

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Штучного інтелекту
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

рівень вищої освіти другий (магістерський)

Прогнозування фінансових часових рядів за допомогою нейронних мереж
(тема)

Виконав:
студент 2 курсу, групи СШМ-22-3
Степаненко П.А.
(прізвище, ініціали)

Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма Системи штучного інтелекту
(повна назва спеціалізації)

Керівник ст.викл. Попов С.В.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____
(підпис)

В.О. Філатов
(прізвище, ініціали)

2024 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)
Кафедра Штучного інтелекту
(повна назва)
Рівень вищої освіти другий (магістерський)
Спеціальність 122 Комп'ютерні науки
(код і повна назва)
Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)
Освітня програма Системи штучного інтелекту
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
« _____ » _____ 20 ____ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Степаненку Павлу Андрійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Прогнозування фінансових часових рядів за допомогою нейронних мереж

затверджена наказом університету від 1 квітня 20 24 р. № 260Ст

2. Термін подання студентом роботи до екзаменаційної комісії 13 червня 20 24 р.

3. Вихідні дані до роботи Науково-технічні публікації та дані різних печатних та інтернет-джерел щодо тематики кваліфікаційної роботи

4. Перелік питань, що потрібно опрацювати в роботі _____

1) Актуальність завдання прогнозування часових рядів _____

2) Аналіз тематики фінансових часових рядів та методів їх прогнозування _____

3) Огляд обраних нейронних мереж для прогнозування часових рядів _____

4) Проектування нейронних мереж для прогнозування фінансових часових рядів _____

5) Порівняння результатів прогнозування нейронних мереж _____

РЕФЕРАТ

Пояснювальна записка: 77 с., 32 рис., 17 табл., 1 дод., 36 джерел.

БАГАТОШАРОВИЙ ПЕРЦЕПТРОН, ДОВГА КОРОТКОСТРОКОВА ПАМ'ЯТЬ, КЛАСИФІКАЦІЯ, МАШИННЕ НАВЧАННЯ, ПРОГНОЗУВАННЯ, РАДІАЛЬНО-БАЗИСНІ ФУНКЦІЇ, РЕГРЕСІЯ, ФІНАНСОВІ ЧАСОВІ РЯДИ, ШТУЧНІ НЕЙРОННІ МЕРЕЖІ.

Об'єкт дослідження – процес прогнозування та аналізу фінансових часових рядів.

Предмет дослідження – програмні засоби та методи аналізу і прогнозування фінансових часових рядів.

Мета роботи – дослідження процесу прогнозування фінансових часових рядів за допомогою нейронних мереж. Проектування нейронних мереж для прогнозування фінансових часових рядів та їх порівняння.

Методи дослідження – дослідження наявних наукових робіт з дотичних тем, аналіз існуючих методів прогнозування, пошук даних для прогнозування та експериментальні дослідження.

У результаті роботи проведено теоретичний аналіз завдання прогнозування, предметної сфери фінансових часових рядів, архітектур нейронних мереж, методи їх оптимізації та метрики якості для їх оцінювання. Для навчання та тестування роботи реалізованих нейронних мереж було використано дані котирувань євро-долар за останні 10 років. В результаті роботи було проаналізовано якість прогнозування часового ряду різними нейронними мережами та зроблено висновки щодо подальших можливостей їх покращення.

ABSTRACT

Master's thesis contains: 77 pp., 32 fig., 17 tabl., 1 ann., 36 references.

ARTIFICIAL NEURAL NETWORKS, CLASSIFICATION, FINANCIAL TIME SERIES, FORECASTING, LONG SHORT-TERM MEMORY, MACHINE LEARNING, MULTILAYER PERCEPTRON, RADIAL BASIS FUNCTIONS, REGRESSION.

The object of research is process of forecasting and analyzing financial time series.

The subject of research is different software tools and methods for analyzing and forecasting financial time series.

The study's purpose is to research the process of forecasting financial time series using neural networks. Designing neural networks for forecasting financial time series and comparing them.

Research methods include an examination of existing scientific works on related topics, analysis of existing forecasting methods, data collection for forecasting, and experimental research.

As a result of the work, a theoretical analysis of the forecasting task, the subject area of financial time series, neural network architectures, methods of their optimization, and quality metrics for their evaluation was conducted. For training and testing the implemented neural networks, data from EUR-USD quotes over the past 10 years were used. As a result of the work, the forecasting quality of the time series by different neural networks was analyzed, and conclusions were drawn regarding further possibilities for their improvement.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів	8
Вступ.....	9
1 Аналіз предметної галузі та постановка задачі.....	11
1.1 Актуальність завдання прогнозування	11
1.2 Часові ряди.....	13
1.2.1 Поняття часового ряду	13
1.2.2 Типи часових рядів	15
1.3 Предметна сфера фінансових часових рядів	17
1.4 Існуючі інструменти прогнозування ЧР	21
1.5 Загальні відомості про ШНМ.....	24
1.6 Постановка задачі дослідження.....	28
2 Прогнозування часових рядів за допомогою нейронних мереж.....	30
2.1 Завдання прогнозування ЧР	30
2.1.1 Завдання класифікації напрямку тренду ЧР	30
2.1.2 Завдання регресії ЧР	31
2.2 Метрики якості та функції втрат	32
2.2.1 Середня квадратична помилка (MSE)	32
2.2.2 Середня абсолютна помилка (MAE).....	33
2.2.3 Крос-ентропійна функція втрат (Cross-Entropy).....	34
2.2.4 Точність класифікації (Accuracy).....	35
2.3 Мережа радіально-базисних функцій (RBF).....	36
2.4 Багатошаровий перцептрон (MLP).....	39
2.5 Мережа довгої короткострокової пам'яті (LSTM)	41
2.6 Навчання нейронних мереж.....	47
3 Результати прогнозування фінансових часових рядів	51
3.1 Алгоритми отримання прогнозу та навчання нейронних мереж.....	51
3.2 Мережа радіально-базисних функцій (RBF).....	53
3.2.1 Архітектура мережі.....	53

3.2.2 Навчання мережі	54
3.2.3 Результати роботи мережі.....	55
3.3 Багатошаровий перцептрон.....	57
3.3.1 Архітектура мережі.....	57
3.3.2 Навчання мережі	59
3.3.3 Результат роботи мережі	60
3.4 Мережа довгої короткострокової пам'яті (LSTM)	61
3.4.1 Архітектура мережі.....	61
3.4.2 Навчання мережі	63
3.4.3 Результат роботи мережі	64
3.5 Порівняльний аналіз результатів прогнозування	65
3.5.1 Класифікація.....	65
3.5.2 Регресія.....	66
3.6 Висновки по розділу	69
Висновки	71
Перелік джерел посилання	73
Додаток А Відомість кваліфікаційної роботи	77

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ЧР – часовий ряд;

ШІ – штучний інтелект;

ШНМ – штучна нейронна мережа;

AI – Artificial Intelligence – штучний інтелект;

CNN – Convolutional Neural Network – згорткова нейронна мережа;

LSTM – Long Short-Term Memory – довга короткострокова пам'ять;

MLP – Multilayer Perceptron – багатошаровий перцептрон;

RBF – Radial Basis Function – радіально-базисна функція;

RNN – Recurrent Neural Network – рекуррентна нейронна мережа.

ВСТУП

Протягом всієї історії людство намагалося поглянути у своє майбутнє. У середні віки люди вгадували, який буде урожай на наступний рік, чекати чи не чекати природних лих або епідемій. Часто ці передбачення були зовсім неточними, проте з часом спостереження за різними процесами приводили до виявлення закономірностей. Наприклад, випадково виявлені природні залежності відображалися в прикметах та інших фольклорних елементах. Таким чином, одні з перших достовірних прогнозів люди навчилися складати для більш ефективного сільського господарства.

Проте у сучасному світі точні прогнози потрібні не лише для ведення сільського господарства, а й для організації будь-якого виробництва, створення прибуткової економіки, підтримки якісного здоров'я, визначення політичного курсу чи для успішної роботи будь-якої іншої сфери діяльності людини. У зв'язку з такими широкими потребами в точних прогнозах люди почали виділяти цілий клас задач прогнозування, методи і способи вирішення яких досліджуються й досі.

Завдання прогнозування поведінки певної системи є актуальним у сучасному світі. Це може бути прогноз погоди, прогноз активності користувачів на сайті або прогноз зміни цін акцій компанії на біржі. Прогнозування є одним із початкових етапів планування різних напрямків (економічного, технічного). Оцінка можливої поведінки системи дозволяє приймати зважені рішення, а також підвищити ефективність виконуваної діяльності та знизити ризики невдачі.

Фінансові ринки є складними системами, на які впливають різноманітні фактори, такі як економічні коливання, політичні події, технологічні зміни та соціокультурні тренди. Ця постійна непередбачуваність створює значні виклики для учасників ринку, оскільки їм потрібно приймати важливі рішення щодо інвестування та торгівлі на основі прогнозів майбутнього руху цін. В умовах такої складності і

нестабільності прогнозування фінансових часових рядів стає критичним елементом для успішного функціонування ринку.

Одним із перспективних підходів до прогнозування фінансових рядів є використання нейронних мереж, які входять до сфери штучного інтелекту. Нейронні мережі мають унікальну здатність до адаптації до складних залежностей у даних та можуть виявити та використати навіть найтонші патерни, що призводить до створення більш точних та надійних прогнозів. Використання нейронних мереж для прогнозування фінансових рядів вже є невід'ємною складовою сучасного фінансового аналізу та прийняття рішень.

Дана дипломна робота спрямована на дослідження та розробку методів прогнозування фінансових часових рядів за допомогою нейронних мереж. Вона охопить аналіз існуючих підходів до прогнозування фінансових рядів, вивчення основних принципів роботи нейронних мереж та їх застосування у фінансовому аналізі. Також робота буде вдосконалена практичною частиною, яка включатиме застосування нейронних мереж для прогнозування фінансових часових рядів на реальних прикладах.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Актуальність завдання прогнозування

Прогнозування є важливим завданням у багатьох областях, включаючи медицину, кліматологію, екологію, транспорт, фінанси, економіку тощо.

У медицині прогнозування може бути використане для прогнозування захворювань, ефективності лікування. Наприклад, прогнозування поширення захворювань може допомогти керуючим організаціям у плануванні необхідної кількості медичних працівників та обладнання. Також воно може допомогти у вирішенні питань, пов'язаних з дистанційним моніторингом стану пацієнтів та запобіганням ускладненням [1].

У кліматології прогнозування може бути використане для визначення запобіжних заходів у разі несприятливих погодних умов, таких як урагани, торнадо, посуха або повені. Наприклад, прогнози погоди можуть допомогти урядам та приватним організаціям приймати рішення про закупівлю продуктів харчування та ліків, організацію евакуації населення [2].

В екології прогнозування може бути використане для дослідження динаміки забруднення повітря, води та ґрунту, аналізу процесів зміни клімату та екологічних параметрів. Наприклад, прогнозування забруднення повітря дозволяє державним органам вживати запобіжних заходів для зменшення викидів забруднюючих речовин, а також організації екстрених заходів при надзвичайних ситуаціях. Також прогнозування клімату може допомогти у прийнятті рішень щодо сільськогосподарського виробництва та інших галузей, що залежать від погодних умов [3].

У транспорті прогнозування використовується для оптимізації розкладу та управління транспортними потоками. Наприклад, дані про трафік та використання громадського транспорту можуть бути застосовані

для прогнозування кількості пасажирів у майбутньому та оптимізації маршрутів та розкладу громадського транспорту [4].

У фінансовій сфері прогнозування є невід'ємною частиною інвестиційного аналізу [5]. Воно допомагає інвесторам та трейдерам приймати обґрунтовані рішення щодо купівлі або продажу активів на фондовому ринку. Наприклад, прогнозування цін на акції, валюти чи товари може допомогти інвесторам визначити, коли краще купити чи продати активи. Також прогнозування рядів може допомогти в управлінні ризиками, оскільки інвестори можуть використати прогнози, щоб ухвалити рішення про те, скільки грошей слід інвестувати у кожен актив [5].

В економіці прогнозування є необхідним інструментом для планування бізнес-стратегії та бюджетування. Прогнози економічних показників, таких як інфляція чи безробіття, допомагають урядам та підприємствам приймати рішення про витрати та доходи, а також визначати, коли необхідно змінювати стратегію бізнесу. Наприклад, прогнози економічних показників можуть допомогти компаніям приймати рішення про розширення бізнесу або, навпаки, коли слід закривати частину підприємства [6].

Перші прогнозуючі моделі з'явилися саме для розв'язання економічних задач, таких як прогнозування динаміки фондового ринку, рівня попиту та пропозиції, зростання цін на товари та послуги, інвестиційних доходів і ризиків тощо. Ефективне розв'язання кожної з описаних вище задач необхідне для сучасної економіки. Саме тому з'являється все більше досліджень у галузі машинного навчання, спрямованих на складання економічних прогнозів. Сотні компаній по всьому світу використовують ці дослідження, щоб збільшити свій прибуток. Навіть держави планують свою економіку, базуючись на таких прогнозах.

Одним із найважливіших економічних завдань є прогнозування динаміки фондових ринків, робота яких напряму впливає не лише на великі корпорації, але й на добробут країн. Аналізу роботи фондових ринків

присвячено багато досліджень, що використовують цілий арсенал методів математичної статистики. Однак саме дослідження в галузі машинного навчання призвели до цікавих відкриттів. Одне з таких досліджень було присвячене прогнозуванню цін акцій на двох найбільших ринках: Нью-Йоркській фондовій біржі та Національній фондовій біржі Індії [7]. В результаті роботи було зроблено висновок, що існує єдина базова динаміка, спільна для обох ринків. До таких висновків привів детальний аналіз чотирьох моделей нейронних мереж: багатошарового перцептрона (MLP), рекурентної нейронної мережі (RNN), мережі довгої короткочасної пам'яті, заснованої на рекурентних нейронних мережах (LSTM), а також згорткової нейронної мережі (CNN). Також прогнози для часових рядів, зроблені за допомогою цих нейронних мереж, виявилися точнішими за прогнози, складені за допомогою інтегрованої моделі авторегресії ковзного середнього (ARIMA).

1.2 Часові ряди

1.2.1 Поняття часового ряду

Поняття часового ряду визначає деякий набір інформації, зібраний за певний проміжок часу, стан процесу, що підлягає дослідженню. Часові ряди можуть бути використані для аналізу та прогнозування поведінки систем у часі. Вони можуть бути представлені у вигляді графіків або таблиць [8].

На відміну від вибірки, дані в якій ніяк не залежать один від одного, часовий ряд залежить від двох параметрів: періоду часу, за який відбувалися зміни аналізованого процесу, та числових показників, що характеризують ці зміни. Таким чином, методи, що використовуються для аналізу вибірок даних, не застосовуються для роботи з часовими рядами. Спостереження за зміною показника в певний проміжок часу називається рівнем ряду.

У фінансовій системі у вигляді часового ряду записуються, наприклад, зміни котирувань акцій компанії за певний проміжок часу. На рисунку 1.1 наведено приклад графіка часового ряду. На цьому зображенні показано графік зміни індексу S&P 500 за один торговий день на фондовій біржі.

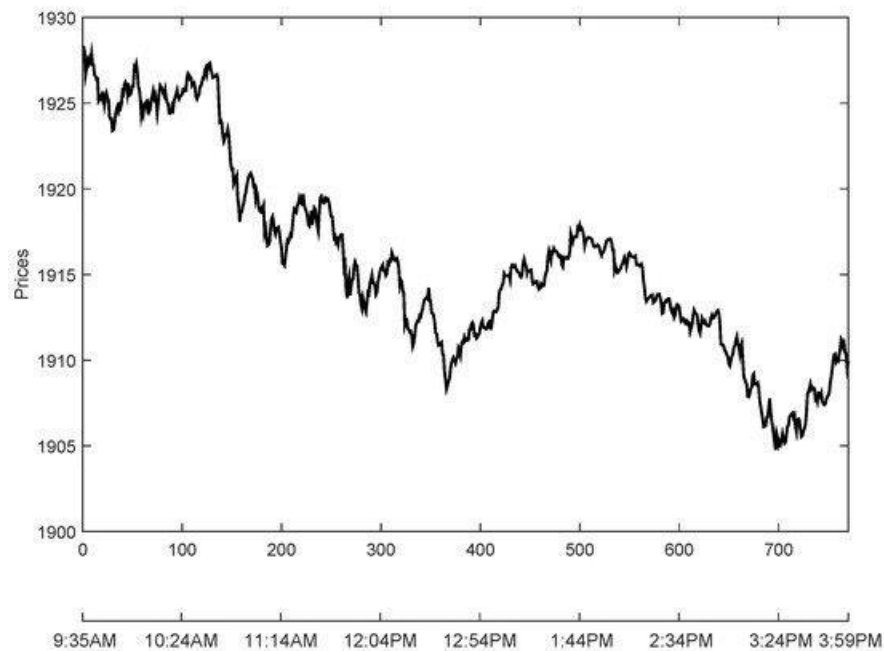


Рисунок 1.1 – Графік зміни індексу S&P 500 за один торговий день на фондовій біржі

Над часовими рядами можна проводити аналіз за допомогою певних сукупностей методів математичного та статистичного аналізу, при цьому даний аналіз може бути спрямований на виявлення наявності можливої структури часових рядів і, в результаті, їх прогнозування.

Аналіз часових рядів складається з виявлення трендів ряду (визначення закономірностей зміни спостережуваного процесу), пошук сезонних компонент (повторюваних циклів), пошук аномальних значень (рівнів, які мають вплив на значення ряду, але при цьому не відповідають можливостям досліджуваної системи, тобто, помилок

технічного або іншого характеру) та розкладання ряду на складові, описані вище: тренд, сезонність та помилку.

Тренд являється систематичною компонентною часового ряду, яка може змінюватися з часом. Трендом називаються не випадкову функцію, яка формується під дією загальних або довготривалих тенденцій, які впливають на часовий ряд. Прикладом тенденції може бути, наприклад, фактор зросту досліджуваного ринку.

При аналізі тренду виділяють чотири його основних типи:

- поліноміальний тренд використовують для опису часових рядів з плавною, повільною динамікою, що змінюється з часом [9];
- експоненційний тренд доцільно використовувати для швидко зростаючих часових рядів [9];
- гармонійний тренд виправдано, якщо в поведінці часового ряду проглядається періодичність [10];
- тренд, виражений логістичною функцією відрізняється від інших тим, що має асимптоту. Часто зустрічається при аналізі часових рядів демографічних показників [10].

1.2.2 Типи часових рядів

Часові ряди класифікуються за наявності основної тенденції на стаціонарні та нестаціонарні. У стаціонарних рядах основний тренд відсутній і математичне очікування такого ряду дорівнює певній константі. На рисунку 1.2 продемонстровано приклад стаціонарного часового ряду.

У нестаціонарних рядах можна виділити тенденцію до розвитку ряду, наприклад, зростаючий, спадаючий ряд або деякі циклічні явища [11]. На рисунку 1.3 представлений приклад нестаціонарного часового ряду зі спадною тенденцією.

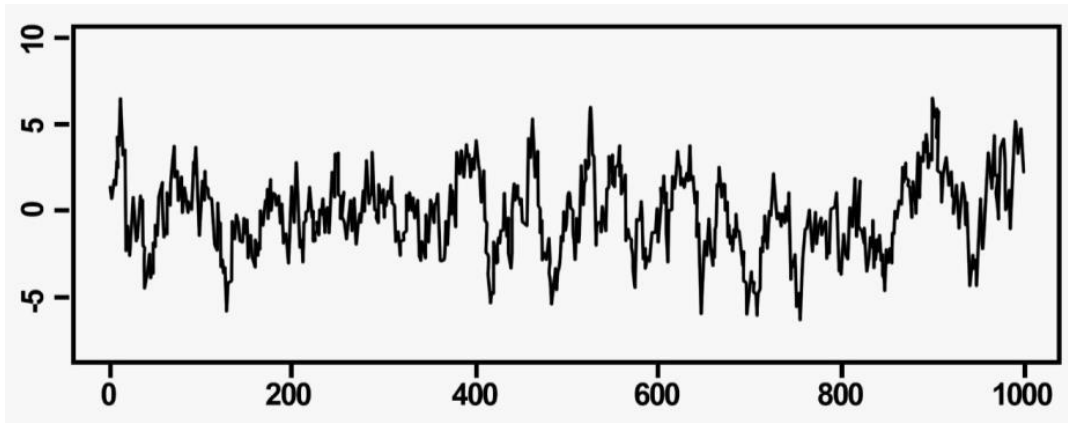


Рисунок 1.2 – Приклад стаціонарного часового ряду

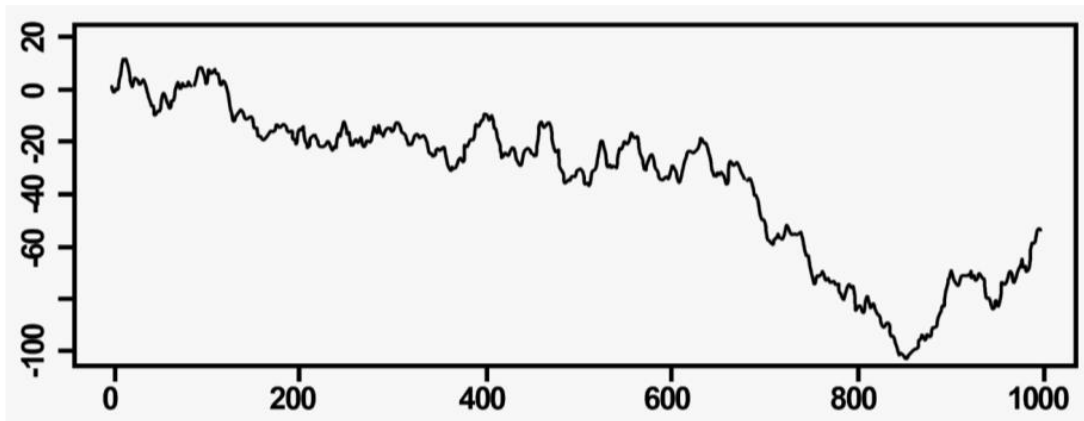


Рисунок 1.3 – Приклад нестационарного спадного часового ряду

За наявності сезонної компоненти часові ряди можна розділити на ряди із сезонністю та без сезонності. Сезонною компонентою називають деяке періодичне явище, що виникає у даних у межах деякого періоду, і є серією циклічних процесів, що характеризуються сезонними явищами. Наприклад, кількість замовлень та загальна активність покупців інтернет-магазину різко збільшуватиметься у передсвяткові періоди та спадатиме після закінчення свят. Часовий ряд активності покупців інтернет-магазину матиме сезонність. На рисунку 1.4 представлений приклад графіка часового ряду із сезонністю.

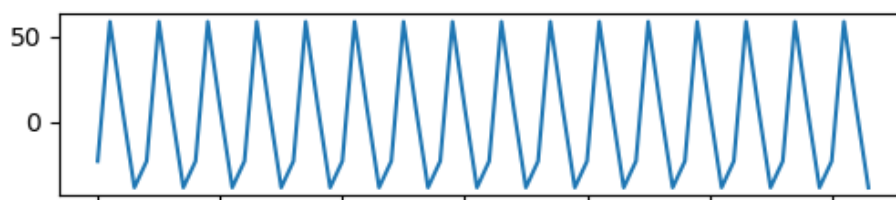


Рисунок 1.4 – Приклад часового ряду із сезонністю

На рисунку 1.5 наведено приклад графіка часового ряду без сезонності. Зміни його значень не обумовлені залежно від будь-якого сезонного явища і є випадковими.

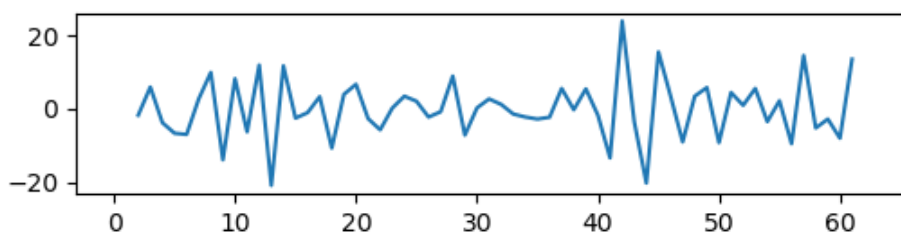


Рисунок 1.5 – Приклад часового ряду без сезонності

Часові ряди можуть мати кілька характеристик одночасно, наприклад, часовий ряд може мати сезонність і бути зростаючим.

1.3 Предметна сфера фінансових часових рядів

Предметна сфера фінансових часових рядів включає в себе вивчення динаміки фінансових показників у часі та їх аналіз з метою прогнозування майбутніх значень. Цей напрямок досліджень має велике значення для фінансової сфери, оскільки відомості про тенденції зміни різних фінансових показників, таких як курси валют, ціни на товари та послуги, обсяги продажів та інші, є важливими для прийняття рішень про інвестування, ризик-менеджмент та планування діяльності підприємств.

Основна мета досліджень у галузі фінансових часових рядів полягає у вивченні різних параметрів цих рядів, зокрема, їхньої структури, тренду, сезонності, циклів, затримки та інших характеристик, що дозволяє зробити висновки про тенденції зміни відповідних фінансових показників.

Фінансові часові ряди – це послідовності фінансових даних, які збираються відповідно до певної часової структури, наприклад, щоденно, щотижня або щомісяця [12]. Дані можуть стосуватися курсів валют, цін на акції, рівня процентних ставок та інших показників фінансової діяльності.

Фінансові часові ряди дозволяють досліджувати динаміку зміни показників фінансової діяльності в часі, аналізувати тренди та сезонність, виявляти кореляції між різними показниками та робити прогнози на майбутнє. Аналізуючи динаміку росту акцій певної компанії за допомогою часових рядів, можна визначити потенційні ризики та можливості для інвесторів.

Курс акцій компаній може змінюватися під впливом багатьох факторів, які можуть бути політичними, фінансовими, економічними, соціальними тощо [13]. Наприклад, зміни у ситуації на світових ринках, кризи, політичні кризи, катастрофи, новини про компанію та інші фактори можуть впливати на курс акцій. Події у світі, пов'язані з епідемією COVID-19, мали значний вплив на курс акцій компаній, особливо тих, що пов'язані з туризмом, авіацією та гостинністю. За період з березня по червень 2020 року, коли велика частина світу була закрита через карантин, курс акцій багатьох компаній впав. Наприклад, курс акцій авіакомпанії American Airlines Group Inc. (однієї з найбільших авіакомпаній у світі, яка надає пасажирські та вантажні авіаперевезення по всьому світу) з 21 лютого 2020 року до 23 березня 2020 року знизився більш ніж 62%, що можна побачити на рисунку 1.6.

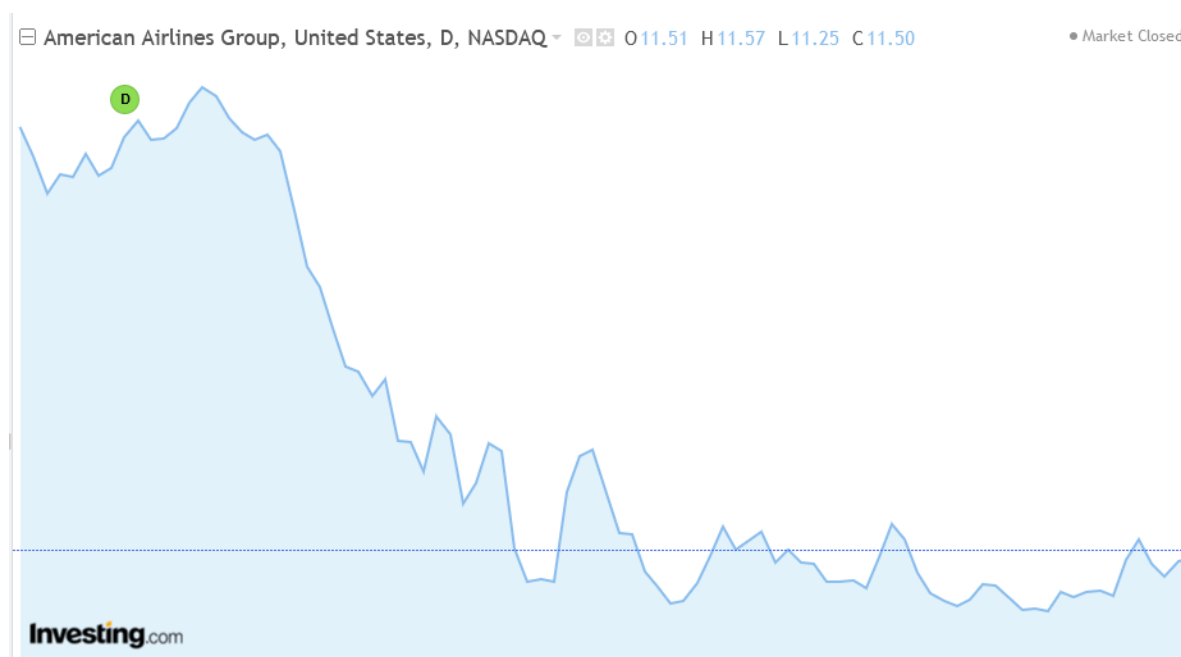


Рисунок 1.6 – Курс акцій авіакомпанії American Airlines Group Inc. з 21 лютого 2020 року до 23 березня 2020 року

Також, курс акцій може змінюватися під впливом фінансових звітів компанії, змін у керівництві компанії, змін у виробничому процесі, змін в правовому регулюванні тощо. Наприклад, коли компанія заявляє про успішність своєї діяльності та публікує позитивний фінансовий звіт, курс акцій може підвищитися, оскільки інвестори будуть вважати її більш привабливою для інвестування.

Один з реальних прикладів пов'язаний з компанією Tesla Inc. У липні 2020 року Tesla опублікувала позитивний фінансовий звіт, згідно з яким компанія заробила високі прибутки в другому кварталі, що перевершило очікування аналітиків. Як наслідок, курс акцій Tesla виросла на 10% в перший день після публікації звіту. Інвестори переконалися в успішності діяльності компанії та побачили перспективи для її подальшого розвитку, що призвело до збільшення попиту на акції Tesla і підвищення їх курсу, що можна побачити на рисунку 1.7.



Рисунок 1.7 – Курс акцій компанії Tesla Inc. в липні 2020 року після опублікування позитивного фінансового звіту

Часові ряди є зручним інструментом для прогнозування, оскільки вони можуть допомогти виявити тренди та сезонність у даних, а також можуть допомогти у прогнозуванні майбутніх значень. Дані в часових рядах складаються зі значень, взятих у різні моменти часу. Така структура дозволяє застосовувати багато різних методів аналізу ЧР [14], [15].

Курс акцій може змінюватися в залежності від таких факторів, як:

- політичні та економічні події: курс акцій може знижуватися у зв'язку з несприятливими рішеннями уряду, такими як введення високих податків або належність до міжнародних санкцій;
- конкуренція на ринку: курс акцій компанії може знижуватися, якщо її конкуренти представляють нові продукти або послуги, які можуть знизити прибуток компанії;
- фінансове здоров'я компанії: курс акцій може змінюватися в залежності від фінансового стану компанії, такого як рівень прибутку, боргові зобов'язання, або інші фінансові показники;

– попит та пропозиція: курс акцій може змінюватися в залежності від зміни попиту та пропозиції на акції компанії. Якщо попит на акції перевищує пропозицію, то ціна зростає, а якщо пропозиція перевищує попит, то ціна падає.

Врахування цих факторів може бути корисним при подальшому прогнозуванні часового ряду. Цей ряд можна проаналізувати, щоб виявити тренди та сезонність у даних, а також щоб побудувати прогноз майбутніх значень курсу акцій. Наприклад, якщо підприємство випускає новий продукт, то можна очікувати зростання попиту на акції компанії, оскільки новий продукт може привести до збільшення прибутку підприємства, що зазвичай позитивно впливає на ціну акцій.

Курси акцій можуть змінюватися через макроекономічні фактори, такі як інфляція, процентні ставки, політика центральних банків та інші подібні події. Наприклад, збільшення процентної ставки може привести до скорочення кількості грошей, доступних для інвестування, що може знизити попит на акції та зменшити їх ціну.

1.4 Існуючі інструменти прогнозування ЧР

Оскільки курси акцій можуть змінюватися залежно від різних факторів, важливо використовувати інструменти для їх прогнозування. Програмні засоби, які дозволяють аналізувати структуру даних часового ряду курсу акцій, виявляти закономірності та здійснювати прогнозування майбутніх значень на основі цих закономірностей, є одним із таких інструментів.

Існує багато систем та програмних сервісів, що дозволяють здійснювати побудову прогнозів часових рядів [16]. Умовно їх можна поділити на кілька типів: статистичні пакети, табличні процесори, пакети візуального проектування, ERP-системи, системи класу Business

Intelligence, когнітивні та академічні системи прогнозування. Розглянемо їх більш детально [17].

Статистичні пакети представлено програмами, які дозволяють використовувати різні методи статистики і Data Mining та відрізняються принципами роботи користувача із пакетом [18]. STATISTICA, IBM SPSS Statistics, Minitab – статистичні програмні комплекси, які містять багато вбудованих засобів прогнозування. Взаємодія з користувачем здійснюється за допомогою меню та панелей інструментів. Програмні комплекси MatLab, SAS, S-Plus вимагають від користувача спеціальних мов програмування. Дані статистичні пакети дозволяють здійснювати побудову моделей прогнозування різного рівня складності.

Табличні процесори є найбільш поширеними засобами економіко-математичного моделювання. Найбільш відомі програмні комплекси у цьому класі MS Excel, MS Excel 365, Corel Quattro Pro. Ці пакети дозволяють створювати та редагувати електронні таблиці, використовуючи вбудовані формули та алгоритми побудови прогнозних моделей.

Пакети візуального моделювання містять набір візуальних елементів для моделювання. Найбільш великими представниками є Microsoft Machine learning і Orange 3. Інтерфейс таких систем аналізу даних є полотном, на який користувач може наносити різні елементи, які характеризують інструменти роботи з даними. Кожен елемент налаштовується та має свої особливості. Результатом роботи з такою програмою є візуальна модель, де представлені етапи роботи з даними (потік даних).

Сучасні системи класу ERP систем дозволяють проводити аналіз даних із використанням різних інструментів. Такі системи допомагають підприємствам автоматизувати основні бізнес-процеси та керувати ними для досягнення оптимальної продуктивності й можуть включати модуль методів машинного навчання з реалізованими методами прогнозування фінансових показників та курсу акцій.

Business Intelligence системи є набором інструментів щодо аналітики, зокрема прогнозування. Прикладом таких систем є Microsoft PowerBI, IBM Cognos Analytics, QlikView, Deductor (Loginom), Contour BI. Їхня архітектура включає модуль інтелектуального аналізу даних, який дозволяє проводити побудову прогнозів досліджуваних показників.

Найбільш яскравим представником класу когнітивних систем є IBM Watson. Ця система дозволяє вирішувати складні завдання з використанням методів штучного інтелекту [19]. З її допомогою можна здійснювати прогнозування з використанням інтелектуальних моделей, а також проводити візуалізацію отриманих результатів.

Клас академічних систем прогнозування розробляється спільною групою вчених і зазвичай має вузьку спеціалізацію. Серед них є системи підтримки прийняття рішення, засновані на експертному методі прогнозування економічних даних, які застосовуються для бізнес-прогнозування. Поширені системи прогнозування на основі аналізу та прогнозування фінансових показників з використанням методів регресії, авторегресії, регресійних дерев рішень, методу опорних векторів, нейронних мереж і бустингу.

Наведені системи дозволяють здійснювати прогнозування часових рядів різної природи. Однак у них є ряд недоліків. Пакети візуального моделювання мають обмежений набір елементів, а також вимагають знання програмування для можливості гнучкого налаштування моделей. Корпоративні системи BI та ERP є дорогими та потребують трудомісткого процесу розгортання та підтримки, а також не є гнучкими. Академічні системи, як правило, вирішують вузьке коло завдань без можливості масштабування.

Когнітивні системи дозволяють провести сценарне прогнозування сукупності показників. Однак при цьому деталі обчислень приховані від аналітика і не всі виявлені моделі зв'язку є очевидними. Статистичні пакети та табличні процесори не дозволяють використовувати системний підхід, а

також потребують трудомістких процесів налаштування моделей та знань мов програмування.

1.5 Загальні відомості про ШНМ

Штучна нейронна мережа (ШНМ) – це математична модель, побудована з урахуванням теорії діяльності мозку, що застосовується для вирішення різних прикладних завдань. Вона імітує роботу нейронів головного мозку – простих однотипних обчислювальних елементів, які обмінюючись сигналами між собою, здатні виконувати складні завдання. Штучні нейрони являють собою функцію, на вхід якої подаються деякі значення, помножені на кориговані параметри – ваги, після чого результати добутку підсумовуються та отримане значення проходить через нелінійну функцію, яка є показником сили збудження нейрона [20].

Штучні нейронні мережі здатні поєднувати у собі безліч таких нейронів і дозволяють вирішувати ряд завдань, таких як: кластеризація, класифікація, регресійний аналіз, комп'ютерний зір, розпізнавання мови та машинний переклад. Особливостями штучних нейронних мереж є навчання і нелінійність, які дозволяють знаходити складні залежності між даними та враховувати досвід для покращення результату своєї роботи [20]. Структура штучної нейронної мережі являє собою спрямований граф, вузлами якого виступають нейрони, а як дуги – зв'язки з ваговими коефіцієнтами. Ця структура показана на рисунку 1.6.

Зрозуміти принцип роботи штучної нейронної мережі можна з прикладу перцептрона – однієї з перших моделей нейронної мережі. Модель перцептрона представлена на рисунку 1.7.

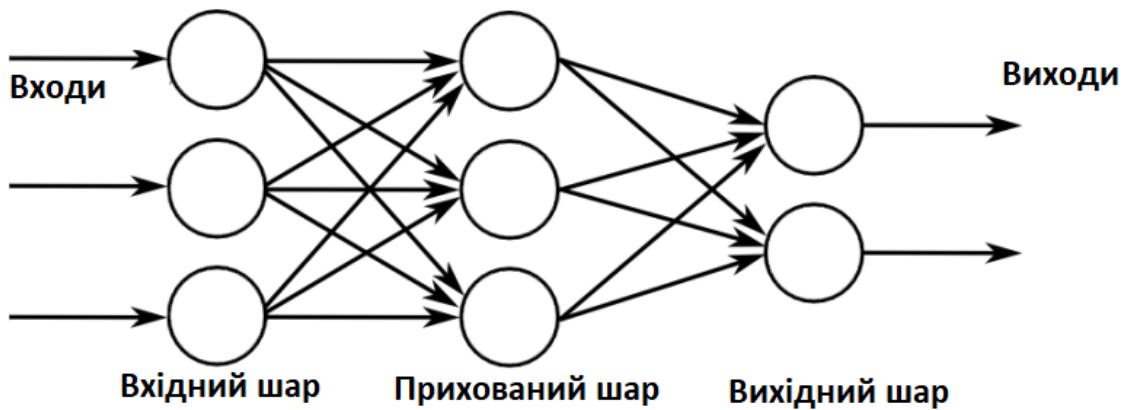


Рисунок 1.6 – Загальна структура ШНМ прямого поширення

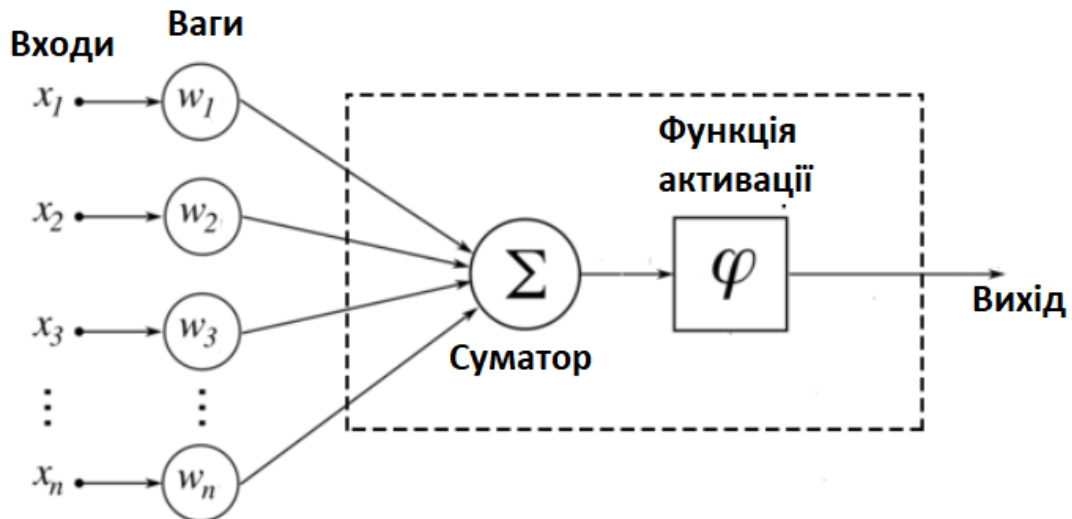


Рисунок 1.7 – Модель перцептрона

Між входом або інакше вхідними нейронами, що отримують сигнал, і суматором знаходяться зв'язки, кожен з яких має свій ваговий коефіцієнт, що визначає значущість відповідного входу на формування вихідного значення. Суматор обчислює зважену суму за допомогою складання вхідних сигналів, помножених на свої вагові коефіцієнти. Ця сума подається у функцію активації, яка визначає вихідне значення нейрона [21].

Функція активації визначає чи повинен бути активований нейрон. Іншими словами, вона здійснює нормалізацію сигналу, що проходить через неї і представляє результат роботи нейронної мережі в потрібному

діапазоні, залежно від типу функції активації [20]. Без цієї функції будь-яка архітектура нейронної мережі являє собою звичайну модель лінійної регресії, оскільки вона дозволяє виконувати нелінійне перетворення для вхідних даних і вирішувати складніші завдання [22]. Різновидів функцій активації дуже багато, основними є: ступінчаста, лінійна, сигмоїдальна, гіперболічний тангенс, функція усіченого лінійного перетворення (ReLU), що представлені на рисунку 1.8.

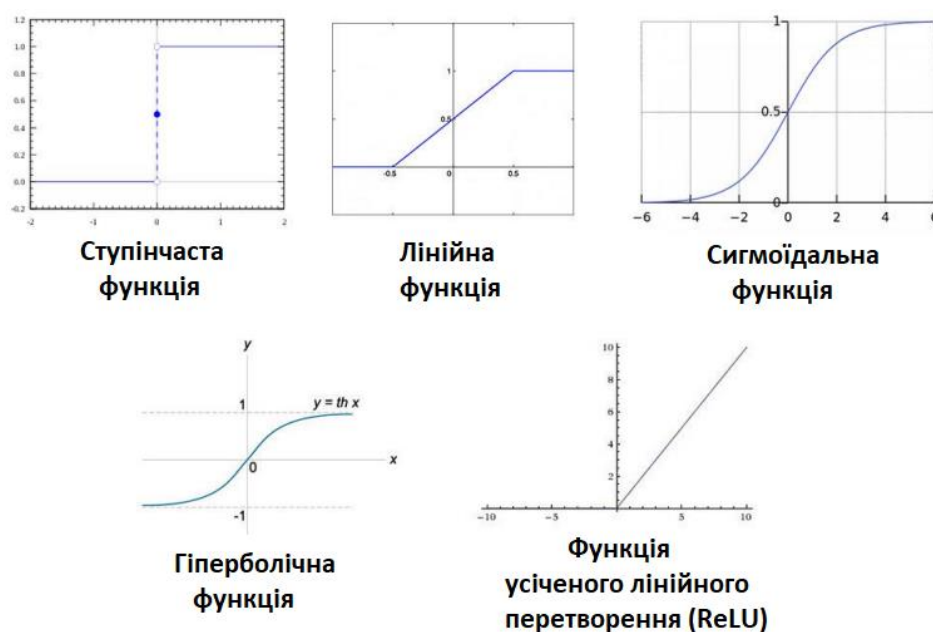


Рисунок 1.8 – Основні функції активації

Функція активації вибирається виходячи з типу завдання, яке вирішується нейронною мережею. Перцептрон є найпростішим видом нейронної мережі, який дозволяє вирішувати найпростіші завдання класифікації, але не призначений для вирішення складніших завдань. Саме для цього були розроблені багатошарові нейронні мережі прямого поширення, які на відміну від перцептрона, складаються з нейронів, згрупованих за шарами. У таких мережах нейрони кожного шару пов'язані з нейронами наступного шару. Усі шари даної нейронної мережі, крім першого і останнього шару називаються прихованими [23].

Однією з важливих складових багатьох архітектур нейронних мереж є використання зміщення у вхідному та прихованих шарах. Зміщення – це нейрон, який не приймає вхідних даних, оскільки його вихідний сигнал завжди дорівнює одиниці. Даний параметр дозволяє зрушувати функцію активації вліво та вправо, що допомагає зробити модель, що навчається, більш гнучкою [24]. Багатошарова нейронна мережа, що містить нейрони зміщення, представлена на рисунку 1.9.

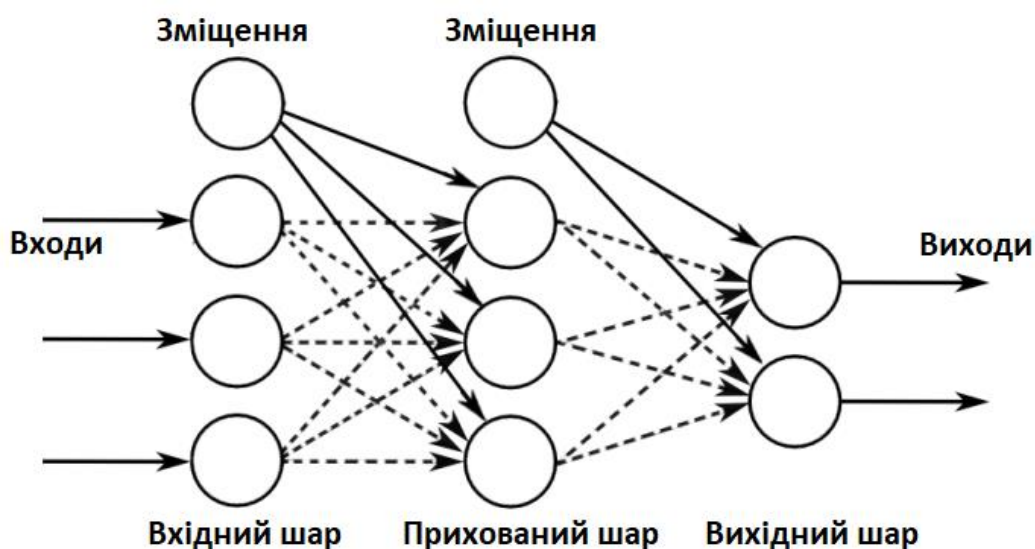


Рисунок 1.9 – Багатошарова нейронна мережа з нейронами зміщення

Таким чином, існує велика кількість різних архітектур нейронних мереж залежно від кількості прихованих шарів, нейронів в даних шарах і видів зв'язків між нейронами, кожна з яких краще справляється зі своїм типом завдань, як наприклад:

- багатошаровий перцептрон (Multilayer Perceptron, MLP) є базовою архітектурою нейронних мереж, де нейрони організовані в шари: вхідний, один або більше прихованих і вихідний. Інформація проходить тільки в одному напрямку - від входу до виходу, без зворотних зв'язків. Ця архітектура використовується для різних задач класифікації та регресії;

- згортова нейронна мережа (Convolutional Neural Network, CNN) призначена для обробки даних з високою просторовою кореляцією, таких як зображення та відео. Вона складається з шарів згортки, які автоматично виділяють ознаки, шарів субдискретизації для зменшення розмірності та повнозв'язних шарів для остаточної класифікації або регресії;
- мережі радіально-базисних функцій (RBF) використовують радіально-базисні функції як функції активації;
- рекурентна нейронна мережа (RNN) здатна обробляти послідовні дані завдяки зворотним зв'язкам, що дозволяють зберігати інформацію про попередні елементи послідовності. Це робить її придатною для задач, таких як обробка тексту, аналіз часових рядів та прогнозування. Довготривала короткочасна пам'ять (Long Short-Term Memory, LSTM) є вдосконаленою версією RNN, що вирішує проблему зникання градієнта за допомогою спеціальних комірок пам'яті. Це дозволяє зберігати важливу інформацію протягом тривалого часу, що є корисним для задач, які потребують довготривалої залежності;
- енкодер-декодер архітектури (Encoder-Decoder Architectures) призначені для перетворення однієї послідовності в іншу, як наприклад, в задачах машинного перекладу. Енкодер обробляє вхідну послідовність і перетворює її в проміжне представлення, яке потім декодер використовує для генерації вихідної послідовності.

1.6 Постановка задачі дослідження

Після вивчення досліджень, присвячених вирішенню завдань прогнозування, зокрема прогнозування часових рядів, можна дійти висновку, що нейронні мережі за багатьма параметрами перевершують класичні методи. Однак величезне розмаїття видів і архітектур нейронних мереж, що застосовуються, свідчить про відсутність єдиної загальної

позиції про те, яка нейромережева модель є оптимальною. На це впливають два основні фактори:

- метою більшості робіт є побудова ШНМ, яка перевершує класичні методи. Отже, під час порівняльного аналізу не завжди розглядаються інші моделі НМ;

- порівняння якості роботи нейронних мереж відбувається під час вирішення вузькоспеціалізованих завдань.

Для проведення оптимального порівняльного аналізу точності прогнозування фінансових часових рядів за допомогою нейронних мереж були обрані 3 найбільш перспективні види нейронних мереж: мережі радіально-базисних функцій (RBF), багатошаровий перцептрон (MLP) та мережа довгої короткострокової пам'яті (LSTM).

Фінансові часові ряди мають унікальні характеристики, такі як нестаціонарність та часові залежності, що робить їх складними для прогнозування. Мережі RBF добре підходять для захоплення нелінійних взаємозв'язків у даних завдяки своїм функціям активації. MLP ефективно моделюють складні закономірності та взаємодії в даних. Мережі LSTM спеціально розроблені для обробки довгострокових залежностей та часових затримок у послідовних даних, що робить їх особливо корисними для захоплення тимчасових залежностей, властивих фінансовим часовим рядам.

2 ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ ЗА ДОПОМОГОЮ НЕЙРОННИХ МЕРЕЖ

2.1 Завдання прогнозування ЧР

Розглянемо часовий ряд $x = (x_1, \dots, x_n)$, що складається з дійсних значень x_i , що описують деякий процес, що протікає в часі. При прогнозуванні такого ряду виділяють дві основні задачі.

2.1.1 Завдання класифікації напрямку тренду ЧР

Завдання класифікації часового ряду полягає у знаходженні «класифікуючої функції» [25] $f(x)$, що задає відображення:

$$f: X \rightarrow Y, \quad (2.1)$$

де $x \in X \subset \mathbb{R}^n$ – простір значень ЧР;

Y – множина аналізованих класів.

При цьому встановлена кінцева підмножина $X' \subset X$, для якої відомий результат відображення. Назвемо таку підмножину навчальною вибіркою:

$$(x, y), \quad x \in X', \quad y \in Y. \quad (2.2)$$

Як одне з найпоширеніших завдань класифікації розглянемо завдання бінарної класифікації визначення напрямку тренду часового ряду. Назвемо тренд зростаючим, якщо для заздалегідь заданого горизонту прогнозування d виконується $x_n < x_{n+d}$, в іншому випадку назвемо тренд спадаючим. Однак з огляду на випадкову складову ряду, що функцію класифікації зручно подати у вигляді $f_d(x) = P(x_n < x_{n+d})$, тобто $f_d(x)$ – ймовірність, з якою часовий ряд x може бути віднесений до класу зростаючого тренду.

Відповідно до ймовірності $1 - f_d(x)$ ряд буде віднесений до класу спадного тренду.

2.1.2 Завдання регресії ЧР

Завдання регресії часового ряду полягає у знаходженні залежності «математичного очікування залежної змінної в залежності від вільної» [26]:

$$f(x) = M(y|x), \quad (2.3)$$

де $x \in X \subset \mathbb{R}^n$ – простір вільних змінних;

$y \in Y$ – простір залежних змінних.

У такому разі, припускаючи, що до складу залежної змінної входить адитивна випадкова величина ε з нульовим математичним очікуванням, отримуємо $y = f(x) + \varepsilon$.

При цьому задано кінцеву підмножину $X' \subset X$ значень вільних змінних та підмножину відповідних їм значень залежних змінних $Y' \subset Y$:

$$(x, y), \quad x \in X', \quad y \in Y'. \quad (2.4)$$

Розглянемо найбільш поширене формулювання завдання регресії часового ряду: прогнозування значення x_{n+d} у межах горизонту прогнозування d . Тоді регресійна функція має задовольняти рівнянню (2.5), а прогноз часового ряду відповідати рівнянню (2.6).

$$f_d(x) = M(x_{n+d}|x); \quad (2.5)$$

$$y = f(x) + \varepsilon. \quad (2.6)$$

2.2 Метрики якості та функції втрат

Одним із найважливіших етапів розв'язання задачі прогнозування ЧР є вибір метрик якості та функцій втрат. Основною метою метрики якості є одержання зрозумілою людині оцінки точності прогнозування моделі. А головне завдання функції втрат полягає у визначенні ступеня відхилення оптимальної моделі. Таким чином, задачу оптимізації моделі можна звести до задачі мінімізації функції втрат.

2.2.1 Середня квадратична помилка (MSE)

Середня квадратична помилка (MSE) є найпоширенішою функцією втрат під час вирішення завдання регресії. MSE – це сума квадратів відхилень прогнозованих значень від істинних [27]:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2, \quad (2.7)$$

де n – кількість значень в істинному наборі y та в прогнозованому наборі \hat{y} .

Графік функції помилки MSE наведено на рисунку 2.1.

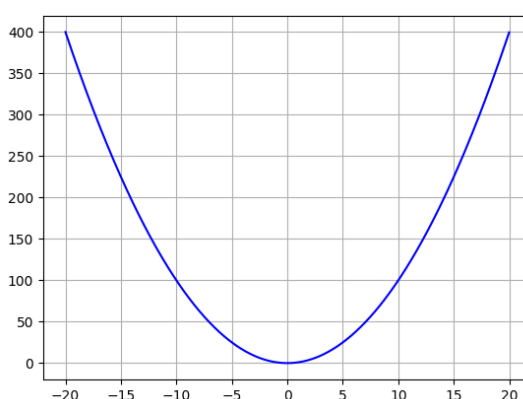


Рисунок 2.1 – Графік функції MSE

Основною перевагою такої функції втрат є чутливість до великих відхилень. Це означає, що мінімізація MSE призведе до зменшення великих помилок прогнозу.

2.2.2 Середня абсолютна помилка (MAE)

Середня абсолютна помилка (MAE) є ще однією функцією втрат, що застосовується під час вирішення завдання регресії. MAE – це сума абсолютних відхилень прогнозованих значень від істинних [30]:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|, \quad (2.8)$$

де n – кількість значень в істинному наборі y та в прогнозованому наборі \hat{y} .

Графік функції помилки MAE наведено на рисунку 2.2.

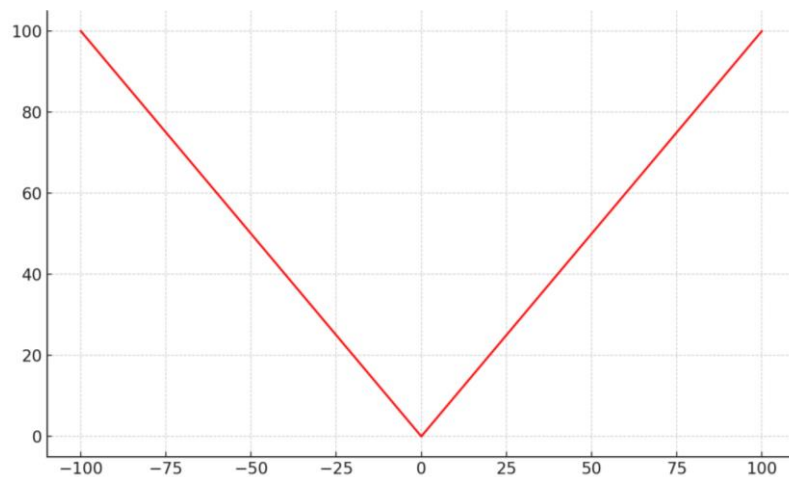


Рисунок 2.2 – Графік функції MAE

MAE на відміну від MSE не має явних переваг, що дозволяють істотно підвищити точність моделі. Однак інтерпретація такої функції втрат як суми модулів помилок є візуально та логічно простішою для розуміння. Таким чином, MAE можна використовувати як метрику якості регресійної моделі.

2.2.3 Крос-ентропійна функція втрат (Cross-Entropy)

Найбільш поширеною функцією втрат при вирішенні завдання класифікації є крос-ентропійна функція помилки (Cross-Entropy) [29]:

$$H(y, \hat{y}) = \sum_{i=1}^n y_i \cdot \log \frac{1}{\hat{y}_i} = - \sum_{i=1}^n y_i \cdot \log \hat{y}_i, \quad (2.9)$$

де n – кількість класів;

$y_i \in \{0,1\}$ – індикатор відношення до класу i ;

\hat{y}_i – прогнозуєма вірогідність відношення до класу i .

У разі вирішення задачі бінарної класифікації функція втрат може бути представлена у вигляді:

$$H(y, \hat{y}) = -(y \cdot \log \hat{y} + (1 - y) \cdot \log(1 - \hat{y})), \quad (2.10)$$

де y – індикатор відношення до класу 1;

\hat{y} – прогнозуєма вірогідність відношення до класу 1.

Враховуючи той факт, що $y \in \{0,1\}$, функцію втрат можна подати у вигляді системи (2.11).

$$H(y, \hat{y}) = - \begin{cases} \log \hat{y}, & y = 1, \\ \log(1 - \hat{y}), & y = 0. \end{cases} \quad (2.11)$$

Таким чином крос-ентропійну функцію втрат можна подати у вигляді сукупності графіків на рисунку 2.3. На цьому графіку можна побачити, що коли передбачена ймовірність правильного класу близька до 1, тоді крос-ентропійна втрата є низькою. І навпаки, коли передбачена ймовірність правильного класу близька до 0, то крос-ентропійна втрата значно зростає.

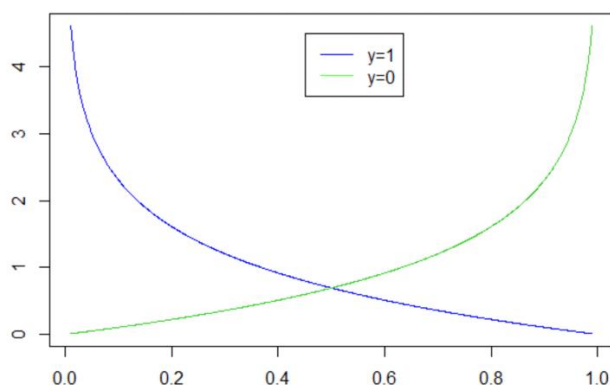


Рисунок 2.3 – Графік Cross-Entropy

2.2.4 Точність класифікації (Accuracy)

Точність (Accuracy) є найбільш наочною метрикою якості під час вирішення завдання класифікації, оскільки відповідає частці правильно класифікованих об'єктів [30].

При вирішенні задачі бінарної класифікації можна зручно скласти таблицю 2.1. В цій таблиці y – істинний клас, \hat{y} – прогнозуємий клас, TP та TN – кількість вірних класифікацій, а FP та FN – кількість помилок при класифікації.

Таблиця 2.1 – Матриця помилок метрики якості Accuracy

	$y = 1$	$y = 0$
$\hat{y} = 1$	TP	FP
$\hat{y} = 0$	FN	TN

Виходячи з таблиці, частку правильно класифікованих об'єктів можна виразити у вигляді формули (2.12).

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (2.12)$$

2.3 Мережа радіально-базисних функцій (RBF)

Основною ідеєю мережі радіально-базисних функцій (RBF) є застосування прихованого шару радіально-симетричних нейронів. Кожен нейрон як функцію активації використовує радіально-базисну функцію для оцінки норми відхилення вхідного вектора від заданого центру нейрона. Найбільш зручною для цього є функція Гауса:

$$\varphi_i(x) = e^{-\frac{\|x-x_i\|^2}{2\cdot\sigma^2}}, \quad (2.13)$$

де x – вхідний вектор;

x_i – заданий центр i -го нейрона;

φ_i – функція активації i -го нейрона;

σ – коефіцієнт нормування.

Таким чином, архітектуру мережі радіально-базисних функцій (RBF) можна представити у вигляді трьох шарів:

- вхідний шар, що складається з n нейронів, кожен з яких приймає на вхід відповідне значення часового ряду і поширює його на вхід кожного нейрона наступного шару;
- прихований шар, що складається з радіально симетричних нейронів, що використовують радіально-базисну функцію Гауса в якості функції активації;
- вихідний шар, що складається з єдиного нейрона та лінійної функції активації при вирішенні задачі регресії, аналогічно рисунку 2.4.

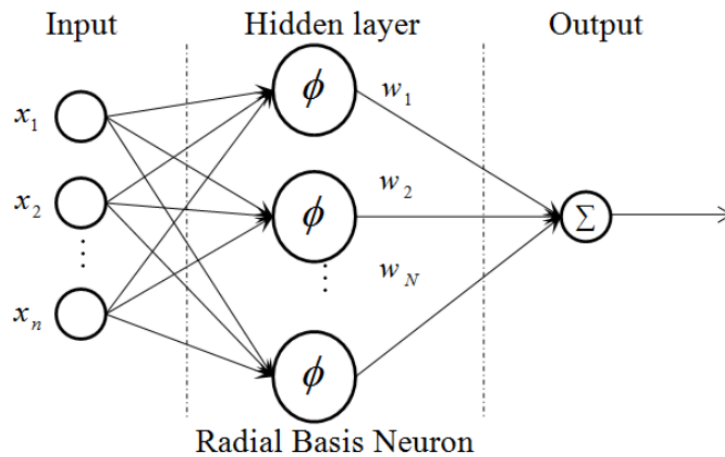


Рисунок 2.4 – Архітектура RBF мережі

В такому випадку результат роботи мережі при вирішенні задачі регресії часового ряду можна подати у вигляді зваженої комбінації результатів роботи радіально-базисних функцій (2.14).

$$y = f_d(x) + \varepsilon = \sum_{i=1}^k w_i \varphi_i(x) + \varepsilon, \quad (2.14)$$

де x – часовий ряд;

y – істинний прогноз;

f_d – регресійна функція;

d – горизонт прогнозування;

k – кількість радіально-симетричних нейронів;

φ_i – функція активації i -го нейрона;

w_i – ваговий коефіцієнт i -го нейрона;

ε – випадкова величина з нульовим математичним очікуванням.

При вирішенні задачі класифікації вихідний шар містить два нейрони та узагальнену логістичну функцію активації (Softmax), що задає розподіл ймовірності між відповідними нейронами класами (2.15).

$$P_i(z) = \frac{e^{z_i}}{\sum_j e^{z_j}}, \quad (2.15)$$

де P_i – ймовірність, з якою ЧР може бути віднесений до класу i ;

z – вектор значень нейронів вихідного шару.

Таким чином, результат роботи мережі, при вирішенні задачі бінарної класифікації для визначення напрямку тренду часового ряду, набуває вигляду:

$$f_d(x) = \frac{e^{z_0}}{e^{z_0} + e^{z_1}}, \quad (2.16)$$

$$1 - f_d(x) = \frac{e^{z_1}}{e^{z_0} + e^{z_1}}, \quad (2.17)$$

$$z_j = \sum_{i=1}^k w_{ij} \cdot \varphi_i(x), \quad (2.18)$$

де f_d – ймовірність, з якою часовий ряд x може бути віднесено до класу зростаючого тренду;

z_j – значення j -го нейрону вихідного шару;

w_{ij} – ваговий коефіцієнт i -го нейрону при розрахунку j -го нейрону вихідного шару.

Останнім кроком побудови мережі є навчання. Найбільш поширений підхід при навчанні мережі радіально-базових функцій є чисельне рішення системи рівнянь складених на основі навчальної вибірки:

$$\begin{aligned} \sum_{i=1}^k w_i \cdot \varphi_i(x_1) &= y_1, \\ &\vdots \\ \sum_{i=1}^k w_i \cdot \varphi_i(x_m) &= y_m, \end{aligned} \quad (2.16)$$

де x_i – часовий ряд з навчальної вибірки;

y_i – істинний прогноз j -го часового ряду;

m – довжина навчальної вибірки.

Однак рішення матричного рівняння за допомогою чисельних методів може бути недоцільним при використанні навчальної вибірки великого розміру або при побудові мережі з великою кількістю симетричних радіально нейронів. Таким чином, для навчання мережі радіальних базисних функцій (RBF) можна вибрати метод стохастичного градієнтного спуску, який є одним з найбільш поширених методів навчання нейронних мереж.

2.4 Багатошаровий перцептрон (MLP)

Головною особливістю архітектури даної нейронної мережі є використання відразу кількох прихованих повнозв'язних шарів нейронів, що можна побачити на рисунку 2.5. Це дозволяє відновлювати набагато складніші залежності, а також збільшує точність роботи мережі. Однак, для апроксимації будь-якого гладкого відображення з будь-якою точністю достатньо двох прихованих шарів з нелінійною функцією активації [31].

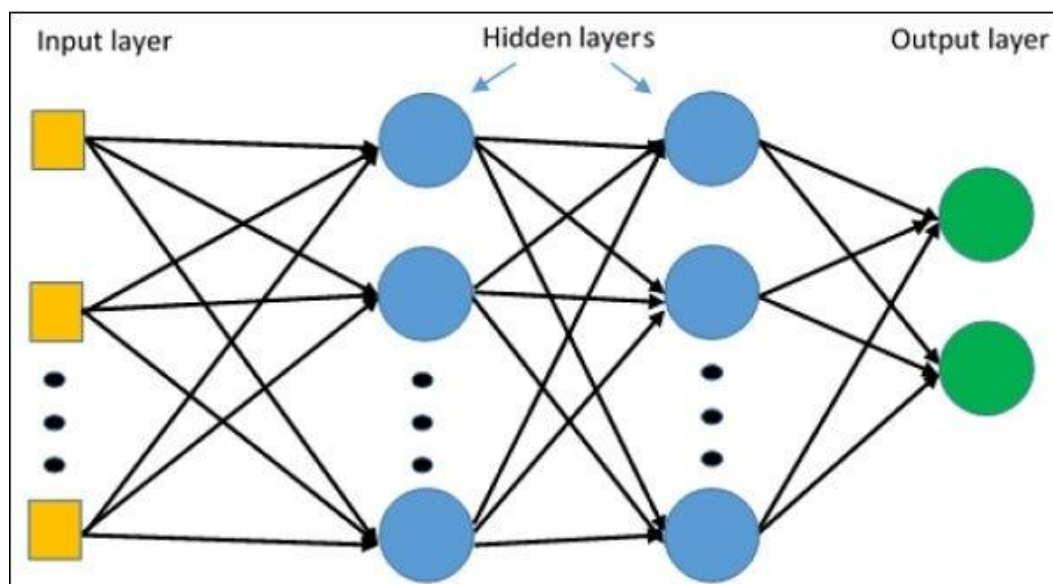


Рисунок 2.5 – Архітектура мережі багатошарового перцептрону MLP

Таким чином, архітектура мережі багатошарового перцептрона MLP складається із 4 шарів.

Вхідний шар, кожний нейрон якого набуває відповідного значення x_i часового ряду і поширює його на вхід кожного нейрона наступного шару.

Два приховані пов'язані шари нейронів, перший з яких містить нелінійну функцію активації – Leaky ReLU (2.17).

$$\varphi(x) = \begin{cases} 0.01x, & x < 0, \\ x, & x \geq 0. \end{cases} \quad (2.17)$$

Обчислення такої функції не вимагає виконання ресурсомістких операцій, крім того, Leaky ReLU не схильний до насичення, а також за останніми дослідженнями «застосування Leaky ReLU істотно підвищує швидкість збіжності градієнтного спуску» [24].

Таким чином, вихідний сигнал кожного нейрона прихованого шару можна подати у вигляді:

$$\varphi\left(\sum_{i=1}^k w_i \cdot x_i + \theta\right), \quad (2.18)$$

де k – кількість нейронів попереднього шару;

w_i – вага зв'язку з i -им нейроном попереднього шару;

x_i – вихідний сигнал i -ого нейрону попереднього шару;

θ – порогове значення.

Вихідний шар при вирішенні задачі регресії складається з єдиного нейрона, що має лінійну функцію активації. У такому разі результат роботи мережі при вирішенні задачі регресії часового ряду можна подати у вигляді матричного рівняння:

$$y = f_d(x) + \varepsilon = \varphi(W_3 \cdot X_2 + \theta_3), \quad (2.19)$$

$$X_2 = \varphi(W_2 \cdot X_1 + \theta_2), \quad (2.20)$$

$$X_1 = \varphi(W_1 \cdot x + \theta_1), \quad (2.21)$$

де x – часовий ряд;

y – істинний прогноз;

f_d – регресійна функція;

d – горизонт прогнозування;

W_1 – матриця ваг зв'язків між вхідним шаром і першим повнозв'язним шаром;

W_2 – матриця ваг зв'язків між першим і другим повнозв'язними шарами;

W_3 – матриця ваг зв'язків між другим повнозв'язним шаром і вихідним шаром;

X_i – вектор вихідних сигналів i -ого повнозв'язного шару;

θ_i – порогове значення з i -ого шару.

Аналогічно мережі радіально-базових функцій (RBF), при розв'язанні задачі бінарної класифікації для визначення напрямку тренду часового ряду, вихідний шар містить два нейрони та узагальнену логістичну функцію активації (Softmax). Також для навчання мережі багатошарового перцептрону (MLP) слід використовувати метод стохастичного градієнтного спуску.

2.5 Мережа довгої короткострокової пам'яті (LSTM)

Головною особливістю архітектури рекурентних нейронних мереж (RNN) є наявність зворотних зв'язків, що дозволяють враховувати контекст у процесі навчання та побудови прогнозу. Однак рекурентну мережу можна розглянути як розгортку, в якій зворотні зв'язки з'єднують копії однієї і тієї ж мережі (рисунок 2.6).

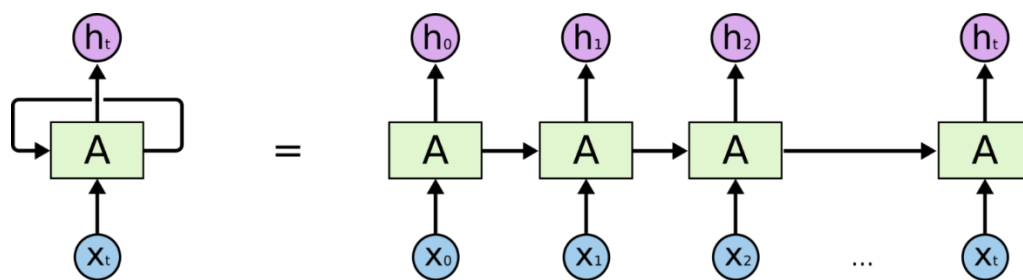


Рисунок 2.6 – Архітектура рекурентної нейронної мережі

Мережа довгої короткострокової пам'яті є особливим різновидом RNN, що складається відразу з чотирьох шарів нейронів, що взаємодіють один з одним [33] (рисунок 2.7).

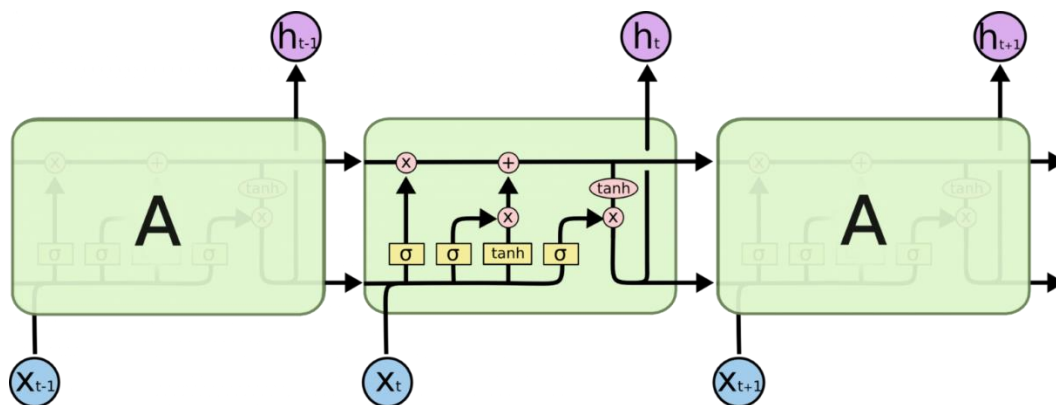


Рисунок 2.7 – Архітектура мережі LSTM

Ключовою компонентою LSTM є потік пам'яті, що є сигналом, що проходить через всі LSTM блоки і агрегує всю необхідну контекстну інформацію. На вхід t -ого по порядку LSTM блоку надходить сигнал C_{t-1} , що містить всю контекстну інформацію, накопичену раніше. Вихідним сигналом t -ого блоку є C_t , що містить контекстну інформацію сигналу C_{t-1} , оновлену з урахуванням вхідного сигналу x_t , а також сигнал h_t , який є результатом роботи поточного блоку (рисунок 2.8).

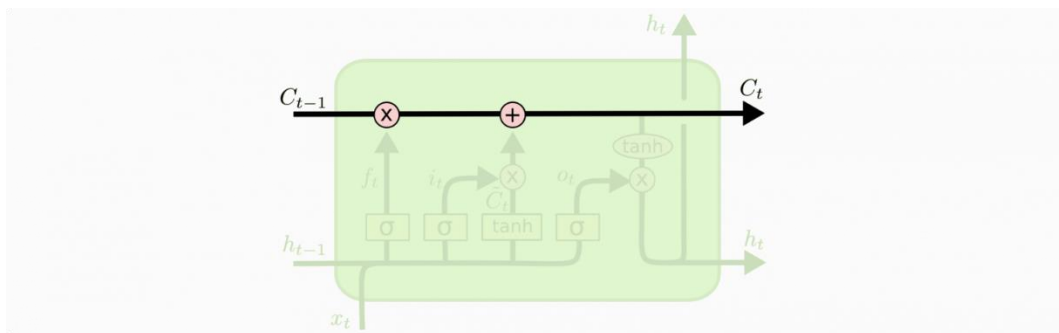


Рисунок 2.8 – Потік LSTM пам'яті

Однією з найважливіших частин мережі LSTM є фільтр, що складається з сигмоїдального шару σ і операції скалярного множення \times . Сигмоїдальний шар перетворює вхідні дані до значення від нуля до одиниці, що сигналізує про те, яка частка інформації повинна бути врахована мережею. LSTM містить відразу три такі фільтри, що дозволяють захищати від перезапису актуальну, оновлювати застарілу та додавати нову інформацію (рисунок 2.9).

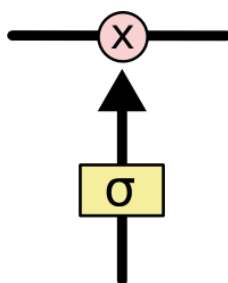


Рисунок 2.9 – Позначення LSTM фільтру

Першим кроком роботи блоку LSTM є фільтрація застарілої інформації. Цим займається перший фільтр (forget gate layer), зображений рисунку 2.10, який приймає на вхід вихідне значення попереднього LSTM блоку h_{t-1} , а також вхідне значення поточного блоку x_t . Результатом цього кроку є фільтраційна функція:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f), \quad (2.22)$$

де W_f – ваги сігмоїдального шару першого фільтру;

b_f – адитивна складова цього шару;

σ – сигмоїдальна функція активації.

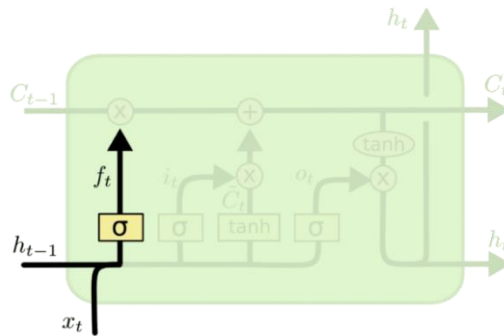


Рисунок 2.10 – Фільтр застарілої пам'яті

Другий крок складається з двох частин (рисунок 2.11), а саме:

– визначення значень, які потребують оновлення за допомогою другого фільтра (input layer gate). Вхідними значеннями цього фільтра є значення h_{t-1} і x_t ;

– побудова нового набору контекстної інформації \tilde{C}_t , за допомогою спеціального шару нейронів, що має гіперболічний тангенс як функцію активації, і приймає на вхід значення h_{t-1} і x_t .

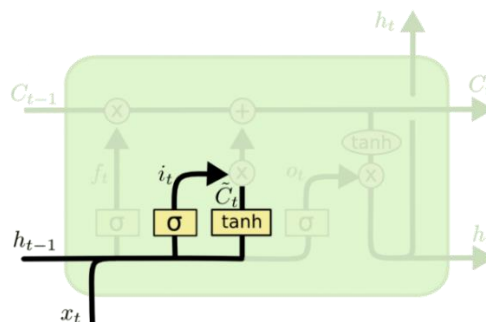


Рисунок 2.11 – Фільтр вибору нового контексту

Результатом роботи цього кроку є функція (2.23), що визначає набір значень, що вимагають оновлення, а також отримується новий набір контекстної інформації (2.24).

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), \quad (2.23)$$

$$\tilde{C}_t = th(W_C \cdot [h_{t-1}, x_t] + b_C), \quad (2.24)$$

де W_i – ваги сигмоїдального шару другого фільтра;

b_i – адитивна складова цього шару;

W_C – ваги спеціального th -шару;

b_C – адитивна складова цього шару;

th – гіперболічний тангенс.

Третій крок роботи блоку LSTM складається з об'єднання результатів роботи попередніх двох кроків і формування нового контекстного сигналу (рисунок 2.12).

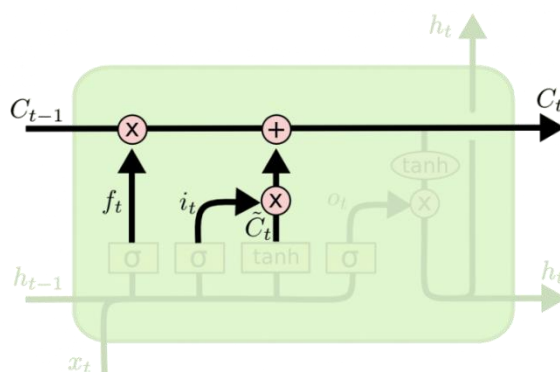


Рисунок 2.12 – Формування нового контексту

$$C_t = f_t \times C_{t-1} + i_t \times \tilde{C}_t, \quad (2.25)$$

де $f_t \times C_{t-1}$ – результат коригування попереднього контексту;

$i_t \times \tilde{C}_t$ – формування та додавання нового контексту.

Четвертий крок роботи блоку LSTM це формування результату h_t роботи блоку t (рисунок 2.12). Це можна зробити аналогічно кроку два, використовуючи сигмоїдальний шар для визначення важливості вихідної інформації (2.25) і спеціального th -шару для її формування (2.26).

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o), \quad (2.26)$$

$$h_t = o_t \times th(C_t), \quad (2.27)$$

де W_o – ваги сигмоїдального шару третього фільтру;

b_o – адитивна складова цього шару.

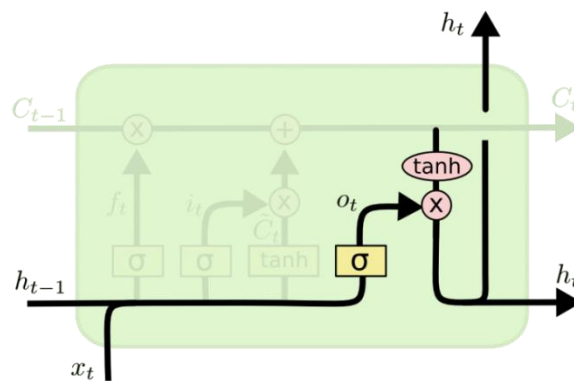


Рисунок 2.12 – Формування результату роботи блоку

Таким чином, архітектура мережі довгої короткострокової пам'яті (LSTM) складається з чотирьох шарів, а саме:

- вхідний шар, кожний нейрон якого набуває відповідного значення x_i часового ряду і поширює його на вхід кожного нейрона наступного шару;
- шар, що містить блоки LSTM;
- шар, що зберігає довгострокову інформацію, яка модифікується за допомогою входів та гейтів;

– вихідний шар, котрий аналогічно до мережі радіально базисних функцій (RBF) та мережі багатошарового перцептронну (MLP), складається з єдиного нейрона та лінійної функції активації при вирішенні задачі регресії часового ряду. А у разі вирішення задачі бінарної класифікації для визначення напрямку тренду ЧР, вихідний шар містить два нейрони та узагальнену логістичну функцію активації (Softmax). Також для навчання мережі довготривалої пам'яті (LSTM) слід використовувати метод стохастичного градієнтного спуску.

2.6 Навчання нейронних мереж

Останнім кроком побудови нейронної мережі є навчання. Для цього необхідно вибрати алгоритм, який оптимізує мережу за допомогою мінімізації функції втрат. Враховуючи, що при вирішенні завдань регресії та класифікації використовувалися лише гладкі функції втрат, то для їх мінімізації можна скористатися класичним методом градієнтного стохастичного спуску [34], описаний у формулі (2.28).

$$w_{t+1} = w_t - \alpha \cdot \frac{\partial L}{\partial w_t}, \quad (2.28)$$

де w – вагові коефіцієнти нейронної мережі;

t – момент часу в процесі навчання мережі;

α – швидкість навчання;

L – функція втрат;

$\frac{\partial L}{\partial w_t}$ – градієнт.

В такому випадку оновлення ваги залежить тільки від поточного значення градієнта, що може призводити до випадкових відхилень у процесі навчання. Однак, використання інерційної оптимізації, заснованої на

згладжуванні градієнта методом експоненціальних ковзних середніх дозволяє стабілізувати процес навчання.

$$w_{t+1} = w_t - \alpha \cdot V_t, \quad (2.29)$$

$$V_t = \beta \cdot V_{t-1} + (1 - \beta) \cdot \frac{\partial L}{\partial w_t}, \quad (2.30)$$

де β – параметр експоненційного згладжування.

Метод адаптивного градієнта (AdaGrad) також дозволяє підвищити якість стохастичного градієнтного спуску за допомогою адаптації швидкості навчання щодо кореня із суми квадратів попередніх градієнтів. Результатом адаптації є загасання градієнта при зменшенні функції втрат та збільшення градієнта при її зростанні.

$$w_{t+1} = w_t - \frac{\alpha}{\sqrt{S_t + \varepsilon}} \cdot \frac{\partial L}{\partial w_t}, \quad (2.31)$$

$$S_t = S_{t-1} + \left(\frac{\partial L}{\partial w_t}\right)^2, \quad (2.32)$$

де ε – малий доданок, необхідний задля запобігання поділу на 0.

Використання методу експоненціальних ковзних середніх також дозволяє покращити метод AdaGrad. Результатом цієї оптимізації є метод адаптивного ковзного середнього градієнтів (RMSprop).

$$S_t = \beta \cdot S_{t-1} + (1 - \beta) \cdot \left(\frac{\partial L}{\partial w_t}\right)^2, \quad (2.33)$$

де β – параметр експоненційного згладжування.

В результаті послідовного покращення та модернізації методу стохастичного градієнтного спуску можна виділити дві основні оптимізації:

- інерційна оптимізація – експоненційне згладжування значень градієнта;
- метод адаптивного ковзного середнього градієнтів (RMSprop), який регулює швидкість навчання за допомогою згладжених значень квадрату градієнта.

Таким чином, метод адаптивної інерції (Adam) є комбінацією цих двох оптимізацій [35]. Цей метод можна представити наступним чином:

$$w_{t+1} = w_t - \frac{\alpha}{\sqrt{\hat{s}_t + \varepsilon}} \cdot V_t, \quad (2.34)$$

$$V_t = \frac{v_t}{1 - \beta_1^t}, \quad (2.35)$$

$$\hat{S}_t = \frac{s_t}{1 - \beta_2^t}, \quad (2.36)$$

$$V_t = \beta_1 \cdot V_{t-1} + (1 - \beta_1) \cdot \frac{\partial L}{\partial w_t}, \quad (2.37)$$

$$S_t = \beta_2 \cdot S_{t-1} + (1 - \beta_2) \cdot \left(\frac{\partial L}{\partial w_t}\right)^2, \quad (2.38)$$

де β_1 – параметр експоненційного згладжування для інерційної оптимізації;
 β_2 – параметр експонентного згладжування для RMSprop.

Проте останні дослідження [36] вказують на те, що додавання імпульсу Нестерова до методу адаптивної інерції (Nadam) виправдане і в більшості випадків Nadam є найбільш точним методом навчання (2.39).

$$w_{t+1} = w_t - \frac{\alpha}{\sqrt{\hat{s}_t + \varepsilon}} \cdot \left(\beta_1 \cdot \hat{V}_t + \frac{1 - \beta_1}{1 - \beta_2^t} \cdot \frac{\partial L}{\partial w_t}\right). \quad (2.39)$$

Таким чином, для вибору оптимального алгоритму навчання розроблених архітектур нейронних мереж було розглянуто кілька оптимізацій методу стохастичного градієнтного спуску. Основуючись на аналізі розглянутих оптимізацій та результатах досліджень, як алгоритм навчання було обрано метод додавання імпульсу Нестерова в метод адаптивної інерції (Nadam).

3 РЕЗУЛЬТАТИ ПРОГНОЗУВАННЯ ФІНАНСОВИХ ЧАСОВИХ РЯДІВ

Як вхідні дані для навчання та тестування нейронних мереж було обрано набір даних, які містять в собі котирування валютної пари євро/долар (EUR/USD) представлені у вигляді чисельних значень курсу, зафіксованих рівно опівночі кожного дня за останні 10 років. Оскільки даний часовий ряд схильний до впливу трендів фондового ринку, то аналіз якості розв'язання задач прогнозування дозволяє оцінити здатність нейронної мережі визначати тенденцію (тренд) ряду.

Дані поділяються на тренувальну та тестову вибірки в пропорції 75% на 25%. Це стандартна практика, що забезпечує достатню кількість даних для навчання моделі та адекватну кількість даних для оцінки її продуктивності.

Попередня обробка даних складається з сегментації часових рядів, складання істинного прогнозу для кожного сегмента та формування навчальної та тестової вибірки:

$$(x, y), x \in X, y \in Y, \quad (3.1)$$

де $x \in X \subset \mathbb{R}^n$ – сегмент часового ряду довжини n ;

$y \in Y$ – істинний прогноз для сегмента x , який визначається виходячи з виду завдання прогнозування та горизонту прогнозування d .

3.1 Алгоритми отримання прогнозу та навчання нейронних мереж

Для проведення порівняльного аналізу точності прогнозування було розроблено алгоритм тестування нейронних мереж, який представлено на рисунку 3.1.

Для навчання мережі (рисунок 3.2) радіально-базових функцій (RBF), багатошарового перцептрона (MLP) та мережі довгої короткострокової пам'яті (LSTM) вибрано метод Nadam, що є модифікацією методу стохастичного градієнтного спуску, який поєднує переваги Adam і Nesterov momentum, забезпечуючи швидше та стабільніше навчання.



Рисунок 3.1 – Алгоритм тестування НМ

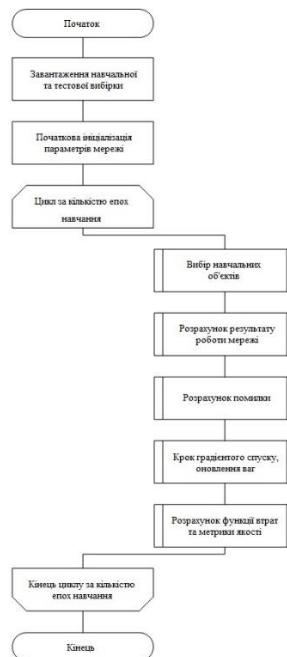


Рисунок 3.2 – Алгоритм навчання НМ

3.2 Мережа радіально-базисних функцій (RBF)

3.2.1 Архітектура мережі

Виходячи з математичної моделі RBF і можливостей, що надають бібліотеки Keras і TensorFlow, опишемо архітектуру мережі для вирішення задачі класифікації та регресії часового ряду:

- вхідний шар збігається за розміром із довжиною вхідного вектора, який складається з 15 фіч, що відповідає даним у проміжку часу в 15 днів;
- прихований шар складається з 150 радіально симетричних нейронів, що містять рівномірно розподілені випадкові центри активації при прогнозуванні котирувань валютної пари євро-долар;
- пакетна нормалізація вихідних значень прихованого шару дозволяє підвищити продуктивність мережі, а також частково замінити механізм регулювання;
- вихідний повнозв'язний шар, що відповідає одному нейрону з лінійною функцією активації у разі вирішення задачі регресії та двом нейронам з узагальненою логістичною функцією активації (Softmax) у разі вирішення задачі класифікації.

Таким чином, для вирішення задачі класифікації та регресії котирувань валютної пари євро-долар дві архітектури мережі RBF, які представлені в таблиці 3.1 та таблиці 3.2.

Ці значення параметрів моделі обрані як компроміс між здатністю мережі навчитися розпізнавати складні патерни та уникати переобчислення. Більша кількість нейронів дозволяє краще моделювати складні функції, але водночас може збільшити ризик перенавчання та збільшити обчислювальні витрати.

Таблиця 3.1 – Архітектура RBF мережі для вирішення завдання класифікації

№	Тип шару	Розмір шару
0	Вхідний шар	n
1	Шар радіально-базисних нейронів	150
2	Пакетна нормалізація	150
3	Вихідний повнозв'язний шар	2
4	Логістична функція активації (Softmax)	2

Таблиця 3.2 – Архітектура RBF мережі для вирішення завдання регресії

№	Тип шару	Розмір шару
0	Вхідний шар	n
1	Шар радіально-базисних нейронів	150
2	Пакетна нормалізація	150
3	Вихідний повнозв'язний шар	1
4	Лінійна функція активації	1

3.2.2 Навчання мережі

Розглянемо графіки навчання мереж RBF за допомогою методу Nadam, представлені на рисунках 3.3–3.4. Можна зауважити, що зі збільшенням кількості епох навчання зменшується функція помилки, а значення метрики якості тестової вибірки поступово стабілізуються. Це свідчить про те, що метод Nadam ефективно знижує функцію помилки під час навчання мережі RBF. Крім того, стабілізація значень метрики якості тестової вибірки вказує на хорошу узагальнюючу здатність моделі та зменшення ризику перенавчання.

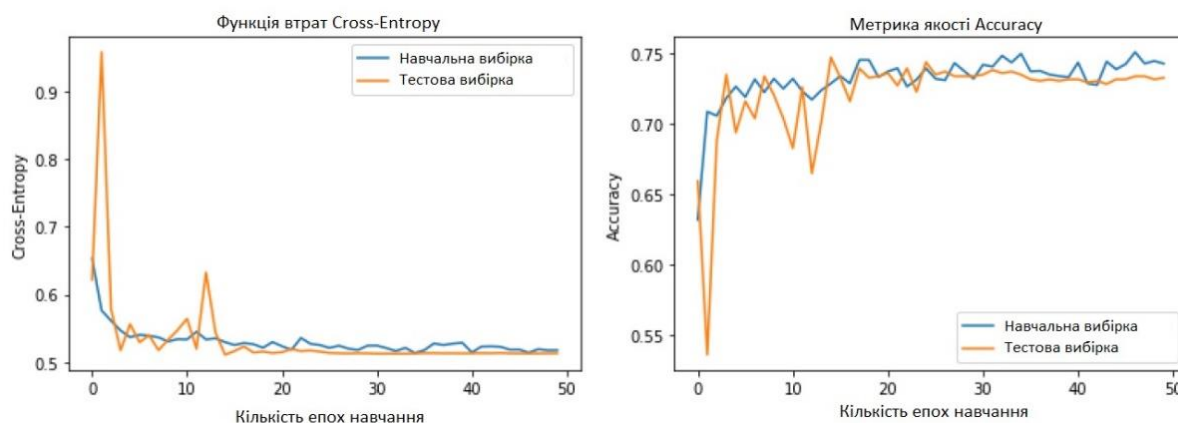


Рисунок 3.3 – Графік навчання мережі RBF для класифікації котирувань валютної пари євро-долар

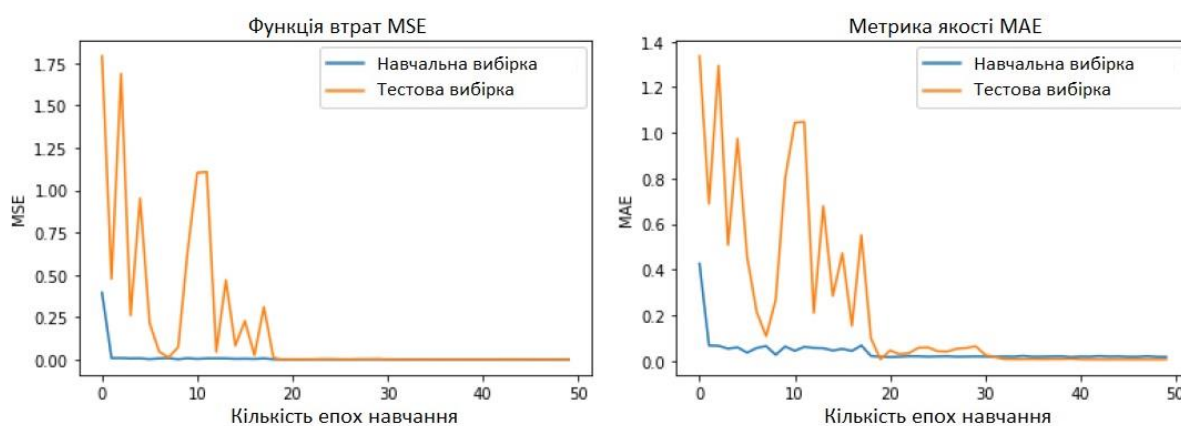


Рисунок 3.4 – Графік навчання мережі RBF для регресії котирувань валютної пари євро-долар

3.2.3 Результати роботи мережі

Для проведення порівняльного аналізу точності прогнозування при розв'язанні задачі класифікації та регресії досліджуємо залежність метрики якості від довжини вхідного сегмента часового ряду n та горизонту прогнозування d . Для цього складемо таблиці 3.3–3.4.

Таблиця 3.3 – Значення Ассурасу при вирішенні завдання класифікації котирувань валютної пари євро-долар за допомогою мережі RBF

d	n		
	10	15	20
1	0,7343	0,7373	0,7348
3	0,7258	0,7294	0,7314
5	0,7131	0,7203	0,7273
7	0,6873	0,7159	0,7211
10	0,6469	0,6917	0,7113

Мережа радіально-базових функцій продемонструвала високу точність розв'язання завдання класифікації часового ряду. З отриманих результатів можна дійти висновку, що збільшення горизонту прогнозування призводить до зменшення точності прогнозів. Також для короткострокового прогнозування достатньо розглядати часові ряди невеликої довжини. З іншого боку, часові ряди невеликої довжини непридатні для довгострокового прогнозування.

Таблиця 3.4 – Значення MAE при вирішенні завдання регресії котирувань валютної пари євро-долар за допомогою мережі RBF

d	n		
	10	15	20
1	0,0038	0,0037	0,0038
3	0,0059	0,0057	0,0058
5	0,0082	0,0075	0,0073
7	0,0106	0,0087	0,0086
10	0,0177	0,0105	0,0101

Мережа RBF продемонструвала високу точність розв'язання задачі регресії часових рядів. При аналізі результатів регресії отримані висновки

про довжину часового ряду та горизонт прогнозування аналогічні висновкам, складеним для завдання класифікації.

3.3 Багатошаровий перцептрон

3.3.1 Архітектура мережі

Виходячи з математичної моделі мережі багатошарового перцептрона та можливостей, що надають нейромережеві бібліотеки Keras та TensorFlow, опишемо архітектуру мережі для вирішення задачі класифікації та регресії часових рядів:

- вхідний шар збігається за розміром із довжиною вхідного вектора, який складається з 15 фіч, що відповідає даним у проміжку часу в 15 днів;
- перший повнозв'язний шар є основним шаром мережі MLP і складається з 100 нейронів;
- пакетна нормалізація вихідних значень першого повнозв'язкового шару, аналогічно мережі RBF, служить частковою заміною механізму регуляризації;
- застосування функції активації Leaky ReLU до вихідних значень першого повнозв'язного шару, додавання нелінійності до моделі нейронної мережі MLP;
- другий повнозв'язний шар реалізує поступове узагальнення результатів роботи першого повнозв'язного шару та складається з 50 нейронів;
- пакетна нормалізація вихідних значень другого повнозв'язкового шару також є частковою заміною механізму регуляризації;
- вихідний повнозв'язний шар, що відповідає одному нейрону з лінійної функції активації у разі вирішення задачі регресії та двом нейронам

з узагальненою логістичною функцією активації (Softmax) у разі вирішення задачі класифікації.

Таким чином, для вирішення задачі класифікації та регресії котирувань валютної пари євро-долар було створено дві архітектури мережі MLP, які представлені в таблиці 3.5 та таблиці 3.6.

Щодо параметрів архітектури мережі, то, загалом, це стандартні значення, які часто використовуються для задач з помірною складністю. Вони забезпечують достатню потужність моделі для навчання, одночасно знижуючи ризик перенавчання. 100 нейронів у першому шарі надають достатню потужність для навчання складних патернів, що характерні для фінансових часових рядів, таких як котирування євро-долар. Це значення є компромісом між потужністю моделі та ризиком перенавчання. Зменшення кількості нейронів у другому шарі до 50 допомагає зменшити складність моделі, сприяючи узагальненню та зменшуючи ризик перенавчання. Це також дозволяє моделі фокусуватися на найважливіших особливостях, виділених першим шаром.

Таблиця 3.5 – Архітектура MLP мережі для вирішення завдання класифікації

№	Тип шару	Розмір шару
0	Вхідний шар	n
1	Повнозв'язний шар	100
2	Пакетна нормалізація	100
3	Функція активації Leaky ReLU	100
4	Повнозв'язний шар	50
5	Пакетна нормалізація	50
6	Вихідний повнозв'язний шар	2
7	Логістичная функція активації (Softmax)	2

Таблиця 3.6 – Архітектура MLP мережі для вирішення завдання регресії

№	Тип шару	Розмір шару
0	Вхідний шар	n
1	Повнозв'язний шар	100
2	Пакетна нормалізація	100
3	Функція активації Leaky ReLU	100
4	Повнозв'язний шар	50
5	Пакетна нормалізація	50
6	Вихідний повнозв'язний шар	1
7	Лінійна функція активації	1

3.3.2 Навчання мережі

Розглянемо графіки навчання мереж MLP за допомогою методу Nadam, які представлені на рисунках 3.5–3.6. Виходячи з аналогічних міркувань для мережі RBF можна зробити висновок, що обидві розроблені архітектури мережі MLP були успішно навчені за допомогою методу Nadam для вирішення задачі прогнозування часових рядів. Під час навчання мережі MLP на вирішення завдання регресії спостерігається швидша збіжність методу Nadam, ніж під час навчання мережі RBF на вирішення аналогічного завдання.

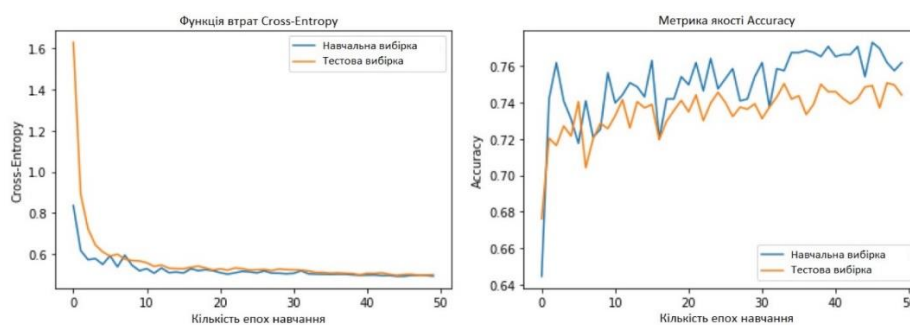


Рисунок 3.5 – Графік навчання мережі MLP для класифікації

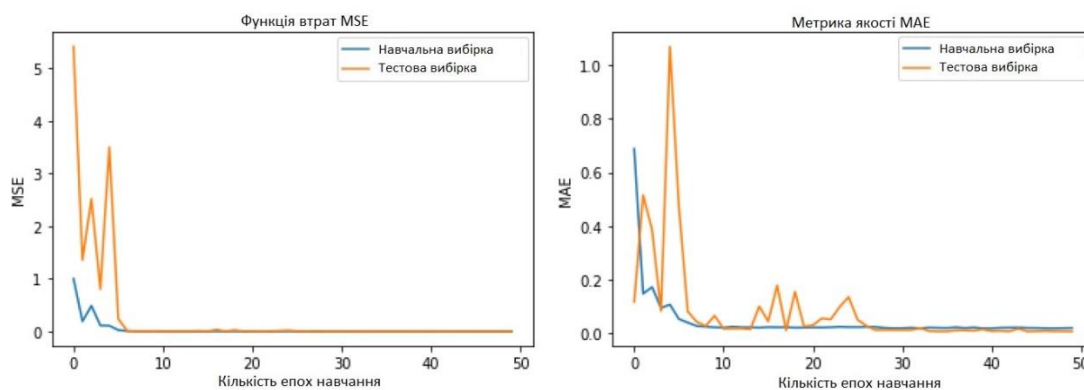


Рисунок 3.6 – Графік навчання MLP для регресії

3.3.3 Результат роботи мережі

Для проведення порівняльного аналізу точності прогнозування при розв'язанні задачі класифікації та регресії досліджуємо залежність якості прогнозування від довжини вхідного сегмента часового ряду і горизонту прогнозування. Для цього складемо таблиці 3.7–3.8.

При вирішенні завдання класифікації котирувань валютної пари євро-долар мережа багатошарового перцептрона продемонструвала точність, яку можна порівняти з мережею RBF.

Таблиця 3.7 – Значення Ассигасу при вирішенні завдання класифікації за допомогою мережі MLP

d	n		
	10	15	20
1	0,7393	0,7386	0,7336
3	0,7313	0,7316	0,7314
5	0,7186	0,7269	0,7256
7	0,6973	0,7147	0,7216
10	0,6525	0,7056	0,7141

Таблиця 3.8 – Значення MAE при вирішенні завдання регресії за допомогою мережі MLP

d	n		
	10	15	20
1	0,0032	0,0032	0,0034
3	0,0057	0,0056	0,0058
5	0,0079	0,0072	0,0074
7	0,0095	0,0087	0,0086
10	0,0162	0,0113	0,0102

Мережа багатошарового перцептрона продемонструвала більш високу точність, ніж RBF мережа при вирішенні задачі регресії для ЧР. Було підтверджено висновки про горизонт прогнозування та довжину часового ряду, отримані при аналізі результатів роботи мережі RBF. Таким чином мережа MLP є більш точною при вирішенні завдань регресії часових рядів.

3.4 Мережа довгої короткострокової пам'яті (LSTM)

3.4.1 Архітектура мережі

Виходячи з математичної моделі мережі LSTM і можливостей, що надають нейромережеві бібліотеки Keras і TensorFlow, опишемо архітектуру мережі для вирішення задачі класифікації та регресії часових рядів:

- вхідний шар збігається за розміром із довжиною вхідного вектора, який складається з 15 фіч, що відповідає даним у проміжку часу в 15 днів;
- прихований шар реалізує контекстну обробку часового ряду та складається з 150 LSTM блоків;

– пакетна нормалізація вихідних значень прихованого шару, аналогічно мережі RBF служить частковою заміною механізму регуляризації;

– вихідний повнозв'язний шар, що відповідає одному нейрону з лінійною функцією активації у разі вирішення задачі регресії та двом нейронам з узагальненою логістичною функцією активації (Softmax) у разі вирішення задачі класифікації.

Таким чином, для вирішення завдання класифікації та регресії котирувань валютної пари євро-долар було розроблено дві архітектури мережі LSTM, представлені в таблиці 3.9 та таблиці 3.10.

Значення кількості нейронів в шарах нейронної мережі достатнє для захоплення часових залежностей у даних фінансових часових рядів. LSTM часто використовуються для задач, де важливі довготривалі залежності, і ця кількість нейронів обирається для балансування між потужністю моделі та обчислювальною ефективністю.

Таблиця 3.9 – Архітектура LSTM мережі для завдання класифікації

№	Тип шару	Розмір шару
0	Вхідний шар	n
1	Шар LSTM блоків	150
2	Пакетна нормалізація	150
3	Вихідний повнозв'язний шар	2
4	Логістичная функція активації (Softmax)	2

Таблиця 3.10 – Архітектура LSTM мережі для завдання регресії

№	Тип шару	Розмір шару
0	Вхідний шар	n
1	Шар LSTM блоків	150
2	Пакетна нормалізація	150

Продовження таблиці 3.10

3	Вихідний повнозв'язний шар	1
4	Лінійна функція активації	1

3.4.2 Навчання мережі

Розглянемо графіки навчання мереж довгої короткострокової пам'яті з допомогою методу оптимізації Nadam, представлені на рисунках 3.7–3.8.

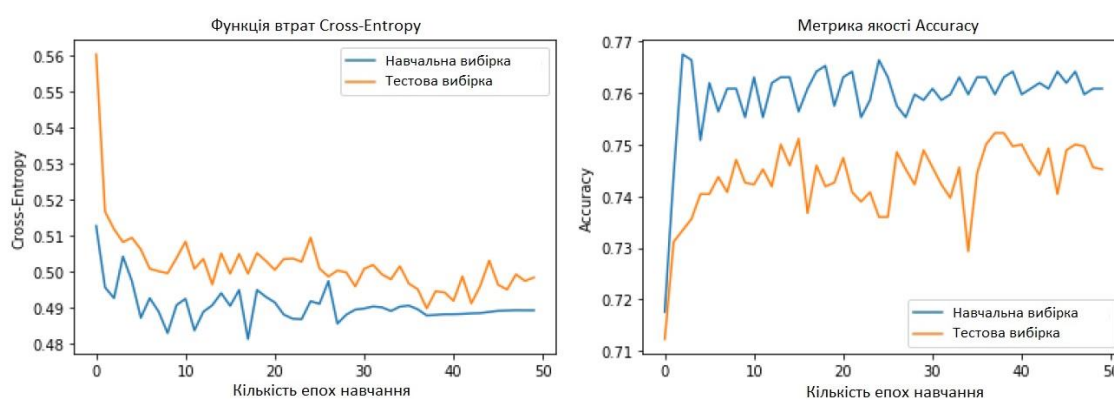


Рисунок 3.7 – Графік навчання мережі LSTM для завдання класифікації котирувань євро-долар

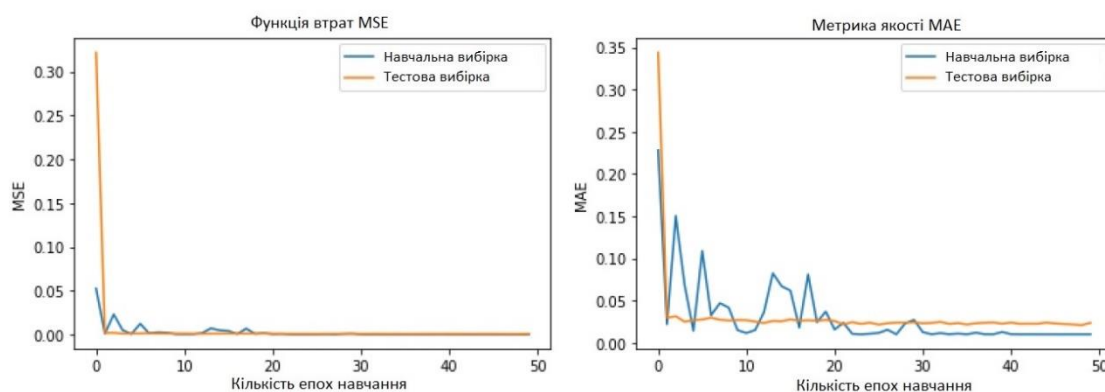


Рисунок 3.8 – Графік навчання мережі LSTM для завдання регресії котирувань євро-долар

Виходячи з аналогічних міркувань для мереж RBF і MLP можна зробити висновок, що розроблені архітектури мережі LSTM були успішно навчені для вирішення задач прогнозування часових рядів. Однак при порівнянні з графіками навчання мереж RBF та MLP спостерігається більш рівномірна збіжність.

3.4.3 Результат роботи мережі

Для проведення порівняльного аналізу точності прогнозування при розв'язанні задачі класифікації та регресії досліджуємо залежність якості прогнозування від довжини вхідного сегмента часового ряду n та горизонту прогнозування d . Для цього складемо таблиці 3.11–3.12.

Таблиця 3.11 – Значення Accuracy при вирішенні завдання класифікації за допомогою мережі LSTM

d	n		
	10	15	20
1	0,7423	0,7428	0,7425
3	0,7374	0,7372	0,7358
5	0,7253	0,7319	0,7303
7	0,7017	0,7247	0,7274
10	0,6625	0,7096	0,7241

Таблиця 3.12 – Значення MAE при вирішенні завдання регресії за допомогою мережі LSTM

d	n		
	10	15	20
1	0,0031	0,0031	0,0032
3	0,0056	0,0056	0,0056

Продовження таблиці 3.12

5	0,0075	0,0072	0,0071
7	0,0089	0,0085	0,0084
10	0,0167	0,0102	0,0101

Мережа довгої короткострокової пам'яті продемонструвала найбільшу точність вирішення завдання класифікації часових рядів. Також при аналізі результатів були отримані висновки про горизонти прогнозування та довжину часового ряду аналогічні висновкам мережі RBF.

При вирішенні завдання регресії часових рядів, мережа довгої короткострокової пам'яті також продемонструвала найбільшу точність. Таким чином, мережа LSTM може бути успішно застосована для вирішення завдань прогнозування часових рядів. Однак при вирішенні завдань регресії мережа MLP лише незначно поступається точності.

3.5 Порівняльний аналіз результатів прогнозування

Для проведення порівняльного аналізу якості прогнозування за допомогою різних нейронних мереж розглянемо результати їхньої роботи для двох значень горизонту прогнозування $d=3$ та $d=7$. Складемо таблиці 3.13 та 3.14 для задачі класифікації та таблиці 3.15–3.16 для задачі регресії.

3.5.1 Класифікація

При вирішенні завдання класифікації котирувань мережа LSTM виявилася найточнішою при всіх значеннях горизонту прогнозування і довжини часового ряду. Точності мереж MLP і RBF можна порівняти при розгляді більш довгих часових рядів, проте мережа MLP більш точна при малій довжині часового ряду. Враховуючи більш просту архітектуру та незначну різницю в точності з мережею LSTM, при вирішенні завдання

короткострокового прогнозування $d=3$ тренда можна використовувати мережі MLP та RBF.

Таблиця 3.13 – Значення Ассурасу при вирішенні завдання класифікації при значенні $d = 3$

Мережа	n		
	10	15	20
RBF	0,7258	0,7294	0,7314
MLP	0,7313	0,7316	0,7314
LSTM	0,7374	0,7372	0,7358

Таблиця 3.14 – Значення Ассурасу при вирішенні завдання класифікації при значенні $d = 7$

Мережа	n		
	10	15	20
RBF	0,6873	0,7159	0,7211
MLP	0,6973	0,7147	0,7216
LSTM	0,7017	0,7247	0,7274

3.5.2 Регресія

Для більш об'єктивного порівняльного аналізу результатів регресії розглянемо таблиці метрик якості та графіки регресії, отримані з урахуванням прогнозів, складених нейронними мережами. Графіки регресії котирувань валютної пари євро-долар наведено на рисунках 3.9–3.11.

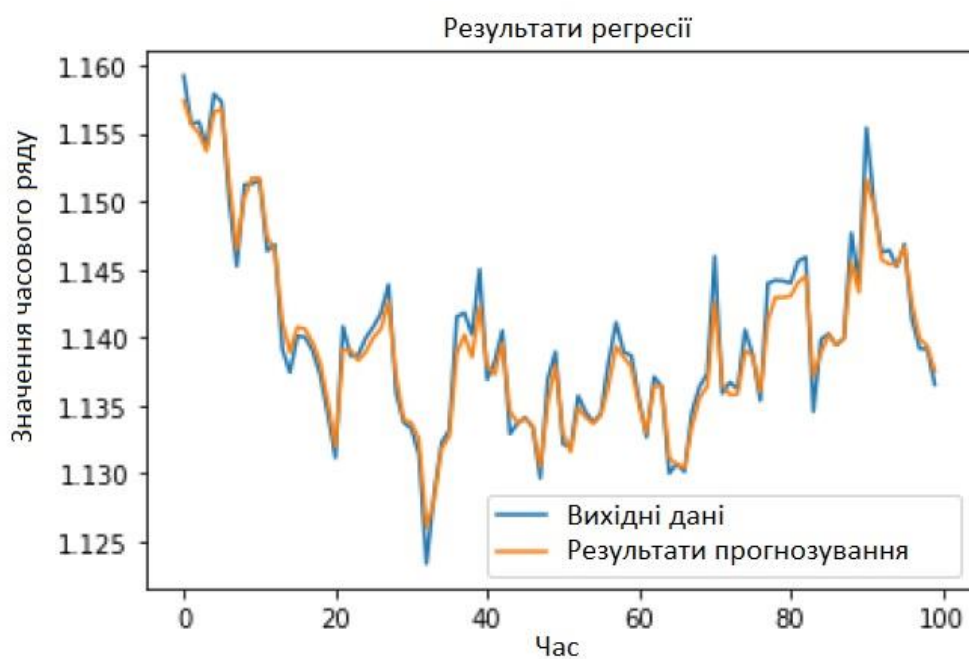


Рисунок 3.9 – Графік регресії за допомогою RBF

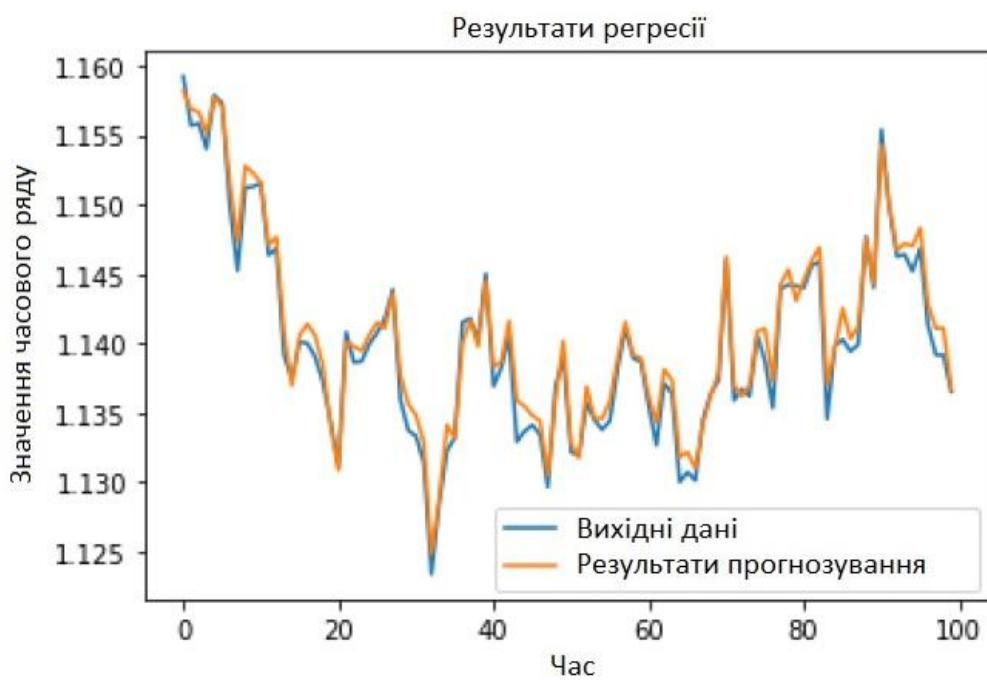


Рисунок 3.10 – Графік регресії за допомогою MLP

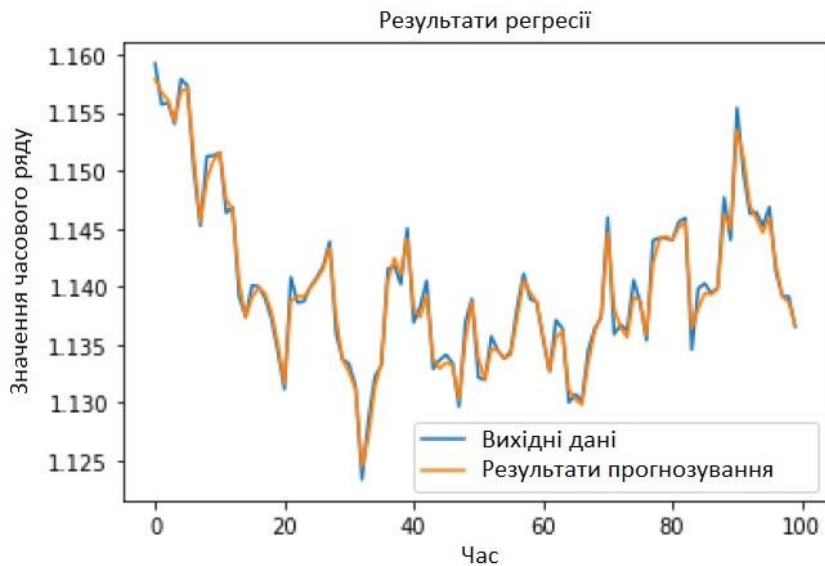


Рисунок 3.11 – Графік регресії за допомогою LSTM

Таблиця 3.15 – Значення MAE при вирішенні завдання класифікації при значенні $d = 3$

Мережа	n		
	10	15	20
RBF	0,0059	0,0057	0,0058
MLP	0,0057	0,0056	0,0058
LSTM	0,0056	0,0056	0,0056

Таблиця 3.16 – Значення MAE при вирішенні завдання класифікації при значенні $d = 7$

Мережа	n		
	10	15	20
RBF	0,0106	0,0087	0,0086
MLP	0,0095	0,0087	0,0086
LSTM	0,0089	0,0085	0,0084

При вирішенні завдання регресії котирувань валютної пари євро-долар всі три мережі продемонстрували порівняні значення точності, але мережа LSTM виявилася найточнішою за малої довжини часового ряду. Таким чином, для регресії часових рядів, схильних до впливу тренду, можуть бути використані всі розглянуті мережі. Однак найбільш оптимальною є мережа довгої короткострокової пам'яті, оскільки для прогнозування їй потрібно менше значень ряду. Враховуючи більш просту архітектуру та незначну різницю в точності з мережею LSTM, при вирішенні задачі регресії тренду можна використовувати мережі MLP та RBF.

З графіків регресії видно, що з мережі радіально-базисних функцій (RBF) характерне згладжування екстремальних значень. У той час як мережа багат шарового перцептрона справляється з екстремальними значеннями, але створює надмірні коливання. Обидві ці проблеми зведені до мінімуму при використанні довгої короткострокової мережі.

3.6 Висновки по розділу

Для проведення порівняльного аналізу точності прогнозування котирувань євро-долар були реалізовані алгоритми отримання прогнозів та навчання нейронних мереж. Вони були створені за допомогою мови програмування Python, із застосуванням бібліотеки машинного навчання TensorFlow та нейромережевої бібліотеки Keras.

Були реалізовані архітектури нейронних мереж, включаючи радіально-базисні функції, багат шаровий перцептрон та мережу довгої короткострокової пам'яті, для розв'язання задач класифікації та регресії. Усі розроблені мережі навчалися з використанням методу оптимізації Nadam. Після цього були отримані прогнози з використанням кожної з розроблених нейронних мереж.

Результати прогнозування часових рядів показали, що мережа довгої короткострокової пам'яті є найбільш точною для розглянутих задач прогнозування. Мережі багат шарового перцептрона та радіально-базисних функцій показали співставну з LSTM точність при регресії часових рядів, які піддаються впливу тренду. Таким чином, найбільш точною нейронною мережею для розв'язання задач прогнозування виявилася мережа LSTM. Водночас, враховуючи більш просту архітектуру, мережі MLP та RBF також можуть застосовуватися залежно від особливостей задачі.

ВИСНОВКИ

В процесі науково-дослідницької роботи над даною кваліфікаційною роботою мною було проведено всебічне дослідження актуальності завдання прогнозування часових рядів та аналіз предметної сфери фінансових часових рядів. Детально розглянуті існуючі методи та інструменти прогнозування часових рядів, а також теоретичні основи штучних нейронних мереж.

Мною було обрано три типи нейронних мереж для реалізації поставленої задачі: радіально-базисні функції (RBF), багат шаровий перцептрон (MLP) та мережу довгої короткострокової пам'яті (LSTM). Було детально розглянуто архітектуру кожної з цих мереж та способів їх використання для прогнозування часових рядів. Також було детально проаналізовано та розглянуто метод оптимізації Nadam, використаний для навчання мереж.

В експериментальній частині роботи були побудовані три архітектури нейронних мереж для прогнозування часового ряду котирувань євро-долар. Усі три нейронні мережі показали добрі результати. Для проведення порівняльного аналізу точності прогнозування були розроблені алгоритми отримання прогнозів та навчання нейронних мереж. Усі розроблені мережі навчалися з використанням методу оптимізації Nadam, після чого були отримані прогнози з використанням кожної з нейронних мереж.

Результати прогнозування показали, що мережа LSTM є найбільш точною для задач прогнозування часових рядів. Мережі MLP та RBF продемонстрували співставну з LSTM точність при регресії часових рядів, які піддаються впливу тренду. Таким чином, мережа LSTM виявилася найточнішою для розв'язання задач прогнозування. Водночас, враховуючи більш просту архітектуру, мережі MLP та RBF також можуть бути ефективно використані залежно від особливостей задачі.

Загалом, результати дослідження підтвердили ефективність використання нейронних мереж для прогнозування фінансових часових рядів, а також важливість вибору оптимальної архітектури та методу оптимізації для досягнення найкращих результатів.

Якщо вести мову про можливі варіанти розвитку цієї роботи, то можна запропонувати, наприклад, розширення набору даних шляхом використання більш тривалих часових рядів та різних фінансових інструментів може допомогти у створенні більш загальних моделей прогнозування. Покращення архітектури мереж через дослідження більш складних архітектур, таких як глибокі нейронні мережі, згорткові нейронні мережі (CNN) або комбінації CNN та LSTM, що може покращити точність прогнозування.

Використання методів автоматичної оптимізації гіперпараметрів, таких як байєсівська оптимізація або генетичні алгоритми, для пошуку найкращих параметрів моделей також є перспективним напрямком. Створення ансамблів моделей, які поєднують прогнози кількох різних нейронних мереж, може забезпечити більш стабільні та точні результати. Інтеграція додаткових макроекономічних та новинних даних як вхідних параметрів для моделей може підвищити точність прогнозування. Розробка моделей, які можуть адаптуватися до змін у часових рядах в реальному часі, буде корисною для динамічних фінансових ринків. Порівняння нейронних мереж з традиційними статистичними методами прогнозування, такими як ARIMA, GARCH, допоможе визначити переваги та недоліки кожного підходу.

Продовження досліджень у зазначених напрямках може сприяти подальшому покращенню точності та надійності прогнозування фінансових часових рядів за допомогою нейронних мереж.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Artificial intelligence and healthcare: Forecasting of medical bookings through multi-source time-series fusion / F. Piccialli та ін. *Information Fusion*. 2021. Т. 74. С. 1–16.
2. Introducing an operational method to forecast long-term regional drought based on the application of artificial intelligence capabilities / M. R. Kousari та ін. *Theoretical and Applied Climatology*. 2015. Т. 127. С. 361–380.
3. Mostafa M. M., Natarajan R. A neuro-computational intelligence analysis of the ecological footprint of nations. *Computational Statistics & Data Analysis*. 2009. Т. 53, № 9. С. 3516–3531.
4. Past, present and prospect of an Artificial Intelligence (AI) based model for sediment transport prediction / H. A. Afan та ін. *Journal of Hydrology*. 2016. Т. 541. С. 902–913. URL: <https://doi.org/10.1016/j.jhydrol.2016.07.048> (дата звернення: 10.05.2024).
5. Popovic D., Palit A. K. Computational Intelligence in Time Series Forecasting. London : Springer-Verlag, 2005. URL: <https://doi.org/10.1007/1-84628-184-9> (дата звернення: 10.05.2024).
6. Chen S.-H., Wang P. P., Kuo T.-W. Computational Intelligence in Economics and Finance: Shifting the Research Frontier. *Computational Intelligence in Economics and Finance*. Berlin, Heidelberg. С. 1–23. URL: https://doi.org/10.1007/978-3-540-72821-4_1 (дата звернення: 10.05.2024).
7. NSE Stock Market Prediction Using Deep-Learning Models / M. Hiransha та ін. *Procedia Computer Science*. 2018. Т. 132. С. 1351–1362. URL: <https://doi.org/10.1016/j.procs.2018.05.050> (дата звернення: 10.05.2024).
8. Chatfield C. Time-series forecasting. *Significance*. 2005. Т. 2, № 3. С. 131–133. URL: <https://doi.org/10.1111/j.1740-9713.2005.00117.x> (дата звернення: 11.05.2024).

9. The 5 Different Trend Lines Explained. *The Data School*. URL: <https://www.thedataschool.com.au/alex-taylor-jackson/the-5-different-trend-lines-explained/> (дата звернення: 11.05.2024).
10. How to Classify Trends in a Time Series Regression Model. *For Dummies*. URL: <https://www.dummies.com/article/business-careers-money/business/accounting/calculation-analysis/how-to-classify-trends-in-a-time-series-regression-model-145932/> (дата звернення: 12.05.2024).
11. Diebold F. X., Mariano R. S. Comparing Predictive Accuracy. *Journal of Business & Economic Statistics*. 1995. Т. 13, № 3. С. 253–263.
12. Часові ряди. Прикарпаття, 2018. 82 с. URL: <https://kstat.pnu.edu.ua/wp-content/uploads/sites/63/2018/04/Часові-ряди.pdf> (дата звернення: 12.05.2024).
13. Теорія статистики. 2-ге вид. Київ : Центр учб. літ., 2013. 592 с.
14. Moroz V., Milovska K. Wavelet analysis and forecasting of financial time series. *Bulletin of the National Technical University "KhPI" A series of "Information and Modeling"*. 2022. Т. 1, № 1-2 (7-8). URL: <https://doi.org/10.20998/2411-0558.2022.02.11> (дата звернення: 13.05.2024).
15. Hamilton J. D. Time Series Analysis. New Jersey : Princeton University Press, 1994. 820 с.
16. Берзлев О. Сучасний стан інформаційних систем прогнозування часових рядів. *Управління розвитком складних систем*. 2013. № 13. С. 78–82.
17. Андрусенко Ю.О. Аналіз основних моделей прогнозування часових рядів. *Збірник наукових праць Харківського національного університету Повітряних сил*. 2020. № 3(65). С. 91-96.
18. Ляшенко О.І., Кравець Т.В., Хрущ Л.З. Застосування пакетів прикладних програм в економетричному моделюванні фінансових часових рядів. *Економіко-математичне моделювання соціально-економічних систем*. 2017. Вип. 22. С. 33-59.

19. Огляд методів машинного навчання в задачі прогнозування фінансових часових рядів / Гурєєва К.М., Кудін О.В. Лісняк Ф.О. *Вісник Запорізького національного університету. Фізико-математичні науки*. 2018. №2. С. 18-28.
20. Rashid T. *Make your own neural network: A gentle journey through the mathematics of neural networks, and making your own using the Python computer language*. 2016. 222 с.
21. Learning M. *Machine Learning Proceedings 1989*. Morgan Kaufmann, 1989. 510 с.
22. Ben-David S., Shalev-Shwartz S. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.
23. Müller A. C., Guido S. *Introduction to Machine Learning with Python: A Guide for Data Scientists*. O'Reilly Media, Incorporated, 2018.
24. *Machine Learning* / М. Mohssen та ін. Boca Raton, 2016. 226 с.
25. Buza K. *Time Series Classification and its Applications. the 8th International Conference*, м. Novi Sad, Serbia, 25–27 черв. 2018 р. New York, New York, USA, 2018. URL: <https://doi.org/10.1145/3227609.3227690> (дата звернення: 18.05.2024).
26. F.M. Zain M., M. Abd S. *Multiple Regression Model for Compressive Strength Prediction of High Performance Concrete. Journal of Applied Sciences*. 2008. Т. 9, № 1. С. 155–160. URL: <https://doi.org/10.3923/jas.2009.155.160> (дата звернення: 18.05.2024).
27. Thompson P. A. *An MSE statistic for comparing forecast accuracy across series. International Journal of Forecasting*. 1990. Т. 6, № 2. С. 219–227. URL: [https://doi.org/10.1016/0169-2070\(90\)90007-x](https://doi.org/10.1016/0169-2070(90)90007-x) (дата звернення: 18.05.2024).
28. Botchkarev A. *A New Typology Design of Performance Metrics to Measure Errors in Machine Learning Regression Algorithms. Interdisciplinary Journal of Information, Knowledge, and Management*. 2019. Т. 14. С. 045–076. URL: <https://doi.org/10.28945/4184> (дата звернення: 19.05.2024).

29. Golik P., Doetsch P., Ney H. Cross-entropy vs. squared error training: a theoretical and experimental comparison. *Interspeech 2013*. ISCA, 2013.
30. Galdi P., Tagliaferri R. Data Mining: Accuracy and Error Measures for Classification and Prediction. *Encyclopedia of Bioinformatics and Computational Biology*. 2019. С. 431–436. URL: <https://doi.org/10.1016/b978-0-12-809633-8.20474-3> (дата звернення: 20.05.2024).
31. Heaton J. Introduction to Neural Networks for Java, Second Edition. St. Louis, Mo : Heaton research, 2008. 439 с.
32. Krizhevsky A., Sutskever I., Hinton G. E. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*. 2017. Т. 60, № 6. С. 84–90. URL: <https://doi.org/10.1145/3065386> (дата звернення: 21.05.2024).
33. Hochreiter S., Schmidhuber J. Long Short-Term Memory. *Neural Computation*. 1997. Т. 9, № 8. С. 1735–1780. URL: <https://doi.org/10.1162/neco.1997.9.8.1735> (дата звернення: 22.05.2024).
34. Amari S.-i. Backpropagation and stochastic gradient descent method. *Neurocomputing*. 1993. Т. 5, № 4-5. С. 185–196.
35. Kingma D. P., Ba J. Adam: A Method for Stochastic Optimization. *3rd International Conference for Learning Representations*, м. San Diego, 23 лип. 2015 р.
36. Dozat T. Incorporating Nesterov Momentum into Adam. *Proceedings of the 27th International Conference on Machine Learning*. 2010. С. 372–376.