

ДВНЗ «Донецький національний технічний університет»
Факультет комп'ютерно-інформаційних технологій та автоматизації
Кафедра прикладної математики та інформатики

«До захисту допущено»

Завідувач кафедри ПМІ

_____ **О. А. Дмитрієва**
(підпис) (ініціали, прізвище)

«_____» _____ 2021 р.

**Випускна кваліфікаційна робота
магістра**

на тему «Прогнозування показників фінансового ринку за допомогою
нейронної мережі на основі LSTM»

Виконав: студент _____ 2 _____ курсу, групи КНМ-20
(шифр групи)

спеціальності 122 Комп'ютерні науки
(шифр і назва спеціальності)

_____ **Мирошніченко І. В.**
(прізвище та ініціали) (підпис)

Керівник _____ **доц. каф. ПМІ, к.т.н., доц. Маслова Н.О.**
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Рецензент _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Рецензент _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Нормоконтроль:

_____ **І. А. Назарова**
(підпис)

_____ (дата)

*Засвідчую, що у цій дипломній
роботі немає запозичень з праць
інших авторів без відповідних
посилань.*

Студент _____
(підпис)

Покровськ – 2021 р.

ДВНЗ «Донецький національний технічний університет»

Факультет комп'ютерно-інформаційних технологій та автоматизації

Кафедра прикладної математики та інформатики

Освітній ступінь магістр

Спеціальність 122 Комп'ютерні науки

(шифр і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри ПМІ

_____/О. А. Дмитрієва/

« » 2021 року

ЗАВДАННЯ

НА ДИПЛОМНУ РОБОТУ СТУДЕНТУ

Мирошниченку Ігнату Васильовичу

(прізвище, ім'я, по батькові)

1. Тема роботи «Прогнозування показників фінансового ринку за допомогою нейронної мережі на основі LSTM»

керівник роботи Маслова Наталя Олександрівна, к.т.н., доцент

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом від « 22 » вересня 20 21 року № 570

2. Строк подання студентом роботи 7 грудня 2021 року

3. Вихідні дані до роботи результати переддипломної практики

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1) поняття штучної нейронної мережі; 2) проектування штучної нейронної мережі; 3) навчання нейронної мережі; 4) результати дослідження

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) робота містить 3 таблиці, 28 рисунків, 0 діаграм, слайдів презентації та інший графічний матеріал, у кількості, достатній для демонстрації сутності роботи

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Назарова Ірина Акопівна	—	—

7. Дата видачі завдання 4 жовтня 2021 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Строк виконання етапів роботи	Примітка
1	Огляд літератури з предметної області	04.10.21-17.10.21	
2	Збір інформації і методичних відомостей щодо застосування нейронних мереж у економічній діяльності	18.10.21-31.10.21	
3	Розробка структури програмного забезпечення, вибір даних для обробки та експериментів.	01.11.21-14.11.21	
4	Тестування системи	15.11.21-30.11.21	
5	Оформлення пояснювальної записки та опис створеної системи. Проходження нормоконтролю	01.12.21-06.12.21	
6	Перевірка пояснювальної записки антиплагіатною системою. Оформлення презентації до роботи, графічного матеріалу та рецензування роботи	07.12.21-20.12.21	
7	Захист роботи	21.12.21	

Студент

(підпис)

Мирошниченко І.В.

(прізвище та ініціали)

Керівник роботи

(підпис)

Маслова Н.О.

(прізвище та ініціали)

АНОТАЦІЯ

Мирошниченко І.В. Прогнозування показників фінансового ринку за допомогою нейронної мережі на основі LSTM / Випускна кваліфікаційна робота на здобуття освітнього ступеня «магістр» за спеціальністю 122 Комп'ютерні науки. – ДВНЗ ДонНТУ, Покровськ, 2021.

Об'єктом дослідження є методики прогнозування показників фінансового ринку.

Предметом дослідження є методи застосування нейронних мереж для прогнозування фінансового ринку.

Мета роботи – розгортання та навчання нейронної мережі для прогнозування показників фінансового ринку.

Основні задачі роботи:

- проаналізувати існуючі методи прогнозування показників фінансового ринку з акцентом на нейронні мережі;
- обрати тип нейронної мережі, найкращій за параметрами точності та глибини прогнозу (параметри – розрахувати);
- розгорнути та навчити мережу на спеціально підготованому наборі даних;
- провести експериментальні дослідження, оцінити якість прогнозу з застосуванням обраної мережі.

У даному проекті проведено розгортання та навчання нейронної мережі обраного типу(LSTM), отримано прогноз основних показників фінансового ринку та проведено їх аналіз.

Ключові слова: штучні нейронні мережі, LSTM, мова Python.

Список публікацій здобувача:

1. Myroshnychenko I.V. Modeling neural merge for forecasting the stock market / I.V. Myroshnychenko, S.O. Popova, G.B. Myroshnychenko // Наукові досягнення та відкриття сучасної молоді. Матеріали міжнародної науково-практичної конференції (2021). – С. 58 – 61.

ABSTRACT

Miroshnichenko I. Forecasting financial market indicators using a neural network based on LSTM / Final qualification work for the degree of "master" in the specialty 122 Computer Science. – SHEI DonNTU, Pokrovsk, 2021.

The object of research is the methods of forecasting financial market indicators.

The subject of research is the methods of using neural networks to predict the financial market.

The purpose of the work is to deploy and train a neural network to forecast financial market indicators.

The main tasks of the work:

- to analyze the existing methods of forecasting financial market indicators with an emphasis on neural networks;
- choose the type of neural network, the best in terms of accuracy and depth of forecast (parameters – calculate);
- deploy and train the network on a specially prepared data set;
- to conduct experimental research, assess the quality of the forecast using the selected network.

In this project, the deployment and training of a neural network of the selected type (LSTM), the forecast of the main indicators of the financial market and their analysis.

Keywords: artificial neural networks, LSTM, Python language.

Publisher's publication list:

1. Myroshnychenko I.V. Modeling neural merge for forecasting the stock market / I.V. Myroshnychenko, S.O. Popova, G.B. Myroshnychenko // Scientific achievements and discovery of modern youth. Proceedings of the international scientific-practical conference (2021). – P. 58 – 61.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів.....	7
Вступ.....	8
Розділ 1 Поняття штучної нейронної мережі	10
1.1 Штучні нейронні мережі	10
1.2 Типи нейронних мереж	11
1.3 Сучасні фінансові ринки, ринок FOREX	15
1.4 Прогнозування фінансових ринків	16
1.4.1 Прогноз і цілі його використання	16
1.4.2 Основні поняття і визначення проблеми прогнозування	18
1.4.3 Методи прогнозування фінансових ринків	22
1.4.4 Використання систем зі штучною «пам'яттю» для вирішення завдань прогнозування	25
1.5 Висновки за розділом.....	27
Розділ 2 Проектування штучної нейронної мережі	28
2.1 Особливості застосування штучних нейронних мереж в різних областях	28
2.2 Прогнозування біржової діяльності	32
2.3 Особливості використання нейронних мереж на біржі	36
2.4 Параметри нейронної мережі	38
2.5 Модель LSTM	39
2.6 Інструменти для імплементації нейронної мережі	41
2.7 Висновки за розділом.....	42
Розділ 3 Навчання нейронної мережі.....	44
3.1 Опис вхідних та вихідних даних	44
3.2 Загальна структура програмної моделі	45
3.3 Висновки за розділом.....	58
4 Результати дослідження	59

4.1 Результати моделі	59
4.2 Порівняння працездатності	61
4.3 Висновки за розділом.....	65
Висновки.....	66
Список використаних джерел.....	68
Додаток А Зауваження нормоконтролера.....	73
Додаток Б Програмний код.....	74
Додаток В Роздатковий матеріал.....	78

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ
І ТЕРМІНІВ

КПП – Карусель постійної помилки

BPTT – Back Propagation Thru Time

CNN – Convolutional Neural Network

LSTM – Long Short-term Memory

MAE – Mean Actual Error

RMSE – Root Mean Square Error

RNN – Recurrent Neural Network

ВСТУП

Штучні нейронні мережі є обчислювальною системою з величезним числом паралельно функціонуючих простих процесорів з безліччю зав'язків.

Вони активно застосовуються там, де звичайні алгоритмічні рішення виявляються неефективними або зовсім неможливими. Практика використання нейромереж показала їх застосовність в таких областях, як прогнозування, виявлення залежностей, ситуаційне управління. Все це може бути застосовано на фінансових ринках.

Численні спостереження за фінансовими ринками засвідчили невідповідність припущень про раціональність інвесторів та інформаційну ефективність ринків реальним умовам, які існують у ряді випадків. У зв'язку з цим стає актуальним побудова нових моделей прогнозування розвитку фінансових ринків, систематизації та вибору ключових особливостей динаміки реальних ринків капіталу.

Об'єктом дослідження є методики прогнозування показників фінансового ринку.

Предметом дослідження є методи застосування нейронних мереж для прогнозування фінансового ринку.

Мета роботи – розгортання та навчання нейронної мережі для прогнозування показників фінансового ринку.

Основні задачі роботи:

- проаналізувати існуючі методи прогнозування показників фінансового ринку з акцентом на нейронні мережі;
- обрати тип нейронної мережі, найкращій за параметрами точності та глибини прогнозу (параметри – розрахувати);
- розгорнути та навчити мережу на спеціально підготованому наборі даних;

– провести експериментальні дослідження, оцінити якість прогнозу з застосуванням обраної мережі.

У даному проекті проведено розгортання та навчання нейронної мережі обраного типу(LSTM), отримано прогноз основних показників фінансового ринку та проведено їх аналіз.

Методи досліджень: моделювання нейронних мереж, аналіз, статистика.

Структура та обсяг роботи. Випускна кваліфікаційна робота обсягом 00 машинописних сторінок, складається з вступу, чотирьох розділів, висновків, переліку використаних джерел, що складається з 37 найменувань. Робота містить 28 рисунки, 00 діаграм, 02 таблиці.

РОЗДІЛ 1.

ПОНЯТТЯ ШТУЧНОЇ НЕЙРОННОЇ МЕРЕЖІ

1.1 Штучні нейронні мережі

Під нейронними мережами маються на увазі обчислювальні структури, які моделюють прості біологічні процеси, зазвичай асоціюються з процесами у нервових клітинах живого організму.

Штучні нейронні мережі – це програмна імплементація нейронних структур людського мозку [1]. Вони є розподіленими та паралельними системами, які дозволяють гнучко навчатися шляхом аналізу позитивних і негативних впливів. Основним елементарним перетворювачем у цих мережах є штучний нейрон, або просто нейрон, названий так за аналогом з його біологічним прототипом [1, 2].

Штучний нейрон, так само як і його природний прототип, біологічний нейрон, має групу синапсів (входів), які з'єднані з виходами інших нейронів, а також аксон – вихідне з'єднання цього нейрона – де сигнал збудження або гальмування нейронних зв'язків надходить на синапси (входи) інших нейронів.

Загальний вигляд нейрона наведено на рисунку 1.1 [1].

На рисунку наведено принцип зв'язків сигналів у штучному нейроні. Де, u_j - сигнал, що надходить від нейрона j ; S_k – скалярний добуток вектору вхідних сигналів і вектору ваг; F_k – функція збудження; u_k – вихідний сигнал нейрона.

Штучний нейрон є основою будь-якої штучної нейронної мережі. Нейрони – це відносно прості схожі елементи, які імітують роботу нейронів мозку [3]. Кожен нейрон характеризується своїм поточним станом, подібним до функціонування нервових клітин мозку, які можуть бути порушені та пригнічені.

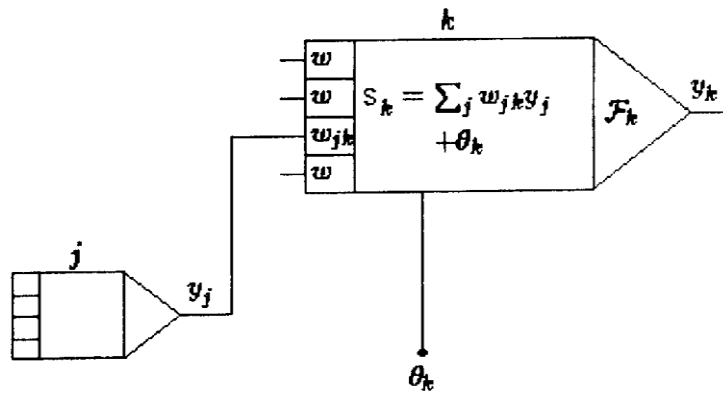


Рисунок 1.1 – Штучний нейрон – елемент штучної нейронної мережі

1.2 Типи нейронних мереж

Існує безліч видів нейронних мереж, які використовуються в залежності від поставлених завдань. Розглянемо різні типи нейронних мереж, такі як, мережі зворотного поширення, повторювані нейронні мережі (RNN), модель Long Short-Term Memory (LSTM).

Більшість використовуваних на практиці нейронних мереж є мережами зворотного поширення – найпопулярніший сучасний алгоритм. Але алгоритм зворотного поширення не позбавлений недоліків [4]. По-перше, немає гарантії, що нейронна мережа зможе навчатися в кінцевому підсумку: зусилля та витрати часу на навчання мережі часто витрачаються даремно. У цьому випадку навчання повторюють без всякої впевненості, що результат буде кращим. Тобто, найкраще рішення не буде отримано внаслідок того, що алгоритм навчання може потрапити в «пастку» так званого локального мінімуму помилки [4].

Рекурентні нейронні мережі (RNN) (рис. 1.2) складають дуже потужний клас обчислювальних моделей, здатних створювати екземпляри майже довільної динаміки. Однак ступінь, в якому цей потенціал можна використати, обмежений методом застосовуваної навчальної процедури. Методи на основі градієнта або регулярне навчання в режимі реального часу мають спільне та велике обмеження [5].

Часова еволюція інтегрального шляху всіх сигналів помилок назад експоненціально залежить від величини ваг. Це означає, що зворотно

розповсюджена помилка швидко наближається до нуля або до нескінченності. Отже, стандартні RNN не можуть навчатися за наявності затримок часу, що перевищують 5-10 дискретних часових кроків між відповідними вхідними подіями та цільовими сигналами. Проблема зникаючої помилки ставить під сумнів, чи може стандартна RNN-мережа продемонструвати практичні переваги над періодичними нейронними мережами прямого поширення [5].

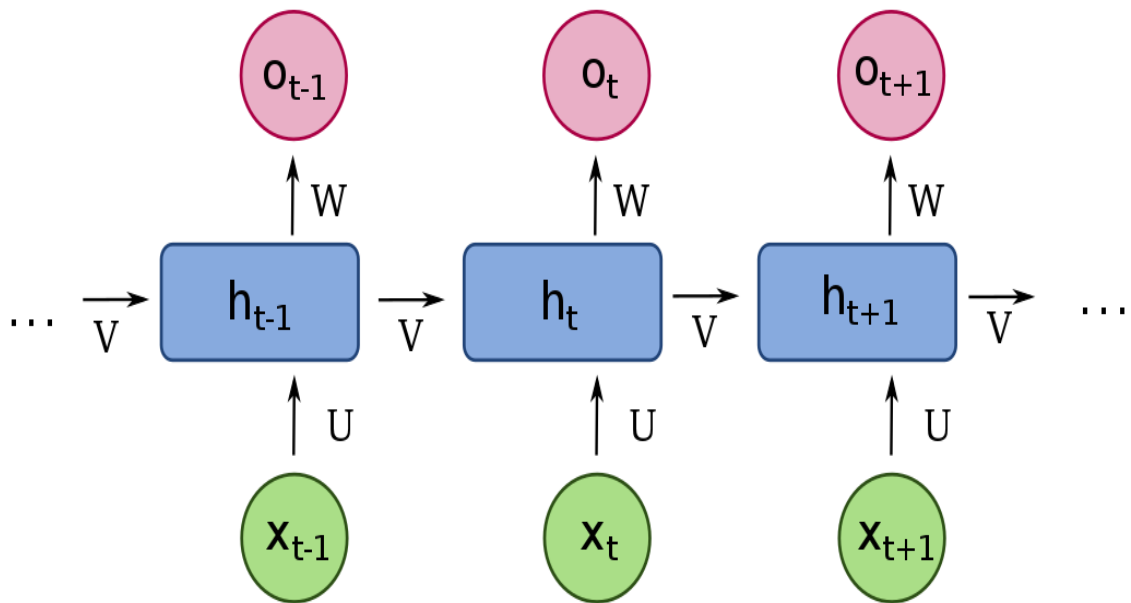


Рисунок 1.2 – Приклад нейронної мережи типу RNN

Проблему подолання мінімальних лагів часу, яка виникає при навчанні RNN-мережі можна вирішити за допомогою використання моделі Long Short-Term Memory (LSTM) (рис 1.3) [6, 7]. Ця модель може навчитися подолати мінімальні затримки часу, що перевищує 1000 дискретних кроків по часу шляху забезпечення постійної помилки через «каруселі з постійними помилками» (КПП) у спеціальних підрозділах, які називаються клітинками. Мультиплікативні блоки шлюзів вчяться відкривати та закривати доступ до осередків.

Алгоритм навчання LSTM локальний у просторі та часі; його обчислювальна складність за крок часу та вагу становить $O(1)$. Він вирішує складні завдання з тривалою затримкою, які ніколи не були вирішені попередніми алгоритмами RNN. Для порівняння LSTM з альтернативними підходами. LSTM не може навчитися правильно обробляти певні часові ряди (дуже довгі або постійні), які апіорі не сегментовані у відповідних навчальних підпоследовностях з чітко десигментованих початків і кінців. Проблема в тому, що безперервний вхідний потік може в кінцевому підсумку призвести до росту внутрішніх значень блоків без обмеження, навіть якщо повторення проблеми буде вимигати час від часу їх скидати.

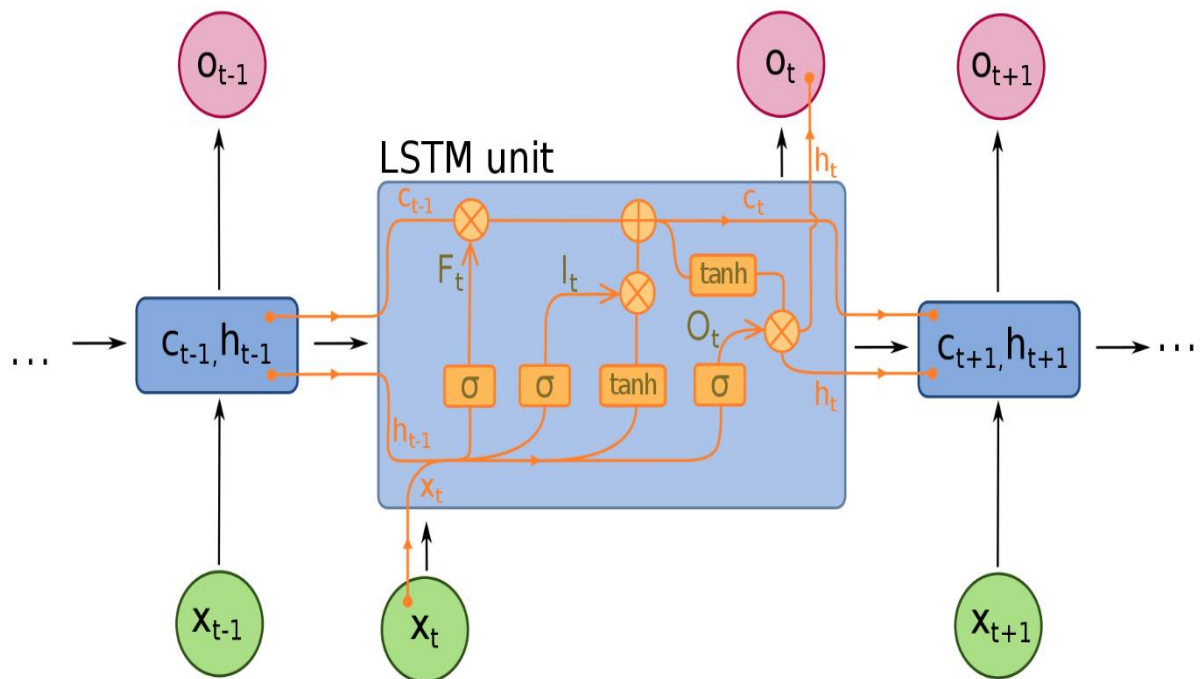


Рисунок 1.3 – Приклад нейронної мережі типу LSTM [7]

Кожен блок комірок пам'яті потребує дві додаткові одиниці (вхідний і вихідний шлюз). В порівнянні для стандартних повторюваних мереж, однак, це не збільшує кількість ваг більше ніж коефіцієнтів, кожна звичайна прихована одиниця замінюється щонайбільше на 3 одиниці в архітектурі LSTM, збільшуючи кількість ваг у 3-2 рази у повністю підключеному випадку [7].

Отже, складність оновлення LSTM за часовий крок і вагу має порядок $O(1)$, по суті така ж, як і для повністю підключеної рекурентної мережі BPTT. Складність зберігання на вагу також становить $O(1)$. Тому складність зберігання не залежить від довжини вхідної послідовності. Отже, розширений LSTM локальний у просторі та часі так само, як і стандартний LSTM.

Розроблено багато інших алгоритмів навчання нейронних мереж, що мають свої переваги, проте, слід зазначити, що всі вони не вільні від обмежень.

Технологія вимагає поліпшення існуючих методів і розширення теоретичних основ, для того щоб нейронні мережі повністю реалізували свої потенційні можливості.

Найбільші проблеми застосування штучних нейронних мереж (ШНМ) пов'язані як з освоєнням непрофільними фахівцями фундаментальних понять ШНМ, так і з освоєнням прийомів нейромережового моделювання специфічних для того чи іншого завдання або предметної області [8].

Відсутність адаптованої документації та навчального курсу, що включають докладний опис методів і прикладів, індексний і предметний покажчики, істотно обмежує застосування ШНМ на фондовій біржі.

В даний час навчальні системи і тренажери по застосуванню ШНМ практично відсутні або недоступні. Це обумовлено тим, що методики використання ШНМ надзвичайно різноманітні.

Області застосування нейронних мереж досить різноманітні - це розпізнавання тексту й мови, семантичний пошук, експертні системи і системи підтримки прийняття рішень, передбачення курсів акцій, системи безпеки, аналіз текстів. Однією з найбільш складних і затребуваних здібностей нейронних мереж є прогнозування [9].

До завдань прогнозування на біржі можна віднести прогнозування короткострокових і довгострокових тенденцій (збір і зберігання статистичних даних; визначення для даного ринку або Інструменту прогнозованої величини і набору факторів, що впливають; обчислення цікавить величини відповідно

до певної функцією, значеннями факторів, що впливають на прогнозований момент і видом прогнозу), а також прогнозування тенденцій фондового ринку.

1.3 Сучасні фінансові ринки, ринок FOREX

Валютний ринок FOREX є найпотужнішим фінансовим інструментом, за допомогою якого можна отримувати високу рентабельність вкладених активів навіть при невеликих коливаннях курсу валют на світовому валютному ринку.

На валютному ринку щодня здійснюються операції по купівлі продажу валюти на 4,5 трильйонів доларів і по обсягам здійснених операцій на ринку він нарощує свій потенціал на 5-7% кожного року [10-13]. Це набагато більше ніж обсяг здійснення операцій будь-якого ринку (товарного, фондового, ф'ючерсного, ринку фінансових послуг, ринку страхових послуг). Сам по собі ринок FOREX має деякі особливості, в першу чергу цей ринок характеризується тим, що в нього немає єдиного центру. Як і для інших фінансових ринків для ринку FOREX притаманно постійний пошук рівноваги. За останніми прогнозами щоденний торговий обсяг фінансових активів на ринку FOREX може зрости до 9 трильйонів доларів вже протягом наступного десятиліття [13].

Доходи від валютних операцій на ринку FOREX складають до 60% від усіх доходів у багатьох банків (в окремих випадках до 80%, як, наприклад, в 1994 р. У Union Bank of Switzerland – одного з найбільших банків світу).

На валютному ринку приймає участь безліч гравців. До їх складу відносяться комерційні і національні банки, валютні біржі, інвестиційні і пенсійні фонди, компанії яким для виконання своїх функцій треба здійснювати зовнішньоторговельні операції, брокерські компанії, приватні особи. Окремо треба сказати, що внаслідок участі в торгах на валютному ринку FOREX національні та комерційні банки отримують доходи до 60% від загальної виручки [13]. Тож можливість застосовувати інструменти нейронних мереж

для прогнозування курсів валют та акцій дає поштовх для розвитку всіх галузей економіки [12].

На даний момент будь-яка банківська система не може бути конкурентоспроможною і існувати без можливості торгівлі на ринку FOREX, так як валютні операції на ринку дозволяють банкам бути незалежними від змін обмінного курсу, якщо активи банку знаходяться в валютах інших країн. На валютний курс на фінансовому ринку впливають різноманітні фактори до яких відносяться: збалансованість взаємних платежів; стан світової економіки та окремих країн; прогнози, зроблені на підставі технічного аналізу валютного ринку; політичні фактори.

Здійснення операцій на ринку FOREX відбувається цілодобово за допомогою інтернет-комунікацій і працює як єдиний механізм. Міжнародні інформаційні системи такі як REUTERS, DOW JONES, CQG, BLOOMBERG, TENFORE та інші використовують для отримання інформації про фінансово-економічний стан на фінансових ринках.

1. 4 Прогнозування фінансових ринків

1.4.1 Прогноз і цілі його використання

Послідовність подій, яка відбувається після прийняття рішення впливає на його ефективність. Можливість прогнозувати впливів некерованих аспектів цих подій на ефективність рішення дозволяє приймати найкращі рішення, тому ефективні системи організації планування та управління зазвичай в соєму арсеналі виконують функцію прогнозування.

Прогнозування – це можливість передбачення майбутніх подій, заснований на аналізі минулого та сьогодення того чи іншого явища або процесу [14]. Метою прогнозування є зниження ризику прийняття неефективних рішень. Прогнозування зазвичай насить суб'єктивний характер, так як на саму подію можуть впливати безліч факторів, які неможливо врахувати в моделі прогнозу, але його похибки (точність прогнозу) залежать від використовуваної системи прогнозування. Надаючи більше інформаційних

даних для формування моделі прогнозу можна підвищити точність прогнозу та зменшити втрати від невизначеності [12].

Оскільки прогнозування ніколи не може повністю усунути ризик прийняття помилкового рішення, необхідно чітко визначити неточність прогнозу. Зазвичай приймають рішення на основі результатів прогнозу (прогноз вважається вірним), з урахуванням можливої похибки в прогнозі [12].

Таким чином ефективна система прогнозування буде тільки в тому випадку, коли вона буде враховувати значення похибки прогнозу. Використання такого підходу в більшій ступені зменшує ризик, який пов'язаний з процесом прийняття рішення.

Саме по собі прогнозування не є самометою у прийнятті рішень з будь-якої проблеми, але ефективна побудована система прогнозування є частиною. Слід зазначити, що прогнозування не є кінцевою метою. Система прогнозування є частиною всеосяжної системи управління і як його підсистема взаємодіє з іншими складовими системи управління рішеннями в умовах невизначеності та відіграє важливу роль у результаті.

Для будь-якої інвестиційної діяльності (а фінансові ринки, це такі ринки на якому обертаються інвестиційні інструменти) характерно прогнозування фінансових часових рядів, як інструмента за допомогою якого приймаються рішення з приводу купівлі-продажу інвестиційних інструментів [8, 10, 12]. Це ствердження впливає з самої економічної сутності поняття інвестування: ми вкладаємо фінансові ресурси зараз заради отримання інвестиційного доходу у майбутньому [15]. Тобто наше рішення про інвестування фінансових ресурсів засновано на принципі передбачення майбутнього. Таким чином, передбачення фінансових тимчасових рядів є основою діяльності всієї інвестиційної індустрії, всіх фондових бірж і небіржових систем для торгівлі фінансовими інструментами.

1.4.2 Основні поняття і визначення проблеми прогнозування

Визначення проблеми прогнозування вимагає її деталізації. Результати прогнозування використовуються для підтримки прийняття рішень, на основі передбачених припущень. Таким чином, характер прийнятих рішень визначає більшість необхідних характеристик системи прогнозування.. Вивчення вирішуваної проблеми має допомогти відповісти на наступні питання [12]:

- що необхідно прогнозувати (об'єкти, явища);
- яким має бути прогноз (його форма);
- які фактори часу повинні бути в ньому враховані;
- яка бажана точність прогнозу.

Щоб визначити те, що нам потрібно прогнозувати, необхідно визначити змінні, які ми будемо аналізувати та прогнозувати. На цьому етапі вкрай важливо визначити необхідний рівень потрібної деталізації. На рівень можливої деталізації прогнозу можуть різні фактори: доступність і точність наведених даних, вартісні витрати на аналіз даних та переваги користувачів щодо результатів прогнозу. При виникненні ситуацій коли набір змінних не може бути визначено конкретно, можна використовувати для прогнозу декілька альтернативних моделей і вибрати ті змінні використання яких буде давати найкращий результат.

Наступний етап при побудові системи прогнозування (рис. 1.4) є визначення параметрів періоду прогнозу, параметрів горизонту прогнозу та параметру інтервалу прогнозу. Час на який складається прогноз і буде прогнозним періодом. Наприклад коли нам потрібно спрогнозувати яким буде курс валют на валютному ринку на наступний день, то в цьому випадку період прогнозу буде 1 доба.

Кількість періодів в межах котрих прогноз буде виконуватися з обумовленої точністю є горизонтом прогнозу. В нашому наведеному прикладі якщо взяти припущення, що прогноз курсу валют потрібен буде кожного дня протягом 10 днів, то горизонт прогнозу буде – 10 днів [16].

Частота послідовності розрахунку нового прогнозу є інтервалом прогнозу. Період прогнозу доволі часто збігається з інтервалом прогнозу. У цьому випадку при побудові наступного прогнозу використовується інформація та вимоги для неї за останній прогнозований період. Якщо горизонт завжди має однакову довжину (період T) прогноз переглядається в кожному періоді то в цьому випадку знову прогнозується вимога для періоду $(T - 1)$ і робиться вихідний прогноз для періоду T [16].

Область діяльності для якої виконується прогноз зазвичай використовує тільки їм притаманні умови прийняття рішень і все це буде впливати на вибір періоду і горизонту прогнозування. При чому повинна виконуватися наступна умова, горизонт прогнозування повинен дорівнювати часу на реалізацію рішення, яке прийняте на основі отриманого прогнозу. Але треба враховувати, що точність прогнозу знижується при збільшенні горизонту прогнозування. Цю проблему зазвичай можливо вирішити за рахунок зменшення часу на впровадження рішення, яке прийняте на основі отриманого прогнозу, що, в свою чергу, дозволить зменшити горизонт прогнозування і відповідно зменшити помилку прогнозової моделі.

Забезпечення інформації про прогнозованої змінної визначається системою обробки даних, тобто операційним режимом. Якщо котирування курсу валют повідомляється щодня, то для погодинного прогнозу протягом доби цих даних недостатньо і інтервал прогнозування – 1 доба буде більш обґрунтованим.

При побудуванні прогнозу треба розуміти відмінність між даними за період і точковими даними. Для прикладу котирування валют середня ціна валюти за добу буде характеризувати період часу, а ціна «закриття валюти» буде характеризувати точкові дані.

Ціна «закриття валюти» і є значенням змінної у конкретний період часу. Різниця між цими двома типами даних важлива в основному для вибору використовуваної системи збору даних, процесу вимірювань і визначення помилки прогнозування.

Третім аспектом прогнозування є необхідна форма прогнозу (рис. 1.4). Зазвичай при прогнозуванні проводиться оцінка очікуваного значення змінної, плюс оцінка варіації помилки прогнозування або проміжку, на якому зберігається ймовірність змісту реальних майбутніх значень змінної. Цей проміжок називається передбачуваним інтервалом [17].

При побудові прогнозних моделей іноді виникає потреба визначенні моменту зміни напрямку руху тренду прогнозу на протилежне, а не самого значення змінної. Врахування цього аспекту дуже важливо на фінансових ринках, так як можливість прогнозування тенденції зміни курсу всього ринку може забезпечити використання страхових важелів для повернення ринку до своєї рівноваги.

Точність прогнозу, який будується під конкретний проект, впливає на всю систему управління проектом. Критерій досягнення оптимальності при роботі в умовах невизначеності і визначає характеристику системи управління прийняття ефективного рішення.

На систему прийняття рішення впливають фактори, які не тільки пов'язані з процесом прийняття рішення, а і такі, що відповідають за процес генерації самої змінної. При формуванні прогнозу нехтувати цим процесом в системі прийняття рішення не є можливим.

В залежності від процесів які відбуваються при прогнозуванні, система, що прогнозує буде різною. Так, прогнозування системи стабільного процесу для якого не важливо критерій зміни в часі буде відрізнятися від прогнозу системи нестійкого процесу з частими фундаментальними змінами [17].

Для системи стабільного процесу прогноз формується на основі використання історичних даних, майбутні спостереження за процесом можуть слугувати критерієм перевірки прогнозу. При формуванні історичних даних для побудови прогнозних процедур необхідно досліджувати і перевіряти на достовірність інформації та показовість цих даних. Для системи нестійкого процесу з частими фундаментальними змінами формування прогнозу відбувається на основі прогнозування змін в самому процесі.

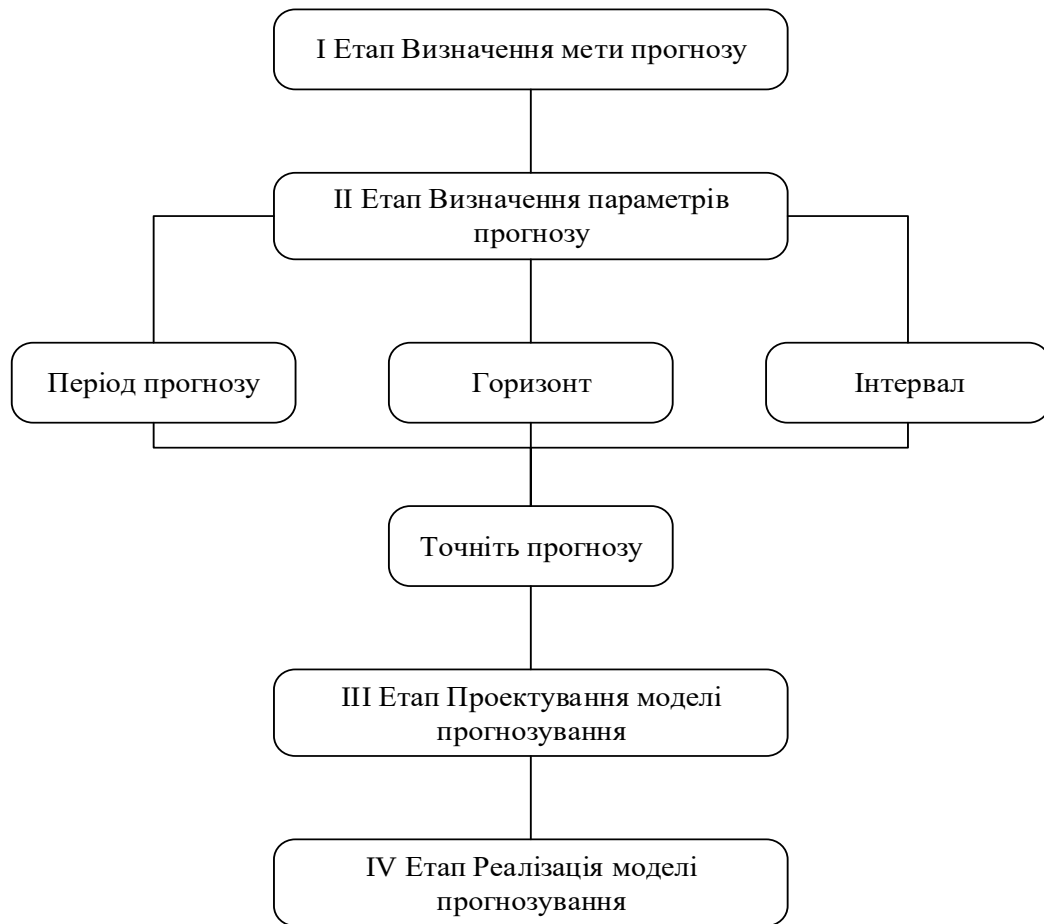


Рисунок 1.4 – Схема проектування моделі прогнозування

Також на систему прогнозування може впливати фактор доступності даних.

Необхідно також враховувати обчислювальні обмеження систем прогнозування. Якщо для вирішення проблеми достатньо прогнозування невеликої кількості змінних – ці передбачення робляться нечасто, система може використовувати більш глибокі методи процедур аналізу. Там, де часто необхідно передбачити велику кількість змінних, особливу увагу слід приділити розробці ефективного управління даними.

Окремо треба сказати що на проблему прогнозування впливає людський фактор. Стосовно проблеми прогнозування людський фактор поділяється на наступні два чинника. По-перше, це можливості людей, які роблять цей прогноз, по -друге, це інтерес людей, які будуть використовувати цій прогноз.

В ідеалі історична інформація аналізується автоматично, і аналітик має можливість на основі отриманого прогнозу робити в ньому якісь модифікації, якщо їх вимагає система. На цьому етапі бажано мати можливість долучати стороннього експерта для оцінки отриманого прогнозу і виявлення можливості і доцільності його модифікації [18]. Потім прогноз надсилається менеджеру, який на його основі приймає остаточне рішення.

Так, наприклад, на фінансовому ринку менеджером буде брокер який приймає рішення щодо купівлі-продажу, на основі отриманого прогнозу, фінансового активу.

1.4.3 Методи прогнозування фінансових ринків

Виділяють три групи методів прогнозування фінансових ринків (див. рис. 1.5).

До першої групи відносяться економіко-математичні методи.

Друга група методів базується на аналізі фундаментальних факторів, основних на фінансово-економічній показниках макро-, і мікроекономіки, фінансового ринку, валютного ринку, окремих компаній. Застосування цих методів для прогнозу та аналізу фінансового ринку в даній кваліфікаційній роботі не розглядається.

Третя група –технічний аналіз до якого відноситься хвильовий аналіз. Технічний аналіз – це поширений підхід у дослідженні фінансового ринку, який спрямований на прогнозування зміни ціни фінансового активу у майбутньому і припущення, що ринок має пам'ять і, отже, майбутні тенденції (тренди) руху курсу фінансового активу будуть залежати від закономірності зміни тренду фінансового активу у минулому.

Об'єктом дослідження в технічному аналізі є графіки, що відображають поведінку індикаторів фінансового ринку. Технічний аналіз використовується для обробки одноденної оперативно-змінюваної інформації. Звичайно, аналітики використовують цей метод частіше, ніж будь-який інший для аналізу оперативної фінансової інформації.

Більшість розроблених методів технічного аналізу використовується в різних програмних продуктах. Усі програмні пакети, засновані на цих методах, відрізняються не тільки за використовуваними методами аналізу, а й за інтерфейсами, функціональністю (онлайн або наприкінці дня), набором основних методів оцінки, фінансовою інформацією тощо.

Побудова графіків таких індикаторів і їх аналіз спільно з графіками руху валютного курсу дає трейдеру багато корисної інформації для прийняття рішень [19].

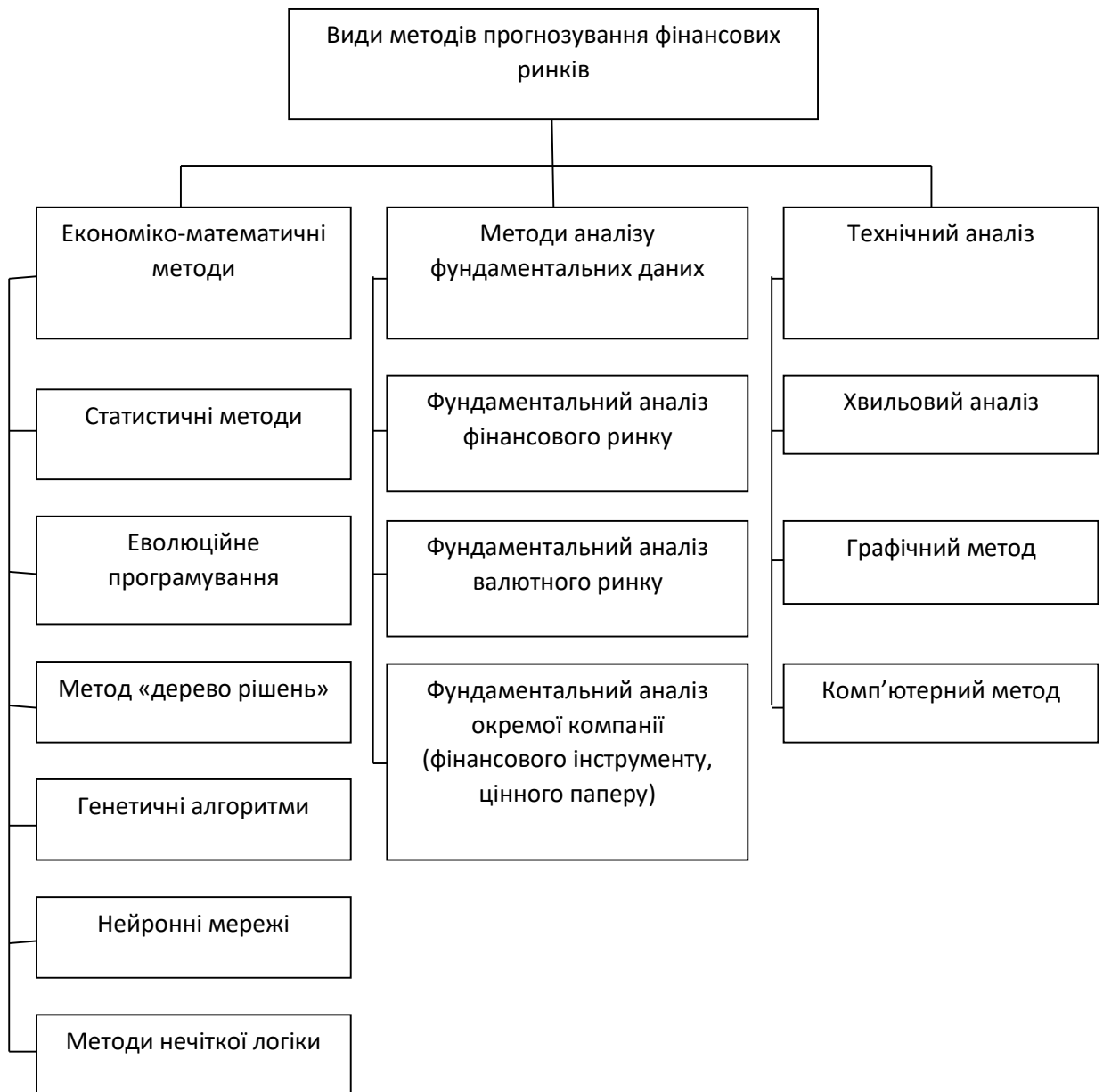


Рисунок 1.5 – Групування методів прогнозування фінансових ринків за видами

Розглянемо більш детально економіко-математичні методи.

До статистичних методів відносяться регресійний, кореляційний, методи статистичних випробувань (метод Монте-Карло), методи аналізу компоненту рядів динаміки, класичне оцінювання невідомих компонентів, використання ARMA-моделей для прогнозування і аналізу показників фінансового ринку та інші [12]. Але застосування цих методів у прогнозуванні індикаторів фінансового ринку вимагає від користувача прогнозного аналізу досвіду в області застосування методів статистики, як для вибору методу аналізу так і для інтерпретації отриманих результатів прогнозу. Це є істотним недоліком при використанні отриманого прогнозу.

Одним із сучасних економіко-математичних методів є еволюційне програмування, як перспективний динамічний напрямок аналізу даних. Ідея методу полягає в тому, щоб записати попередні гіпотези на внутрішній мові програмування. Потім система знаходить програму, яка найбільш точно виражає необхідну залежність, починає самостійно її адаптувати під задану задачу, а потім вибирає найуспішнішу з числа модифікованих програм. Незважаючи на всі можливості методу, оперативне прогнозування не є його силою, програмна реалізація еволюційного програмування ще досі недосконала.

Метод «Дерева рішень» досить умовно можна віднести до систем прогнозу швидкозмінних індикаторів фінансового ринку, використання цього методу для фінансових ринків підходить швидше для системи класифікацій індикаторів. Однак для оперативного аналізу фінансових потоків він малопридатний.

Генетичні алгоритми – цей метод досить успішно використовується для вирішення комбінаторних завдань, а також завдань пошуку оптимальних варіантів. Коротко схему методу можна описати як вибір кращих рішень по раніше формалізованим критеріям, при цьому процес оптимізації нагадує природну еволюцію. Але у методу є ряд недоліків, наприклад складність формалізації критеріїв відбору. Крім того, в цілому методика оптимізована на

клас задач, що трохи відрізняється від оперативного прогнозу швидкозмінливих фінансових показників.

Сьогодні все більше аналітиків фінансового ринку використовують у своїй роботі штучні нейронні мережі. За принципом дії штучної нейронної мережі вирішується задача оцінки якості впливу постійно мінливих зовнішніх умов і окремих досліджуваних параметрів на фінансовий ринок. Перед використанням, нейронну мережу необхідно навчити на прикладі, тобто коли відомі входні параметри і відомий результат, мережа змушена буде реагувати якомога ближче до правильної відповіді.

Одним із методів вирішення проблем прогнозування є метод нечіткої логіки. Методи нечіткої логіки необхідні в ситуаціях, коли досліджувана система не має чітких правил функціонування і тому її поведінку неможливо прогнозувати стандартними методами. Використання методу нечіткої логіки дозволяє прогнозувати глобальні зміни на фінансових ринках.

1.4.4 Використання систем зі штучною «пам'яттю» для вирішення завдань прогнозування

Однією із гіпотез експертів аналізу та прогнозуванню на фінансовому ринку є допущення того, що історія підйому та краху фінансових бірж повторюється. Тож аналітики виходять з того, що можна прогнозувати індикатори фінансового ринку ґрунтуючись на історичні дані розвитку фінансових активів на біржі. Прогнозування індикаторів фінансового ринку може здійснюватися як відомими фінансово-економічними методами, так і методами до яких відносяться мережеві нейронні технології.

На фінансових ринках відбувається безліч подій, кожен з яких вже проаналізували у минулому. Брокери роблять свої прогнози виходячи з інформаційних даних як вів себе фінансовий актив на протязі досліджуваного періоду. Таким чином всі учасники фінансового ринку формують свою пам'ять на основі існуючих тенденцій.

Брокери пам'ятають курси фінансових активів і все це впливає на їх рішення щодо інвестування фінансових ресурсів у той, чи інший актив, тобто пам'ять про те як себе вів фінансовий актив у минулому впливає на рішення в сьогоденні. Ціни на фінансові ресурси представляються зазвичай в виді графіків і ґрунуються на тенденції руху активів на цих графіках, формується попит та пропозиція на фінансовий актив.

Аналіз графіків дорівнює аналізу поведінки учасників ринку. Графічні аналітики фінансового ринку вивчають поведінку ринку, щоб визначити поточні ціни фінансового активу. Відповідно до аксіоми «повторення історії ринку», їхня мета – знайти можливість отримання прибутку від вкладення сьогоденні фінансових ресурсів у актив, повторюючи ці моделі.

Якщо допустити припущення, що аксіома «повторення історії ринку» вірна для будь-якої теорії, Якщо припустити, що історія на ринку повторюється, отже, має сенс розробити систему, яка може «пригадати» минулі ринкові ситуації та їх відповідні наслідки (тобто їх продовження), щоб пізніше порівняти їх із поточними ринковими ситуаціями [19].

Здатність «запам'ятати» нерозривно пов'язана з системами, що реалізують принципи обробки даних через нейронні мережі. Загально прийнято, що система прогнозування штучної нейронної мережі (ШНМ) вивчає великі обсяги інформації, в якій система може розпізнавати залежності, які не можуть бути ідентифіковані іншими методами обробки інформації.

Створення бази даних, в якій ринкові ситуації будуть певним чином допоможе при складанні прогнозу, давало б можливість вирішити завдання прогнозу, але для складання прогнозу усі записи, кількість яких повинна бути достатньої, для можливості проводити прогнозування, повинні бути перевірені на досягнення необхідної точності прогнозу. Така ідея створення бази даних зумовлена обмеженістю доступу до даних, складністю критеріїв порівняння інформації та інше не є можливою, через складність її реалізації.

1.5 Висновки за розділом

У першому розділі проведено дослідження щодо сутності використання нейронних мереж для прогнозування фінансового ринку. Проведено аналіз можливих видів нейронних мереж а також їх оптимальності при прогнозуванні фінансових показників, виявлено що для цієї задачі долучно використовувати мережі типу RNN а саме типу LSTM, через те що у даній моделі присутня функція забування що дозволяє моделювати більш точні прогнози.

Розглянуто структуру та технічні відмінності від стандартної рекурентної нейронної мережі а також порівняно складність пробігу та розповсюдженню помилки.

Проведено аналіз моделей довгострокового прогнозу фінансового ринку та інструменти що можуть застосовуватися для побудування показників фінансового ринку за якими буде формуватися модель. Розглянуті основні поняття та визначення прогнозування на фінансовому ринку. Сформовано схема проектування моделі прогнозування. Визначено фактори та основні параметри прогнозування фінансового ринку.

Розглянуто та проведено аналіз методів прогнозування фінансових ринків, а саме базові принципи технічного, фундаментального та економіко-математичного аналізу.

Обґрунтовано використання систем зі штучною пам'яттю у вирішенні завдань прогнозування індикаторів фінансового ринку.

РОЗДІЛ 2

ПРОЕКТУВАННЯ ШТУЧНОЇ НЕЙРОННОЇ МЕРЕЖІ

2.1 Особливості застосування штучних нейронних мереж в різних областях

Потенційними областями застосування штучних нейронних мереж є ті, де людський інтелект малоефективний, а традиційні обчислення трудомісткі або фізично неадекватні (тобто не відображають або погано відображають реальні фізичні процеси і об'єкти). Дійсно, актуальність застосування нейронних мереж багаторазово зростає тоді, коли з'являється необхідність вирішення погано формалізованих задач.

Типові завдання, які вирішуються за допомогою нейронних мереж і нейрокомп'ютерів наведені на рис. 2.1. Перелік типових завдань на рис. 2.1 не є вичерпаним та постійно розширюється та удосконалюється.



Рисунок 2.1 – Типові завдання нейронної мережі

Розглянемо особливості застосування нейронних мереж, які показують їх переваги порівняно з іншими існуючими методами при виборі моделі прогнозу.

По-перше це ефективність у вирішенні неформальних або погано формалізованих проблем. З відомих переваг методів на основі нейронних мереж слід виділити найпривабливішу – відсутність необхідності суворої математичної специфікації моделі, що особливо цінно при прогнозуванні погано формалізованих процесів. Відомо, що більшість фінансових, ділових та інших подібних завдань погано формалізовані.

По-друге – стійкість до частих змін навколишнього середовища. Переваги нейронних мереж стають очевидними, коли часто змінюються «правила гри»: середовище, в якому існує прогнозований процес, а також характер впливу факторів, що впливають. Тому нейронні мережі найкраще підходять для вирішення таких завдань, як прогнозування тенденцій фондового ринку, що характеризуються впливом цілого набору факторів, що постійно змінюються.

Третє – ефективність при роботі з великою кількістю суперечливої інформації. Нейронні мережі будуть краще там, де аналізується багато даних, в яких приховано закономірності. У цьому випадку також автоматично враховуються різні нелінійні взаємодії між факторами, що впливають. Це особливо важливо під час попереднього аналізу або вибору вихідних даних, виявлення «відсутніх фактів» або грубих помилок у прийнятті рішень.

Четверте – результативність при роботі з неповною інформацією. Доцільно використання нейронних мереж в задачах з неповною або «зачумленою» інформацією, а також в задачах, для яких характерні інтуїтивні рішення.

Уже сьогодні штучні нейронні мережі використовуються в багатьох областях. Останнім часом робляться активні спроби об'єднання штучних нейронних мереж і експертних систем. У такій системі штучна нейронна мережа може реагувати на більшість щодо простих випадків, а всі інші

передаються для розгляду експертною системою. В результаті складні випадки приймаються на більш високому рівні, при цьому, можливо, зі збором додаткових даних або навіть із залученням експертів.

Області застосування нейронних мереж досить різноманітні – це розпізнавання тексту й мови, семантичний пошук, експертні системи і системи підтримки прийняття рішень, передбачення курсів акцій, системи безпеки, аналіз текстів. Здібність нейронних мереж до прогнозування і передбачення подій є однією із можливостей прогнозування. Адже сама можливість здійснювати прогнози на основі використання нейронних мереж є сучасним інструментом у застосуванні інформаційних технологій (як сучасної управлінської технології) у процесі прийняття управлінських рішень. Ефективність управлінського рішення оцінюється на основі подій, що відбуваються після його прийняття. Передбачення некерованих аспектів таких подій до прийняття рішення дозволяє зробити найкращий вибір, який не був би таким успішним без прогнозу.

Недостатність інформації, щодо можливої кількості та якості вихідних даних, зміна середовища під впливом суб'єктивних факторів все це впливає на побудову прогнозової моделі. Прогноз завжди робиться з певною похибкою, яка залежить від використовуваної моделі прогнозу та цілісності вихідних даних. Чим більше вхідних джерел інформації, які використовуються у моделі прогнозу, тим менше будуть втрати від невизначеності у прийнятті рішень. З невеликою або нижчою вибіркою навчання навіть найкращий алгоритм не дасть достатніх результатів, оскільки нейронна мережа не може навчатися без повного набору даних [21].

Втрати, пов'язані з прогнозом, такі, що додаткові витрати не зменшать певною мірою втрати. Це все пояснюється тим, що у будь-якої моделі прогнозування, яка б вона не була «ідеальною», неможливо зменшити похибку прогнозу нижче певного рівня. Таким чином, для побудови ефективної моделі прогнозу, дуже важливе визначення помилки прогнозу. Оцінка прогнозу на

основі критерію достовірності помилки прогнозу може значно знизити ризик при прийнятті рішення.

Розглянемо кілька особливо яскравих і цікавих прикладів використання нейронних мереж в різних областях (рис. 2.2).

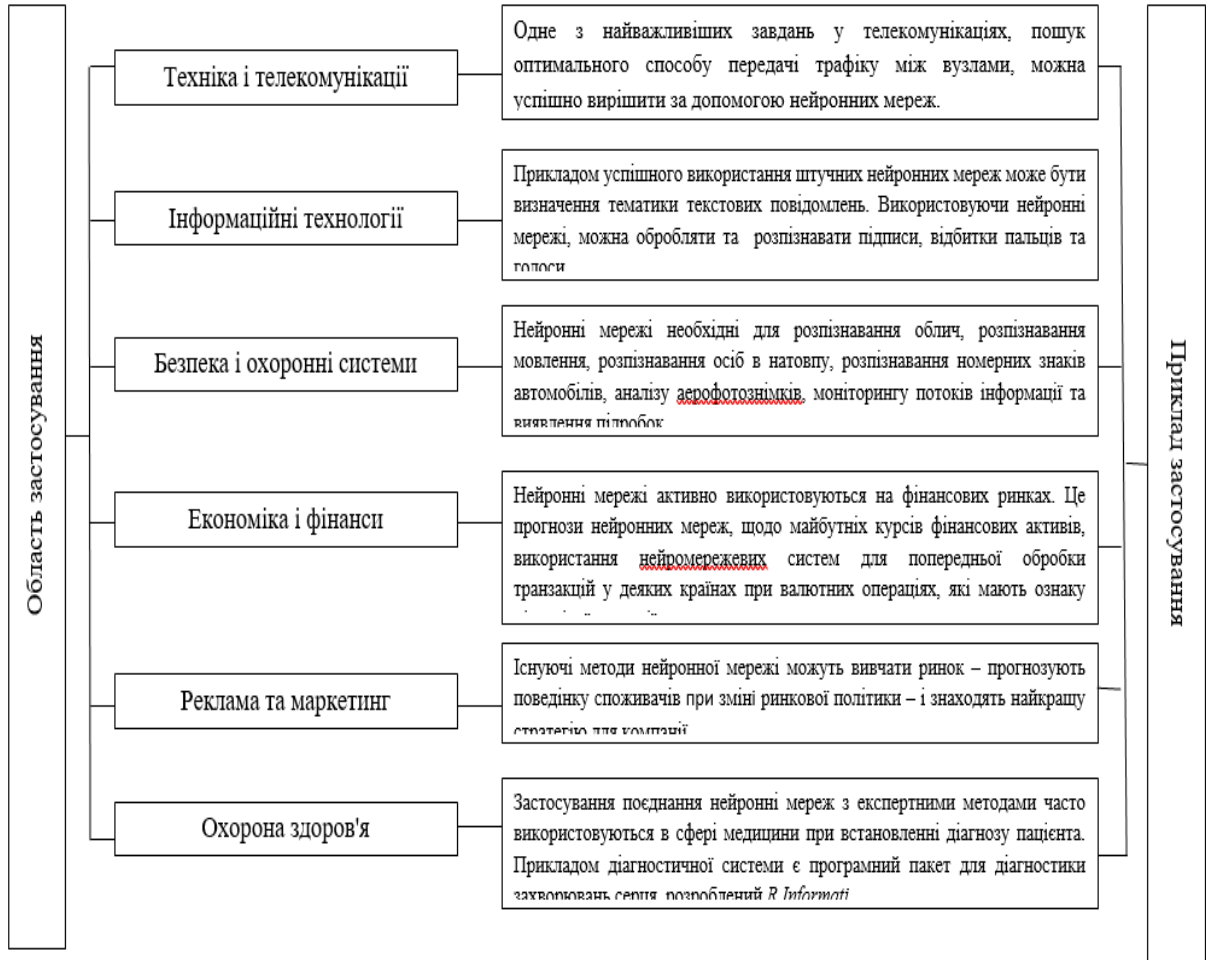


Рисунок 2.2 – Використання нейронних мереж в різних областях науки та техніки

Наведені приклади у рис. 2.2 демонструють, що технології нейронних мереж можна використовувати майже в усіх сферах техніки, науки та суспільства. Повсюдне проникнення нейронних технологій в інші сфери життя є лише питанням часу. Впровадження нових наукових технологій, безумовно, складний процес, але практика показує, що інвестиції не тільки приносять результат, а й приносять відчутну користь користувачам.

2.2 Прогнозування біржової діяльності

Сфера економіки, менеджменту, бізнесу та фінансів є перспективною сферою щодо застосування методу нейромережевого моделювання при побудові прогнозних моделей. Особливого розвитку застосування методу нейромережевого моделювання отримало саме в сфері прогнозування індикаторів фінансового ринку. Так, зараз роблять короткострокові та довгострокові прогнозні тенденції на фінансовому ринку саме з використанням методів нейромережевого моделювання. Ці моделі використовують на таких фінансових ринках як:

- фондовий ринок;
- ф'ючерсний та форвардний ринок;
- ринок облігацій;
- ринок похідних фінансових інструментів;
- валютний ринок FOREX та інше.

За допомогою штучних нейронних мереж робляться прогнози платоспроможного попиту; прогнози ризиків кредитування як на національному міжбанківському ринку так і на міжнародному, сюди ж можна віднести прогнози платоспроможного попиту. Підприємства та корпорації розробляють свої стратегії розвитку базуючись на прогнозах продажів та виручки. Всі ці прогнози (ефективні) дають результат у вигляді додаткового прибутку, який в решті решт також повертається на фінансові ринки.

Для всіх видів фінансових ринків одним із головних завдань є прогнозування короткострокових і довгострокових тенденції розвитку. Завдання автоматизованої системи прогнозування, з використанням штучної нейронної мережі, тенденцій на фінансових ринках, полягає в тому, щоб проаналізувати низку факторів, які впливають на наступний висновок щодо короткострокового чи довгострокового поведінкового прогнозу очікуваної вартості.

Очікувану вартість можливо оцінити за наступними прогнозними показниками рентабельності та ціні. Так автоматизована система може

наприклад, формувати прогноз тенденцій за показниками середньозваженої ціни фінансового активу, або за показником ціни закриття та відкриття фінансового активу тощо. Крім того, можна прогнозувати як показники, визначені для всієї групи інструментів або певного ринку в цілому, так і показники, визначені лише для одного інструменту фінансового ринку.

Рентабельність можна визначити як для набору інструментів, так і для кожного окремо; цінові показники визначаються для кожного конкретного інструменту. Цілями прогнозу (прогнозової вартості) у сфері фінансових ринків можуть бути, наприклад, середньозважена дохідність безкупонних облігацій (за групою інструментів), середньозважена ціна акцій, курс долара, курс щодо гривні та інші.

Як вихідні дані (фактори впливу) для такого прогнозу можуть бути використані різноманітні макро- та мікроекономічні показники, інформація з торгових майданчиків, дані інформаційно-торговельних агентств, експертні оцінки експертів. Кількість факторів, що впливають на прогноз, залежить від ринкового інструменту, що розглядається, та конкретної ринкової ситуації (часу). Це означає, що одні фактори впливають на всі фінансові ринки, інші – лише на певні.

Крім того, вплив факторів на ринки з часом може змінюватися (змінюються ринкові тенденції). Оскільки в певний час існує чіткий взаємозв'язок між фінансовими ринками та ринковими інструментами, рекомендується використовувати один ринок або його інструменти для використання інформації про тенденції на інших ринках як джерела даних. Наприклад, при прогнозуванні ціни відкриття «сьогодні» для будь-якого інструменту на всіх фінансових ринках, цей прогноз сильно залежить від ціни закриття «вчора», і обидва ці значення можуть працювати, як передбачалося.

Процедура виконання короткострокового прогнозу відрізняється від процедури довгострокового на першому і четвертому етапах. У разі короткострокового прогнозу вважається, що всі беруть участь в ньому значущі впливають фактори на прогнозовану дату відомі і зберігаються в базі даних.

Горизонт короткострокового прогнозу не перевищує 3-4 дня. У разі довгострокового прогнозу вважається, що значущі впливають фактори на прогнозовану дату невідомі і повинні бути вказані очікувані значення і похибки. Відповідно похибка визначення прогнозованої величини істотно збільшується (чим далі горизонт прогнозу, тим більше похибка визначення факторів, що впливають і ймовірність помилки аналітика). Горизонт довгострокового прогнозу, як правило, перевищує 3-4 дня.

Прогнозування короткострокових і довгострокових тенденцій фондового ринку передбачає наступні кроки (див. рис. 2.3).

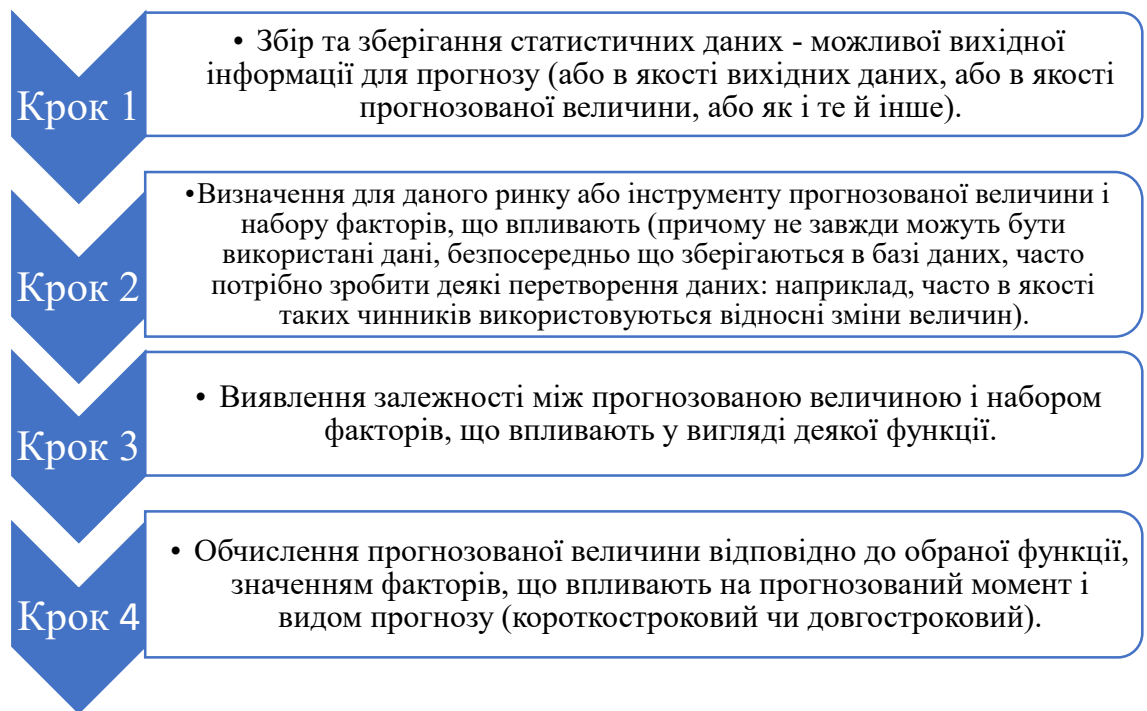


Рисунок 2.3. – Кроки прогнозування тенденцій на фондовому ринку

Як вже вказувалось раніше, прогнозування на основі використання методу нейронних мереж на фондовому ринку це перспективний напрямок розвитку в сфері фінансів. Стандартний підхід до побудови системи прогнозування без використання нейронних мереж має ряд недоліків. Основний з яких є неможливість вирішення проблеми побудови актуальної системи прогнозування при зміні умов торгівлі на фондовому ринку через деякий період часу, а сам прогноз будувався при припущенні, що умови

будуть незмінними, тобто гуртувався на «.. жорстко фіксованому наборі «правил гри»...» [21]. Ще один із недоліків використання стандартного підходу при прогнозуванні ситуації на фінансовому ринку є вимога миттєвого прийняття рішення учасником торгів на біржі, а сам прогноз не може бути побудованим для кожної миттєвої ситуації.

Розглянемо, деякі особливості дій на фондовому ринку. Трейдер, який приймає рішення про купівлю-продаж акцій, має доступ до одного або кількох електронних джерел інформації (Reuters, Dow Jones Telerate, Bloomberg, Tenfore). Він спостерігає поточні значення цін фінансового активу і аналізує графіки індексів, що його цікавлять на світових фондових біржах, основні крос-курси валют і інші показники валютного, фондового і кредитного ринків в багатовіконному середовищі з різним ступенем деталізації. При цьому його рішення базується на використанні статистичної оцінки активу (математичними та графічними методами). На ухвалення його рішення про покупку чи продаж акцій впливають макроекономічні, соціально-політичні події, які з'являються в текстовому полі монітора кожні 5-10 хвилин і супроводжуються різними чутками та коментарями експертів. Також трейдер має додаткову інформацію, наприклад, звіти Національного банку України, інші важливі джерела щодо ключових ринкових показників [22].

При цьому, при ухваленні рішенні про купівлю-продаж фінансового активу, обов'язково приймаються до уваги психологічний аспект дій конкуруючих трейдерів, які можуть вплинути на тенденції подій на фондовому ринку. Наприклад, о 16.00 багато українських трейдерів внутрішньо готові до змін тенденції поведінки індексу Доу-Джонса на Нью-Йоркській фондовій біржі, яка відкривається о 16.30 за Київським часом.

Також операції на фондовому ринку мають такі суттєві особливості:

- ринкові процеси дуже різноманітні в часі;

Наприклад, стан і тенденції на осінньому фондовому ринку значно відрізняється від стану літнього фондового ринку того ж року. Тому при

побудові прогнозу з використанням нейронних мереж не доцільно формувати великі за обсягом навчальні вибірки.

- неоднорідністю даних при формуванні вибірки для навчання нейронної мережі;
- наявністю неінформативних показників з відносно невеликою кількістю статистичних даних.

В цілому, завдання короткострокового прогнозу котирувань акцій нехай і з використанням нейронних мереж представляється досить складною задачею, особливо на стрімко змінюючому українському фондовому ринку.

Так, наприклад, Компанія Alela Corp. розробила для прогнозу зміни біржових індексів Dow Jones, S & P500 і Merval. нейромережеву систему, яку застосовують трейдери на всіх світових фінансових біржах. Особливістю застосування цієї системи є її безкоштовність для користувачів. Питому вага вірних прогнозів сформованих цією системою зіставляє 80%. Для умов фінансового ринку це гарний результат достовірності прогнозів. Трейдери використовують цю безкоштовну систему прогнозу в якості додаткової інформації при прийнятті рішень щодо купівлі-продажу активів.

Японські компанії, що діють на ринку цінних паперів, також широко застосовують нейронні мережі (компанія Mitsubishi). Для входу нейронної мережі використовувалася інформація про ділову активність декількох організацій, отримана за 33 роки, включаючи також оборот, попередню вартість акцій, рівні доходу і інше. [23]. Дана нейронна мережа самонавчальна на реальних прикладах і вона характеризується високою точністю прогнозування, а також швидкодією. Загальна результативність прогнозу в порівнянні з системами, що використовують стандартні статистичні підходи, покращилася на 19,0% [24].

2.3 Особливості використання нейронних мереж на біржі

Для того щоб мати можливість прогнозувати будь-що всі системи повинні мати необхідний мінімум спостережень (понад п'ятдесят і навіть ста).

Однак проблема прогнозування складається з того, що існує безліч завдань для котрих неможливо отримати статистичні данні (їх або зовсім не існує в потребуємому вигляді, або данні мають характер фінансової таємниці).

Крім зазначеного недоліка з отриманням інформаційних даних, для моделей прогнозу на основі нейронних мереж, існує проблема часової тривалості прогнозу для досягнення задовільного результату. Ця проблема не настільки важлива при вивченні невеликої кількості часових послідовностей, але система прогнозування зазвичай включає від кількох сотень до кількох тисяч часових послідовностей.

Нейронні мережі зазвичай можуть навчати і неспеціалісти та використовувати їх для вирішення проблем, але експерти, які володіють знаннями щодо моделювання нейронних мереж, можуть надійно інтерпретувати результати нейронної мережі та численно оцінювати важливість отриманих прогнозів.

Розглянемо також ряд особливостей і труднощів, пов'язаних з використанням програмних продуктів неромережевого моделювання.

На фондовому ринку лише деякі фахівці вдало справляються зі ефективним налаштуванням потужного нейросимулятора. Особливо це стосується тих випадків, коли прогнозування включає кілька малозначущих факторів впливу. І фахівці повинні правильно інтерпретувати результати налаштування нейронної мережі. Для ефективного використання нейронного симулятора також потрібно розуміти суть процесу моделювання нейронної мережі.

При використанні нейронної мережі необхідно враховувати вплив детермінованою періодичної функції, в теорії часових рядів ця функція називається «адитивною сезонною компонентною» і яка визначається методами спектрального аналізу.

Тривалість сезонної складової становить від 7 до 14 днів. Можна вважати, наприклад, що в перші два-три дні кожного місяця внутрішні ціни акцій мають тенденцію до зростання. А в середині місяця будуть дні, коли на

грошовий ринок впливають договірні зобов'язання – купувати та продавати валюти за заздалегідь визначеною ціною тощо. Сезонні компоненти можна автоматично додавати до будь-якого стовпця електронної таблиці з даними на етапі прогнозу, і отже тому сезонні компоненти враховуються в нейронному симуляторі під час оцінки прогнозованого збільшення котирувань акцій.

Практика роботи з нейронними симуляторами на фінансових ринках демонструє та свідчить про те, що створення та підтримка великих, постійно оновлюваних і добре структурованих баз даних фінансових активів, макроекономічних та політичних даних мають велике значення. Тому що все це дуже впливає на прогнозовану ситуацію і якість прогнозу. Ситуація на ринку постійно змінюється і набір ключових факторів впливу (або їх порядок всередині цього набору впливу) безперервно змінюється у часі. У зв'язку з цим нейронні мережі час від часу потребують перенастроювання та перенавчання [25].

Наявність деталізованої документації особливо важливо при роботі з нейронними стимуляторами. Документація зазвичай містить детальний опис методів та прикладів, індексний і предметний покажчик та навчальний курс. Деякі розробники нейронних симуляторів мають телефонні та інтернет-«гарячі лінії», а також проводять семінари для користувачів щодо того, як ефективно працювати з нейронними симуляторами.

2.4 Параметри нейронної мережі

У кожній нейронній мережі присутні декілька шарів, за найменшим повинні бути присутні два шари для вхідних та вихідних даних відповідно. У мережі типу LSTM мінімальна кількість шарів складає чотири, два на вже зазначені вхідні та вихідні данні та ще два прихованих один з яких буде виконувати роль шару пам'яті.

Кожний нейрон у мережі має функцію активації, є можливість налаштувати мережу так щоб на кожному шару нейрони мали різну функцію активації, іноді це може бути корисним залежно від виду мережі. У випадку з

мережею типу LSTM зазвичай використовують стандартні типи активації тангенсом та сигмоїдна, але у випадку роботи із показниками фінансового ринку доречніше використовувати тангенсну функцію активації [26].

Вибір функції гіперболічний тангенс активації зумовлено як вже зазначено типом даних, що будуть проходити через нейронну мережу. Оскільки ціни на акції компанії зазвичай не змінюються на великі показники відносно процентного відношення від їх минулого значення за період часу торгів (зазвичай день), то проблема зникаючого градієнта є не суттєвою [27].

2.5 Модель LSTM

Основна одиниця в прихованому шарі мережі LSTM це блок пам'яті, який містить одну або більше комірку пам'яті і пару адаптивних, мультиплікативних одиниць ливникових яких ворота входу і вихід для всіх осередків в блоці. Кожна комірка пам'яті має у своїй основі періодично самозв'язану лінійну одиницю під назвою Карусель постійної помилки (КПП), активацію якої ми називаємо станом комірки. КПП вирішують проблему зникаючої помилки: за відсутності нових вхідних або помилкових сигналів до комірки, локальна помилка зворотного втручання КПП залишається постійною, ні зростаючим, ні занепадаючим. КПП захищена як від вперед внаслідок активації, так і від помилки внаслідок затримки за допомогою вхідних і вихідних воріт відповідно. Коли ворота закриті (активація навколо нуля), це не має значення входи і шуми не проникають в клітку, і стан клітин не обурюється решти мережі.

На рисунку 2.4 показано блок пам'яті з однієї клітини. Стан клітин, підшкірно, оновлюється на основі його поточного стану і трьох джерел input: netc вводиться в саму комірку, тоді як netin і netout є входами для вхідних і вихідних гейтів.

Прогноз складатиметься з семи значень - по одному на кожен день майбутнього тижня.

Як правило, при виконанні інтервального прогнозування кожен прогнозований тимчасовий крок оцінюється окремо. Це корисно з кількох причин:

- щоб відзначити рівень кожного тимчасового кроку (наприклад, день «+1» проти дня «+3»);
- для порівняння якості моделей з різним тимчасовим кроком (наприклад, моделі з високою оцінкою прогнозу на «день +1» і моделі з високою оцінкою прогнозу на «день +5»).

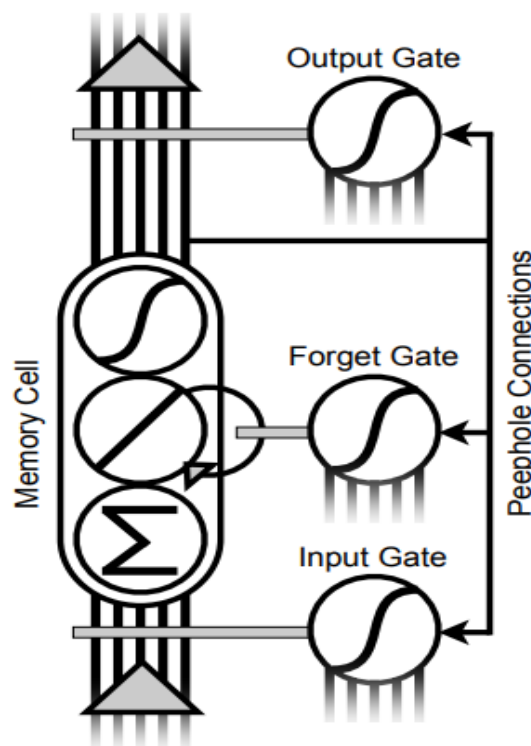


Рисунок 2.4 – Схема блоку пам'яті у нейроні LSTM

Було б корисним мати метрику похибки в тих же одиницях (в тому ж масштабі). При цьому і квадратний корінь з середньоквадратичної помилки (root mean square error – RMSE), і середня абсолютна помилка (mean absolute error – MAE) відповідають цьому масштабу, проте в керівництві буде використовуватися саме квадратний корінь з середньоквадратичної помилки, оскільки ця метрика використовується частіше. До того ж, на відміну від

середньої абсолютної помилки, вона більше штрафує модель за помилки прогнозу. Таким чином, метрикою для розглянутої проблеми в цьому посібнику буде RMSE, що розраховується для кожного прогнозного тимчасового кроку (з 1-го по 7-мій день).

Ця ж метрика буде використовуватися в якості результуючої оцінки за всіма прогнозованими днями, що враховує підсумки роботи моделі. Це допоможе у виборі кращої моделі [28].

2.6 Інструменти для імплементації нейронної мережі

Keras це бібліотека для мови програмування python яка спеціалізована для роботи з нейронними мережами високого рівня та глибинного навчання. Базується поверх tensorflow. Keras має функції які полегшують написання нейронної мережі такі як вказування кількості шарів мережі, встановлення функції активації, регулювання функцій цілі та оптимізації. Також Keras має підтримку для рекурентних та згорткових нейронних мереж. Також Keras має можливість постачати данні для навчання мережі що не знаходяться чи не можуть знаходитися у пам'яті машини за допомогою спеціальної функції, це може бути корисно при розвиненні ідеї використання нейронних мереж для прогнозування показників фінансового ринка шляхом подачі інформації напряду з хостінгу де розміщені дані про ціну акцій за певний день чи час.

Python це мова програмування високого рівня що відрізняється високою модулярністю що дозволяє модифікувати його для вирішення певних задач. Мова Python працює майже на усіх платформах. Також мова підтримує динамічну типізацію даних.

Python є мультипарадигмальною мовою програмування повністю підтримуюче об'єктно орієнтоване програмування, узагальнене, функціональне та метапрограмування.

Велика можливість модифікацій укупі з дизайном що сприяє розробці програм та додатків спрямованих на самонавчання та модифікацію робить мову Python однією з найпопулярніших мов програмування для написання

нейронних мереж. У наявності декілька спеціалізованих бібліотек що спрямовані на полегшення роботи при проектуванні та навчанні нейронних мереж. Можливо привести приклад декількох бібліотек спрямованих на роботу написання нейронної мережі: Theano, TensorFlow, Keras, Lasagne, Neon, Caffe.

Spyder це кроссплатформна середовище програмування для роботи з мовою програмування Python, має інтегровані бібліотеки пов'язані насамперед з науковою діяльністю та має можливість розширення за допомогою встановлення сторонніх плагінів. Сама середовище поширюється через дистрибутив Python Anaconda на більшість платформ Windows, MacOS та велику долю дистрибутивів Linux. Має такі функції як робота з синтаксисом програми, інтроспекцію програмного коду, довершення коду, підтримку декількох консолей одночасно, можливість модифікувати та шукати змінні за допомогою графічного інтерфейсу, налагоджувач на основі IPdb для покрокового виконання та статичний аналіз коду програми за допомогою Pylint.

2.7 Висновки за розділом

У даному розділі проведено аналіз методів та можливостей імплементації нейронних мереж у різних видах діяльності а саме біржовій, виявлено відмінність цікавих факторів при короткостроковому прогнозі та його довгостроковому аналогу, а саме можливість зміни факторів що впливають на довгостроковий прогноз після його кінцевого моделювання, це зумовлює внесення заздалегідь більш великого фактору помилки при побудові моделі на довгий період часу.

Обрано специфічну модель нейронної мережі типу LSTM, за її можливість коректувати себе при випадку зникаючого градієнту за допомогою блоку пам'яті. Також мережі такого типу спеціалізуються на обробці часових рядів з невідомим періодом часу між ключовими подіями, що робить їх придатними при роботі із біржовою активністю.

Обумовлено вибір мови програмування Python для реалізації та навчанні нейронної мережі через її технічні характеристики що сприяють роботі при написанні нейронних мереж а також наявності великої кількості бібліотек для роботи з нейронними мережами. Також обрано бібліотеку Keras для мови програмування Python через її відкритий доступ, відносну легкість роботи та наявності спеціалізованих функції для створення та праці із нейронними мережами типу LSTM. Обрано середу розробки Spyder через відкритий доступ та інтегровані бібліотеки що пов'язані із науковою діяльністю.

Сформовано параметри нейронної мережі а саме кількість шарів, кількість виходів, кількість нейронів та функція їх активації.

РОЗДІЛ 3

НАВЧАННЯ НЕЙРОННОЇ МЕРЕЖІ

3.1 Опис вхідних та вихідних даних

В якості вхідних даних застосовано цінові показники курсу акцій компанії Apple за період у чотири роки. Попередньо дані розміщені на хостінгу Yahoo finances (рис. 3.1).

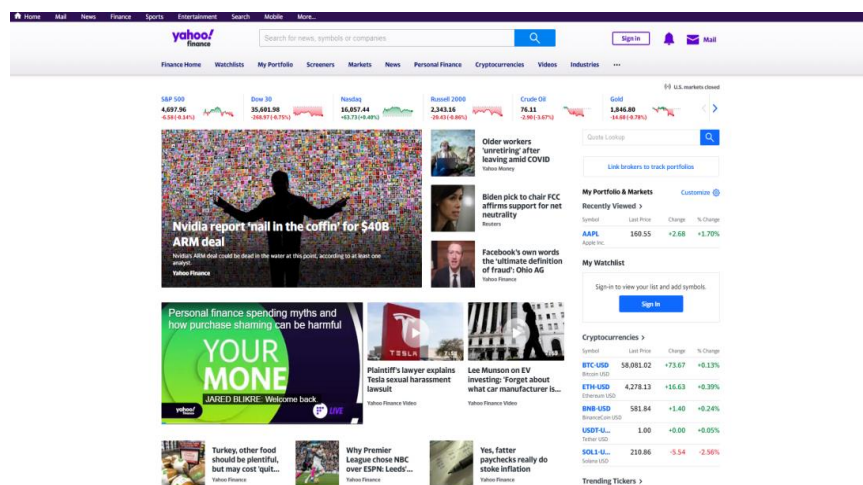


Рисунок 3.1 – Вид хостінгу Yahoo finances

Попередньо на екрані сайту відображається графічна інтерпретація вхідних даних (рис. 3.2).

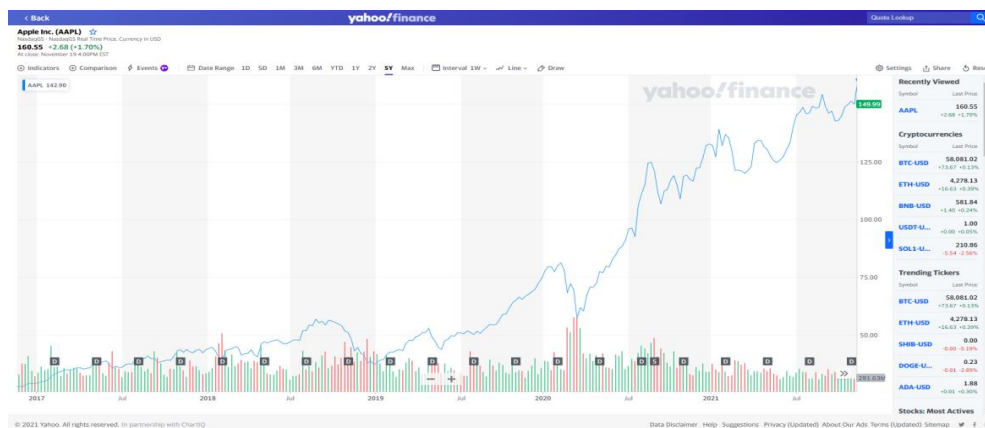


Рисунок 3.2 – Ціна на акції компанії Apple

При завантаженні отримуємо файл формату csv з відповідними чисельними значеннями цінових показників курсу акцій (рис. 3.3), які застосовуються у подальшому як вхідні дані для навчання нейронної мережі та побудови прогнозу. Загальний обсяг становить близько двох гігабайтів даних.

Apple Inc. (AAPL)
NasdaqGS - NasdaqGS Real Time Price. Currency in USD

160.55 +2.68 (+1.70%) **160.99** +0.44 (+0.27%)
At close: November 19 4:00PM EST After hours: 07:59PM EST

Summary Chart Conversations Statistics **Historical Data** Profile Financials Analysis Options Holders Sustainability

Time Period: Nov 20, 2020 - Nov 20, 2021 Show: Historical Prices Frequency: Daily Apply

Currency in USD [Download](#)

Date	Open	High	Low	Close*	Adj Close**	Volume
Nov 19, 2021	157.65	161.02	156.53	160.55	160.55	117,147,500
Nov 18, 2021	153.71	158.67	153.05	157.87	157.87	137,827,700
Nov 17, 2021	151.00	155.00	150.99	153.49	153.49	88,807,000
Nov 16, 2021	149.94	151.49	149.34	151.00	151.00	59,256,200
Nov 15, 2021	150.37	151.88	149.43	150.00	150.00	59,222,800
Nov 12, 2021	148.43	150.40	147.48	149.99	149.99	63,632,600
Nov 11, 2021	148.96	149.43	147.68	147.87	147.87	41,000,000
Nov 10, 2021	150.02	150.13	147.85	147.92	147.92	65,187,100
Nov 09, 2021	150.20	151.43	150.06	150.81	150.81	56,787,900
Nov 08, 2021	151.41	151.57	150.16	150.44	150.44	55,020,900
Nov 05, 2021	151.89	152.20	150.06	151.28	151.28	65,414,600
Nov 05, 2021	0.22 Dividend					
Nov 04, 2021	151.52	152.13	150.54	150.96	150.74	60,394,600

[load/AAPL?period1=1605896465&period2=1637432465&interval=1d&events=history&includeAdjustedClose=true](#)

Рисунок 3.3 – Вигляд вхідних даних перед завантаженням

Увесь об'єм вхідних даних буде поділений на три частини. Перша частина буде використовуватися для навчання мережі. Друга є валідаційною частиною. Третя застосована при тестуванні.

Результатом роботи нейронної мережі прогноз побудований на основі вхідних даних

3.2 Загальна структура програмної моделі

На рисунку 3.4 наведено загальний алгоритм роботи програми.

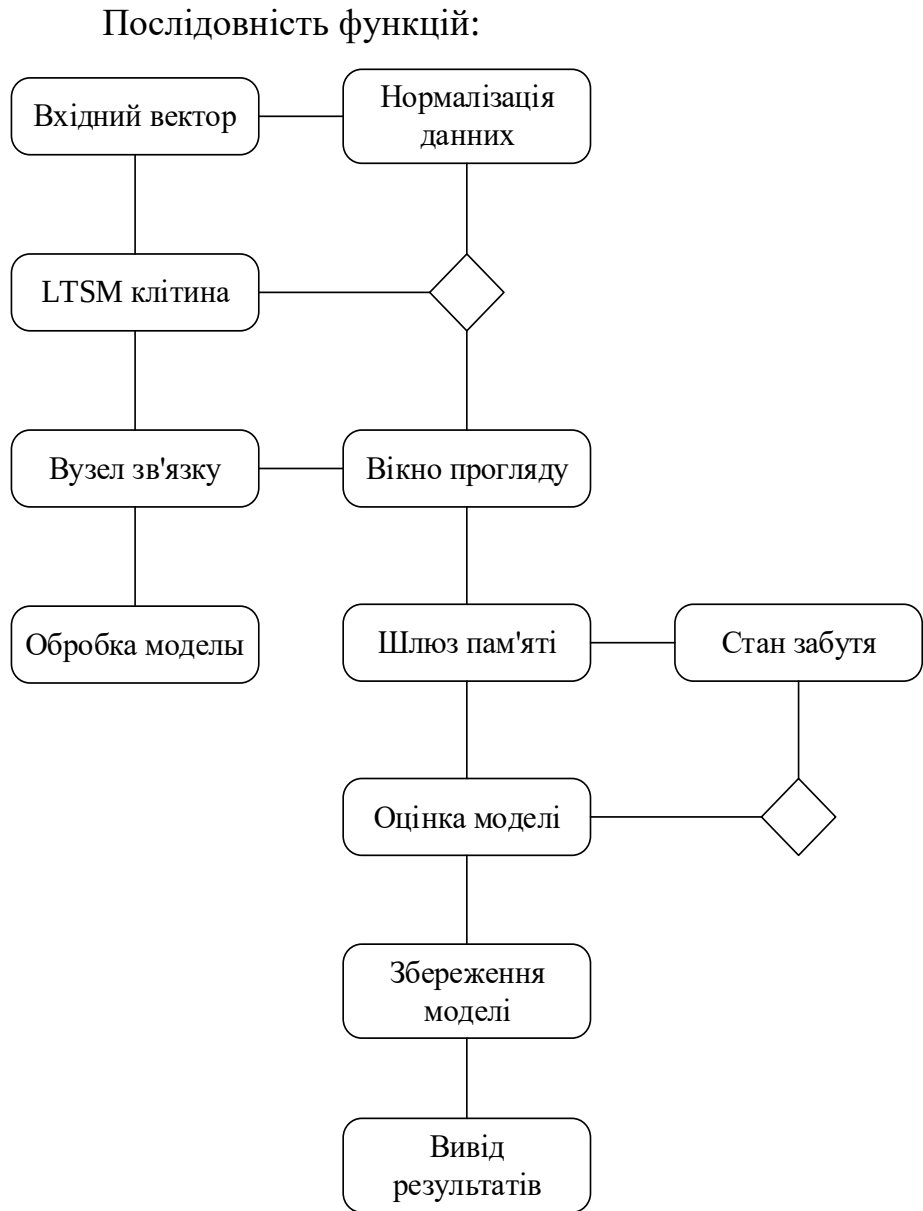


Рисунок 3.4 – Опис програмної реалізації

Функція `evaluate_forecasts ()`, представлена нижче на рис. 3.5, реалізує вищепописане рішення.

Моделі будуть оцінюватися з використанням схеми відомої як «валідацією з наростаючим розміром блоку» (Walk-Forward Validation).

З урахуванням того, що метою моделювання на тиждень вперед, то подаються на вхід моделі фактичні щотижневі дані можна використовувати в якості основи для прогнозування майбутніх щотижневих значень. Даний підхід знаходить широке застосування на практиці.

```

def evaluate_forecasts(actual, predicted):
    scores = list()
    # calculate an RMSE score for each day
    for i in range(actual.shape[1]):
        # calculate mse
        mse = mean_squared_error(actual[:, i], predicted[:, i])
        # calculate rmse
        rmse = sqrt(mse)
        # store
        scores.append(rmse)
    # calculate overall RMSE
    s = 0
    for row in range(actual.shape[0]):
        for col in range(actual.shape[1]):
            s += (actual[row, col] - predicted[row, col])**2
    score = sqrt(s / (actual.shape[0] * actual.shape[1]))
    # plot forecasts vs observations
    for j in range(predicted.shape[1]):
        show_plot(actual[:, j], predicted[:, j], j + 1)
    return score, scores

```

Рисунок 3.5 – Блок коду evaluate_forecasts

Аргументами функції є навчальна і валідаційну частини набору даних в форматі щотижневих даних, а також додатковий аргумент `n_input`, що визначає кількість попередніх спостережень, які модель буде використовувати в якості вхідних даних параметр `n_input` найкраще представити як історію. Кожного тимчасового інтервалу для прогнозування.

Поряд з цим в даній функції викликаються дві допоміжні функції: `build_model()`, яка використовується для конструювання і навчання моделі на навчальних даних, і `forecast()`, що застосовується для виконання прогнозів під час вступу нових щотижневих даних.

У зв'язку з цим переважне використання моделей полягає в тому, щоб сконструювати і навчити їх один раз на історичних даних, і далі

використовувати для прогнозування кожного кроку валідації з наростаючим розміром блоку. Моделі є статичними, тобто не оновлюються під час їх оцінювання.

У цьому полягає відмінність від інших моделей, які швидше навчаються, коли модель може бути переоснащена або оновлена на кожному кроці валідації з наростаючим розміром блоку під час надходження нових даних (рис. 3.6).

```
def evaluate_model(train, test, n_input):
    ...# fit model
    ...model = build_model(train, n_input)
    ...# history is a list of weekly data
    ...history = [x for x in train]
    ...# walk-forward validation over each week
    ...predictions = list()
    ...for i in range(len(test)):
        ...# predict the week
        ...yhat_sequence = forecast(model, history, n_input)
        ...# store the predictions
        ...predictions.append(yhat_sequence)
        ...# get real observation and add to history for predicting the next week
        ...history.append(test[i, :])
    ...# evaluate predictions days for each week
    ...predictions = array(predictions)
    ...score, scores = evaluate_forecasts(test[:, :, 0], predictions)
    ...return score, scores
```

Рисунок 3.6 – Блок коду валідації

Функція `summarize_scores()` відображає ефективність моделі в вигляді строки зі значеннями (рис. 3.7).

Далі визначається конфігурація моделі і проводиться її навчання.

По суті, дана проблема інтервального прогнозування часових рядів відноситься до авторегресії.

```
def summarize_scores(name, score, scores):
    s_scores = ', '.join(['%.1f' % s for s in scores])
    print('%s: [%.3f] %s' % (name, score, s_scores))
```

Рисунок 3.7 – Блок коду summarize_scores

Це означає, що хороший результат може бути отриманий в тому випадку, коли такі сім днів є деякою функцією спостережень попередніх. Дана обставина поряд з відносно невеликим обсягом даних обумовлює застосування «невеликої», простий по конфігурації, моделі.

З урахуванням цього ми сконструюємо модель з декількома шарами. У ролі першого прихованого шару виступить шар LSTM з 200 нейронами (кількість нейронів в прихованому шарі не пов'язане з тривалістю тимчасового інтервалу у вхідних послідовностях). За шаром LSTM слід повнозв'язний шар з 200 нейронами, призначення якого інтерпретувати виявлені шаром LSTM ознаки. Нарешті, вихідний шар буде безпосередньо виводити вектор з сімома прогностичними елементами - по одному на кожен день у вихідний послідовності.

За функцію втрат прийнята середньоквадратична помилка (MSE), оскільки вона відповідає масштабу прийнятої раніше метриці (корінь з середньоквадратичної помилки, RMSE). В якості оптимізатора прийнятий алгоритм Adam (варіант стохастичного градієнтного спуску), навчання моделі триватиме 70 епох, розмір пакета складається з 16-ти зразків.

Невеликий розмір пакета і стохастична природа алгоритму означають, що одна і та ж модель буде по-різному відображати вхідні і вихідні дані при повторному навчанні. Це призведе до того, що після виконання оцінки моделей, їх результати можуть відрізнятися один від одного. Ви можете спробувати виконати навчання моделі кілька разів і в підрахунках підсумкових результатів використовувати середнє значення.

Представлена нижче функція build_model () (рис. 3.8) готує навчальну частину даних, визначає конфігурацію моделі, виконує її навчання і повертає

навчену модель, яку потім можна використовувати для виконання прогнозів.(описати кількість циклів та епох).

```
def build_model(train, n_input):  
    ...# prepare data  
    ...train_x, train_y = to_supervised(train, n_input)  
    ...# define parameters  
    ...verbose, epochs, batch_size = 0, 70, 16  
    ...n_timesteps, n_features, n_outputs = train_x.shape[1], train_x.shape[2],  
train_y.shape[1]  
    ...# define model  
    ...model = Sequential()  
    ...model.add(LSTM(200, activation='relu', input_shape=(n_timesteps,  
n_features)))  
    ...model.add(Dense(100, activation='relu'))  
    ...model.add(Dense(n_outputs))  
    ...model.compile(loss='mse', optimizer='adam')  
    ...# fit network  
    ...model.fit(train_x, train_y, epochs=epochs, batch_size=batch_size,  
verbose=verbose)  
    ...return model
```

Рисунок 3.8 – Блок коду побудови моделі

Мережі довгострокової пам'яті, які зазвичай називають просто «LSTM» – це особливий тип RNN, здатний вивчати довгострокові залежності. Вони були представлені, і були вдосконалені та популяризовані багатьма людьми. Вони надзвичайно добре працюють над широким спектром проблем і зараз широко використовуються.

LSTM явно розроблені, щоб уникнути довгострокової проблеми залежності. Запам'ятовування інформації протягом тривалого періоду часу – це практично їхня поведінка за замовчуванням.

Всі рекурентні нейронні мережі мають вигляд ланцюга повторюваних модулів нейронної мережі. У стандартних RNN цей повторюваний модуль матиме дуже просту структуру, таку як один шар з активацією тангенс.

LSTM також мають таку ланцюжкову структуру, але повторюваний модуль має іншу структуру. Замість того, щоб мати один шар нейронної мережі, є чотири, які взаємодіють дуже особливим чином.

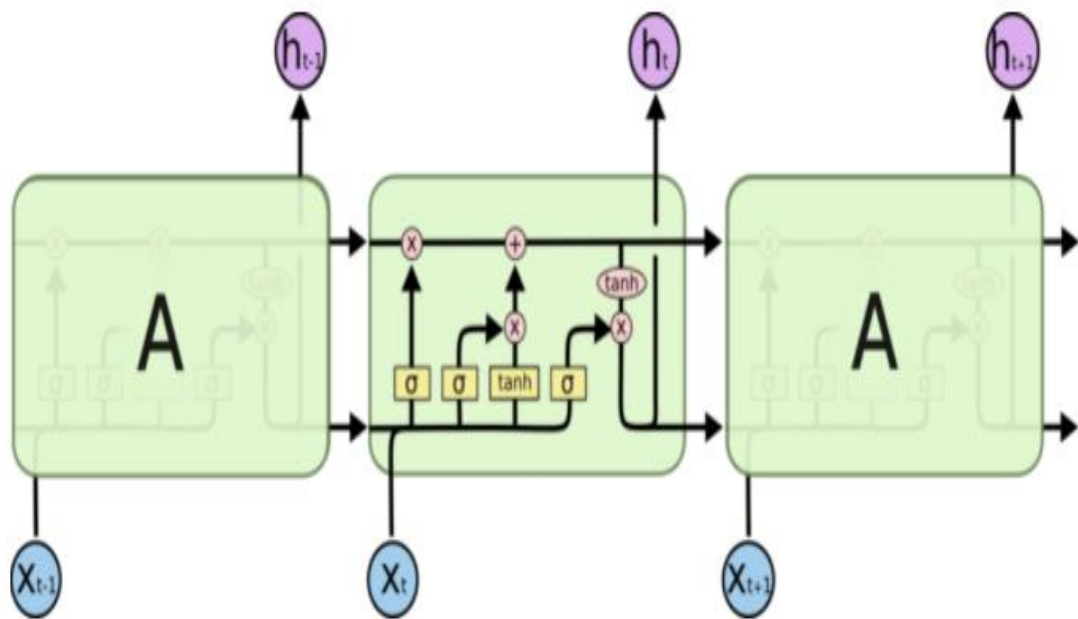


Рисунок 3.9 – Повторювана модель LSTM

У наведеній вище діаграмі кожен рядок несе цілий вектор від виходу одного вузла до входу інших. Рожеві кола представляють точкові операції, такі як додавання векторів, а жовті прямокутники – це вивчені шари нейронної мережі. Злиття рядків позначає конкатенацію, а розгалуження рядків означає, що його вміст копіюється, а копії спрямовуються в різні місця.

Перший крок у нашому LSTM – вирішити, яку інформацію ми збираємось викинути зі стану клітинки. Це рішення приймається сигмовидним шаром, який називається «шар шлюзів забуття». Він розглядає h_{t-1} і x_t і виводить число від 0 до 1 для кожного числа в стані комірки C_{t-1} . 1 означає

«повністю зберегти це», а 0 означає «повністю забути це». Коли ми бачимо новий предмет, ми хочемо забути показники старого предмета.

$$f_t = \sigma(w_f[h_{t-1}, x_t] + b_f) \quad (3.1)$$

Наступний крок – вирішити, яку нову інформацію ми буде зберігати стан клітинки. Це має дві частини. По-перше, сигмовидний шар, який називається «вхідним шаром шлюзу», вирішує, які значення ми будемо оновлювати. Далі шар $\tanh$ створює вектор нових значень-кандидатів, $C_{\sim t}$, які можна додати до стану. На наступному кроці ми об'єднаємо ці два, щоб створити оновлення стану.

Тепер настав час оновити старий стан комірки, C_{t-1} , на новий стан комірки C_t . Попередні кроки вже вирішили, що робити, нам просто потрібно це зробити насправді.

Ми множимо старий стан на f_t , забуваючи речі, які вирішили забути раніше. Потім додаємо його c_t . Це нові значення-кандидати, масштабовані відповідно до того, наскільки ми вирішили оновити значення кожного стану.

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_t) \quad (3.2)$$

$$\tilde{c}_t = \tanh(w_c[h_{t-1}, x_t] + b_c) \quad (3.3)$$

Нарешті, нам потрібно вирішити, що ми збираємося вивести. Цей вихід буде заснований на нашому стані пам'яті, але буде відфільтрованою версією. Спочатку ми запускаємо сигмовидний шар, який вирішує, які частини стану клітинки ми збираємося вивести. Потім ми переносимо стан комірки через $\tanh$ (щоб значення становили між -1 і 1) і множимо його на вихід сигмоподібних шлюзів, щоб ми виводили лише ті частини, які ми вирішили.

$$o_t = \sigma(w_o[h_{t-1}, K_\epsilon] + b_o) \quad (3.4)$$

$$h_t = o_t \tanh(C_t) \quad (3.5)$$

Початкова версія блоку LSTM включає декілька елементів пам'яті, вхідних і вихідних елементів.

Навчання проводилося з використанням суміші реального часу. Повторне навчання і зворотне поширення через час (BPTT) [29,30]. Тільки градієнт осередку був поширений назад у часі, а градієнт для інших повторювань з'єднання був відкинутий.

Остання модифікація до стандартного LSTM була зроблена Грейвсом і Шмідхубером [31]. Представлено це дослідження навчання повного зворотного поширення в часі (BPTT). Мережі LSTM з архітектурою, описаною в Розділі II, та представив результати за бенчмарком TIMIT [32]. Використання повний BPTT мав додаткову перевагу, що градієнти LSTM можна перевірити за допомогою кінцевих різниць, що робить практичною реалізація більш надійною (рис. 3.10).

```
def split_dataset(data):
    # split into standard weeks
    train, test = data[1:-328], data[-328:-6]
    # restructure into windows of weekly data
    train = array(split(train, len(train)/7))
    test = array(split(test, len(test)/7))
    return train, test

def show_plot(true, pred, title):
    fig = pyplot.subplots()
    pyplot.plot(true, label='Y_original')
    pyplot.plot(pred, dashes=[4,3], label='Y_predicted')
    pyplot.xlabel('N samples', fontsize=12)
    pyplot.ylabel('Instance value', fontsize=12)
    pyplot.title(title, fontsize=12)
    pyplot.grid(True)
    pyplot.legend(loc='upper right')
    pyplot.show()
```

Рисунок 3.10 – Блок коду розподілу вхідних даних

З моменту його введення стандартний LSTM був найбільшим часто використовувана архітектура, але були запропоновані і інші варіанти. Перед запровадженням повного навчання BPTT, Gers та ін [33] використовував метод навчання, заснований на Extended Фільтрація Калмана, яка дозволила навчати LSTM деякі патологічні випадки ціною високих обчислень складність.

Шмідхубер та ін. [34] запропоновано використовувати гібрид метод на основі еволюції замість BPTT для навчання але зберіг стандартну архітектуру LSTM. Bayer [29] розробили різні архітектури блоків LSTM які максимізують пристосованість до контекстно-залежних граматики. Більший дослідження такого роду пізніше було проведено Jozefowicz et al. [35]. Сак та ін. [37] представив лінійний проекційний шар, який проектує вихід рівня LSTM вниз перед повторним і прямим зв'язків з метою зменшення кількості параметрів для Мережі LSTM з великою кількістю блоків.

Представлена нижче функція `build_model()` готує навчальну частину даних, визначає конфігурацію моделі, виконує її навчання і повертає навчену модель, яку потім можна використовувати для виконання прогнозів (рис. 3.11).

```
def build_model(train, n_input):
    # prepare data
    train_x, train_y = to_supervised(train, n_input)
    # define parameters
    verbose, epochs, batch_size = 0, 70, 16
    n_timesteps, n_features, n_outputs = train_x.shape[1], train_x.shape[2],
    train_y.shape[1]
    # define model
    model = Sequential()
    model.add(LSTM(200, activation='relu', input_shape=(n_timesteps,
    n_features)))
    model.add(Dense(100, activation='relu'))
    model.add(Dense(n_outputs))
    model.compile(loss='mse', optimizer='adam')
    # fit network
    model.fit(train_x, train_y, epochs=epochs, batch_size=batch_size,
    verbose=verbose)
    return model
```

Рисунок 3.11 – Блок підготовки навчальних даних

Мережі довгострокової пам'яті, які зазвичай називають просто «LSTM» – це особливий тип RNN, здатний вивчати довгострокові залежності. Вони були представлені Hochreiter & Schmidhuber, і були вдосконалені та популяризовані багатьма людьми в наступних роботах.¹ Вони надзвичайно добре працюють над широким спектром проблем і зараз широко використовуються.

LSTM явно розроблені, щоб уникнути довгострокової проблеми залежності. Запам'ятовування інформації протягом тривалого періоду часу – це практично їхня поведінка за замовчуванням (рис. 3.12).

```
# train the model
def build_model(train, n_input):
    # prepare data
    train_x, train_y = to_supervised(train, n_input)

    # define parameters
    verbose, epochs, batch_size = 0, 70, 16
    n_timesteps, n_features, n_outputs = train_x.shape[1], train_x.shape[2],
    train_y.shape[1]

    # define model
    model = Sequential()
    model.add(LSTM(200, activation='relu', input_shape=(n_timesteps,
    n_features)))
    model.add(Dense(100, activation='relu'))
    model.add(Dense(n_outputs))
    model.compile(loss='mse', optimizer='adam')

    # fit network
    model.fit(train_x, train_y, epochs=epochs, batch_size=batch_size,
    verbose=verbose)
    return model
```

Рисунок 3.12 – Блок тренування моделі


```

# make a forecast
def forecast(model, history, n_input):
    # flatten data
    data = array(history)
    data = data.reshape((data.shape[0]*data.shape[1], data.shape[2]))
    # retrieve last observations for input data
    input_x = data[-n_input:, 0]
    # reshape into [1, n_input, 1]
    input_x = input_x.reshape((1, len(input_x), 1))
    # forecast the next week
    yhat = model.predict(input_x, verbose=0)
    # we only want the vector forecast
    yhat = yhat[0]
    return yhat

```

Рисунок 3.12 лист 2 – Блок тренування моделі продовження

Через введення навченого параметру масштабування для нахилу функцій активації шлюзів змогли підвищити продуктивність LSTM в автономному наборі даних розпізнавання рукописного введення. В тому, що вони називають динамічною пам'яттю кори головного мозку покращено швидкість конвергенції LSTM шляхом додавання повторюваних з'єднань між шлюзами одного блоку (але не між блоками).

Лінійна функція активації

$$\phi(v) = a + v'b.$$

Зрізаний лінійний вузол

$$\phi(v) = \max(0, a + v/b).$$

Логістична

$$\phi(v) = (1 + \exp(-a - v'b))^{-1}.$$

Чо та ін. [32] запропонував спрощений варіант LSTM архітектура під назвою Gated Recurrent Unit (GRU). Вони використовували ні з'єднання вічка, ні функції активації виходу, і об'єднав вхід і шляз забуття в шляз оновлення. Нарешті, їхній вихідний шляз (так званий шляз скидання) виконує лише шляз повторювані підключення до входу блоку (W_z). Chung та ін. [33] провели початкове порівняння між GRU та Vanilladef zipped() (рис. 3.13) LSTM, це привело до неоднозначних результатів.

```

...."""Return the path to the zipped pack to upload."""
...name=CONFIG.get("packname") or "Unknown Pack"
...if native_os!=HOST_OS:
.....name+= "-" + HOST_OS
...return dist(name.format(pack_ver(warn=False)) + ".zip")
¶
¶
def base(*paths):
...."""Return the path to the persistent content directory."""
...return os.path.join("base", *paths)
gh.f.bkbnb.jf.d.byr.lkd.vjlbdba.rjv.eh.dppvj
¶
def components(*paths):
...."""Return the path to the downloaded components cache dir."""
...return os.path.join("components", *paths)

```

Рисунок 3.13 – Блок коду Vanilladef

Моделі довгострокової пам'яті є надзвичайно потужними моделями часових рядів. Вони можуть передбачити довільну кількість кроків у

майбутнє. Модуль (або клітинка) LSTM має 5 основних компонентів, які дозволяють моделювати як довгострокові, так і короткострокові дані.

Стан комірки (c_t) – це внутрішня пам'ять клітинки, яка зберігає як короткочасну, так і довгострокову пам'ять (вектор стану клітини).

Прихований стан (h_t) – це вихідна інформація про стан, розрахована за значенням поточний вхід, попередній прихований стан і поточний вхід комірки, які ви в кінцевому підсумку використовуєте для прогнозування майбутніх цін на фондовому ринку. Крім того, прихований стан може вирішити отримати лише короткочасну чи довгострокову або обидва типи пам'яті, збережені в стані комірки, щоб зробити наступне передбачення.

Вхідний шлюз (i_t) – визначає, скільки інформації з поточного входу надходить до стану комірки

Forget gate (f_t) – визначає, скільки інформації з поточного входу та попереднього стану клітинки перетікає в поточний стан комірки.

Вихідний шлюз (o_t) – вирішує, скільки інформації з поточного стану комірки переходить у прихований стан, щоб у разі потреби LSTM міг вибрати лише довгострокові або короткострокові та довгострокові спогади.

3.3 Висновки за розділом

У даному розділі технічно реалізовано програмну частину для створення штучної нейронної мережі. Представлено діаграму роботи програми по створенню та навчанні нейронної мережі.

Представлено де і як були отримані тестові данні. Розглянуто функцію нормалізації та обробки вхідних даних для нейронної мережі.

Створено та описано діяльність функції генерації нейронної мережі. Описано структуру навчання нейронної мережі. Описано процес відбору мереж із точність, яка дозволяє їх використовувати.

РОЗДІЛ 4

РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

4.1 Результати моделі

Після коректної роботи програми ми отримаймо на виході готову нейронну мережу із певними параметрами точності та втрати градієнта (рис. 4.1). Але це не означає що вона буде коректно працювати із іншим набором даних, так звана проблема переобучення. Саме для цього і було потрібно ділити вхідні данні на три частини до однієї з яких, а саме валідаційної, мережа не буде мати доступу у період навчання. Після завершення навчання мережу слід знов протестувати вже на цих даних.

```

Train for 200 steps, validate for 50 steps
Epoch 1/10
200/200 [=====] - 2s 11ms/step - loss: 0.4075 - val_loss:
0.1351
Epoch 2/10
200/200 [=====] - 1s 4ms/step - loss: 0.1118 - val_loss:
0.0360
Epoch 3/10
200/200 [=====] - 1s 4ms/step - loss: 0.0490 - val_loss:
0.0289
Epoch 4/10
200/200 [=====] - 1s 4ms/step - loss: 0.0444 - val_loss:
0.0257
Epoch 5/10
200/200 [=====] - 1s 4ms/step - loss: 0.0299 - val_loss:
0.0235
Epoch 6/10
200/200 [=====] - 1s 4ms/step - loss: 0.0317 - val_loss:
0.0224
Epoch 7/10
200/200 [=====] - 1s 4ms/step - loss: 0.0287 - val_loss:
0.0206
Epoch 8/10
200/200 [=====] - 1s 4ms/step - loss: 0.0263 - val_loss:
0.0200
Epoch 9/10
200/200 [=====] - 1s 4ms/step - loss: 0.0254 - val_loss:
0.0182
Epoch 10/10
200/200 [=====] - 1s 4ms/step - loss: 0.0228 - val_loss:
0.0174

```

Рисунок 4.1 – Втрата градієнта та збільшення точності при навчанні

Таблиця 4.1 – Точність побудованих мереж

Номер моделі	% точності моделі	% точності моделі після валідації	% вірного передбачення знаку
7	80,1	76,35	80,2
23	79,64	76,28	85,3
36	81,05	77,14	83,23
54	85,39	80,23	82,2

В таблиці 4.1 представлені результати тестування на валідаційних даних. Як можна бачити точність нейронної мережі на нових даних нижче ніж на даних при навчанні. Це зумовлено додатковою втратою градієнта після навчання (рис. 4.2) .

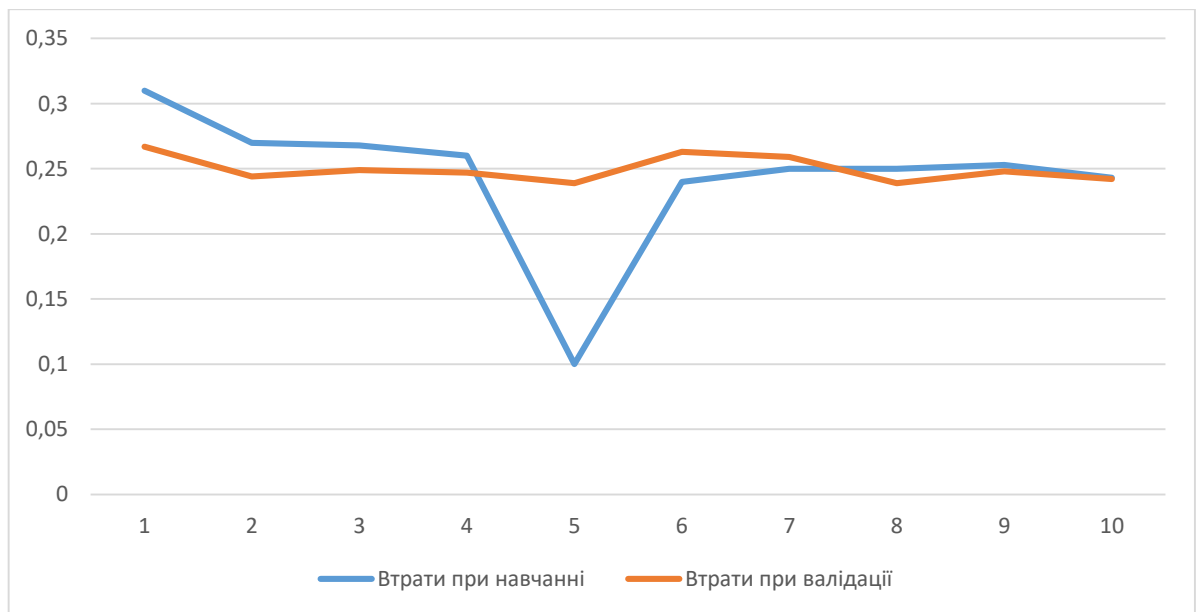


Рисунок 4.2 – Втрати градієнта при навчанні та валідації

Слід зазначити що у нашому випадку є два параметри які ми можемо передбачати, це як і точність самого прогнозу, так і напрямок зміни значення. Якщо брати це до уваги то стає можливо отримати нейронну мережу яку

можна застосовувати для прогнозування навіть при тому що точність цієї моделі знаходиться у 70-75% (рис 4.3).

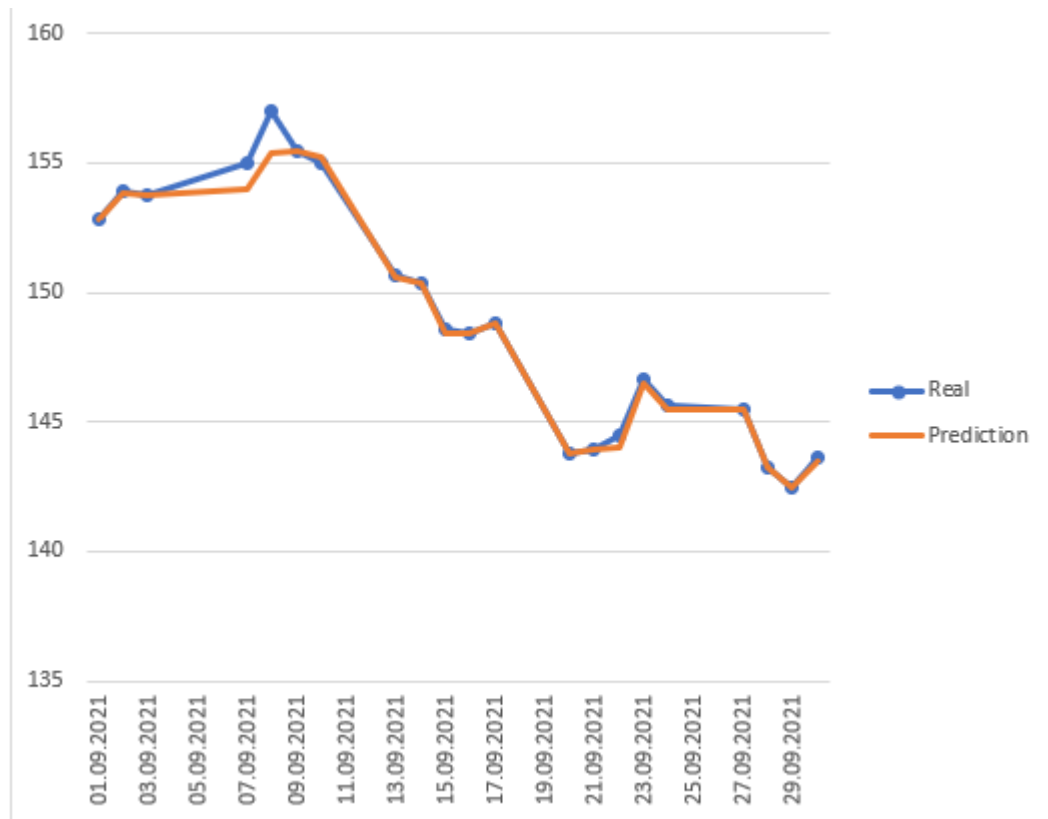


Рисунок 4.3 – Приклад роботи нейронної мережі

На прикладі вище представлено що навіть якщо модель не зможе точно передбачити ціну, то якщо модель точно передбачає зміну тренду вона все одно може бути використана.

Але слід зазначити що проводити підвищення точності та відсіювання валідації все одно потрібно.

4.2 Порівняння працездатності

Для прикладу справності праці нейронної мережі можна порівняти її результат із роботою інших нейронних мереж що виконували схожі функції (таблиця 4.2). Для проведення порівняльного аналізу обрано моделі Medianprice Close [23] Price [24], Momentum [30]. Далі наведено таблицю із назвою нейронної мережі, її тип та головні параметри.

Таблиця 4.2 – Характеристики ШНМ – порівняльна таблиця

Модель	Кількість шарів	Набір даних	Тип мережі	% точності моделі	% вірного передбачення знаку
Medianprice Close Price	5	1173	RNN	80,61	88
Momentum(All TADAWUL prices)	4	1170	CNN	79,5	86
All TADAWUL prices	3	1500	RNN	83,61	88
Mr & CP	3+1	1170	LSTM	83,42	87
MACD (low price)	3	30512	LSTM	83,89	83
Створена мережа	3	1460	LSTM	80,23	82

У порівнянні із нейронною мережею моментум яка має тип мережі CNN (рис. 4.4) ми можемо бачити що навіть у кращому випадку точність прогнозу не перевищила 80%, це першим чином пов'язано із типом цієї мережі, згорткові нейронні мережі погано підходять для прогнозування поведінки ринку. Однак слід звернути увагу на те що мережа, як і в нашому випадку, доволі добре передбачала зміну напрямку тренду.

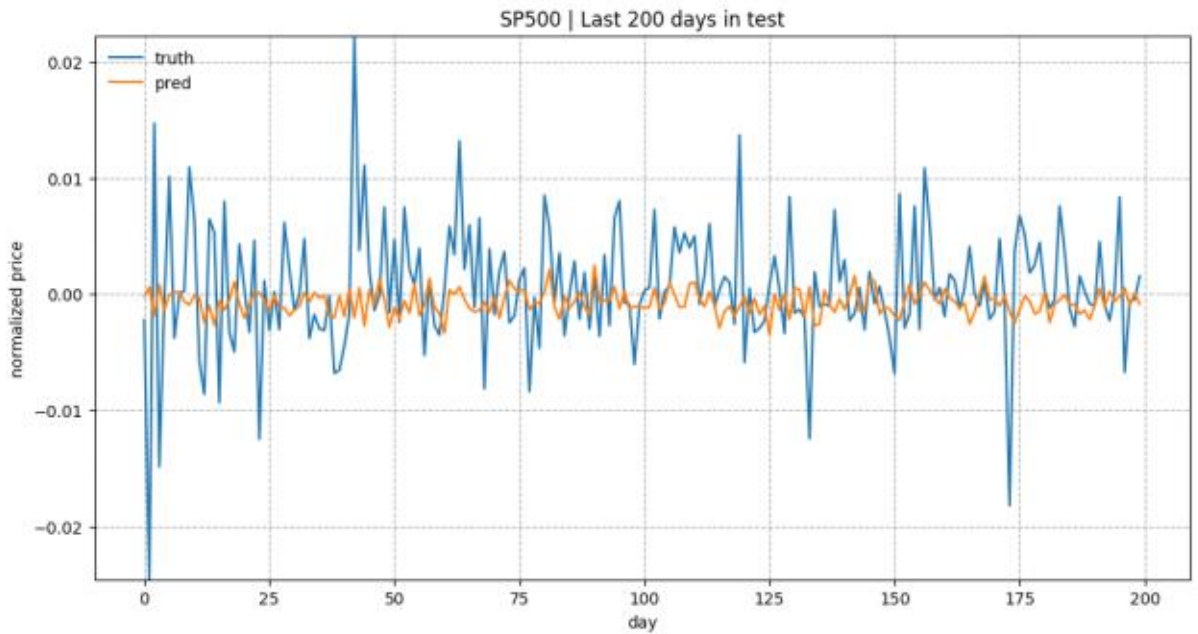


Рисунок 4.4 – Нейронна мережа моментум із типом CNN

Якщо порівнювати із нейронними мережами типу RNN (рис. 4.5) то можна побачити доволі схожі результати як і у плані точності прогнозу так і у можливості передбачити зміну тренду.

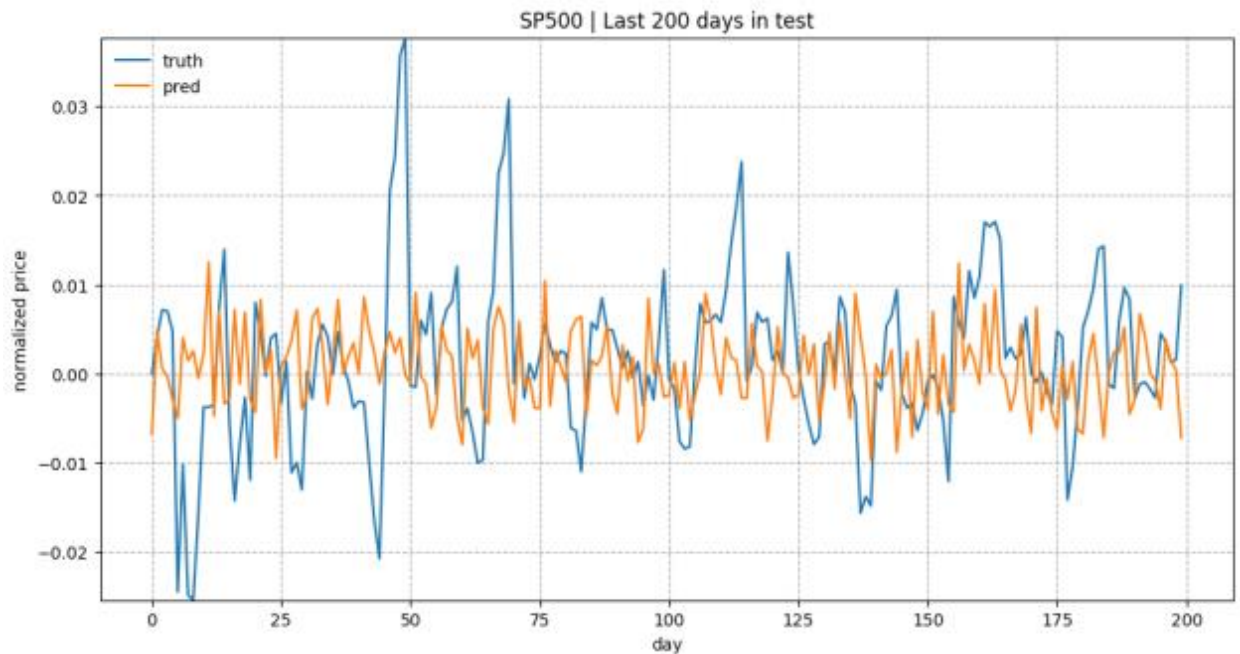


Рисунок 4.5 – Нейронна мережа із типом RNN

Слід зазначити що схожість результатів обумовлено фактом того що нейронні мережі типу LSTM є модифікацією нейронних мереж RNN. Ми можемо бачити схожість мереж у неспроможності передбачити великі стрибки амплітуди на коротких періодах часу.

Хоча по показникам точності наша модель у найкращому випадку наближається до показників звичайних нейронних мереж типу RNN, все одно можна порівняти із нейронною мережею типу LSTM, для прикладу обрано мережу MACD [24] (рис. 4.6).

Як можна бачити присутня схожа проблема неспроможності передбачити великі стрибки по амплітуді, але точність на передбачення невеликих змін та зміни тренду помітно кращі.

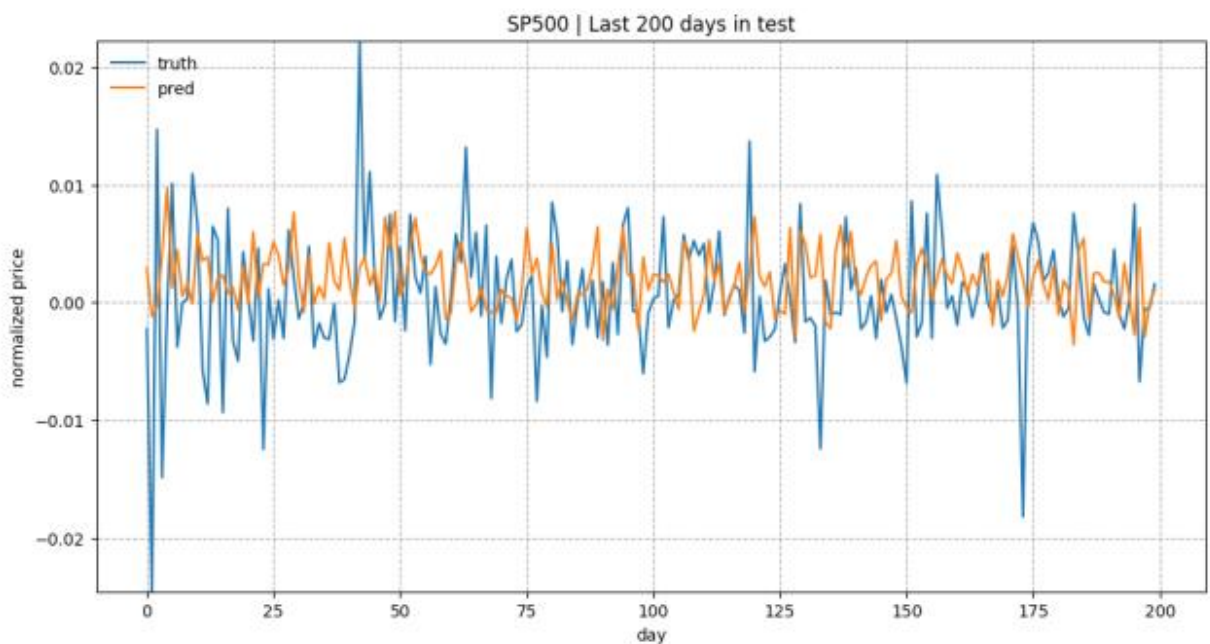


Рисунок 4.6 – Нейронна мережа MACD із типом LSTM

З усіх прикладів можна вивести що усі прогнози зроблені нейронними мережами досить консервативні у плані зміни амплітуди та досить добре передбачають зміну тренду, звідси можна зробити висновок що наша створена нейронна мережа є конкуренто спроможною.

4.4 Висновки за розділом

У цьому розділі приведено приклад роботи навченої нейронної мережі.

Приведено статистику точності найкращих нейронних мереж отриманих при навчанні.

Приведено статистику нейронних мереж, що намагалися вирішити схожу проблему.

ВИСНОВКИ

У роботі проведено дослідження щодо сутності використання нейронних мереж для прогнозування фінансового ринку. Проведено аналіз можливих видів нейронних мереж а також їх оптимальності при прогнозуванні фінансових показників, виявлено що для цієї задачі долучно використовувати мережі типу RNN а саме типу LSTM, через те що у даній моделі присутня функція забування що дозволяє моделювати більш точні прогнози.

Розглянуто структуру та технічні відмінності від стандартної рекурентної нейронної мережі а також порівняно складність пробігу та розповсюдженню помилки.

Проведено аналіз моделей довгострокового прогнозу фінансового ринку та інструменти що можуть застосовуватися для побудування показників фінансового ринку за якими буде формуватися модель. Розглянуті основні поняття та визначення прогнозування на фінансовому ринку. Сформовано схема проектування моделі прогнозування. Визначено фактори та основні параметри прогнозування фінансового ринку.

Розглянуто та проведено аналіз методів прогнозування фінансових ринків, а саме базові принципи технічного, фундаментального та економіко-математичного аналізу.

Обґрунтовано використання систем зі штучною пам'яттю у вирішенні завдань прогнозування індикаторів фінансового ринку.

Проведено аналіз методів та можливостей імплементації нейронних мереж у різних видах діяльності а саме біржовій, виявлено відмінність цікавих факторів при короткостроковому прогнозі та його довгостроковому аналогу, а саме можливість зміни факторів що впливають на довгостроковий прогноз після його кінцевого моделювання, це зумовлює внесення заздалегідь більш великого фактору помилки при побудові моделі на довгий період часу.

Обрано специфічну модель нейронної мережі типу LSTM, за її можливість коректувати себе при випадку зникаючого градієнту за допомогою блоку пам'яті. Також мережі такого типу спеціалізуються на обробці часових рядів з невідомим періодом часу між ключовими подіями, що робить їх придатними при роботі із біржовою активністю.

Обумовлено вибір мови програмування Python для реалізації та навчання нейронної мережі через її технічні характеристики що сприяють роботі при написанні нейронних мереж а також наявності великої кількості бібліотек для роботи з нейронними мережами. Також обрано бібліотеку Keras для мови програмування Python через її відкритий доступ, відносну легкість роботи та наявності спеціалізованих функцій для створення та праці із нейронними мережами типу LSTM. Обрано середу розробки Spyder через відкритий доступ та інтегровані бібліотеки що пов'язані із науковою діяльністю.

Сформовано параметри нейронної мережі а саме кількість шарів, кількість виходів, кількість нейронів та функція їх активації.

Технічно реалізовано програмну частину для створення штучної нейронної мережі. Представлено діаграму роботи програми по створенню та навчанні нейронної мережі.

Представлено де і як були отримані тестові данні. Розглянуто функцію нормалізації та обробки вхідних даних для нейронної мережі.

Створена та описано діяльність функції генерації нейронної мережі. Описано структуру навчання нейронної мережі. Описано процес відбору мереж із точність, яка дозволяє їх використовувати.

Приведено приклад роботи навченої нейронної мережі. Приведено статистику точності найкращих нейронних мереж отриманих при навчанні.

Приведено статистику нейронних мереж що намагалися вирішити схожу проблему.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Neural Networks Tutorial – A Pathway to Deep Learning [Електронний ресурс] // Adventures in Machine Learning. – 2017. – Режим доступу до ресурсу: <http://adventuresinmachinelearning.com/neural-networks-tutorial/>.
2. Нейронні мережі - шлях до глибокого навчання [Електронний ресурс] // Codeguida. – 2017. – Режим доступу до ресурсу: <https://codeguida.com/post/739>.
3. Карванський С. С. Евристичний метод тренування штучної нейронної мережі : дис. магістра : 123 Комп'ютерна інженерія / Карванський С. С. – Київ, 2018. 87с.
4. Пулеко І. В. Основні типи нейронних мереж / І. В. Пулеко // Штучний інтелект в задачах кібербезпеки / І. В. Пулеко. – Житомир: Державний університет "Житомирська політехніка", 2019. – С. 57–72.
5. Дранишников Л.В. Інтелектуальні методи в управлінні: навчальний посібник / Л. В. Дранишников. – Кам'янське: ДДТУ, 2018. – 416 с.
6. Експериментальний аналіз застосування нейронних мереж для керування траєкторією польоту БПЛА / О. Д.Фесенко, Р. О. Беляков, Г. Д. Радзівілов, В. С. Гулій. // Збірник наукових праць ВІТІ. – 2020. – №1. – С. 97–112.
7. Зорін Ю. М. Конспект лекцій з предмету «Комп'ютерні системи штучного інтелекту» [Електронний ресурс] / Ю. М. Зорін – Режим доступу до ресурсу: <http://m.scs.kpi.ua/course/view.php?id=107>
8. Прохоров Д. П. Система прогнозування фінансових ринків із використанням рекурентних нейронних мереж : дис. магістра : 126 Інформаційні системи та технології / Прохоров Д. П. – Київ, 2018. – 125 с.
9. Яхьяева Г. Э. Основы теории нейронных сетей / Г. Э. Яхьяева. – Москва: Национальный открытый университет, 2016. – 200 с

10. Ставицький О. В. Використання нейронних мереж для прогнозування в фінансовій сфері / О. В. Ставицький. // Актуальні проблеми економіки та управління. – 2017. – №11. – Режим доступу до ресурсу: <http://ape.fmm.kpi.ua/article/view/102584>

11. Степанов В. А. Фондовый рынок и нейросети [Електронний ресурс] / В. А. Степанов // Мир ПК – 1998. – Режим доступу до ресурсу: <http://www.osp.ru/pcworld/1998/12/159835>

12. Иванов Д. В. Прогнозирование финансовых рынков с использованием искусственных нейронных сетей [Електронний ресурс] / Д. В. Иванов – Режим доступу до ресурсу: <http://forex-mmci.ru/D.Ivanov>

13. Зайцев О. В. Знайомство з валютним ринком FOREX [Електронний ресурс] / О. В. Зайцев, Т. В. Дворянова // Вісник наукових праць СумДУ. – 2020. – Режим доступу до ресурсу: https://visnyk.fem.sumdu.edu.ua/issues/1_2020/1_2020_20_Zaitsev.pdf

14. Прогнозування [Електронний ресурс] // Wikipedia – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D0%BD%D0%BE%D0%B7%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F#:~:text=%D0%9F%D1%80%D0%BE%D0%B3%D0%BD%D0%BE%D0%B7%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F%20%E2%80%94%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D1%81%20%D0%BF%D0%B5%D1%80%D0%B5%D0%B4%D0%B1%D0%B0%D1%87%D0%B5%D0%BD%D0%BD%D1%8F%20%D0%BC%D0%B0%D0%B9%D0%B1%D1%83%D1%82%D0%BD%D1%8C%D0%BE%D0%B3%D0%BE%20%D1%81%D1%82%D0%B0%D0%BD%D1%83,%D0%BF%D1%80%D0%B5%D0%B4%D0%BC%D0%B5%D1%82%D0%B0%20%D1%87%D0%B8%20%D1%8F%D0%B2%D0%B8%D1%89%D0%B0%20%D0%B2%20%D0%BF%D0%B5%D1%80%D1%81%D0%BF%D0%B5%D0%BA%D1%82%D0%B8%D0%B2%D1%96>

15. Гуляєва Н. М. Системна єдність фінансування і використання фінансових ресурсів в інвестиційному процесі / Н. М. Гуляєва. // Інвестиції: практика та досвід. – 2017. – №7. – С. 32–35.

16. Моць Е. О. Розробка системи прогнозування результатів економічної діяльності підприємства зв'язку: маг. дисерт: Інформаційні управляючі системи та технології / Е.О. Моць. – Покровськ, 2017. – 98 с.

17. Бокс Дж., Дженкинс Г. Анализ временных рядов прогнозирования и управления: пер. с англ. А.Л. Левшина / Дж. Бокс, Г. Дженкинс – М.: Мир, 1974. – 362 с.

18. Сергеев В.А. Использование нейросетей в прогнозировании фондового рынка / В.А. Сергеев [Электронный ресурс]/ Азимут научных исследований: экономика и управление. – 2018. – № 4(25). – С. – 280-282. // Режим доступа до ресурсу: <https://cyberleninka.ru/article/n/ispolzovanie-neyrosetey-v-prognozirovanii-fondovogo-rynka>.

19. Сохацька О. М. Фундаментальний та технічний аналіз цін товарних та фінансових ринків / О. М. Сохацька, І. В. Роговська-Ішук, С. І. Вінницький. – Київ: Кондор, 2012. – 305 с.

20. Коцур Д. Використання нейронних мереж - перспективна сфера науки і суспільства / Д. Коцур. // XVIII Международная научно-практическая интернет-конференция «Проблемы и перспективы развития науки в начале третьего тысячелетия в странах СНГ». – 2013. – Режим доступа до ресурсу: <http://oldconf.neasmo.org.ua/node/139>

21. Конюхова О.В., Лапочкина К.С. Применение нейронных сетей в экономике и актуальность их использования при составлении краткосрочного прогноза бюджета [Электронный ресурс] / О.В. Конюхова, К.С. Лапочкина // Информационные ресурсы, системы и технологии : Сетевое научное издание – Режим доступа до ресурсу : <http://irsit.ru/article150>.

22. Pang, X., Zhou, Y., Wang, P. et al. An innovative neural network approach for stock market prediction// J Supercomput 76, 2098–2118 (2020). <https://doi.org/10.1007/s11227-017-2228-y>

23. C. S. Vui, G. K. Soon, C. K. On, R. Alfred and P. Anthony, "A review of stock market prediction with Artificial neural network (ANN),// IEEE International Conference on Control System, Computing and Engineering, 2013, pp. 477-482, doi: 10.1109/ICCSCE.2013.6720012.

24. Talal Alotaibi, Amril Nazir, Roobaea Alroobaea, Moteb Alotibi, Fasal Alsubeai, Abdullah Alghamdi, Thamer Alsulimani Saudi Arabia Stock Market Prediction Using Neural Network// International Journal on Computer Science and Engineering 9(2):62-70.

25. Солдатов О.П. Применение нейронных сетей для решения задач прогнозирования [Электронный ресурс]/ О.П. Солдатов, В.В.Семенов// Режим доступа до ресурсу:<http://elibrary.lt/resursai/Uzsienio%20leidiniai/MFTI/2006/136.pdf>

26. Овакимян А.С. Нейросетевое прогнозирование временных рядов. [Электронный ресурс]/ А.С. Овакимян, С.Г Саркисян. М.А. Зироян, В.И. Тинякова // Режим доступа до ресурсу: <http://ysu.am/files/Paper4.pdf>

27. Мицель А. А. Прогнозирование динамики цен на фондовом рынке/ А.А. Мицель // Известия Томского политехнического университета Известия ТПУ. – 2006. – Т. – 309, № 8. – С. – 197-201.

28. Сафронов, Ю.Ю. Прогнозирование котировок на финансовых рынках с помощью искусственных нейронных сетей/ Ю.Ю. Сафронов // Банковский вестник. – 2016. – №10. – (639) . – С.61-65.

29. Лысов К. А. Прогнозирование ценовых колебаний и долгосрочных трендов на финансовых рынках. Маг.диссер. Санкт-Петербургский Государственный Университет. – 2018. – 53 с. [Электронный ресурс]/ К.А. Лысов // - Режим доступа до ресурсу: https://dspace.spbu.ru/bitstream/11701/12457/1/lysov_vkr.pdf

30. Selvamuthu, D., Kumar, V. & Mishra, A. Indian stock market prediction using artificial neural networks on tick data// Financ Innov 5, 16 (2019). DOI: 10.1186/s40854-019-0131-7

31. Wohler C, Anlauf JK. An adaptable time-delay neural-network algorithm for image sequence analysis// IEEE Trans Neural Netw. 1999;10(6):1531-6. DOI: 10.1109/72.809100. PMID: 18252656.

32. Галещук С. І. Штучні нейронні мережі у прогнозуванні валютного ринку / С. І. Галещук. // Вісник Київського національного торговельно-економічного університету. – 2016. – №3. – С. 101–114.

33. Карпа Д. М. Нейромережеві засоби прогнозування споживання енергоресурсів / Д. М. Карпа, І. Г. Цмоць, Ю. В. Опотяк. // Науковий вісник НЛТУ України. – 2018. – С. 140–146.

34. Харинович-Яворська Д. О. Застосування нейромережевих технологій для прогнозування конкурентної стратегії торговельних підприємств [Електронний ресурс] / Д. О. Харинович-Яворська. // Міжнародний науковий журнал "Інтернаука". Серія : Економічні науки.. – 2017. – №2. – С. 25–30. – Режим доступу до ресурсу: http://nbuv.gov.ua/UJRN/mnjie_2017_2_6

35. Сєдай А. В. Використання нейронних мереж для моделювання та прогнозування фінансової діяльності транспортного підприємства [Електронний ресурс] / А. В. Сєдай. – 2015. – Режим доступу до ресурсу: <http://publications.ntu.edu.ua/eut/2015-02/115-120.pdf>.

36. Дудар З. В. . Порівняння методів прогнозування часових рядів. [Електронний ресурс] / З. В. Дудар, М. С. Широкопетлева, О. А. Пономаренко. // Бионика интеллекта. – 2018. – С. 41–47. – Режим доступу до ресурсу: <https://journals.indexcopernicus.com/api/file/viewByFileId/645667.pdf>

37. Новікова В. В. Прогнозування внутрішнього валового продукту на основі апарату штучних нейронних мереж. / В. В. Новікова. // Економіка та управління АПК. – 2014. – №1. – С. 5–11.

Додаток А

Зауваження нормоконтролера

Таблиця А.1 – Зауваження нормоконтролера

[illegible]

Додаток Б

Програмний код

```
# -*- coding: utf-8 -*-
"""
Created on Sun Sep 26 11:11:52 2021
@author: ignat
"""

# univariate multi-step lstm
from math import sqrt
from numpy import split
from numpy import array
from pandas import read_csv
from sklearn.metrics import mean_squared_error
from matplotlib import pyplot
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Flatten
from keras.layers import LSTM
# split a univariate dataset into train/test sets
def split_dataset(data):
    # split into standard weeks
    train, test = data[1:-328], data[-328:-6]
    # restructure into windows of weekly data
    train = array(split(train, len(train)/7))
    test = array(split(test, len(test)/7))
    return train, test
def show_plot(true, pred, title):
    fig = pyplot.subplots()
    pyplot.plot(true, label='Y_original')
    pyplot.plot(pred, dashes=[4, 3], label='Y_predicted')
    pyplot.xlabel('N_samples', fontsize=12)
    pyplot.ylabel('Instance_value', fontsize=12)
    pyplot.title(title, fontsize=12)
    pyplot.grid(True)
    pyplot.legend(loc='upper right')
    pyplot.show()
## evaluate one or more weekly forecasts against expected values
def evaluate_forecasts(actual, predicted):
    scores = list()
    # calculate an RMSE score for each day
    for i in range(actual.shape[1]):
```

```

    # calculate mse
    mse = mean_squared_error(actual[:, i], predicted[:, i])
    # calculate rmse
    rmse = sqrt(mse)
    # store
    scores.append(rmse)
# calculate overall RMSE
s = 0
for row in range(actual.shape[0]):
    for col in range(actual.shape[1]):
        s += (actual[row, col] - predicted[row, col])**2
score = sqrt(s / (actual.shape[0] * actual.shape[1]))
# plot forecasts vs observations
for j in range(predicted.shape[1]):
    show_plot(actual[:, j], predicted[:, j], j + 1)
return score, scores
# summarize scores
def summarize_scores(name, score, scores):
    s_scores = ', '.join(['%.1f' % s for s in scores])
    print('%s: [%.3f] %s' % (name, score, s_scores))
# convert history into inputs and outputs
def to_supervised(train, n_input, n_out=7):
    # flatten data
    data = train.reshape((train.shape[0]*train.shape[1], train.shape[2]))
    X, y = list(), list()
    in_start = 0
    # step over the entire history one time step at a time
    for _ in range(len(data)):
        # define the end of the input sequence
        in_end = in_start + n_input
        out_end = in_end + n_out
        # ensure we have enough data for this instance
        if out_end <= len(data):
            x_input = data[in_start:in_end, 0]
            x_input = x_input.reshape((len(x_input), 1))
            X.append(x_input)
            y.append(data[in_end:out_end, 0])
        # move along one time step
        in_start += 1
    return array(X), array(y)
# train the model
def build_model(train, n_input):
    # prepare data
    train_x, train_y = to_supervised(train, n_input)
    # define parameters

```

```

verbose, epochs, batch_size = 0, 70, 16
n_timesteps, n_features, n_outputs = train_x.shape[1], train_x.shape[2],
train_y.shape[1]
# define model
model = Sequential()
model.add(LSTM(200, activation='relu', input_shape=(n_timesteps,
n_features)))
model.add(Dense(100, activation='relu'))
model.add(Dense(n_outputs))
model.compile(loss='mse', optimizer='adam')
# fit network
model.fit(train_x, train_y, epochs=epochs, batch_size=batch_size,
verbose=verbose)
return model
# make a forecast
def forecast(model, history, n_input):
    # flatten data
    data = array(history)
    data = data.reshape((data.shape[0]*data.shape[1], data.shape[2]))
    # retrieve last observations for input data
    input_x = data[-n_input:, 0]
    # reshape into [1, n_input, 1]
    input_x = input_x.reshape((1, len(input_x), 1))
    # forecast the next week
    yhat = model.predict(input_x, verbose=0)
    # we only want the vector forecast
    yhat = yhat[0]
    return yhat
# evaluate a single model
def evaluate_model(train, test, n_input):
    # fit model
    model = build_model(train, n_input)
    # history is a list of weekly data
    history = [x for x in train]
    # walk-forward validation over each week
    predictions = list()
    for i in range(len(test)):
        # predict the week
        yhat_sequence = forecast(model, history, n_input)
        # store the predictions
        predictions.append(yhat_sequence)
        # get real observation and add to history for predicting the next week
        history.append(test[i, :])
    # evaluate predictions days for each week
    predictions = array(predictions)

```

```

    score, scores = evaluate_forecasts(test[:, :, 0], predictions)
    return score, scores
# load the new file djghue iie zpoe ziii oood
dataset = read_csv('price_value.csv', header=0,
                   infer_datetime_format=True, parse_dates=['datetime'],
                   index_col=['datetime'])
# split into train and test
train, test = split_dataset(dataset.values)
# evaluate model and get scores
n_input = 7
score, scores = evaluate_model(train, test, n_input)
# summarize scores
summarize_scores('lstm', score, scores)
# plot scores
days = [ 'mon', 'tue', 'wed', 'thr', 'fri', 'sat', 'sun']
fig = pyplot.subplots()
pyplot.plot(days, scores, marker='o', label='lstm')
pyplot.show()

```

Додаток В
Роздатковий матеріал

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
Кафедра прикладної математики та інформатики

Прогнозування показників фінансового ринку за допомогою нейронних мереж

Мирошніченко Ігнат
КНМ-20

1

Рисунок В.1 – Перший аркуш презентації

План доповіді

1. Актуальність
2. Економічні методи прогнозування
3. Проназування за допомогою нейронної мережі
4. Приклади моделей
5. Недоліки нейронних мереж для прогнозування

2

Рисунок В.2 – Другий аркуш презентації

Актуальність

- Численні спостереження за фінансовими ринками засвідчили невідповідність припущень про раціональність інвесторів та інформаційну ефективність ринків реальним умовам, які існують у ряді випадків. У зв'язку з цим стає актуальним побудова нових моделей прогнозування розвитку фінансових ринків, систематизації та вибору ключових особливостей динаміки реальних ринків капіталу.

3

Рисунок В.3 – Третій аркуш презентації

Економічні методи прогнозування

При побудові економетричної моделі використовуються два типи даних:

- дані, що характеризують сукупність різних об'єктів в певний момент або період часу;
- дані, що характеризують один об'єкт за ряд послідовних моментів або періодів часу.

Моделі, побудовані за даними першого типу, називаються просторовими моделями. Моделі, побудовані на основі другого типу даних, називаються моделями часових рядів

4

Рисунок В.4 – Четвертий аркуш презентації

Проназування за допомогою нейронної мережі

- Нейронні мережі можуть міняти свою поведінку в залежності від стану навколишнього середовища. Після аналізу вхідних сигналів вони самоналагоджувальна і навчаються, щоб забезпечити правильну реакцію. Навчена мережа може бути стійкою до деяких відхилень вхідних даних, що дозволяє їй правильно передбачати тренд, що містить різні аномалії та відхилення.

5

Рисунок В.5 – П'ятий аркуш презентації

Порівняння тестів літератури

Модель	Кількість шарів	Набір даних	Тип мережі	% точності моделі	% вірного передбачення знаку
Medianprice Close Price	5	1173	RNN	80,61	88
Momentum(All TADAWUL prices)	4	1170	CNN	79,5	86
All TADAWUL prices	3	1500	RNN	83,61	88
Mp & CP	3+1	1170	LSTM	83,42	87
MACD (low price)	3	30512	LSTM	83,89	83

6

Рисунок В.6 – Шостий аркуш презентації

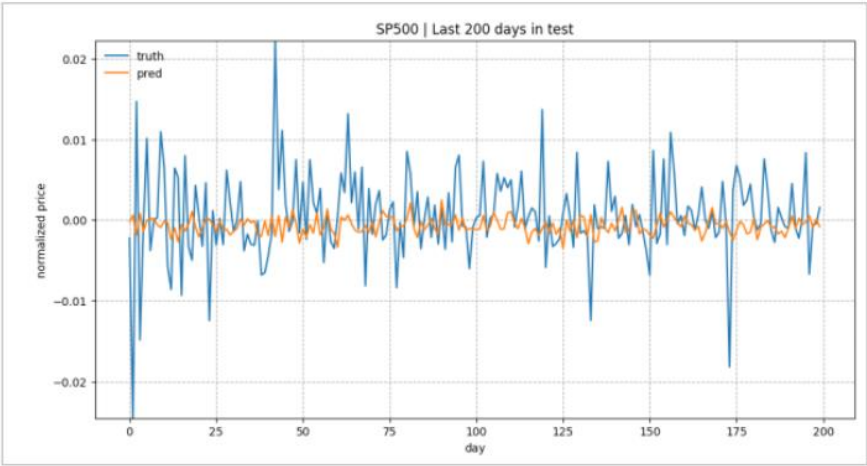
Точність побудованих мереж

Номер моделі	% точності моделі	% точності моделі після валидації	% вірного передбачення знаку
7	80,1	76,35	80,2
23	79,64	76,28	85,3
36	81,05	77,14	83,23
54	85,39	80,23	82,2

Номер слайда

Рисунок В.7 – Сьомий аркуш презентації

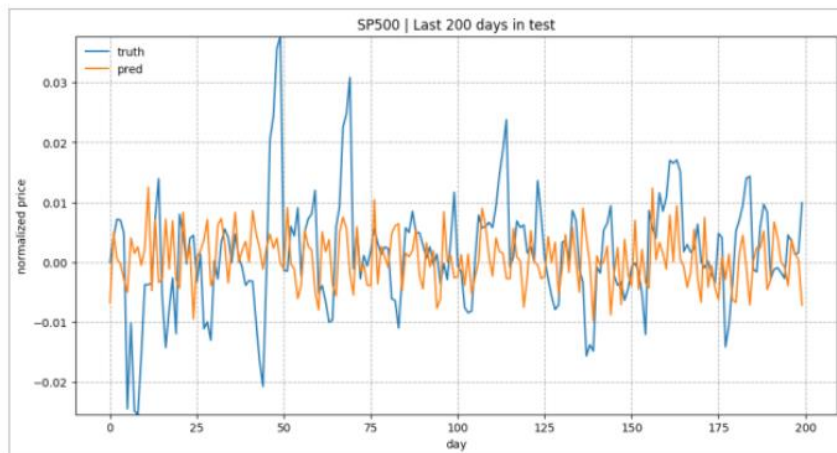
Модель Momentum з типом нейронної мережі CNN



8

Рисунок В.8 – Восьмий аркуш презентації

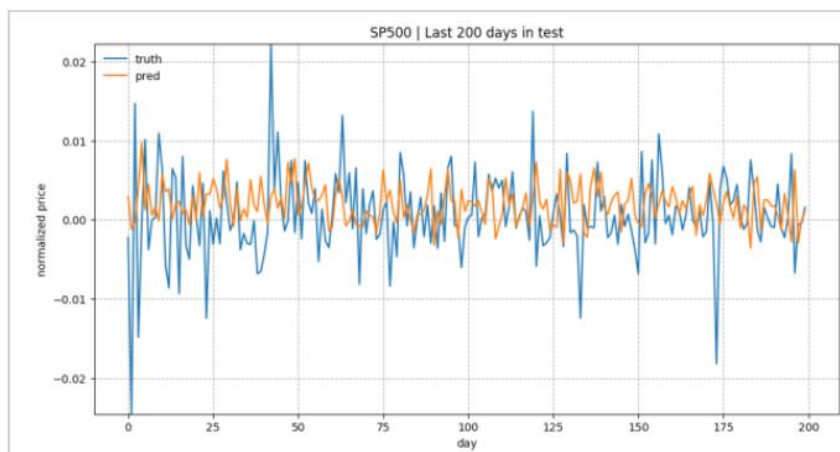
Модель Medianprice Close Price з типом нейронної мережі RNN



9

Рисунок В.9 – Дев'ятий аркуш презентації

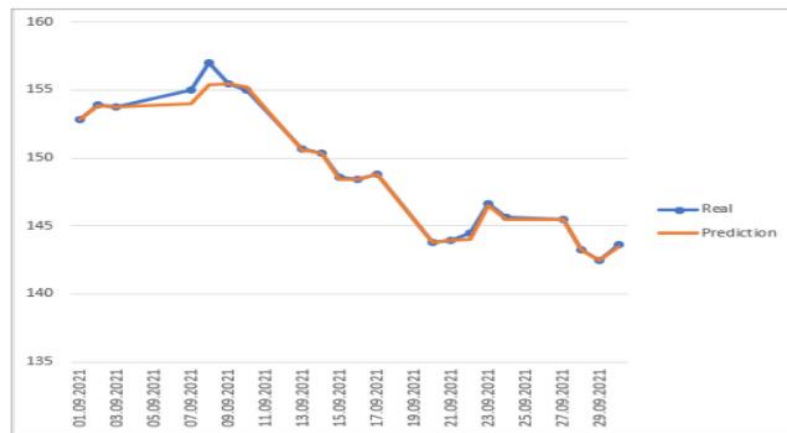
Модель MACD (low price) з типом нейронної мережі LSTM



10

Рисунок В.10 – Десятий аркуш презентації

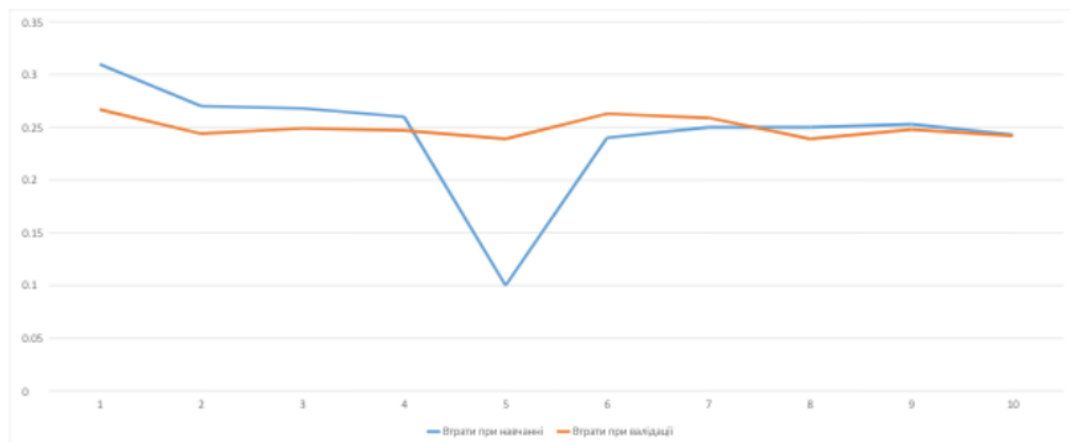
Приклад роботи нейронної мережі



11

Рисунок В.11 – Одинадцятий аркуш презентації

Втрати градієнта при навчанні та валідації



12

Рисунок В.12 – Дванадцятий аркуш презентації

Недоліки використання нейронних мереж

- Як можна бачити із графіків моделі мереж мають проблему із передбаченням амплітуди зміни індексу, це може призвести до великих витрат.
- Також слід зазначити що в усіх випадках даними для ряду слугували тільки показники цього самого ряду, що може призвести до ситуації де зміни показників інших рядів, що впливають на зміни показників даного ряду, викличуть зміни які модель не зможе передбачити

13

Рисунок В.13 – Тринадцятий аркуш презентації

Висновки

Нейронні мережі мають багато переваг понад класичними методами прогнозування показників фінансового ринку, а також можуть включати у себе ті данні що зазвичай не включають до створення традиційних моделей.

Але слід зазначити що точність нейронних мереж страждає від непередбаченості самого ринку

14

Рисунок В.14 – Чотирнадцятий аркуш презентації