

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет комп'ютерної інженерії та управління
(повна назва)

Кафедра Комп'ютерних інтелектуальних технологій та систем
(повна назва)

АТЕСТАЦІЙНА РОБОТА
Пояснювальна записка

Другий (магістерський) рівень
(рівень вищої освіти)

Прогнозування часових рядів на основі використання
штучних нейронних мереж

(тема)

Виконав: студент II курсу, групи КІТм-19-1
спеціальності (напряму підготовки) _____

123 – Комп'ютерна інженерія

(код і повна назва спеціальності, напряму)

освітньої програми _____

Комп'ютерні інтелектуальні технології

(повна назва освітньої програми)

Корецька П.С.

(підпис)

(прізвище, ініціали)

Керівник: _____ проф. Корабльов М.М.

(підпис)

(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри КІТС _____

(підпис)

Руденко О.Г.

(прізвище, ініціали)

2020 р.

Харківський національний університет радіоелектроніки

Факультет _____ комп'ютерної інженерії та управління
Кафедра _____ електронних обчислювальних машин
Рівень вищої освіти _____ другий (магістерський)
Спеціальність (напрямок) _____ 123 – Комп'ютерна інженерія
(код і назва)
Освітня програма _____ Комп'ютерні інтелектуальні технології
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

“ _____ ” _____ 20__ р.

ЗАВДАННЯ
НА АТЕСТАЦІЙНУ РОБОТУ

студентові _____ **Корецькій Поліні Сергіївні**
(прізвище, ім'я, по батькові)

1. Тема роботи Прогнозування часових рядів на основі використання штучних нейронних мереж

затверджена наказом по університету від “ 11 ” листопада 2020 р. № 1582 Ст

2. Термін подання студентом роботи до екзаменаційної комісії _____ 10 грудня 2020 р.

3. Вхідні дані до роботи _____

- 1) часові ряди, які прогноуються, можуть бути довільної природи;
- 2) прогнозування часових рядів виконувати з використанням ШНМ;
- 3) в якості даних для моделювання використовувати дані фінансового ринку продаж;
- 4) мова програмування – Python.

4. Перелік питань, що потрібно опрацювати в роботі 1) Огляд предметної області;

- 2) аналіз предмету дослідження;
- 3) дослідження методів прогнозування часових рядів;
- 4) побудова прогнозних моделей;
- 5) оцінка основних характеристик продукту;
- 6) моделювання прогнозних моделей;
- 7) висновки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) Демонстраційні матеріали. Плакати - 17 арк. ф. А4

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Огляд стану проблеми та постановка задачі	01.09-14.09	
2	Аналіз літератури за напрямком магістерської роботи	14.09-28.09	
3	Вибір методів рішення для реалізації та їх обґрунтування	28.09-12.10	
4	Моделювання прогнозних моделей	12.10-02.11	
5	Оформлення пояснювальної записки	02.11-23.11	
6	Підготовка графічного матеріалу	23.11-07.12	

Дата видачі завдання 11 листопада 2020 р.

Студент _____
(підпис)

Керівник роботи _____
(підпис)

проф. Корабльов М.М.
(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка атестаційної роботи: 81 с., 21 рис., 12 табл., 2 дод., 19 джерел.

ЧАСОВИЙ РЯД, АВТОРЕГРЕСІЯ, ЗГЛАДЖУВАННЯ, НЕЙРОННІ МЕРЕЖІ, ЗГОРТКОВІ НЕЙРОННІ МЕРЕЖІ, РЕКУРЕНТНІ НЕЙРОННІ МЕРЕЖІ

Предмет дослідження – методи прогнозування часових рядів: традиційні моделі авторегресії ARMA, ARIMA, експоненційне згладжування, а також методи інтелектуального аналізу даних з використанням штучних нейронних мереж та глибинного навчання – згорткові та рекурентні нейронні мережі.

Метою атестаційної роботи є проаналізувати предмет дослідження, виявити параметри впливу на ефективність та точність деяких моделей, що використовуються для аналізу та прогнозування вартості акцій.

У ході виконання атестаційної роботи проводився аналіз методів прогнозування часових рядів, було розглянуто класичні методи прогнозування часових рядів та методи прогнозування на основі використання штучних нейронних мереж, було проведено порівняльний аналіз розглянутих методів прогнозування, визначено їх переваги та недоліки, описано архітектуру розробленої програми, та остаточні порівняльні результати роботи всіх моделей.

ABSTRACT

Explanatory note of attestation work: 81 pages, 21 figures, 12 tables, 2 appendix, 19 sources.

TIME SERIES, AUTOREGRESSION, SMOOTHING, NEURAL NETWORKS, ROLLED NEURAL NETWORKS, RECURRENT NEURAL NETWORKS

Subject of research - time series prediction methods: traditional models of autoregression ARMA, ARIMA, exponential smoothing, as well as methods of data mining using artificial neural networks and deep learning - convolutional and recurrent neural networks.

The purpose of certification work is to analyze the subject of research, to identify the parameters of the impact on the efficiency and accuracy of some models used to analyze and forecast the value of shares.

During the attestation work the analysis of time series forecasting methods was performed, classical time series forecasting methods and forecasting methods based on the use of artificial neural networks were considered, comparative analysis of the considered forecasting methods was performed, their advantages and disadvantages were identified, the developed program architecture was described, and final comparative results of all models.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	8
ВСТУП	9
1 АНАЛІЗ МЕТОДІВ ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ.....	11
1.1 Основні поняття часових рядів.....	11
1.2 Методи згладжування.....	12
1.2.1 Просте ковзне середнє.....	12
1.2.2 Зважене ковзне середнє.....	14
1.2.3 Експоненціальне ковзне середнє.....	14
1.2.4 Подвійне експоненціальне ковзне середнє	15
1.2.5 Потрійне експоненційне згладжування	17
1.3 Методи авторегресії.....	19
1.3.1 Проста модель авторегресії (AR - Autoregression)	20
1.3.2 Модель авторегресії — ковзного середнього (ARMA - Autoregressive moving average)	21
1.3.3 Модель авторегресії — інтегрованого ковзного середнього (ARIMA - Autoregressive integrated moving average)	21
1.4 Сезонна модель Бокса-Дженкінса	22
1.5 Постановка задачі дослідження.....	23
2 НЕЙРОМЕРЕЖЕВЕ ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ	25
2.1 Штучна нейронна мережа	26
2.2 Штучний нейрон	28
2.3 Перцептрон	29
2.4 Багатошаровий перцептрон.....	30
2.5 Згорткові нейронні мережі.....	31
2.6 Рекурентна нейронна мережа	35
2.7 Мережа Кохонена.....	36

3 РЕАЛІЗАЦІЯ МЕТОДІВ ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ	37
3.1 Вхідні дані.....	37
3.2 Дані для дослідження.....	39
3.3 Опис метрик.....	40
3.3.1 Середня абсолютна похибка (MAE).....	40
3.3.2 Середньоквадратична помилка (MSE).....	41
3.4 Реалізація методів згладжування.....	41
3.5 Реалізація методів авторегресії.....	46
3.6 Реалізація MLP	47
3.7 Реалізація згорткових нейронних мереж (CNN).....	49
3.8 Реалізація рекурентних нейронних мереж (RNN)	50
4 ОБҐРУНТУВАННЯ ВИКОРИСТАННЯ ПРОГРАМНОГО ПРОДУКТУ	54
4.1 Функції програмного продукту	54
4.2 Обґрунтування системних параметрів програмних продуктів	57
4.3 Аналіз експертної оцінки параметрів	60
4.4 Аналізування рівня якості відбору функцій.....	63
ВИСНОВОК.....	65
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	67
ДОДАТОК А.....	69
ДОДАТОК Б	78

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

MAE – Mean Absolute Error – Середня абсолютна помилка

MSE – Mean Squared Error – Середня квадратична помилка

MA – Moving Average – Ковзне середнє

EMA – Exponential Moving Average – Експоненціальне ковзне середнє

ANN – Artificial Neural Networks – Штучні нейронні мережі

CNN – Convolutional Neural Networks – Згорткові нейронні мережі

RNN – Recurrent Neural Networks – Рекурентні нейронні мережі

LSTM – Long Short-Term Memory – Довга короткочасна пам'ять

ARMA – Autoregressive moving average – Авторегресивне ковзне середнє

ARIMA – Autoregressive integrated moving average – Авторегресивне інтегроване ковзне середнє

AIC – Інформаційний критерій Акайке

GPU – Графічний процесор

MLP Multilayer Perceptron – багат шарових перцептрон

ВСТУП

Сьогодні завдання прогнозування часових рядів стає особливо актуальним у різних сферах людської діяльності: у природничих науках - прогнози кількості опадів, забруднення природних ресурсів, показники різних біологічних та біохімічних показників; в економічній науці - прогнозувати поточні коливання цін на акції, щотижневі та щомісячні продажі, курси валют, річний обсяг виробництва тощо.

Фондовий ринок є одним з найважливіших секторів ринкової економіки, оскільки він надає компаніям доступ до капіталу, дозволяючи інвесторам купувати акції компанії. Прогнозувати роботу фінансових ринків - одне з найскладніших завдань в економіці. Прогнози повинні враховувати низку фізичних, психологічних, раціональних та ірраціональних чинників поведінки тощо. Усі ці аспекти дозволяють зробити висновок, що ціни на акції дуже мінливі і їх дуже важко передбачити з високою точністю. Однак це завдання є актуальним для всього світу та для міжнародної економіки в цілому, оскільки здатність прогнозувати точну вартість акцій тісно пов'язана з економічною віддачею населення. діловий капітал, державний або особистий капітал та формування більш раціональної економічної поведінки.

За останні 70 років було проведено багато досліджень щодо здатності прогнозувати економічні часові ряди. Вони працюють над визначенням того, які активи компанія збільшить у вартості, а який бізнес буде прибутковим при інвестуванні. Це означає, що прибутковість завжди перевищує середню ринкову віддачу і підтримує такий самий рівень ризику, як і решта ринку. Для цього давно використовується базовий аналіз. Базовим аналізом є інтерпретація, оцінка та прогнозування аналітичних показників на макро- та мікрорівні та очікування учасників ринку, які можуть прямо чи опосередковано впливати на динаміку ціни фінансових активів. По-перше, такі дослідження забирають багато часу, а по-друге, такий аналіз обов'язково

підпорядковується упередженості аналітиків. Тому в останній час багато спеціалістів в області фінансів використовують методи аналізу даних, зокрема прогнозування часових рядів, для розв'язання цієї задачі.

Предметом дослідження є методи прогнозування часових рядів. Метою даної роботи є аналіз вивченого, вивчення відомих і найпопулярніших методів прогнозування часових рядів та визначення параметрів, що впливають на ефективність і точність деяких моделей. Метод дослідження полягає у побудові прогнозних моделей та аналізі результатів. Пояснювальна записка складається з 4 частин. У першій частині розглядаються існуючі підходи до прогнозування часових рядів. У другій частині розглядаються методи передбачення нейронної мережі, що використовуються для побудови передбачень. Третя частина описує архітектуру розробленої програми та кінцеві результати порівняння всіх обраних моделей після аналізу.

1 АНАЛІЗ МЕТОДІВ ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ

1.1 Основні поняття часових рядів

Часовий ряд – використання випадкового процесу, сукупності результатів подальших спостережень.

Як правило, метою статистичного аналізу використання часових рядів є побудова моделі серії, яка може бути використана для пояснення поведінки серії та складання прогнозів на майбутні періоди часу.

Аналіз часових рядів починається з побудови та вивчення графіку. Якщо стабільність часових рядів незрозуміла, спочатку слід визначити його нестатичну складову. Процес виявлення тенденцій та інших елементів ланцюга, що призводять до статичних збурень, може проходити у багато етапів. Кожен з них враховує певну кількість залишків, отриманих відніманням обраної моделі тренду з вихідної серії або результатом різниці та інших перетворень серії. На додаток до гістограми, ознаками нестабільності в часових рядах може бути функція самокореляції, а не нуля (крім дуже великих значень гістерезису) та наявність яскраво виражених піків на частотах. діаграма низького циклу. За допомогою функції інтеграції монет також досліджуються внутрішні зв'язки між елементами в часових рядах.

У вибіркових опитуваннях найпростіші числові властивості описової статистики (середнє, медіана, відхилення, стандартне відхилення, асиметричні коефіцієнти та надмірність) часто дають досить повну картину зразок інформації. Графічні методи аналізу зображень та зразків відіграють лише допоміжну роль, що забезпечує краще розуміння місця та концентрації даних та законів розподілу.

Роль графічних методів в аналізі часових рядів зовсім інша. Табличне представлення часових рядів та описова статистика, як правило, не

дозволяють зрозуміти природу процесу, тоді як багато графіків можна скласти з графіка для часових зображень. Надалі їх можна перевірити та уточнити розрахунками.

Людське око цілком визначає графік часових рядів:

- наявність тренду та його характер;
- наявність сезонних та циклічних компонентів;
- ступінь затримки або безперервності змін послідовних значень рядів після усунення тренду (цей показник можна оцінити на основі характеру та величини кореляції між наступними елементами. близькість лінії).

Таким чином, аналіз графіка числа часто визначає подальший напрямок аналізу.

1.2 Методи згладжування

Згладжування – важливий і популярний метод прогнозування фінансових ринків. Як правило, різні методи згладжування базуються на концепції ковзних середніх. Це допомагає зменшити вплив випадкових компонентів на часові ряди. У цьому розділі будуть представлені 5 загальних методів згладжування прогнозів для часових рядів. Основне припущення цих методів полягає в тому, що коливання попередніх значень полягають у тому, що випадкові відхилення від деяких плавних кривих можуть бути екстрапольовані для отримання прогнозу.

1.2.1 Просте ковзне середнє

Просте ковзне середнє (англ. Simple Moving Average – SMA) - один з найпростіших і найпопулярніших показників у технічному аналізі. SMA - це звичайна середня ціна за певний період часу. SMA належить до групи показників, що слідує за трендом, що допомагає визначити початок нового тренду та його кінець, за його кутом можна визначити силу (швидкість), що

використовується як основа (або фактор згладжування) для багатьох інших технічних показників. Ковзну середню інколи називають лінією тренду.

Формула для простого ковзного середнього:

$$SMA = \frac{\sum_{i=1}^n P_i}{n} \quad (1.1)$$

де P_i ринкова ціна;

n - ключовий параметр - тривалість згладжування або цикл SMA (сума цін, включених у розрахунок ковзного). Цей параметр іноді називають ступенем середньої змінної.

Просте ковзне середнє - це звичайне середнє значення цін за певний проміжок часу. SMA - це індекс, який оцінює ціну рівноваги за певний період, чим коротший SMA, тим коротший час рівноваги. Розраховуючи середню ціну, вона завжди дотримується основної ринкової тенденції та відфільтровує невеликі коливання. Чим нижчий SMA (коротка ковзна середня), тим швидше він виявляє нову тенденцію, але в той же час створює більше помилкової волатильності, і навпаки, чим вище параметр (довга ковзна середня), тим новіша тенденція повернеться. рухатися повільніше, але з декількома помилками.

Недоліки методу ковзного середнього:

Коли використовується метод SMA для торгівлі по тренду, затримка входу та виходу з тренду, як правило, дуже значна, тому в більшості випадків більша частина руху тренду втрачається.

Одним з найсерйозніших недоліків методу SMA є те, що він робить однаковий акцент як на нових, так і на старих цінах, хоча доцільно припустити, що нові ціни важливіші, оскільки вони відображають жорстку ринкову ситуацію, ніж зараз.

1.2.2 Зважене ковзне середнє

Зважене ковзне середнє (англ. Weighted Moving Average – WMA). Одним з недоліків SMA є вказівка на середнє значення однакових зважених цін, незалежно від того, ближче вони чи далі від поточної. Цей пропуск виправляється зваженою ковзною середньою. Отже, WMA - це популярна модифікація SMA з вибраними шкалами, завдяки чому останні ціни мають більшу вагу.

Формула зваженого ковзного середнього:

$$WMA = \frac{\sum_{i=1}^n P_i * W_i}{\sum_{i=1}^n W_i} \quad (1.2)$$

де - P_i значення ціни i - періодів тому, (i сьогодні = 1);

W_i - значення ваги ціни до періоду.

Зважене ковзне середнє - це волатильність ціни за певний період. Як інструмент аналізу, він усуває деякі недоліки звичайного ковзного, але не усуває їх повністю.

1.2.3 Експоненціальне ковзне середнє

Експоненціальне ковзне середнє (англ. Exponential Moving Average – ЕМА) зменшує помилку і перевершує останні ціни порівняно з іншими. Цей метод дозволяє швидше реагувати на поточні зміни ціни порівняно з SMA. Шлях остаточної ціни залежить від періоду ковзної середньої. Чим коротший інтервал ЕМА, тим важча кінцева ціна.

Експоненціальне ковзне середнє можна визначити двома способами - як відсоток від ковзної середньої або періодичної ковзної середньої. Отже, у відсотках від ковзання єдиним параметром є вага (у відсотках), а за один період - період КС. Основна формула наступна:

$$EMA = \frac{EMA_{i-1} * (n-1) + 2 * P_i}{n+1} \quad (1.3)$$

де EMA – експонентна ковзна середня;

P_i – значення ціни в i -му періоді;

n – період розрахунку;

EMA_{i-1} - значення попереднього періоду EMA .

Слід зазначити, що теоретично всі ціни використовуються для розрахунку ковзної протягом усього періоду побудови, і, незважаючи на те, що вплив старої ціни з часом зникає, він не зникає повністю. Вплив від старої ціни зникає швидше при коротшій EMA порівняно з довшою ціною. У реальному вираженні різниця між SMA та EMA не велика, навіть якщо вона присутня. Передбачається, що експоненціальне ковзне все-таки краще відображає ринкову ціну за подібних умов, оскільки вплив кожної попередньої ціни експоненційно зменшується із віддаленням від поточної ціни.

1.2.4 Подвійне експоненціальне ковзне середнє

Подвійне експоненціальне згладжування є продовженням експоненціального згладжування, яке чітко підтримує тенденції в односторонніх часових рядах. На додаток до альфа-параметра для управління коефіцієнтом вирівнювання рівня додано коефіцієнт вирівнювання, який контролює зменшення ефекту зміни тренду, відомий як бета-версія (b). Цей метод підтримує тенденції, які змінюються по-різному: додавання та множення залежно від того, чи є тенденція лінійною чи експоненціальною.

Адитивний тренд: експоненціальне подвійне згладжування з лінійним трендом.

Мультиплікативний тренд: експоненціальне подвійне вирівнювання з експоненціальним переміщенням.

Подвійне експоненціальне згладжування з адитивним трендом називається моделлю лінійного тренду Холта, названою на честь розробника методу Чарльза Холта.

Цей метод включає рівняння прогнозування та два рівняння згладжування (одне для рівня та одне для тренда):

Рівняння прогнозу:

$$\hat{y}_{t+h|t} = l_t + hb_t \quad (1.4)$$

Рівень рівняння:

$$l_t = \alpha y_t + (1 - \alpha)(l_{t-1} + b_{t-1}) \quad (1.5)$$

Рівняння тренду:

$$b_t = \beta^*(l_t - l_{t-1}) + (1 - \beta^*)b_{t-1} \quad (1.6)$$

де l_t позначає оцінку рівня ряду за часом t , b_t позначає оцінку тенденції (нахилу) рядів у часі t , α - параметр згладжування для рівня $0 \leq \alpha \leq 1$, та β^* параметр згладжування для тренду $0 \leq \beta^* \leq 1$.

Як і у випадку з простим експоненціальним згладжуванням, рівняння рівня тут показує, що l_t середнє зважене спостереження y_t , і навчання прогнозу на один крок вперед t має вигляд $l_{t-1} + b_{t-1}$. Рівняння тренду показує що b_t середньозважене значення оціненої тенденції за часом t , на основі $l_t - l_{t-1}$, та b_{t-1} попередня оцінка тренду.

Функція прогнозу вже не однорідна, а упереджена. Прогноз для наступних кроків h дорівнює остаточної оцінці плюс остаточно оціночне значення тренду. Отже, прогноз є лінійною функцією.

Прогноз Холта передбачає стійку (вгору або вниз) тенденцію на

невизначений час у майбутньому. Експериментальні дані показують, що ці методи зазвичай передбачувані, особливо для довгих прогнозів. Керуючись цим спостереженням, Гарднер і Маккензі ввели в 1985 р. параметр, який в якийсь момент у майбутньому зменшить ("приглушує") тенденцію до прямої лінії. Методи, що включають демпфований тренд, виявилися дуже успішними, і, мабуть, найбільш популярними індивідуальними методами, коли прогнози необхідні автоматично для багатьох серій.

У поєднанні з параметрами згладжування α і β^* (зі значеннями від 0 до 1, як у методі Холта), цей метод також включає параметр демпфування $0 < \varphi < 1$. Рівняння виглядатимуть так:

$$\begin{aligned}\hat{y}_{t+h|t} &= l_t + (\varphi + \varphi^2 + \dots + \varphi^h)b_t; \\ l_t &= \alpha y_t + (1 - \alpha)(l_{t-1} + \varphi b_{t-1}); \\ b_t &= \beta^*(l_t - l_{t-1}) + (1 - \beta^*)\varphi b_{t-1}.\end{aligned}\tag{1.7}$$

Якщо $\varphi = 1$ то метод такий самий, як і лінійний метод Холта. Для значень від 0 до 1 φ пригнічує тенденцію, щоб у майбутньому він міг наблизитися до константи. Насправді прогнози сходяться до $l_T + \frac{\varphi b_t}{(1-\varphi)}$, де $h \rightarrow \infty$ для любого $0 < \varphi < 1$. Це означає, що короткостроковий прогноз є більш складним, тоді як довгостроковий прогноз залишається незмінним. На практиці це рідко буває менше 0,8, оскільки демпфування дуже сильно впливає на нижчі значення. Значення, близьке до 1, означає, що модель демпфування не може бути відокремлена від моделі демпфування. З цих причин ми, як правило, обмежуємося як мінімум 0,8 і максимум 0,98.

1.2.5 Потрійне експоненційне згладжування

У 1960 році методологію Холта вдосконалив один із його учнів, Вінтерс. Окрім тренду, Вінтерс зазначив, що слід враховувати сезонність.

Метод Холт-Вінтерса (потрійне експоненційне згладжування) включає рівняння прогнозу та три рівняння згладжування - одне для рівня l_t , одне для тренду b_t та одне для сезонного компонента s_t , з параметрами згладжування α, β^* , та γ відповідно. Параметри, що використовуються для позначення частоти сезонності. Наприклад, для щоквартальних та щомісячних даних.

Існує два варіанти цього підходу, які відрізняються за характером сезонної складової [8]. Метод адитивний є кращим, коли сезонні зміни майже постійні протягом ряду, тоді як мультиплікативний метод кращий, коли сезонні зміни змінюються пропорційно рівню в ланцюгу. У методі додавання сезонна складова виражається як абсолютна величина за шкалою серії спостережень, а в рівнянні рівня серія сезонно коригується відніманням сезонної складової. Щороку склад сезону буде приблизно нульовим. У мультиплікативному методі сезонна складова представлена відносною кількістю (відсотками), а серія сезонно коригується діленням сезонної складової. Протягом кожного року сезонний склад буде приблизно m .

Для методу Холта-Вінтерса рівняння буде виглядати так:

$$\begin{aligned}\hat{y}_{t+h|t} &= l_t + hb_t + s_{t+h-m(k+1)} \\ l_t &= \alpha(y_t - s_{t-m}) + (1 - \alpha)(l_{t-1} + b_{t-1}); \\ b_t &= \beta^*(l_t - l_{t-1}) + (1 - \beta^*)b_{t-1}; \\ s_t &= \gamma(y_t - l_{t-1} - b_{t-1}) + (1 - \gamma)s_{t-m}\end{aligned}\tag{1.8}$$

Де k є цілою частиною $\frac{h-1}{m}$, яка гарантує, що оцінки сезонних індексів. Весь розділ наводив оцінку сезонних показників і використовувався для прогнозів останнього року вибірки. Рівняння рівня показують середньозважені значення для сезонних спостережень та прогнозів погоди поза сезоном. Рівняння тенденції подібне до лінійного методу Холта. Рівняння сезону показує середньозважене значення між індексом поточного сезону та сезонним індексом того самого сезону роком раніше (попередні

періоди часу).

Рівняння сезонного компоненту:

$$s_t = \gamma(y_t - l_{t-1} - b_{t-1}) + (1 - \gamma^*)s_{t-m} \quad (1.9)$$

Якщо ми замінимо l_t з рівняння згладжування рівня форми рівнянь вище, отримуємо:

$$s_t = \gamma^*(1 - \alpha)(y_t - l_{t-1} - b_{t-1}) + [1 - \gamma^*(1 - \alpha)]s_{t-m} \quad (1.10)$$

що є подібним до рівняння згладжування сезонних компонентів $\gamma = \gamma^*(1 - \alpha)$. Загальне обмеження параметра $0 \leq \gamma^* \leq 1$, що можна трактувати як $0 \leq \gamma \leq 1 - \alpha$.

Форма рівняння для мультиплікативного методу Холта-Вінтерса, має такий вигляд:

$$\begin{aligned} \hat{Y}_{t+h|t} &= (l_t + hb_t)s_{t+h-m(k+1)}; \\ l_t &= \alpha \frac{y_t}{s_{t-m}} + (1 - \alpha)(l_{t-1} + b_{t-1}); \\ b_t &= \beta^*(l_t - l_{t-1}) + (1 - \beta^*)b_{t-1}; \\ s_t &= \gamma \frac{y_t}{l_{t-1} + b_{t-1}} + (1 - \gamma)s_{t-m} \end{aligned} \quad (1.11)$$

де α параметр згладжування даних, $0 < \alpha < 1$, β параметр згладжування тренду, $0 < \beta < 1$, та γ параметр згладжування сезонності, $0 < \gamma < 1$.

1.3 Методи авторегресії

Модель авторегресії є ефективним інструментом для розуміння та прогнозування майбутніх значень часових рядів, включаючи регресію

змінної щодо попередніх значень у минулому. Важливість моделей авторегресії полягає в їх універсальності, а також у їх здатності описувати майже всі статичні особливості в часових рядах. Частини самовідновлення цих моделей описують, як подальші спостереження впливають один на одного з часом, тоді як у ковзних середніх реєструють деякі неспостережувані потрясіння, що дозволяють моделювати різні явища, які можна спостерігати в різних предметних областях, від біології до економіки.

1.3.1 Проста модель авторегресії (AR - Autoregression)

Регресійна модель показує, що значення вихідної змінної залежить лише лінійно від власних попередніх значень та від випадкових шумів.

$AR(p)$ позначає авторегресійну модель порядку p . $AR(p)$.

Модель математично визначається як:

$$y_n = \sum_{i=1}^p a_i e_{n-i} + e_n \quad (1.12)$$

де a_i - параметри моделі, e_n - білий шум.

Найпростішим процесом AR є $AR(0)$, який не має залежності між компонентами. На значення прогнозу впливають лише шум або помилки.

Для $AR(1)$ з коефіцієнтом a_1 , тільки попереднє значення часового ряду впливає на значення прогнозу.

У випадку якщо a_1 близький до 0, то процес буде виглядати як білий шум, але якщо a_1 за модулем буде наближатися до 1, то вихідне значення буде отримувати більший внесок від попередніх значень ряду порівняно з шумом[3].

1.3.2 Модель авторегресії — ковзного середнього (ARMA - Autoregressive moving average)

Модель ARMA характеризується стохастичним процесом, який використовує два компоненти авторегресії (AR) та ковзного середнього (MA).

Частина AR стосується регресії змінної відповідно до її власних попередніх значень. Частина MA розглядає помилки моделювання як лінійна комбінація помилок, які мали місце в минулому.

$ARMA(p, q)$ характеризує модель з p авторегресійними компонентами і q компонентами для ковзного середнього:

$$y_n = \sum_{i=1}^p a_i y_{n-i} + \sum_{i=1}^q b_i e_{n-i} + e_n, \quad (1.13)$$

де a_i, b_i - параметри моделі, e_n - білий шум.

Модель ARMA підходить, коли система має функцію низки непомічених потрясінь та власної поведінки. Наприклад, ціни на акції можуть значно змінюватися залежно від певної довідкової інформації, а також показувати ефект середньої прибутковості внаслідок певних дій учасників ринку.

1.3.3 Модель авторегресії — інтегрованого ковзного середнього (ARIMA - Autoregressive integrated moving average)

Модель ARIMA є варіантом моделі ARMA. Він складається з трьох компонентів:

- авторегресія (AR) використовує залежність від взаємозв'язку між спостереженнями та іншими спостереженнями із затримкою часу;
- компонент інтегрований (I) призначає вхідні спостереження для

створення фіксованих часових рядів;

- ковзна середня (МА) використовує відношення спостережень до відхилення від ковзних середніх попередніх спостережень.

Кожен із цих компонентів визначає параметри моделі. Використовується стандартний запис $ARIMA(p, d, k)$, де параметри вказують:

- p кількість спостережень за затримкою включається в модель;
- d кількість застосувань відмінностей у вхідних спостереженнях.

Параметр також відомий як ступінь диференціювання;

- q розмір вікна ковзної середньої або порядок ковзного середнього.

Модель формули ARMA використовується для використання моделі ARIMA, але замість y_t подається $\Delta y_t = y_t - y_{t-1}$.

Можна використовувати значення 0 для кожного параметра, яке вказує на відсутність відповідного компонента моделі. Тобто модель ARIMA може бути налаштована на виконання функцій моделі ARMA або навіть простих AR, I або MA.

Варто також зазначити, що AR та MA є лінійними моделями, які працюють на статичних часових рядах, тоді як Model I є препроцесором для зменшення серії до статичних, якщо це необхідно.

1.4 Сезонна модель Бокса-Дженкінса

Модель Бокса-Дженкінса, Autoregressive integrated moving average (ARIMA) - це модель, яка об'єднує авторегресію-ковзного середнього, модель аналізу часових рядів та методологію. Це продовження моделі ARMA для нестационарних часових рядів, що можна зробити стаціонарним, взявши відмінності в порядковому номері від вихідного часового ряду (так звані інтегровані або різницево-стаціонарні тимчасові ряди).

Сезонна модель Бокса-Дженкінса має наступний вигляд:

$$\Delta^d X_t = c + \varepsilon_t + \sum_{i=1}^p a_i \Delta^d X_{t-i} + \sum_{j=0}^q b_j \varepsilon_{t-j}, \quad (1.14)$$

де a_i, \dots, a_p – коефіцієнти авторегресії;

b_0, \dots, b_q – коефіцієнти ковзаючого середнього;

c – константа;

ε_t – білий шум;

Δ^d - оператор різниці часового ряду порядку d (бере різниці в першому порядку - спочатку з часового ряду, потім з досягнутої першої різниці, потім з квадрата тощо).

Також дана модель інтерпретується як $ARMA(p + d, q)$ – модель з d одиничними корінням. При $d = 0$ маємо звичайні ARMA моделі.

Підхід ARIMA до часових рядів - це перш за все оцінка стійкості ланцюга. Різні тести показують наявність окремих коренів та порядок інтегрування часових рядів (зазвичай обмежуються першим або другим порядком). Потім, якщо це необхідно (якщо порядок інтеграції перевищує 0), візьміть перетворений рядок з відмінністю від відповідного замовлення та моделі ARMA, побудованої для перетвореної моделі, за умови, що отриманий процес є статичним, на відміну від першого непостійного процесу.

1.5 Постановка задачі дослідження

Метою магістерської атестаційної роботи є аналіз предмету дослідження, дослідження відомих та найбільш поширених математичних методів прогнозування часових рядів та прогнозування часових рядів на основі нейронної мережі, виявлення параметрів впливу на ефективність та точність прогнозів деяких моделей. Потрібно визначити недоліки та переваги

запропонованих методів прогнозування. У ході аналізу для експериментального дослідження були обрані такі математичні методи прогнозування як:

- методи згладжування;
- методи авторегресії;

та такі методи прогнозування на основі нейронної мережі як:

- багат шаровий перцептрон (MLP);
- Згортоква нейронна мережа (CNN);
- Рекурентна нейронна мережа (RNN).

Необхідно побудувати прогнози моделей та зробити порівняльний аналіз отриманих результатів, описати архітектуру розробленої програми та остаточні порівняльні результати роботи всіх моделей, які було обрано. Провести оцінку ключових особливостей програмного продукту, розробленого для прогнозування часових рядів. Розробити модель у середовищі розробки Jupyter Notebook, використавши мову програмування Python.

2 НЕЙРОМЕРЕЖЕВЕ ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ

Нейромережеві методи - це математичні моделі, засновані на штучних нейронних мережах. Нейромережеві моделі можна вважати простими математичними моделями, що визначають функцію $f: X \rightarrow Y$ або розподіл над X , або над X та Y . Іноді моделі тісно пов'язані з певним правилом навчання.

Важливим є завдання побудови мережі. Це питання вирішується у два етапи:

- вибір конфігурації мережі;
- навчальна мережа (підбір ваги).

На першому етапі слід вибрати конфігурацію нейронної мережі та вирішити наступні проблеми:

- які нейрони ми хочемо використовувати (кількість входів, передавальні функції);
- як зв'язати їх між собою;
- що взяти за мережеві входи та виходи.

Це завдання на перший погляд здається складним, але не потрібно винаходити нейронну мережу. Існує десятки різних архітектур нейронних мереж, і ефективність багатьох з них доведена математичною статистикою. Найпопулярнішими та вивченими архітектурами є багатошаровий перцептрон, нейронна мережа із загальною регресією, мережа Кохонена, мережа Хопфілда, мережа Хеммінга та ін.

На другому етапі необхідно вивчити обрану мережу, тобто підібрати такі значення її ваг, щоб мережа працювала коректно. На практиці нейронних мереж кількість масштабів може становити десятки тисяч, тому навчання є дійсно складним процесом. Для багатьох архітектур були розроблені спеціальні алгоритми навчання, які дозволяють регулювати ваги мережі певним чином. Навчання може бути як у формі контрольованого тренування

(коли мережевий результат постійно порівнюється з очікуваним результатом), так і неконтрольованого (не використовувати зовнішні впливи для регулювання ваг та внутрішнього контролю їх ефективності, шукати регулярність або тенденції вхідних сигналів та налаштовуватись на тренувальну функцію). Для прогнозування використовуються контрольовані моделі навчання.

2.1 Штучна нейронна мережа

Штучна нейронна мережа - це мережа простих елементів, званих нейронами, які отримують вхід, змінюють свій внутрішній стан (збудження) на цьому вході і виробляють залежний вихід. введення та збудження. Мережа утворюється шляхом з'єднання виходів певних нейронів із входами інших нейронів шляхом формування графіку спрямованої ваги. Вага, як і функції обчислювання збудження, може змінюватися в процесі, який називається навчання, який регулюється правилами навчання.

У звичайних додатках ANN сигнал про зв'язок між штучними нейронами є дійсним числом, і вихід кожного штучного нейрона обчислюється з використанням нелінійної суми входів. Штучні нейрони та зв'язки часто мають адаптаційну вагу під час навчання. Вага збільшує або зменшує силу підключеного сигналу. Нейрони можуть мати такий поріг, що сигнал надсилається лише в тому випадку, якщо накопичений сигнал перевищує цей поріг. Штучні нейрони зазвичай розташовуються шарами. Різні класи можуть виконувати різні типи перетворень свого введення. Сигнали рухаються від першого шару (вхід) до останнього шару (вихід), можливо, після багаторазового проходження шарів (наприклад, у рекурсивній нейронній мережі).

Для опису нейронних мереж використовується низка усталених термінів, які можна тлумачити по-різному в різних джерелах, а саме:

- структура нейронної мережі - спосіб взаємодії нейронів у нейронній

мережі;

- архітектура нейронних мереж - структура нейронних мереж та типи нейронів;

- модель нейронної мережі - метод навчання та використання, який іноді включає архітектурну концепцію.

Цей набір параметрів (структура, архітектура, модель) буде називатися конфігурацією нейронної мережі, тобто конфігурація нейронної мережі є визначальними параметрами:

- кількість входів;
- кількість виходів;
- кількість проміжних шарів;
- інформація, що надається на вхід;
- інформація, отримана на виході;
- функція розподілу;
- зв'язок між шарами;
- правила навчання;
- виберіть вхідної та вихідної інформації.

Визначення конфігурації нейронних мереж, яка ефективно вирішить проблему, є тривалим і складним процесом.

Різні моделі нейронних мереж можуть бути реалізовані на основі архітектури і навпаки. Оскільки всі штучні нейронні мережі засновані на концепції нейронів, функцій зв'язку та передачі, між різними структурами або архітектурами нейронних мереж є схожість.

Відповідно до методу комунікації, існує два основних типи нейронних мереж:

- нейронні мережі прямого поширення;
- рекурентні нейронні мережі.

Нейронні мережі прямого поширення - це тип нейронної мережі, в якій сигнали рухаються в одному напрямку, від вхідного шару нейронів, через приховані шари до вихідного шару. Результат - обробка сигналу. Зворотніх

зв'язків немає.

Рекурентна нейронна мережа є та, де зв'язки між вузлами утворюють орієнтований цикл. Це створює внутрішній стан у мережі, що дозволяє демонструвати динамічну поведінку з часом. На відміну від нейронної мережі для передачі, RNM можуть використовувати внутрішню пам'ять для обробки довільних послідовностей введення. Це стосується таких завдань, як несегментоване безперервне розпізнавання рукописного вводу та розпізнавання мови.

2.2 Штучний нейрон

Штучний нейрон - це одиниця для обробки інформації в нейронній мережі. Поведінка нейрона структурована таким чином: є $m + 1$ входів, значення яких дорівнюють $x_0, x_1, x_2, \dots, x_m$, а значення їх ваг рівні, w_0, w_1, \dots, w_m , при цьому перший вхідний елемент, як правило, являє собою фіксоване значення зміщення $x_0 = 1$. Графічно нейрон та його зв'язки показані на рисунку 2.1.

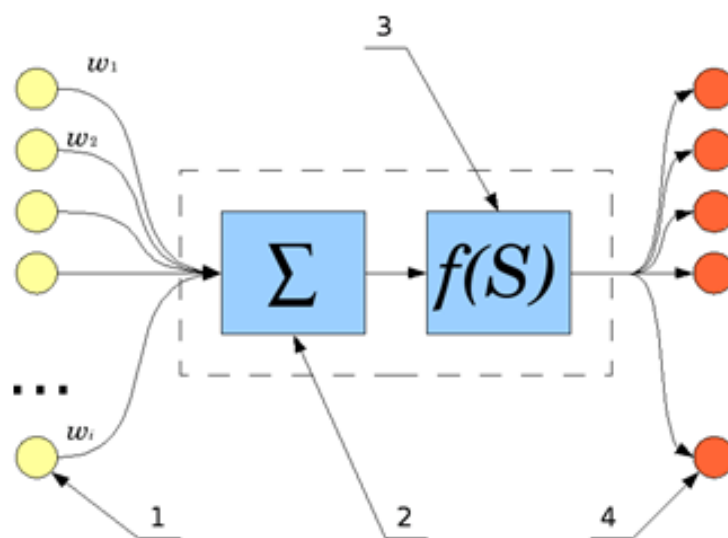


Рисунок 2.1 - Схема штучного нейрону

Тоді вихідне значення нейрона є значенням функції активації зі зваженої суми вхідних значень:

$$y = f(S), S = \sum_{i=0}^m w_i x_i, \quad (2.1)$$

де x_i – входи (1), w_i – ваги, S – суматор (2), f – передатна функція (функція активації) (3), y – вихід/виходи штучного нейрону (4).

Під час реалізації програмного продукту штучні нейрони називаються «процесними елементами» або «процесорами» і вкладають в них більше, ніж базові штучні нейрони, які описані вище.

Функція суматора може бут складною, наприклад, вибирати мінімум, максимум, середнє арифметичне, добуток або обчислювати за допомогою іншого алгоритму. Багато програмних реалізацій використовують власну функцію суматора.

Результат функції суматора перетворюється на вихідний сигнал передатної функції. У передатній функції, яка визначає вихід нервової клітини, загальне значення порівнюється з пороговим значенням (як правило, діапазон $[0, 1][−1, 1]$ чи іншим способом) за допомогою певного алгоритму.

Слід використовувати нелінійну передатну функцію, оскільки прямолінійні функції обмежені, а вихід пропорційний введенню. Використання функцій лінійної передачі було проблемою в ранніх мережевих моделях, і їх обмеження та неефективність доведено.

У існуючих нейронних мережах тангенціальна сигмоїдна, синусова, гіпербола та ін Вони використовуються як функція клітини.

2.3 Перцептрон

Розенблатт сконструював перший у 1958 р. нейронний комп'ютер, перцептрон, намагаючись імітувати навчання людини. Дендрити нейрона моделюються за вагами, що множать на вхідні значення. Крім того, додається

значення зміщення для моделювання необхідного потенціалу активації нейрона. Потім множинні значення та зміщення підсумовуються в тілі клітини і пропускаються через функцію активації для отримання виходу, що представляє швидкість випалу в нейроні. Описана архітектура показана на рисунку 2.2.

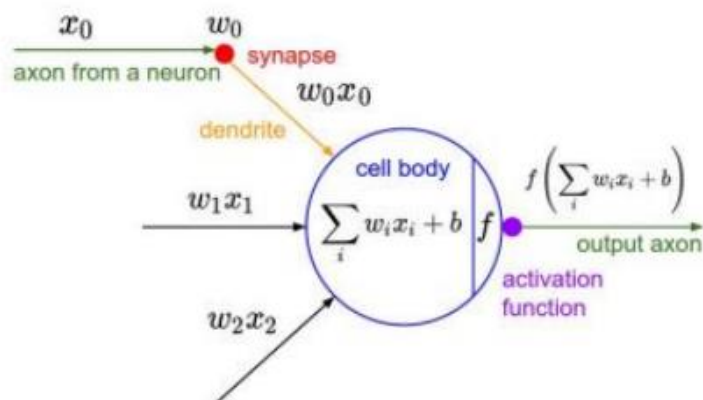


Рисунок 2.2 – Модель перцептрона з вагами w_i , входами x_i та зсувом b .

Перцептрон - це лінійний класифікатор, визначений вагами w_i , зміщенням b і функцією активації f . Можна об'єднати зміщення з вагами за допомогою однорідних координат, тобто покласти зміщення в нижній частині вагового вектора і додати постійну 1 до вхідного вектора x . Можуть використовуватися різні функції активації, але оригінальний перцептрон використовує функцію крокової важкої сторони, що призводить до двійкового виводу. Тому перцептрон є лінійним класифікатором, і його ваги визначають гіперплан як лінійну межу рішення між класами.

2.4 Багатошаровий перцептрон

Багатошаровий перцептрон - (Multilayer Perceptron - MLP) має три або більше шарів. Використовується для класифікації даних, які не можна

розділити лінійно. Це повністю пов'язаний тип нейронної мережі. Це пов'язано з тим, що кожен вузол у шарі пов'язаний з кожним вузлом у наступному шарі.

Багатошаровий перцептрон використовує нелінійну тригерну функцію (насамперед дотичну або логістичну гіперболічну функцію). Ось так виглядає багатошарова персона (рис. 2.3).

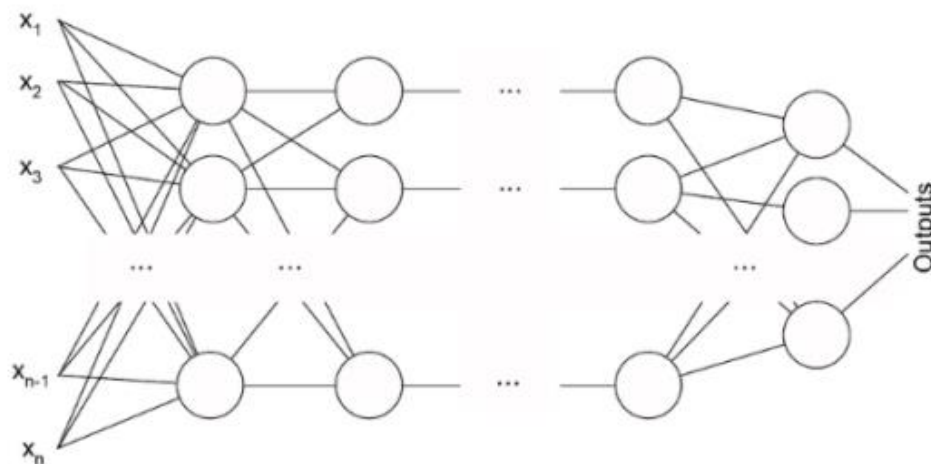


Рисунок 2.3 - Багатошаровий перцептрон

Цей тип нейронної мережі широко використовується в технологіях розпізнавання мови та машинного перекладу.

2.5 Згорткові нейронні мережі

Згорткові нейронні мережі (CNN) - це підгрупа алгоритмів глибокого навчання, які зосереджуються на завданнях комп'ютерного зору та обробки зображень. Системні мережі CNN надихнуті тим, як мозок обробляє візуальну інформацію. Хубель і Визель були першими, хто запропонував глибокі моделі, схожі на структуру зорової кори. Ці моделі ототожнювали прості клітини з локальними сприйнятливими полями, схожими на фільтри або ядра, і складні комірочки, схожі на об'єднання шарів. Перший CNN був

введений в неокогнітроні Фукусіми [5]. Пізніше CNN були покращені французьким вченим Яном ЛеКуном.

Ян ЛеКун, який застосував стохастичні градієнти на основі градієнта для розпізнавання документів і був дуже успішним для розпізнавання рукописних завдань. Основним недоліком у дослідженні та розробці CNN в 1990-х і 2000-х роках була необхідна обчислювальна потужність для їх широкого застосування до зображень з високою роздільною здатністю. Однак це змінилося з 2010 року. Існує три причини, завдяки чому глибокі мережі стали успішними:

- більша обчислювальна здатність завдяки закону Мура, особливо це стосується сьогоднішніх графічних процесорів;
- більше навчальних даних;
- нові та кращі алгоритми.

Зі збільшенням обчислювальної потужності дослідники описали нові способи більш ефективного тренування згорткових нейронних мереж, що дозволяло створювати більш глибокі мережі.

Останнім часом продуктивність значно збільшилась завдяки безлічі баз даних зображень, наближаючись або навіть перемагаючи продуктивність людини, наприклад щодо розпізнавання цифр ($< 0,25\%$) [6] та багатьох інших завдань з розпізнавання шаблонів, найбільш важливих проблем візуальної класифікації та інших.

Загалом, CNN дуже добре класифікують такі об'єкти, як конкретні породи собак та котів на основі дрібнозернистих деталей, тоді як люди мають проблеми з цим. Недоліком глибокого навчання є те, що для розпізнавання об'єктів у реалістичних умовах потрібні дуже великі набори даних для навчання. В даний час найкращі CNN борються з невеликими або плоскими об'єктами, або спотвореними цифровими фільтрами. Недавнє дослідження виявило відмінності того, як глибокі нейронні мережі та людина розпізнають об'єкти. У цьому дослідженні зображення кодуються таким чином, який людина не в змозі розпізнати, але глибокі нейронні мережі виявляли

правильний клас об'єкта майже зі 100% точністю [10].

Для тренування дуже складних згорткових нейромереж на великих наборах даних, останнім часом, з'явилася тенденція до збільшення кількості шарів та розмірів шарів при використанні відсіву [34] для вирішення проблеми надмірного оснащення. Крижевський та Сегеди, творці двох найбільш відомих мереж, що використовуються в дослідженнях, а саме AlexNet [6] та GoogLeNet [7], підкреслюють, наскільки важливо використовувати високий коефіцієнт випадання під час тренувань.

Ще одна тенденція - зробити мережі дуже глибокими, тобто мережі мають багато шарів. Зейлер експериментував у з глибиною CNN і дійшов висновку, що продуктивність мережі сильно залежить від кількості шарів. Виміряна продуктивність значно знижується, навіть якщо видаляється один згортковий шар. Він зазначає, що глибина мережі важливіша, ніж будь-яка інша архітектурна складова мережі.

Отже, вибір базової архітектури є ключем до ефективності мережі. Це також можна побачити на GoogLeNet, який був опублікований у 2014 році та має глибину 22 шари, не рахуючи шарів об'єднання. У 2012 році AlexNet розпочав ажіотаж навколо прогресу в глибокому навчанні, зокрема, з мережею, що складається з 8 шарів. Хоча вона досягла найсучасніших результатів, коли вона була опублікована, відтепер багато інших мереж перевершили результативність.

Обмежуючим фактором розміру CNN є об'єм пам'яті, наявної в поточних одиницях обробки графічних процесорів (GPU), і переносний час на навчання. До того, як розпочали широко використовувати GPU для навчання глибоких мереж, загальний час навчання процесів був більшим у 9 разів. Бібліотека cuDNN від Nvidia, яка оптимізована для швидкої обробки зображень та ефективних операцій згортання в графічних процесорах, додатково прискорює час навчання з майже в 9 разів. Таким чином, сучасні графічні процесори з найновішими встановленими бібліотеками CUDA та cuDNN прискорюють час обчислення в 17 разів порівняно з сьогоднішніми

процесорами. Залежно від кількості параметрів мережі та розміру набору даних, навчання згорткової нейронної мережі на графічному процесорі з випадково ініційованими вагами може іноді займати кілька днів або навіть і тижнів.

Згорткова нейронна мережа (CNN) використовує варіацію багатошарових перцептронів (рис. 2.4). CNN містить один або декілька згорткових шарів. Ці шари можуть бути повністю пов'язані між собою або об'єднані.

Перш ніж передати результат наступному шару, згортковий шар використовує згорнуту операцію на вході. Завдяки цій згортковій роботі мережа може бути набагато глибшою, але зі значно меншими параметрами.

Завдяки цій функції перетворена нейронна мережа ефективно відображає результати в розпізнаванні зображень та відео, власній обробці голосу та системах переходу.

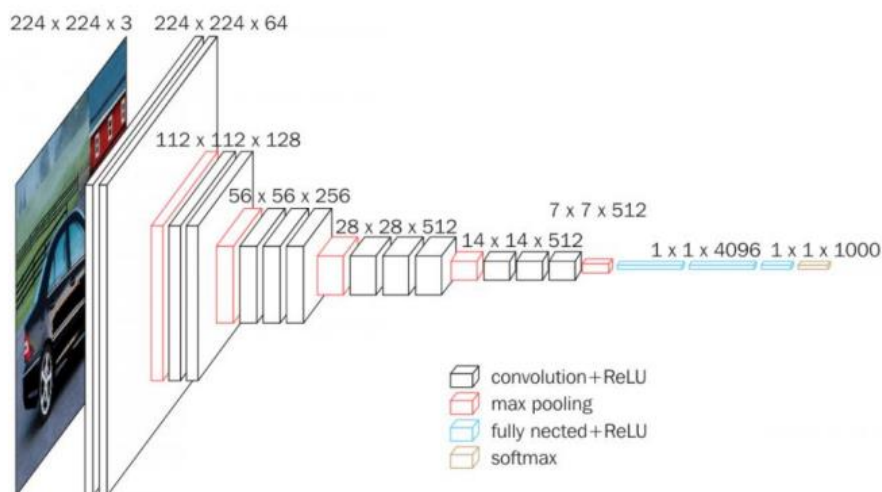


Рисунок 2.4 - Заміна архітектури нейронної мережі

Згорткові нейронні мережі показують чудові результати в семантичному розборі та виявленні парафрази. Вони також застосовуються при обробці сигналів та класифікації зображень, де відмінно себе

проявляють.

2.6 Рекурентна нейронна мережа

Рекурентна нейронна мережа - це тип штучної нейронної мережі, де виходи на певному рівні зберігаються і повертають вхід. Це допомагає передбачити результати занять.

Перший шар формується так само, як і в мережі подачі. Тобто продукт додає ваги та ознак. Однак у наступних шарах починається ітераційний процес у нейронних мережах.

З кожного кроку синхронізації до наступного кроку кожен вузол запам'ятовує інформацію, яку мав на попередньому кроці. Іншими словами, кожен вузол виконує функцію комірки при обчисленні та операції. Нейронні мережі починають розширюватися спочатку, як зазвичай, але запам'ятовування інформації, можливо, доведеться використовувати пізніше.

Якщо прогнозування є неправильним, система буде навчатися та працювати на основі правильного передбачення під час відтворення. Цей тип нейронної мережі є дуже ефективним у технології синтезу мовлення. Ось як виглядає рекурентна нейронна мережа (рис. 2.5).

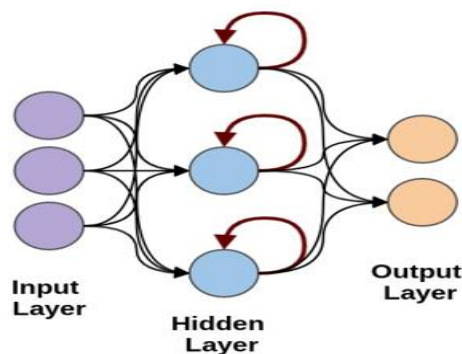


Рисунок 2.5 - Рекурентна нейронна мережа

2.7 Мережа Кохонена

Мережа була розроблена Тойво Кохоненом на початку 1980-х років і суттєво відрізняється від обговорених раніше тим, що вона використовує неконтрольований метод навчання та навчання, що включає лише значення вхідних змінних.

Мережа визначає кластери у навчальних даних та розподіляє їх у відповідні кластери. Якщо наступна мережа зустріне набір даних, який не відповідає жодному відомому шаблону, вона призначає його новому кластеру. Якщо дані містять теги класів, мережа може вирішити проблеми з класифікацією. Мережа Кохонена також може використовуватися у задачах, де класи відомі - перевагою мережі буде можливість виявлення подібності між різними класами.

Такі мережі добре підходять для завдань кластеризації, але не для прогнозів, тому вони не будуть детально обговорюватися.

Тому дослідження щодо використання нейронних мереж для прогнозування проблем показало можливість та перспективність використання цих методів.

У розділі 2 було розглянуто математичні основи нейронних мереж, а саме перцептрон, багатошаровий перцептрон, мережа Кохонена, згорткові нейронні мережі та рекурентні нейронні мережі. Був проведений аналіз методів, порівняльний аналіз, визначені переваги та недоліки. На основі цих методів для реалізації прогнозування часових рядів було обрано дві нейронні мережі, а саме згорткова нейронна мережа (CNN) та рекурентна нейронна мережа (RNN). Суттєвим є правильно обрані вхідні дані. В розділі 1 також зазначені інші підходи на прикладі відомих дослідницьких робіт в цій сфері.

3 РЕАЛІЗАЦІЯ МЕТОДІВ ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ

Цей розділ містить результати обчислень, пов'язаних із навчальним процесом, а також детальний опис прикладних досліджень та архітектуру програмного забезпечення.

Мовою програмування було обрано Python 3.0. Перевагою цього варіанту є наявність широкого вибору бібліотек і фреймворків для роботи з даними, а також стислість і читабельність кодів (особливо корисно для завдань машинного навчання).

Бібліотека Statsmodels використовується для застосування методів авторегресії та згладжування. Це бібліотека Python, яка дозволяє користувачам вивчати дані, оцінювати статистичні моделі та проводити статистичні тести. Statsmodels побудована поверх багатьох бібліотек NumPy та SciPy та інтегрується з бібліотекою даних Pandas.

Усі нейронні мережі реалізовані бібліотекою глибокого навчання Keras. Кожна побудована нейронна мережа (ANN) вимагає для навчання 100-200 епох.

3.1 Вхідні дані

Одним з найважливіших елементів побудови нейронної мережі є визначення даних, на яких мережа буде тренуватися. У цій роботі метою більшості цих мереж є як можна точніше спрогнозувати ціни на акції та вирішити, коли купувати чи продавати цінні папери на основі минулих ринкових даних. Завдання - вирішити, які показники та вхідні дані використовувати та зібрати достатню кількість навчальних даних для підготовки відповідної системи.

Вхідними даними можуть бути обсяг, ціна або щоденна зміна ціни, але ANN може також приймати вхідні дані як технічні показники (ковзна

середня, показник лінії тренду тощо) або базові показники (ціна власної акції, індекс економічного середовища тощо).

Хорошим прикладом нейронної мережі є система JSE [6], яка імітує роботу Йоганнесбурзької фондової біржі. Ця система має 63 показники з різних категорій, щоб отримати загальне уявлення про ринкове середовище. Значення цих 63 вхідних показників можна розділити на такі класи:

1. основні продажі, прибутковість, ціна;
2. технічні ковзні середні, тенденції продажів тощо;
3. індекси ринку JSE для різних галузей: золота, металів;
4. міжнародний промисловий індекс Доу Джонса та інші;
5. ціна / курс золота;
6. відсотки;
7. економічні показники загальний експорт та імпорт.

Хоча система ANN JSE врешті-решт прийняла всі 63 входи, аналіз показав, що багато з них були зайвими. Під час навчання автори цієї моделі видалили 20 змінних, зменшили кількість прихованих вузлів з 21 до 14, мало вплинули на продуктивність системи та скоротили час навчання.

Інші нейромереві системи можуть використовувати інші типи вхідних даних. Більш проста система може використовувати лише попередні ціни на акції. Система, розроблена Юном [7], базується на частоті ключових фраз, що використовуються у звіті акціонерів правління. Дослідники з Валчанського інституту технологій Джайбай, Аргіді та Апте поєднали цифровий аналіз даних та аналіз новинної інформації як вхідні дані для прогнозування вартості акцій [8].

Визначення правильного введення - це перший крок у навчанні мережі. Другим кроком є подання вхідних даних таким чином, що дозволяє мережі навчитися належним чином, не перетренуючись. У цій роботі, щоб забезпечити більш об'єктивну порівняльну характеристику класичних методів та методів прогнозування на основі використання нейронних мереж, попередні вхідні значення вибираються як вхідні значення.

3.2 Дані для дослідження

Для виконання цієї роботи дані про запаси Google завантажувались з офіційного веб-сайту NASDAQ за п'яти років, починаючи з 1 квітня 2014 р. [11]

NASDAQ (National Association of Securities Dealers Automated Quotation — Автоматизовані котирування Національної асоціації дилерів цінних паперів) - американська фондова біржа, яка спеціалізується на акціях високотехнологічних компаній (електроніка, програмне забезпечення тощо). Понад 3300 компаній зареєстровано на одній з трьох найбільших фондових бірж США. Члени Nasdaq - провідні компанії у таких галузях, як високі технології, комерція, телекомунікації, фінансові послуги, транспорт, медіа та біотехнології.

Набори даних - це щоденна інформація про ціни акцій на початок та кінець дня (рис. 3.1), максимальні та мінімальні ціни акцій за день та кількість проданих акцій (рис. 3.2).

	date	close	volume	open	high	low
0	5/10/19	1164.27	1314546	1162.38	1172.600	1142.50
1	5/9/19	1162.38	1185973	1166.27	1169.660	1150.85
2	5/8/19	1166.27	1309514	1174.10	1180.424	1165.74
3	5/7/19	1174.10	1551368	1189.39	1190.440	1161.04
4	5/6/19	1189.39	1563943	1185.40	1190.850	1166.26

Рисунок 3.1 - Вхідні дані



Рисунок 3.2 - Таблиця щоденних цін

3.3 Опис метрик

Загалом статистичні та нестатистичні дані використовуються для аналізу цінових проблем.

Нестатистичні показники ефективності включають річний прибуток для певної моделі, а також коефіцієнт звернень або кількість випадків, коли модель правильно прогнозує ріст або спад ринку.

Для аналізу використовуються різні статистичні дані, такі як середня абсолютна помилка (MAE), середньоквадратичний квадрат (RMSE), середнє квадратне прогнозування помилок (MSPE), коефіцієнт кореляції та прямокутний коефіцієнт кореляції, а також мінімальна остаточна помилка прогнозування Акайке.

У цій роботі аналізуються всі моделі та методи з використанням вимірювань MAE та MSE.

3.3.1 Середня абсолютна похибка (MAE)

MAE вимірює середню абсолютну похибку в наборі прогнозів для неперервних змінних.

Припустимо, що \hat{y}_t - прогноз значення часового ряду у періоді t , тоді метрика задається формулою:

$$MAE = \sum_t |y_t - \hat{y}_t| \quad (3.1)$$

MAE - це лінійна оцінка, що означає, що всі індивідуальні відмінності вимірюються в середньому однаково.

3.3.2 Середньоквадратична помилка (MSE)

MSE - це різниця між прогнозом та відповідними спостережуваними значеннями, кожним квадратом, а потім середнім значенням вибірки. Оскільки помилки є квадратними, перш ніж їх усереднювати, MSE робить великий акцент на великих помилках. Це означає, що MSE є найбільш корисним, коли великі помилки особливо небажані, що відповідає цілям цієї роботи.

Припускаючи, що \hat{y}_t - прогноз значення часового ряду у періоді t , тоді метрика задається формулою:

$$MSE = \sum_t (y_t - \hat{y}_t)^2 \quad (3.2)$$

Обидва метрики, MSE та MAE, можуть приймати значення 0 до ∞ і без урахування напрямку помилки. Чим нижче значення показника, тим точніше прогноз.

3.4 Реалізація методів згладжування

У цій роботі розглядаються п'ять методів згладжування, кожен з яких застосовується до різних параметрів. Кожен із наведених методів є ускладненням попереднього методу, тому очевидно, що можна отримати

більш точні результати.

Метод ковзного середнього використовується для різних значень параметра N за попередню кількість разів, врахованих при побудові прогнозу (рис. 3.3). Було підтверджено, що із зменшенням довжини вікна N модель показує більш точні результати на прототипі, вказуючи властивість останніх даних впливати на майбутні прогнози.

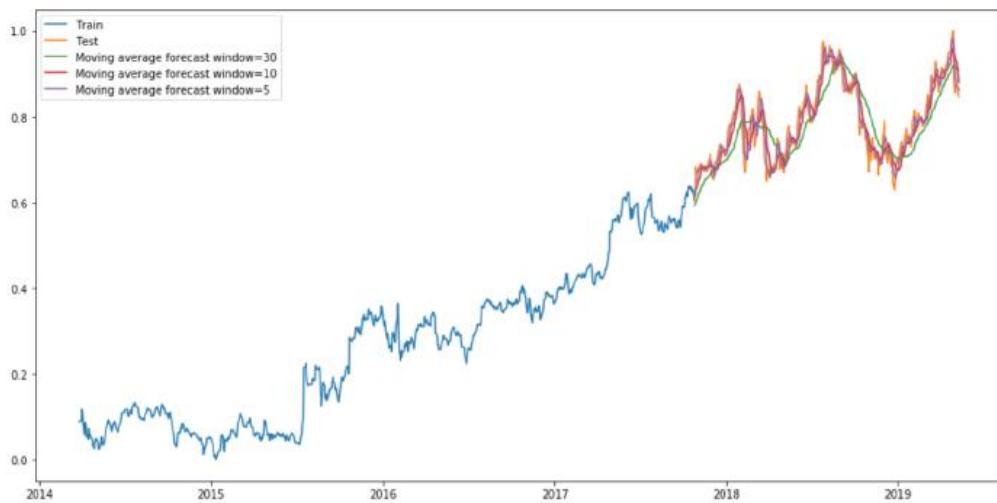


Рисунок 3.3 - Графік прогнозу методом простого ковзного середнього

Для методу зваженого середнього із бібліотеки Pandas був використаний метод ewm (рис. 3.4). Відповідно до документа, відповідні вагові коефіцієнти розраховується за такою формулою:

$$\alpha = 1 - e^{\frac{\log(0,5)}{half\ life}} \quad (3.3)$$

Тобто параметром налаштування моделі стало *half life*. Це дало наступні результати для різних значень *half life* :

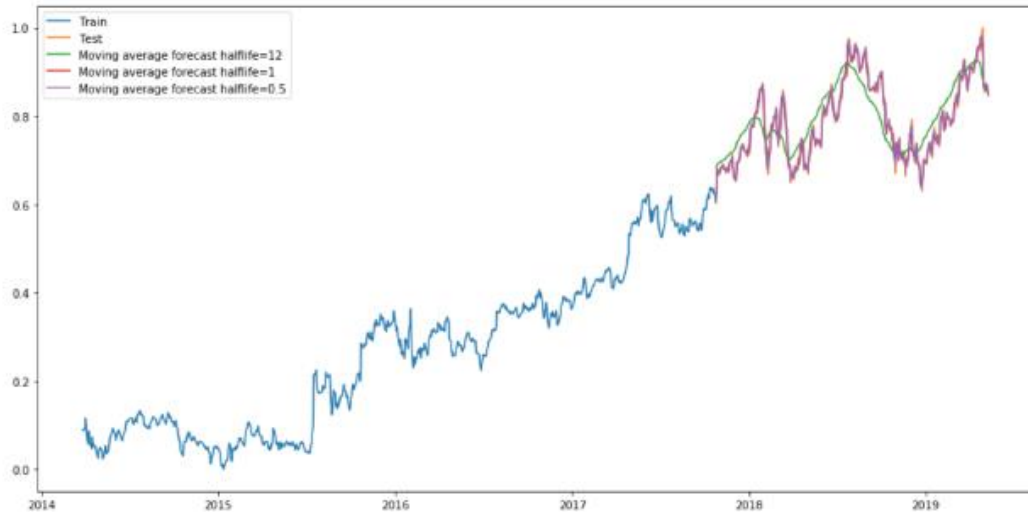


Рисунок 3.4 - Графік прогнозу методом зваженого ковзного середнього

Подібний підхід застосовується для експоненціального ковзного середнього (рис. 3.5), де рівень коефіцієнта α згладжування встановлюється як параметр, який є ступенем втрати ваги від 0 до 1 до.

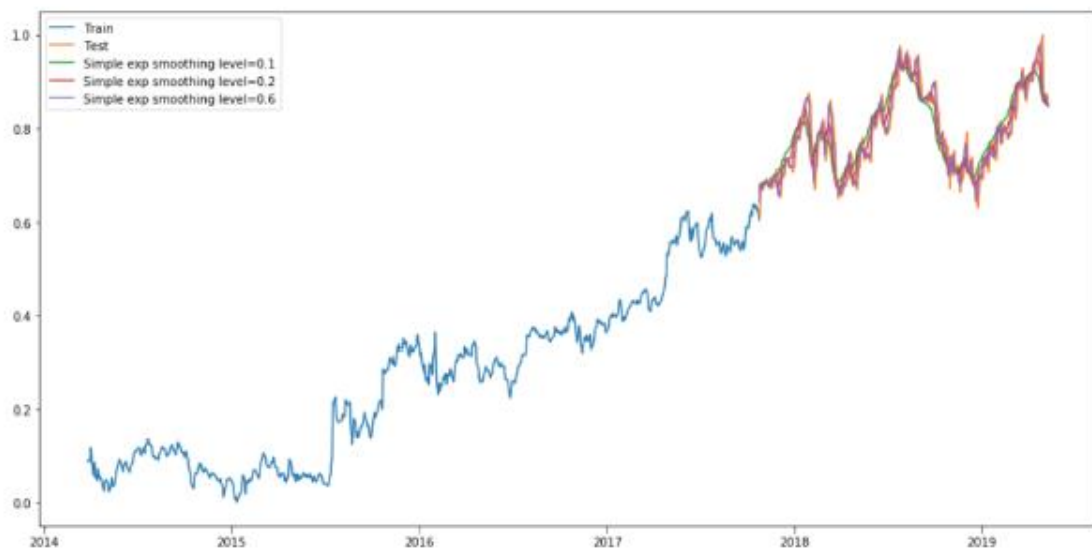


Рисунок 3.5 - Графік прогнозу методом експоненціального ковзного середнього

Чим нижчий рівень згладжування, тим точніше прогнозування, оскільки кожне попереднє значення важче.

Метод подвійного експоненціального згладжування передбачає встановлення додаткового параметра β , який відповідає за згладжування тренду (рис. 3.6). Комбінація пари α та β коригувало точність та якість прогнозу.

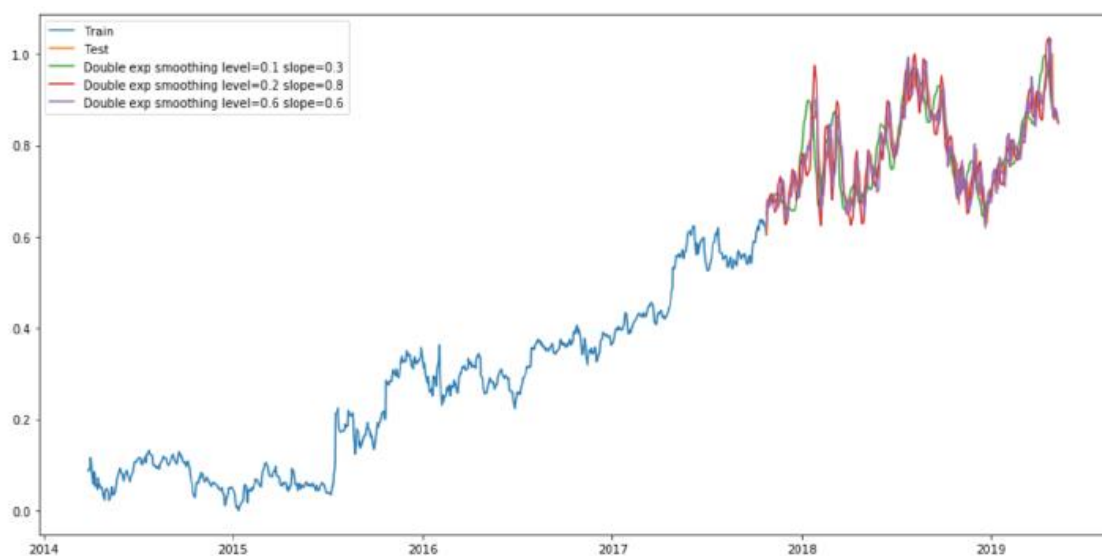


Рисунок 3.6 - Графік прогнозу методом подвійного експоненціального ковзного середнього

Як було зазначено у розділі 1, експоненціальне ковзне середнє або метод Голта-Вінтерса використовується як для тенденцій, так і для сезонних даних і додає коефіцієнт рівноваги для сезонної складової γ .

У роботі розглядається модель адитивних тенденцій та сезонної компоненти часових рядів. За допомогою функції `ExponentialSmoothing` в модулі `tsa.holtwinters` ви можете вказати модель відповідно до кількості сезонних періодів часу, доступних у даних. Отримані результати показані на рисунку 3.7.

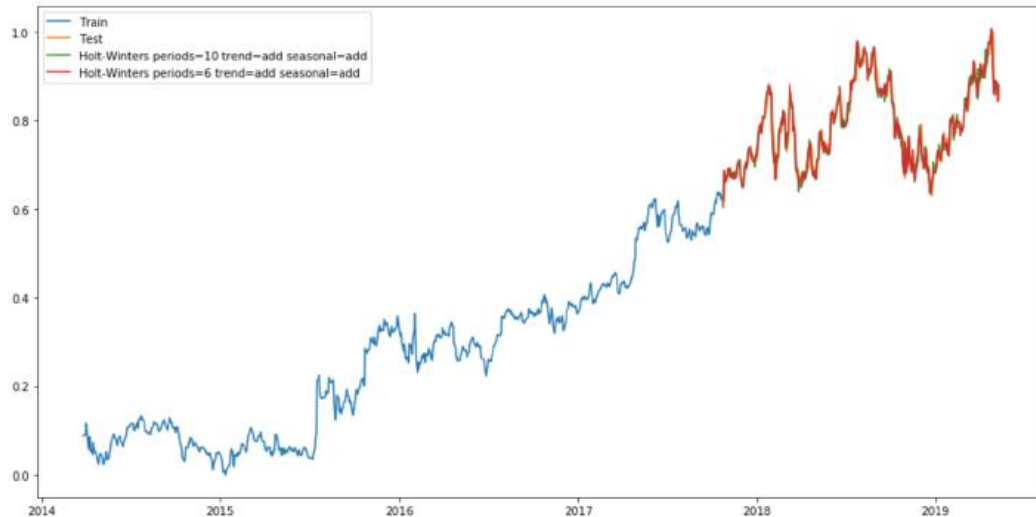


Рисунок 3.7 - Графік прогнозу методом Голта-Вінтерса

Отже, у цьому випадку зменшення кількості періодів призводить до більш точного прогнозу вартості акцій.

Результати всіх методів згладжування наведені в таблиці 3.1.

Таблиця 3.1 - Результати застосування методів згладжування

Метод	Характерні параметри	Test MSE	Test MAE
Ковзне середнє	Вікно N = 30	0.0029	0.0455
	Вікно N = 10	0.0012	0.0281
	Вікно N = 5	0.0005	0.0180
Зважене ковзне середнє	half-life = 12	0.0025	0.0419
	half-life = 1	0.0001	0.0089
	half-life = 0.5	0.000026	0.0037
Експоненціальне ковзне середнє	Рівень згладжування $\alpha = 0.1$	0.0018	0.0340
	Рівень згладжування $\alpha = 0.2$	0.0011	0.0259
	Рівень згладжування $\alpha = 0.6$	0.00049	0.0165

Продовження таблиці 3.1

Метод	Характерні параметри	Test MSE	Test MAE
Подвійне експоненціальне ковзне середнє	$\alpha = 0.1, \beta = 0.3$	0.00244	0.0388
	$\alpha = 0.2, \beta = 0.8$	0.00163	0.0322
	$\alpha = 0.6, \beta = 0.6$	0.00054	0.0177
Потрійне експоненціальне ковзне середнє	period = 10	0.00045	0.0162
	period = 6	0.00042	0.0154

3.5 Реалізація методів авторегресії

Як було зазначено в розділі 1, у роботі розглядаються 3 класичних методи авторегресії. Кожен із наступних методів давав кращі результати та мав більш складну математичну основу. Для цих методів коефіцієнти Акайке дозволяють програмному забезпеченню вибрати модель, яка забезпечує найкращий прогноз. Отже, у випадку цього набору даних виявлена модель ARIMA (2,1,1).

Результати всіх трьох методів зведені в порівняльну таблицю 3.2. З таблиці видно, що найкращу точність прогнозування часових рядів із використанням методу авторегресії дає модель інтегрованого ковзного середнього.

Таблиця 3.2 – Результати реалізації методів авторегресії

Метод	Характерні параметри	Test MSE	Test MAE
Проста модель авторегресії	$p = 2$	0.000347	0.013322
Модель авторегресії — ковзного середнього	$p = 3, q = 1$	0.000348	0.013346
Модель авторегресії — інтегрованого ковзного середнього	$p = 2, q = 1, d = 1$	0.000345	0.013335

3.6 Реалізація MLP

Головною функцією багатошарового перцептронну є її архітектура, а саме кількість прихованих шарів, вузлів та функцій активації. Крім того, оскільки результат навчання частково залежить від ініціалізації змінних, кожна модель навчається 5 разів окремо, і всі характеристики таблиці порівняння усереднюються. Для більш детального вивчення моделі було впроваджено 10 архітектур, їх ефективність показана в таблиці 3.3:

Таблиця 3.3 – Результати навчання MLP

#	Модель	Кількість параметрів	Training MAE	Training MSE	Test MAE	Test MSE	Алгоритм	Функція активації
1	MLP 20-10-1	221	0.0645	0.0074	0.090	0.014	adam	relu
2	MLP 20-20-1	441	0.0462	0.0038	0.052	0.005	adam	relu
3	MLP 20-100-1	2201	0.0438	0.0038	0.037	0.002	adam	relu
4	MLP 20-500-1	11001	0.0398	0.0032	0.036	0.002	adam	relu
5	MLP 30-500-1	16001	0.0387	0.0029	0.035	0.002	adam	relu
6	MLP 10-500-1	6001	0.0354	0.0025	0.034	0.002	adam	relu
7	MLP 20-100-1	2201	0.0035	0.0428	0.032	0.002	adam	tanh
8	MLP 20-100-1	2201	0.0593	0.0064	0.046	0.004	sgd	tanh
9	MLP 20-100-1	2201	0.0668	0.0078	0.093	0.014	sgd	relu
10	MLP 20-100-1	2201	0.0572	0.0064	0.059	0.005	rmsprop	relu

Відповідно до результатів у цій таблиці порівняння, ви можете

повністю відстежувати залежність якості прогнозу від кількості вузлів у прихованому шарі, методу оптимізації та функції активації. Модель MLP 20-100-1 дає найкращі результати, використовуючи метод оптимізації Adam та гіперболічний тангенс як активуючу функцію. Це графік ціни акцій, запропонованої цією моделлю (рис. 3.8), а також графік похибки прогнозу (рис. 3.9).



Рисунок 3.8 - Графік прогнозу MLP-20-100-1

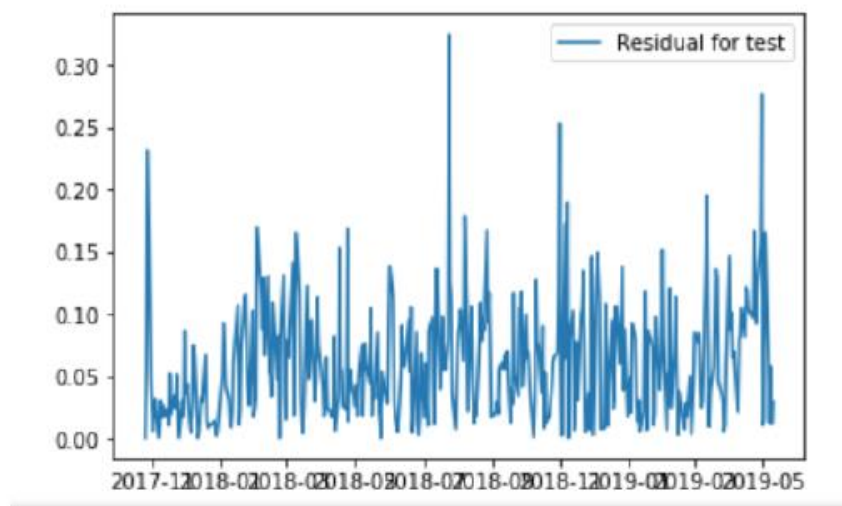


Рисунок 3.9 - Графік похибки MLP-20-100-1

3.7 Реалізація згорткових нейронних мереж (CNN)

На відміну від багатошарового перцептрона, згорткові нейромережі мають більш складну структуру, оскільки вони вимагають калібрування кількості та типу шарів, кількості вузлів у кожному шарі, методу оптимізації, функції активації та числа. кількості фільтрів. Таким чином, термін CNN-20-200-3 позначатиме 20 вхідних вузлів як перший шар, 200 фільтрів 3x3, шар MaxPooling і 2 fully-connected (повністю з'єднаних) шари. У цій роботі ми розглянули 7 різних архітектур CNN і дали наступні результати в таблиці 3.4:

Таблиця 3.4 – Результати навчання CNN

#	Модель	Кількість параметрів	Training MAE	Training MSE	Test MAE	Test MSE	Алгоритм	Функція активації
1	CNN 20-256-3	199,937	0.0212	9.4573e-04	0.1243	0.0216	adam	relu
2	CNN 20-256-5	201,367	0.0182	7.4266e-04	0.174	0.0388	adam	relu
3	CNN 10-256-3	198,657	0.0238	0.0011	0.096	0.0137	adam	Relu
4	CNN 20-256-3	199,937	0.0513	0.0050	0.29	0.110	sgd	relu
5	CNN 20-500-3	756,501	0.0515	0.0050	0.2592	0.0890	sgd	relu
6	CNN 20-300-3	273,901	0.0342	0.0021	0.121	0.0282	prmsprop	relu
7	CNN 20-300-3	273,901	0.0414	0.0030	0.255	0.079	prmsprop	tanh

Найкращі результати показує модель CNN 20-256-3 з методом оптимізації Adam та функцією активації ReLU. Це графік курсу акцій, пропонується цією моделлю (рис. 3.10).



Рисунок 3.10 - Графік прогнозу CNN 20-256-3

3.8 Реалізація рекурентних нейронних мереж (RNN)

Модель LSTM (Long Short-Term Memory Units) була обрана для використання рекурентних прогнозів нейронних мереж.

LSTM допомагає зберегти помилки, які можуть поширюватися з плином часу та класів. Підтримуючи більш постійні помилки, вони дозволяють більш детально вивчити повторювані мережі за кілька кроків [11]. LSTM містить інформацію за межами звичайного потоку повторюваної мережі в закритій комірці. Інформацію можна зберігати, записувати або читати з комірки, наприклад дані в пам'яті даних. Мережа вирішує, що і коли дозволяти читати, писати та видаляти через відкриті та закриті порти. На відміну від цифрової пам'яті комп'ютера, ці порти є аналоговими, використовуються при основному множенні сигмовидної лінії, всі в діапазоні 0 – 1. Ці порти працюють на основі сигналів. вони отримують і подібно до вузлів нейронної мережі, вони фіксують або передають інформацію на основі джерела та вводу. які вони фільтрують,

використовуючи власні ваги.

Для аналізу параметрів, що впливають на прогноз ціни акцій, було використано 5 варіантів архітектури LSTM, які зведені в таблицю 3.5 нижче. Символ LSTM-20-30-30-30-1 використовується для ідентифікації нейронних мереж LSTM, з початковими даними 20 днів тому, кількістю 30 вузлів на першому, другому та третьому шарі LSTM та одним шаром fully-connected (повністю пов'язаним) з отриманим виходом елемента.

Таблиця 3.5 – Результати навчання LSTM

#	Модель	Кількість параметрів	Training MAE	Training MSE	Test MAE	Test MSE	Алгоритм	Функція активації
1	LSTM 20-30-30-30-1	18,63 1	0.0074	1.0564e -04	0.019 8	0.000 5	adam	relu
2	LSTM 20-30-30-40-1	22,68 1	0.0079	1.8716e -04	0.018 8	0.000 5	adam	relu
3	LSTM 30-30-30-30-1	18,63 1	0.0106	1.9467e -04	0.015 9	0.000 4	adam	relu
4	LSTM 10-30-30-30-1	18,63 1	0.0099	1.7086e -04	0.015 0	0.000 3	adam	relu
5	LSTM 10-30-30-30-1	18,63 1	0.0082	1.2895e -04	0.015 22	0.000 36	sgd	relu

Найкращі результати показала модель LSTM 10-30-30-30-1 з методом оптимізації Adam та функцією активації ReLU та з мінімальною затримкою вікна. Це графік курсу вартості акцій, пропонується цією моделлю (рис. 3.11).

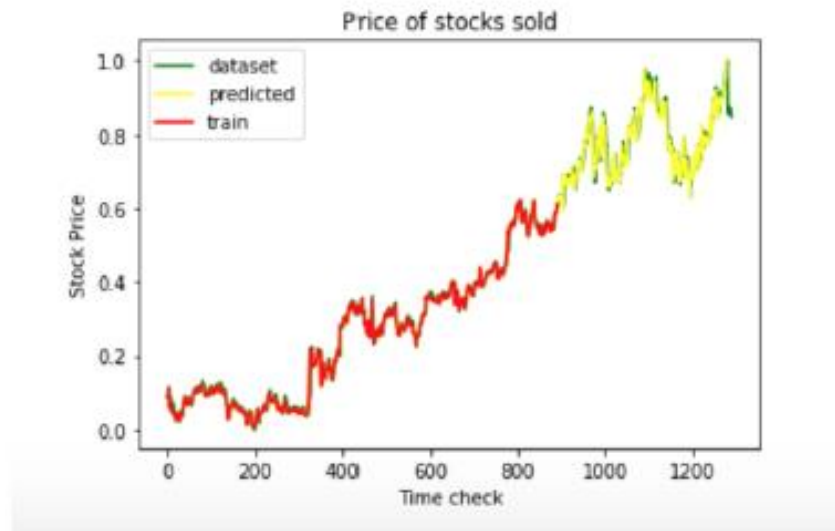


Рисунок 3.11 - Графік прогнозу LSTM 10-30-30-30-1

Порівняно з класичними методами, ця робота може підтвердити перспективи та можливість використання нейронних мереж для більш глибокого вивчення їх застосування на фінансових ринках.

Загалом, застосування класичних алгоритмів та алгоритмів машинного навчання у частині 3 може бути узагальнено в таблиці 3.6.

Таблиця 3.6 – Порівняльна таблиця реалізованих методів

Моделі	Переваги	Недоліки
Згладжування	Здатність обробляти тенденції змінних рівнів і компоненти сезонності	Вразливі до екстремальних значень
Авторегресія	Можна легко автоматизувати	Сильні обмеження в припущеннях
Штучні нейронні мережі - ANN	Можливість обробки складних нелінійних шаблонів. Висока точність прогнозу	Потребує велику кількість даних.

З порівняльної таблиці реалізованих методів можна побачити, що штучні нейронні мережі дають високу точність прогнозу та можливість обробки складних нелінійних шаблонів. Недоліком використання штучних нейронних мереж є те, що вони потребують велику кількість даних. Моделі авторегресії можна легко автоматизувати, але вони мають сильні обмеження в припущеннях. Перевагами моделей згладжування є здатність обробляти тенденції змінних рівнів і компоненти сезонності. Недоліком моделей згладжування є те, що вони вразливі до екстремальних значень.

4 ОБҐРУНТУВАННЯ ВИКОРИСТАННЯ ПРОГРАМНОГО ПРОДУКТУ

4.1 Функції програмного продукту

Основною функцією F_0 є використання програмного продукту, який аналізує вхідні дані та будує свою модель для прогнозування вартості акцій. Виходячи з конкретної мети, ми можемо виділити такі основні особливості програмного продукту:

- F_1 – вибір мови програмування;
- F_2 – вибір бібліотеки для глибинного навчання нейронних мереж;
- F_3 – візуалізація даних.

Кожна основна функція може мати ряд варіантів розподілу.

Функція F_1 :

- мова програмування Python;
- мова програмування R.

Функція F_2 :

- бібліотека Keras;
- бібліотека scikit-learn.

Функція F_3 :

- за допомогою переносу даних в Excel ;
- програмно за допомогою бібліотеки Matplotlib.

Варіанти реалізації основних функцій призводять до морфологічного відображення системи (рис. 4.1).

На основі цієї карти побудовано позитивну та негативну матриці для варіацій основних функцій, показаних у таблиці 4.1.

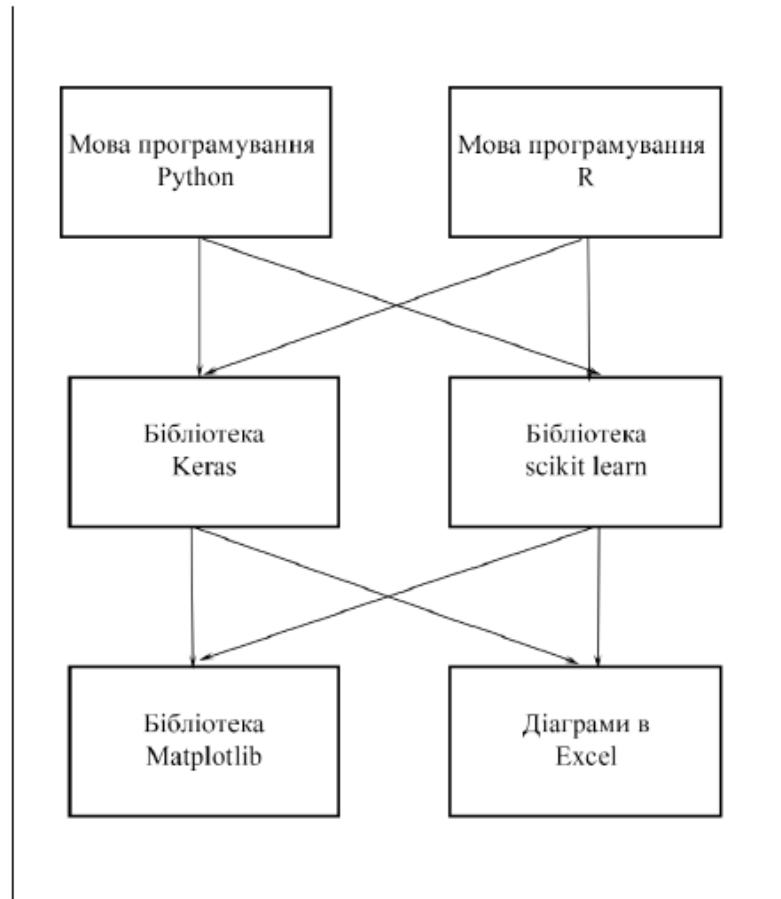


Рисунок 4.1 - Морфологічна карта

Таблиця 4.1 - Матриця інь і ян

Основні функції	Варіанти реалізації	Переваги	Недоліки
F1	А	Створювати нові моделі з нуля легко - розраховувати матрицю та оптимізувати бібліотеку даних	Менша бібліотека машинного навчання
	Б	Наявність великої кількості бібліотек	Кодування займає більше часу

Продовження таблиці 4.1

Основні функції	Варіанти реалізації	Переваги	Недоліки
F2	А	Широкий функціонал для глибинного навчання	Складніший у вивченні
	Б	Простий і недорогий у використанні навіть для початківців	Виконує лише основні функції
F3	А	Швидкість реалізації	Потрібні додаткові навички кодування
	Б	Зрозумілий інтерфейс	Обмежена функціональність

На основі аналізу позитивно-негативної матриці ми робимо висновок, що під час розробки програмного продукту деякі варіанти реалізації функцій слід опустити, оскільки вони не відповідають поставленим цілям перед програмним продуктом. Ці варіації позначені на морфологічній карті.

1. Функція F_1

Оскільки для розробки програмного забезпечення потрібен чіткий і зрозумілий код, дуже важливим фактором є наявність коду, що читається, щоб мінімізувати час, необхідний для написання та читання програми. Тому варіант б) слід відхилити для того, щоб задовольнити ці вимоги;

2. Функція F_2

Оскільки основним завданням продукту є розробка методів прогнозування, що вимагають швидкого застосування основних математичних операцій, використання бібліотек із широкими функціональними можливостями, що створює ці функції актуальною проблемою. Тому альтернативу б) слід відхилити;

3. Функція F_3

Досягнута візуалізація не відіграє великої ролі в цьому програмному продукті, оскільки якість зображення підтримується обома альтернативами, ми вважаємо, що варіанти а) та б) варто розглянути.

Тому ми розглянемо такі варіанти реалізації програмного продукту:

$$\begin{aligned} F_{1a} - F_{2a} - F_{3a} \\ F_{1a} - F_{2a} - F_{3b} \end{aligned} \quad (4.1)$$

Для оцінки якості ознак, які були оцінені, була обрана система параметрів, описана нижче.

4.2 Обґрунтування системних параметрів програмних продуктів

Існують вимоги до основних функцій, що виконуються в програмному продукті, що визначають основні параметри виробу, які будуть використовуватися в майбутньому для розрахунку коефіцієнтів технічного класу.

Потрібно ввести такі параметри:

X1 – точність розв'язку;

X2 – потенційний об'єм програмного коду;

X3 – час обробки даних алгоритмом;

X4 – кількість додатково імпортованих бібліотек;

X1: Показує точність реалізованого прогнозу, тобто відсоток похибки.
 X2: відображає кількість програмного коду, яку програмісти генерують безпосередньо. X4: Відображає час виконання програмного коду. X4: Відображає кількість додаткових бібліотек, необхідних для розповсюдження продукту.

Таблиця 4.2 - Основні параметри програмного продукту

Назва параметра	Умовні позначення	Одиниці виміру	Значення параметра		
			гірші	середні	кращі
Точність розв'язку	X1	%	10	5	1
Потенційний об'єм програмного коду	X2	кількість строк коду	1200	800	600
Час обробки даних алгоритмом	X3	с	1500	900	300
Кількість додатково імпортованих бібліотек	X4	одиниці	10	7	5

Згідно з таблицею 4.2 будуються графічні характеристики параметрів (рис. 4.2 - 4.5).

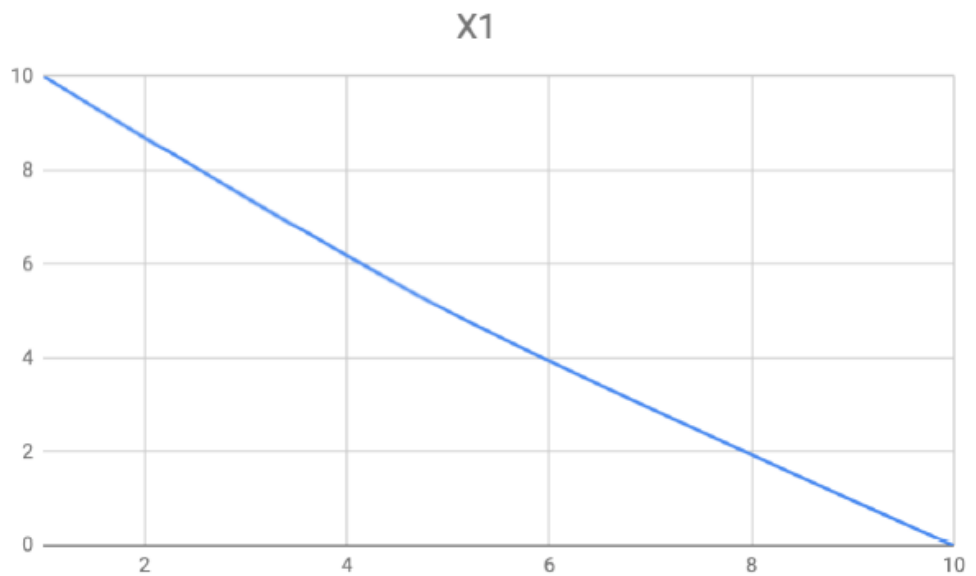


Рисунок 4.2 – X1, точність розв'язку.

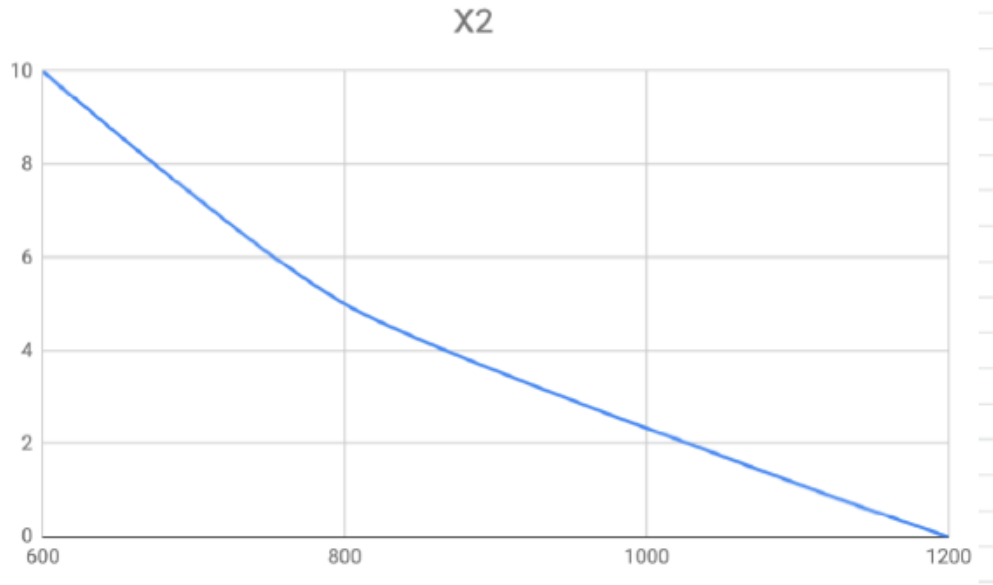


Рисунок 4.3 – X2, потенційний об'єм програмного коду.

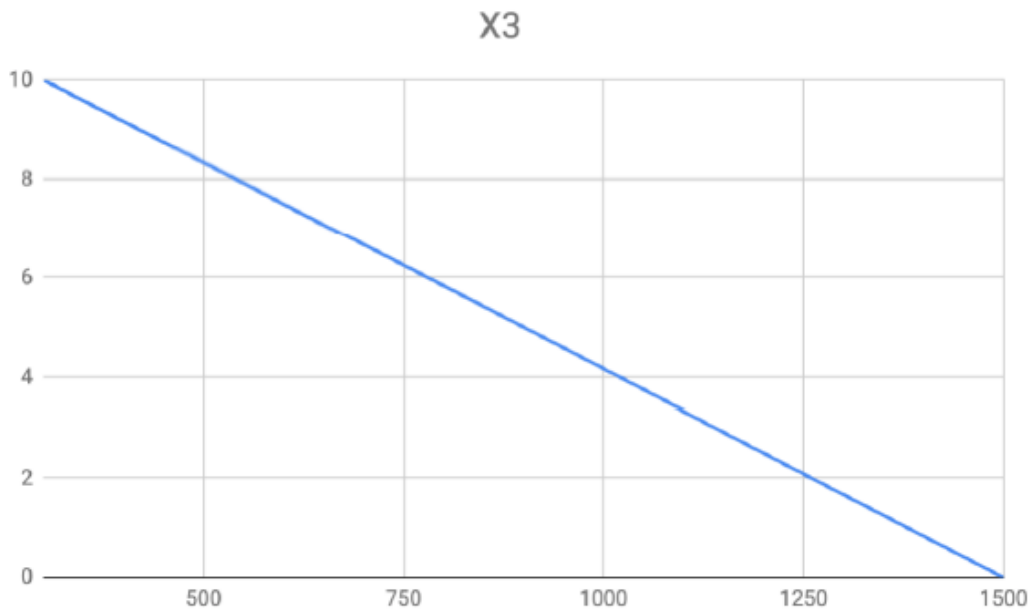


Рисунок 4.4 – X3, час обробки даних алгоритмом.

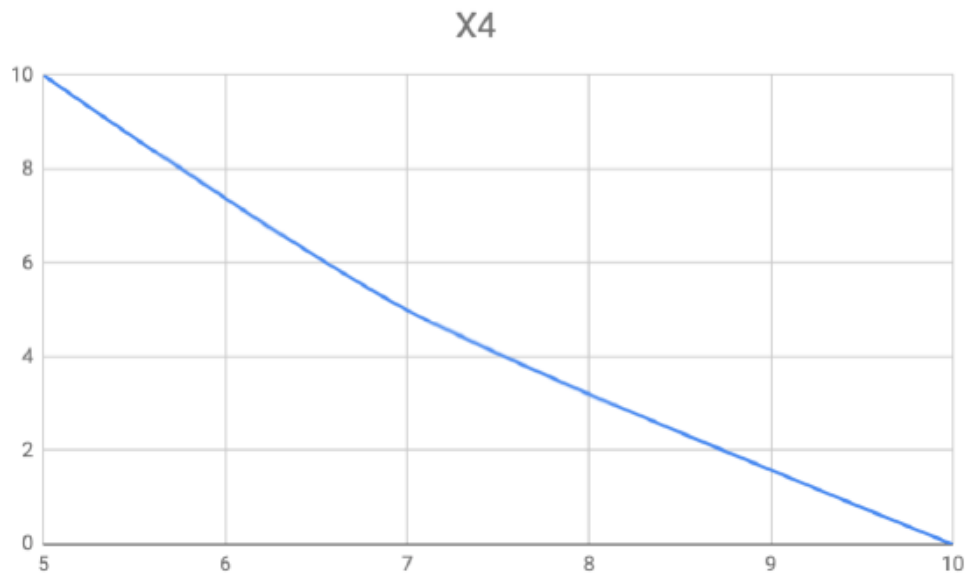


Рисунок 4.5 – X4, кількість додатково імпортованих бібліотек.

4.3 Аналіз експертної оцінки параметрів

Після детального обговорення та аналізу кожен експерт оцінює важливість кожного параметра для певної мети - розробки програмного продукту для досягнення найбільш точних результатів для пошуку параметрів моделі для адаптивного прогнозування та обчислення прогнозованих значень.

Рівень значущості кожного параметра визначається методом попарного порівняння. Визначення коефіцієнтів значимості включає:

- визначати значення параметрів, присвоюючи різні діапазони;
- Перевірити, чи експертні оцінки придатні для подальшого використання;
- визначити оцінку попарної пріоритетності параметрів;
- обробляти результати та визначати коефіцієнти значимості.

Результати експертної оцінки наведені в таблиці 4.3.

Таблиця 4.3 - Результати параметрів оцінки

Параметр	Назва параметра	Вимір	Ранг параметра за оцінкою експерта							Сума рангів	Δ_i	2 Δ_i
			1	2	3	4	5	6	7			
X1	Точність розв'язку	%	1	2	1	1	1	1	1	8	-9.5	90.25
X2	Потенційний об'єм програмного коду	кількість рядків коду	3	3	4	4	3	3	4	24	6.5	42.25
X3	Час обробки даних алгоритмом	с	2	1	2	3	2	2	2	14	-3.5	12.25
X4	Кількість додатково імпортованих бібліотек	одиниці	4	4	3	2	4	4	3	24	6.5	42.25
	Разом		10	10	10	10	10	10	10	70	0	187

Щоб перевірити надійність експертних оглядів, вирішіть наступне:

а) Сума рангів для кожного параметра та загальна кількість рангів обчислюється за такою формулою:

$$R_i = \sum_{j=1}^N r_{ij} R_{ij} = \frac{Nn(n+1)}{2} = 70, \quad (4.2)$$

де N – число експертів, n – кількість параметрів;

б) середня сума рангів обчислюється за формулою:

$$T = \frac{1}{n} R_{ij} = 17,5; \quad (4.3)$$

в) Відхилення суми рангів кожного параметра від середнього за рангом обчислюється за такою формулою:

$$\Delta_i R_i - T; \quad (4.4)$$

г) загальна сума квадратів відхилення розраховується за формулою:

$$S = \sum_{i=1}^N \Delta_i^2 = 187. \quad (4.5)$$

Розраховуємо коефіцієнт узгодження:

$$W = \frac{12S}{N^2(n^3-n)} = \frac{12 \cdot 187}{49(64-4)} = \frac{2244}{2940} = 0.76 > W_k = 0.67. \quad (4.6)$$

Порядок можна вважати надійним, оскільки було встановлено, що коефіцієнт узгодженості перевищує стандартний. Використовуючи результати оцінки, ми порівнюємо всі параметри в парах і вносимо результати в таблицю 4.4.

Таблиця 4.4 – Попарне порівняння параметрів

Параметри	Експерти							Кінцева оцінка	Числове значення
	1	2	3	4	5	6	7		
X1 і X2	<	<	<	<	<	<	<	<	0.5
X1 і X3	<	>	<	<	<	<	<	<	0.5
X1 і X4	<	<	<	<	<	<	<	<	0.5
X2 і X3	>	>	>	>	>	>	>	>	1.5
X2 і X4	<	<	>	>	<	<	>	<	0.5
X3 і X4	<	<	<	>	<	<	<	<	0.5

На основі чисельних оцінок переваги ми створимо матрицю $A = \|a_{ij}\|$.

Для кожного параметра ми розраховуємо вагу K_{B_i} за такою формулою:

$$K_{bi} = \frac{\sum_{i=1}^N a_{ij}}{\sum_{i=1}^N \sum_{i=1}^N a_{ij}}. \quad (4.7)$$

Відносні оцінки обчислюються кілька разів, поки наступні значення трохи не відрізняться від попередніх значень (менше 2%). На другому та наступних етапах відносні оцінки обчислюються за такими формулами:

$$K_{bi} = \frac{\sum_{i=1}^N a_{ij} b_j}{\sum_{i=1}^N \sum_{i=1}^N a_{ij} K_j}. \quad (4.8)$$

Як показано в таблиці 4.5, різниця між значеннями ваги не перевищує 2%, тому не доведеться повторювати її знову і знову.

Таблиця 4.5 - Розрахунок параметрів ваги

Параметри x _i	Параметри x _j				Перша ітерація		Друга ітерація		Третя ітерація	
	1	2	3	4	b _i	K _{bi}	1 b _i	1 K _{bi}	2 b _i	2 K _{bi}
X1	1	0.5	0.5	0.5	2.5	0.156	9.25	0.156	34.175	0.158
X2	1.5	1	1.5	0.5	4.5	0.281	16.25	0.275	59.125	0.273
X3	1.5	0.5	1	0.5	3.5	0.218	12.25	0.207	44.875	0.207
X4	1.5	1.5	1.5	1	5.5	0.345	21.25	0.36	77.875	0.36
Всього					16	1	59	1	216	1

4.4 Аналізування рівня якості відбору функцій

Ми визначаємо рівень якості для кожного варіанту, який виконує основні функції.

Таблиця 4.6 - Розрахунок показників якості для основних функцій

Основні функції	Варіант реалізації функції	Параметри	Абсолютне значення параметр а	Бальна оцінка параметра	Коефіцієнт вагомості параметра	Коефіцієнт рівня якості
F1	A	X4	6	9	0.36	3.24
F2	A	X1	4	8	0.158	1.264
		X3	900	5	0.207	1.035
F3	A	X2	700	6	0.273	1.638
	Б	X2	1000	3	0.273	0.819

Відповідно до таблиці 4.6, визначаємо рівень якості кожного варіанту:

$$K_{K1} = 3.24 + 1.264 + 1.035 + 1.638 = 7.177.$$

$$K_{K2} = 3.24 + 1.264 + 1.035 + 0.819 = 6.358. \quad (4.9)$$

Як видно з розрахунку, перший варіант кращий, де коефіцієнт технічної оцінки є найбільш важливим.

Таким чином, у розділі 4 було розглянуто функції програмного продукту, побудовано позитивно-негативну матрицю варіантів основних функцій. Було проведено ранжування, яке можна вважати надійними, оскільки було встановлено, що коефіцієнти узгодженості перевищують стандарти. Визначено рівень якості кожного варіанту виконання основних функцій.

ВИСНОВОК

У даній магістерській атестаційній роботі було розглянуто поширені методи прогнозування часових рядів та методи прогнозування часових рядів на основі використання штучних нейронних мереж, проведено їх порівняльний аналіз та визначено їх ефективність. На основі порівняльного аналізу було визначено напрямок дослідження та розроблено остаточне завдання.

Штучні нейронні мережі (ANN) мають перевагу прогнозування часових рядів, оскільки вони мають потенціал для вирішення складних прогнозних задач. Важливою особливістю штучних нейронних мереж, що беруть участь в задачах прогнозування часових рядів, є їх здатність до нелінійного моделювання без припущень щодо статистичного розподілу часових рядів. Кожна модель адаптивно формується на основі даних. З цієї причини штучні нейронні мережі комп'ютеризовані та самоадаптивні.

У цьому дослідженні порівнювалися показники прогнозування між нейромережевими та класичними методами прогнозування часових рядів. Було розглянуто математичні основи нейронних мереж, а саме перцептрон, багат шаровий перцептрон, згорткові нейронні мережі та рекурентні нейронні мережі. Був проведений аналіз методів прогнозування часових рядів, їх порівняльний аналіз, визначені переваги та недоліки. На основі цих методів для реалізації прогнозування часових рядів було обрано дві нейронні мережі, а саме згорткова нейронна мережа (CNN) та рекурентна нейронна мережа (RNN).

Моделювання проводилось у середовищі розробки Jupyter Notebook за допомогою мови програмування Python.

Аналіз показує, що моделі нейронних мереж, представлені в цьому дослідженні, демонструють кращу точність прогнозування, тим самим підтверджуючи перспективи та можливість використання штучних

нейронних мереж на фінансових ринках. На точність прогнозування моделі впливає безліч різних факторів, залежно від архітектури та вхідних даних.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Lawrence R. Using Neural Networks to Forecast Stock Market Prices. New York; London: Oxford University Press, 1997. 326 p.
2. Бідюк П.І. Аналіз часових рядів: навчальний посібник. К: Політехніка, 2010. 317 с.
3. Steel A. Predictions in Financial Time Series Data: Doctoral dissertation / University at Albany. Albany, NY, 2014. 192 p.
4. Adhikari R. An Introductory Study on Time Series Modeling and Forecasting.
5. Journal of Artificial Intelligence. 2009. Vol. 42, No. 5. P.856–874.
6. Nwankpa C., Ijomah W. Activation Functions: Comparison of Trends in Practice and Research for Deep Learning. Journal of Artificial Intelligence. 2018. Вип. 1 (9). URL: <https://arxiv.org/pdf/1811.03378.pdf> (дата звернення: 17.04.2019).
7. Robert J. Van Eyden The Application of Neural Networks in the Forecasting of Share Prices. New York: Finance and Technology Publishing, 1996. 326 p.
8. Yoon Y., G. Swales Applying Artificial Neural Networks to Investment Analysis. London: Taylor & Francis, 2011. 80 с.
9. Jaybhay K. Stock Market Prediction Model by Combining Numeric and News Textual Mining. International Journal of Computer Applications. 2012. Vol. 6, No. 19. P. 18–20.
10. Володин С. Н. Прогнозирование динамики курсовых стоимостей акций фондового рынка с применением резонансных систем искусственного интеллекта: дис. ... канд. техн. наук: 01.05.03 / Высшая школа экономики. Москва, 2017. 113 с
11. Гипотеза эффективного рынка: стаття. URL: <https://postnauka.ru/longreads/62491> (дата звернення: 25.10.2020).

12. Петерс Эдгар. Хаос и порядок на рынках капитала: учебное пособие. М.: Мир, 2000. – 336 с.

13. George Routledge Gibson. The stock exchanges of London, Paris, and New York: a comparison [Текст] / George Routledge Gibson // New York; London: G.P. Putnam, 1889. – 155 с.

14. Мэрфи Джон Дж. Технический анализ фьючерсных рынков. Теория и практика: учебное пособие. «Альпина Паблишер», 2011. – 616 с.

15. Efficient Market Hypothesis: Is The Stock Market Efficient? URL: <https://www.investopedia.com/articles/basics/04/022004.asp> (дата звернения: 01.11.2020).

16. Швагер Джек. Технический анализ. Полный курс: учебное пособие. «Альпина Паблишер», 2001. – 805 с.

17. No such thing as risk free: статья. URL: <https://www.economist.com/blogs/freeexchange/2011/11/financial-market> (дата звернения: 15.11.2020).

18. Подробно про технический анализ: статья. URL: <http://berg.com.ua/tech/technical-analysis-in-detail/> (дата звернения: 03.11.2020).

19. Что такое Фундаментальный анализ: статья. URL: <http://allfi.biz/glossary/eng/F/fundamentalanalysis.php> (дата звернения: 11.11.2020).