дії.

Поділимо набір даних на тренувальний та тестувальний у відношенні 8:2. Розділення набору даних на тренувальний та тестувальний є важливим кроком при навчанні моделей машинного навчання. Основною причиною цього є необхідність оцінити точність та загальну ефективність моделі перед її застосуванням на нових, невідомих даних. Основна ідея полягає в тому, що ми використовуємо тренувальний набір даних для навчання моделі. Модель адаптується до даних, вивчає закономірності та робить передбачення на цих даних. Однак, важливо перевірити, наскільки добре модель засвоїла ці закономірності та наскільки загальними є її передбачення.

Для цього ми використовуємо тестувальний набір даних, який модель не бачила під час тренування. Ми подаємо цей набір даних на вхід моделі і оцінюємо її точність на нових даних. Це дає нам уявлення про те, наскільки добре модель здатна узагальнювати свої навички та зробити передбачення на невідомих даних.

Реалізуємо логіку нейронної мережі на мові Python без використання бібліотек мащинного навчання.

2.3 Аналіз результатів

Для початку визначимо гіперпараметри, кількість нейронів та кількість прихованих шарів. Далі будемо їх змінювати і порівнювати результат. Оцінювати точність результата будемо за допомогою згаданого вище MSE.

Отже, epochs (кількість епох) = 100000,

learning_rate (α) = 0.5,

input_size(вхідні дані) = 7,

hidden_sizes (кількість нейронів в прихованих шарах) = [4].

Початкові значення вагів випадкові нормально розподілені із середнім 0 і стандартним відхиленням 1. Спробуємо спочатку зробити з одним прихованим шаром і 4 нейронами в ньому. Візьмемо кожне 4 значення у вхідних даних (56 рядків), тому що структура нейронної мережі дуже проста і не здатна знаходити складні закономірності.

Для порівняння я беру 4 похибки:

TRAIN: MSE PRED - похибка прогнозування на тренувальних даних,

MSE YEST - похибка між значенням заданого дня і минулого в тренувальних даних.

TEST: MSE TEST - похибка прогнозування на тренувальних даних

MSE YEST - похибка між значенням заданого дня і минулого в тестувальних даних.

Ми хочемо досягти, щоб похибка прогнозування на тестувальних даних була меншою, ніж похибка між значенням ціни акції і її значенням день назад.

```
TRAIN
MSE PRED 5.032264317367699
MSE YEST 7.440581073831706
TEST
MSE PRED 6.3501277213840437
MSE YEST 7.575696678205958
```

(Мал. 6) Результат

Як бачимо, похибка як в тренувальних даних, так і в тестувальних є досить великою. Спробуємо змінити параметр hidden_sizes.

Тепер hidden_sizes = [4, 4, 4]. Тобто маємо 3 прихованих шари по 4 нейрони в кожному. Все ще беремо кожне 4 число (56 рядків) і все інше також залишимо як було.

```
TRAIN
MSE PRED 0.09659379020443463
MSE YEST 7.54957577470883
TEST
MSE PRED 5.460918927566864
MSE YEST 8.05755253457502
```

(Мал. 7) Результати

Також можемо побачити, що хоча похибка не дуже гарна, але майже в 60% випадків нейронна мережа вгадує напрям руху ціни:

```
SIDE GUESS [1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0]
58.333333333333336 %
```

(Мал. 8) Напрям руху

Результат вийшов значно кращим. Бачимо, що похибка на тренувальних даних досить маленька, а на тестувальних менша, ніж похибка вчорашнього дня. Результат досягнений ускладненням структури нейронної мережі. Він доволі непоганий, враховуючи те, що це все ще проста нейронна мережа без

модифікацій. Вона не враховує значення минулого дня, тобто послідовність даних, а просто будує модель на основі вхідних даних.

2.4 Теоретична частина. Рекурентна нейронна мережа

Рекурентна нейронна мережа (англ. recurrent neural network, RNN) є типом штучної нейронної мережі, яка призначена для роботи з послідовними даними, такими як текст, звук, часові ряди тощо.

Модель Ізінга, розроблена Ленцом та Ізінгом в 1925 році, мала архітектуру, подібну до рекурентних нейронних мереж (RNN). Вона використовувалася для моделювання фізичних систем. Ця модель мала обмежену здатність моделювати складні залежності і не була придатною для навчання. Пізніше, у 1972 році, Шунічі Амарі модифікував модель Ізінга, перетворивши її на адаптивну мережу, яку він назвав мережею Хопфілда. Ця модель мала здатність навчатися і зберігати інформацію про залежності між вхідними даними. У 1986 році Девід Румельхарт представив багатошарову мережу Хопфілда, яка використовувала більш складну архітектуру з кількома шарами нейронів. Далі відбувалось досить багато модифікацій цього алгоритму для посилення його ефективності.

Основна ідея рекурентних нейронних мереж полягає в тому, щоб враховувати контекст і залежності між елементами послідовності. Завдяки своїй внутрішній пам’яті, RNN можуть зберігати важливу інформацію про вхідні дані, які вони отримали. Ця особливість дозволяє їм досягати високої точності у передбаченнях майбутніх подій. У RNN кожен шар мережі має ваги, які можуть використовуватися для збереження і передачі інформації від одного кроку до наступного. Це дозволяє RNN "пам'ятати" попередній стан і використовувати його для врахування контексту при обробці наступних вхідних даних. Завдяки своїм рекурентним зв'язкам, нейронні мережі здатні до

формування глибокого розуміння послідовності та контексту, що робить їх ефективними порівняно з іншими алгоритмами.

Тепер перейдемо до теоретичної частини рекурентної нейронної мережі.

З точки зору архітектури, RNN зазвичай складається з трьох основних компонентів:

1. Вхідний рівень: вхідний рівень RNN отримує послідовні вхідні дані. Кожен елемент вхідної послідовності зазвичай представляється у вигляді вектора.

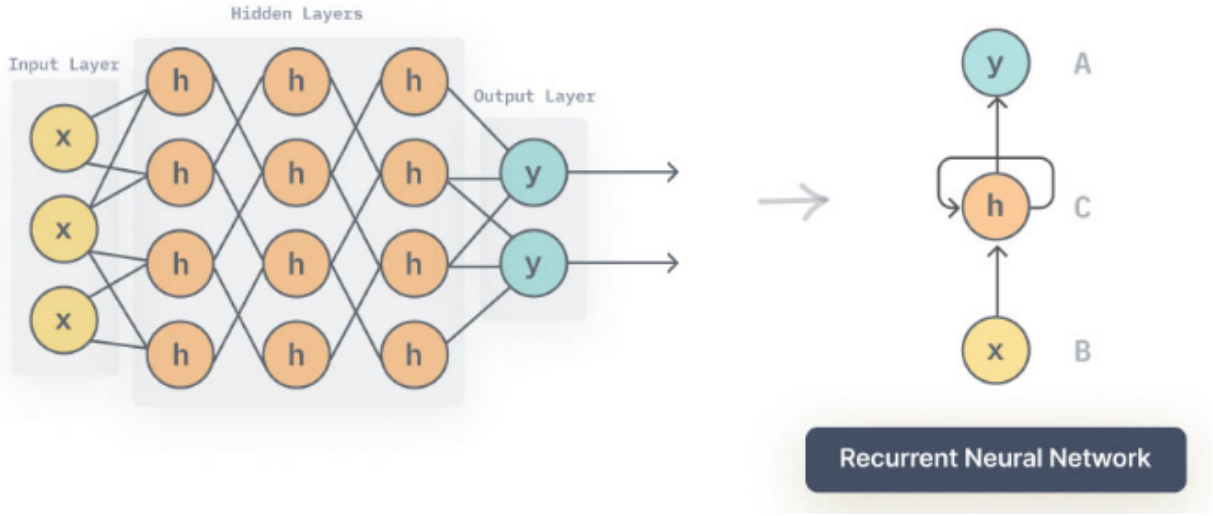
2. Прихований рівень (рівні): приховані рівні RNN містять повторювані з'єднання, які дозволяють мережі зберігати пам'ять про минулі вхідні дані. Прихований шар отримує як поточний вхід, так і попередній прихований стан як вхідні дані. Прихований стан оновлюється та передається на наступний часовий крок, ефективно фіксуючи інформацію з попередніх введень.

3. Ваги: Подібно до інших нейронних мереж, RNN має ваги, пов'язані зі зв'язками між її шарами. Ці ваги визначають вплив входів на прихований стан і вихід.

4. Функція активації: на кожному кроці часу введення та попередній прихований стан поєднуються та передаються через функцію активації. Найпоширенішою функцією активації в RNN є функція гіперболічного тангенса (tanh), яка здавлює комбінований вхідний сигнал у діапазон [-1, 1]. Також можна використовувати інші функції активації, такі як ReLU або вже відому нам функцію сигмоїда.

5.Сума: комбіновані вхідні дані (вхідні дані + попередній прихований стан) зазвичай лінійно перетворюються шляхом застосування вагових коефіцієнтів і зміщень перед проходженням через функцію активації. Точні деталі цього перетворення залежать від конкретної архітектури та варіанту RNN, що використовується.

6. Вихід: Вихід RNN можна отримати на кожному кроці часу або лише на останньому кроці часу. Це залежить від конкретного завдання. Наприклад, у завданнях від послідовності до послідовності, таких як машинний переклад, вихідні дані зазвичай отримують на кожному кроці часу та представляють передбачуваний вихід у цій конкретній позиції послідовності.



(Мал. 9) Спрощений вид на рекурентну нейронну мережу

2.5 Реалізація рекурентної нейронної мережі

Позначимо вхід на кроці часу t як x(t), прихований стан на кроці часу t як h(t), а вихід на кроці часу t як y(t).

1. Рівняння оновлення прихованого стану:

Прихований стан на кожному кроці часу обчислюється на основі поточного введення та попереднього прихованого стану. Математично рівняння оновлення прихованого стану можна представити так:

$$h(t) = f(W_x \cdot x(t) + W_h \cdot h(t-1) + b_h), \qquad (2.5.1)$$

де $h(t)$ - вектор прихованого стану на кроці часу t,

$x(t)$ - вхідний вектор на кроці часу t,

$h(t\ -\ 1)$ - прихований вектор стану на попередньому часовому кроці,

W_x - вагова матриця для вхідних даних,

W_h - вагова матриця для прихованого стану,

b_h - вектор зміщення,

$f(\cdot)$ - функція активації, як правило, нелінійна функція, як-от сигмоїда або функція гіперболічного тангенсу.

Це рівняння поєднує поточний вхідний сигнал $x(t)$, попередній прихований стан $h(t\ -\ 1)$ і зсув b_h за допомогою вагових матриць W_x і W_h.

2. Вихідне рівняння:

Вихід на кожному кроці часу обчислюється на основі прихованого стану. Математично вихідне рівняння можна представити у вигляді:

$$y(t)\ =\ g(W_{hy}\cdot\ h(t)\ +\ b_y), \qquad (2.5.2)$$

де $y(t)$ - вихідний вектор на кроці часу t,

$h(t)$ - вектор прихованого стану на кроці часу t,

W_{hy} - вагова матриця для результату,

b_y - вектор зміщення,

$g(\cdot)$ - функція активації, яка перетворює вихід.

4. Зворотне поширення помилки в часі (Backpropagation Trough Time):

Під час навчання параметри RNN (ваги та зміщення) вивчаються шляхом мінімізації функції втрат, яка вимірює різницю між прогнозованим виходом $y(t)$ і цільовим виходом для кожного кроку часу. Градієнти функції втрат щодо параметрів обчислюються за допомогою алгоритму зворотного поширення помилки. BPTT передбачає поширення градієнтів назад у часі для оновлення параметрів на кожному часовому кроці.

Обчислимо градієнти функції втрат щодо параметрів RNN.

Спочатку ми обчислюємо градієнт втрат відносно виходу на кожному кроці часу:

$$\partial L \,/\, \partial y(t) \tag{2.5.3}$$

Тоді ми можемо обчислити градієнт втрат щодо прихованого стану на кожному кроці часу:

$$\partial L(t)/\partial h(t) \;=\; (\partial L(t)/\partial y(t) \cdot \partial y(t)/\partial h(t)) + (\partial L(t+1)/\partial h(t) \cdot \partial h(t+1)/\partial h(t))$$

Для останнього кроку T ми маємо:

$$\partial L(T)/\partial h(T) = \partial L(T)/\partial y(T) \cdot \partial y(T)/\partial h(t)$$

Розрахуємо градієнти відносно параметрів RNN:

$$\partial L \,/\, \partial W_x \;=\; \sum(\partial L/\partial h(t) \cdot \partial h(t)/\partial W_x)$$

$$\partial L \,/\, \partial W_h \;=\; \sum(\partial L/\partial h(t) \cdot \partial h(t)/\partial W_h)$$

$$\partial L \,/\, \partial b \;=\; \sum(\partial L/\partial h(t) \cdot \partial h(t)/\partial b)$$

$$\partial L \,/\, \partial W_{hy} \;=\; \sum(\partial L/\partial y(t) \cdot \partial y(t)/\partial W_{hy})$$

$$\partial L \,/\, \partial b_y \;=\; \sum(\partial L/\partial y(t) \cdot \partial y(t)/\partial b_y)$$

5. Параметри оновлення:

Після обчислення градієнтів щодо параметрів ми можемо оновити параметри за допомогою алгоритму оптимізації, такого як градієнтний спуск. Наприклад, розглянемо формули оновлення для вагової матриці W_x і вектора зміщення b_h:

$$W_{x\,new} \;=\; W_x \;-\; \alpha \cdot \partial L/\partial W_x,$$

$$b_{h\,new} \;=\; b_h \;-\; \alpha \cdot \partial L/\partial\, b_h,$$

де α – швидкість навчання (learning rate)

Подібним чином можна застосувати ту саму формулу оновлення для інших параметрів у RNN, таких як W_h, W_{hy} і b_y, замінивши параметр і відповідний градієнт.

2.6 Аналіз результатів

Спочатку ми визначимо гіперпараметри моделі - кількість нейронів та кількість прихованих шарів. Потім ми будемо змінювати ці гіперпараметри і порівнювати результати. Для оцінки точності моделі використаємо середньоквадратичну помилку (MSE), яку ми вже згадували раніше.

Отже, epochs (кількість епох) = 10000,

learning_rate (α) = 0.1,

input_size(вхідні дані) = 7,

hidden_sizes (кількість нейронів в прихованих шарах) = [20].

Початкові значення вагів випадкові нормально розподілені із середнім 0 і стандартним відхиленням 0.01. Спробуємо зробити з одним прихованим шаром і 20 нейронами в ньому. Візьмемо підряд 56 рядків, тому що структура нейронної мережі дуже проста і не здатна знаходити складні закономірності.

Для порівняння я беру знову 4 похибки, як і в п. 2.3**.**

```
TRAIN
MSE PRED 5.032264317367699
MSE YEST 7.440581073831706
TEST
MSE PRED 6.3501277213840437
MSE YEST 7.575696678205958
```

(Мал. 10) Результат

```
SIDE GUESS [1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0]
66.66666666666666 %
```

(Мал. 11) Напрям руху

Результат вийшов трохи кращим ніж в звичайній нейронній мережі. Похибка на тренувальних не така маленька, бо модель починає прогнозувати без "пам'яті", а отримує її по ходу. На тестувальних даних отримали хороший результат - кращий за вчорашній день і за звичайну нейронну мережу. Напрям руху ціни також був вгаданий краще - майже 70% - це відбувається тому що нейронна мережа має "пам'ять" і розуміння тренду. Загалом результат непоганий, враховуючи те, що це все ще проста структура.

2.7 Порівняння точності двох моделей

З п. 2.3 та 2.6 можемо сказати, що рекурентна нейронна мережа дійсно показує кращі результати для прогнозування цін акцій. Обидві архітектури нейронних мереж можуть бути застосовані для вирішення завдань прогнозування часових рядів, таких як ціни акцій, але результати можуть змінюватись в залежності від різних факторів.

Точність нейронної мережі та рекурентної нейронної мережі залежить від кількох факторів, включаючи якість даних, обрану архітектуру мережі, кількість нейронів та епох, кількість самих даних та багато іншого. Жодна з цих архітектур не гарантує абсолютну точність при прогнозуванні цін акцій, оскільки вони обидві є моделями машинного навчання і схильні до помилок і обмежень.

ВИСНОВКИ

У цій роботі було проведено прогнозування цін акцій компанії Google з використанням нейронних мереж. Спочатку було визначено вхідні дані, що включали різні фактори, які можуть впливати на ціни акцій та проведена кореляція для визначення, які параметри підходять найбільше для навчання нейронних мереж.

Потім були побудовані прогнози двома методами: нейронними мережами та рекурентними нейронними мережами. Нейронні мережі є потужним інструментом для аналізу складних залежностей у даних і можуть бути ефективними в прогнозуванні цін акцій. Рекурентні нейронні мережі мають здатність вловлювати тимчасові залежності в часових рядах і можуть бути особливо корисними при прогнозуванні цін акцій. Впродовж тренування та тестування даних були визначені параметри, необхідні для навчання мережі, такі як функції активації: сигомоїда, гіперболічний тангенс, та оптимізації: MSE. Також були визначені гіперпараметри, такі як кількість епох, швидкість начання та ін. Вони впродовж всього процесу змінювались задля покращення результату.

Для порівняння точності цих методів було використано оцінку MSE (середня квадратична помилка), яка дозволяє оцінити розбіжність між прогнозованими значеннями та фактичними значеннями цін акцій. Шляхом порівняння MSE для нейронних мереж та рекурентних нейронних мереж можна зробити висновки про те, яка архітектура справляється з задачею краще.

З проведеного аналізу можна встановити, що рекурентна нейронна мережа демонструє нижчу середню квадратичну помилку, що вказує на кращу точність прогнозування цін акцій. Також можна оцінити інші показники, такі як точність прогнозування напряму руху цін акцій (наприклад, ймовірність передбачення підвищення чи зниження ціни).

В цілому, результати цієї роботи підтверджують ефективність використання нейронних мереж у прогнозуванні цін акцій та надають практичну основу для застосування даних методів у реальних фінансових завданнях. Однак, необхідно враховувати, що точність прогнозів може змінюватись в залежності від конкретних умов ринку, доступних даних та інших факторів. Подальші дослідження та експерименти можуть зробити додатковий внесок у розвиток та покращення цих методів прогнозування цін акцій.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

[1]https://www.nasdaq.com/market-activity/stocks/goog

[2] Goodfellow, I., Bengio, Y., & Courville, A. (2016). "Deep Learning". MIT Press.

[3] Rivera, D. E., & Winer, E. H. (2018). "Recurrent Neural Networks for Short-Term Time-Series Forecasting". Springer.

[4] Brownlee, J. (2018). "Recurrent Neural Networks in Python: Recurrent Neural Networks for Deep Learning, Sequence Prediction, and Natural Language Processing". Machine Learning Mastery.

[5] Nielsen, M. (2015). "Neural Networks and Deep Learning". Determination Press.

[6] Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). "Learning representations by back-propagating errors". Nature, 323(6088), 533-536.

[7] https://fred.stlouisfed.org/series/FEDFUNDS

[8] https://tradingeconomics.com/united-states/unemployment-rate

[9] https://tradingeconomics.com/united-states/consumer-price-index-cpi

[10] https://buklib.net/books/30294/

[11] https://www.investopedia.com/terms/r/rsi.asp

[12] https://www.investopedia.com/terms/m/macd.asp

[13] https://www.investopedia.com/terms/s/stochasticoscillator.asp

[14]https://www.investopedia.com/ask/answers/112814/how-average-directional-index-adx-calculated-and-what-formula.asp

[15]https://www.statisticssolutions.com/free-resources/directory-of-statistical-analyses/pearsons-correlation-coefficient/

[16]https://en.wikipedia.org/wiki/Neural_network

[17]https://machinelearningmastery.com/an-introduction-to-recurrent-neural-networks-and-the-math-that-powers-them/

[18]https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks

Відгук

на кваліфікаційну роботу бакалавра на тему:

«Прогнозування вартості акцій методами машинного навчання»

студентки 4-го курсу факультету комп'ютерних наук та кібернетики

Київського національного університету імені Тараса Шевченка

Бурі Діани Тарасівни

Кваліфікаційна робота присвячена прогнозуванню цін акцій великих компаній методами машинного навчання (на приклалі компанії Google). Було досліджено використання звичайних та рекурентних нейронних мереж.

Студентка провела тренування та тестування з метою визначення оптимальних параметрів для навчання мережі, таких як функції активації та крок навчання, адже такий підхід дозволє досягти більш високої точності прогнозів. Особлива увага приділена гіперпараметрам, які були налаштовані впродовж всього процесу з метою покращення результатів.

Під час роботи над дипломом студентка продемонструвала здатність досліджувати теоретичні основи алгоритмів, здатність до самостійного мислення та вміння застосовувати теоретичний матеріал в практичних задачах. Вважаю, що кваліфікаційна робота студентки Бурі Діани Тарасівни відповідає всім вимогам, які висуваються до бакалаврських робіт, і заслуговує на оцінку «відмінно», а її автор заслуговує на присвоєння кваліфікації бакалавра.

Асистент кафедри обчислювальної математики  Сергій ДЕНИСОВ

Рецензія

на кваліфікаційну роботу бакалавра на тему:
«Прогнозування вартості акцій методами машинного навчання»
студентки 4-го курсу факультету комп'ютерних наук та кібернетики
Київського національного університету імені Тараса Шевченка
Бурі Діани Тарасівни

Кваліфікаційна робота Бурі Діани Тарасівни присвячена прогнозуванню цін акцій компанії Google та являє собою дослідження використання нейронних мереж та рекурентних нейронних мереж у фінансовому аналізі.

В роботі було проведено аналіз вхідних даних, включаючи різні фактори, які можуть впливати на ціну акцій. Був застосований кореляційний аналіз. Студентка побудувала прогнози за допомогою двох методів - нейронних мереж та рекурентних нейронних мереж. Цей вибір є обґрунтованим, оскільки нейронні мережі відомі своєю здатністю аналізувати складні залежності в даних, тоді як рекурентні нейронні мережі здатні враховувати тимчасові залежності в часових рядах. Також проведено експерименти щодо зміни параметрів навчання та гіперпараметрів з метою досягнення найкращих результатів. Порівняння середньоквадратичної похибки між моделями дозволило зробити обґрунтований висновок про те, що рекурентна нейронна мережа показує трохи кращу точність в прогнозуванні цін акцій. Також прогнозування напряму руху цін акцій були зроблені для більш повного розуміння ефективності методів.

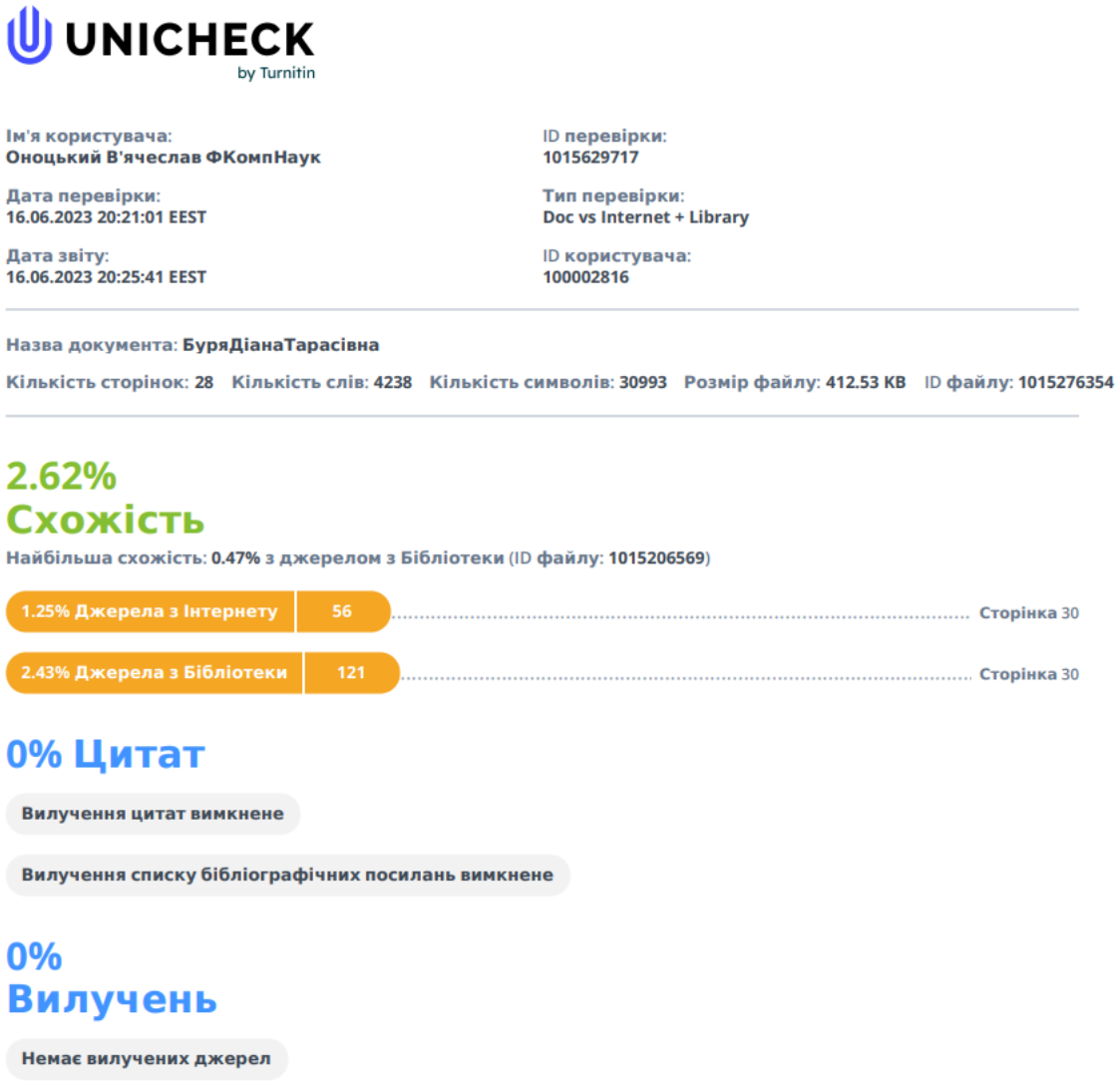
Вважаю, що кваліфікаційна робота студентки Бурі Діани Тарасівни відповідає всім вимогам, які висуваються до бакалаврських робіт і заслуговує на оцінку «відмінно», а її автор заслуговує на присвоєння кваліфікації бакалавра.

Кандидат технічних наук,
доцент  Катерина ГОЛУБЄВА

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА

СИСТЕМА ЗАПОБІГАННЯ ТА ВИЯВЛЕННЯ АКАДЕМІЧНОГО ПЛАГІАТУ

Довідка про оригінальність кваліфікаційної роботи за освітнім рівнем бакалавр



Експертна оцінка роботи науковим керівником :

Робота студенки 4-го курсу Бурі Діани Тарасівни «Прогнозування вартості акцій методами машинного навчання» виконана самостійно, при цьому цитування та запозичення становить 2.62% та не перевищують норму.

Науковий керівник: _____ Денисов С.В.

(підпис) (ПІБ)

Оператор: _____ Оноцький В.В.

(підпис) (ПІБ)