

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ТАРАСА ШЕВЧЕНКА  
Факультет інформаційних технологій  
Кафедра інтелектуальних технологій**

**ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА  
БАКАЛАВРА  
НА ТЕМУ**

**Система прогнозування курсу валют на основі аналізу  
інформації з соціальних мереж**

Галузь знань **12 «Інформаційні технології»**

Спеціальність **122 «Комп'ютерні науки»**

Освітня програма **«Комп'ютерні науки»**

Освітній рівень: бакалавр

Виконав студент 4 курсу, групи КН-41

Богуцький О. І.

(прізвище та ініціали)

Керівник

Красовська Г. В.

(прізвище та ініціали)

кандидат технічних наук, доцент

(науковий ступінь, звання)

Випускна кваліфікаційна робота бакалавра допущена до захисту  
рішенням кафедри *інтелектуальних технологій*

Протокол № 13 від 05.06.2023 р.

зав. кафедри  доц. Іларіонов О. Є.

Київ - 2023

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет інформаційних технологій

Кафедра інтелектуальних технологій

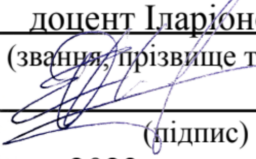
Спеціальність 122 «Комп'ютерні науки»

ЗАТВЕРДЖУЮ:

Зав. кафедри інтелектуальних технологій

доцент Іларіонов О. Є.

(звання, прізвище та ініціали)

  
(підпис)

«15» лютого 2023р. \_\_\_\_\_

**ЗАВДАННЯ**

**НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ**

Богуцькому Олегу Ігоровичу

(прізвище, ім'я, по батькові)

1. Тема роботи

Система прогнозування курсу валют на основі аналізу інформації з соціальних мереж

2. Термін здачі студентом закінченої роботи

28 травня 2023 року

3. Вихідні дані до роботи набір даних з соціальних мереж, набір даних з курсом валют

4. Зміст пояснювальної записки (перелік питань, що їх належить розробити) вступ, аналіз існуючих методів та моделей прогнозування курсу валют за допомогою інформації, наявної у відкритому доступі, проектування системи прогнозування курсу валют, реалізація системи прогнозування курсу валют, висновки, джерела, додатки


5. Перелік презентаційного матеріалу (з точним зазначенням обов'язкових презентацій)

Актуальність, мета, об'єкт, предмет, завдання (2 слайди), аналіз існуючих систем (8 слайдів), проектування системи (10 слайдів), реалізація системи і аналіз результатів (3 слайди), висновок (1 слайд)

7. Дата видачі завдання  
13 лютого 2023 року


---

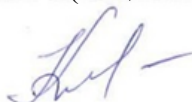
Керівник  /Красовська Г. В./  
(підпис) (ініціали та прізвище)

Завдання прийняв до виконання  /Богучький О. І./  
(підпис) (ініціали та прізвище)

### КАЛЕНДАРНИЙ ПЛАН

Пор. №	Назва етапів випускної кваліфікаційної роботи	Термін виконання етапів випускної кваліфікаційної роботи	Примітка
1	Вивчення матеріалів за темою роботи	13.02.2023 - 26.02.2023	Виконано
2	Підготовка першого розділу	20.02.2023 - 01.03.2023	Виконано
3	Підготовка другого розділу	02.03.2023 - 01.04.2023	Виконано
4	Розробка програмного застосунку	02.04.2023 - 15.04.2023	Виконано
5	Підготовка третього розділу	16.04.2023 - 01.05.2023	Виконано
6	Оформлення розділів відповідно до ДСТУ	02.05.2023 - 15.05.2023	Виконано
7	Підготовка презентації та доповіді	16.05.2023 - 28.05.2023	Виконано

Студент  /Богучький О. І./  
(підпис) (ініціали та прізвище)

Керівник випускної кваліфікаційної роботи  /Красовська Г. В./  
(підпис) (ініціали та прізвище)

### Анотація

**Богуцький Олег Ігорович** виконав випускню кваліфікаційну роботу на тему «Система прогнозування курсу валют на основі аналізу інформації з соціальних мереж» за спеціальністю 122 - «Комп'ютерні науки».

У випускній кваліфікаційній роботі проведено аналіз сучасних систем прогнозування курсу валют, запропоновано підхід до прогнозування, що покращує такі системи, розроблено інформаційне і програмне забезпечення, що виконує процедуру прогнозування курсу валют на основі семантичного аналізу повідомлень з соціальних мереж (Twitter) та фінансових показників SMA та RSI.

**Ключові слова:** прогнозування курсу валют, повідомлення з соціальних мереж, Twitter, рекурентна нейронна мережа, LSTM, фінансові показники, SMA, RSI.

### Summary

The degree project «Currency rate forecasting system based on analysis of information from social networks» has been completed by **Bohutskyi Oleh** specialty 122 - «Computer Science».

In the final qualification work, an analysis of modern exchange rate forecasting systems was carried out, an approach to forecasting that improves such systems was proposed, information and software was developed, which performs the exchange rate forecasting procedure based on the semantic analysis of messages from social networks (Twitter) and financial indicators of SMA and RSI.

**Keywords:** currency rate forecasting, messages from social networks, Twitter, recurrent neural network, LSTM, financial indicators, SMA, RSI.

## ЗМІСТ

ВСТУП	5
РОЗДІЛ 1. АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ ТА МОДЕЛЕЙ ПРОГНОЗУВАННЯ КУРСУ ВАЛЮТ ЗА ДОПОМОГОЮ ІНФОРМАЦІЇ, НАЯВНОЇ У ВІДКРИТОМУ ДОСТУПІ	8
1.1 Аналіз проблематики та актуальність	8
1.2 Аналіз існуючих алгоритмів передбачення курсу валют на основі аналізу інформації з соціальних мереж	8
1.3 Архітектура LSTM	13
1.4 Фінансові показники для аналізу динаміки зміни курсу валют	17
1.5 Вимоги до інформаційної системи	19
1.6 Висновки до першого розділу	19
РОЗДІЛ 2. ПРОЕКТУВАННЯ СИСТЕМИ ПРОГНОЗУВАННЯ КУРСУ ВАЛЮТ	21
2.1 Функціональний аналіз системи прогнозування курсу валют	21
2.2 Архітектура інформаційної системи	27
2.3 Проектування програмного користувацького інтерфейсу	28
2.4 Алгоритм прогнозування курсу валют	29
2.5 Висновки до другого розділу	34
РОЗДІЛ 3. РЕАЛІЗАЦІЯ СИСТЕМИ ПРОГНОЗУВАННЯ КУРСУ ВАЛЮТ	36
3.1 Вибір засобів розробки інформаційної системи	36
3.2 Опис структури програмного забезпечення системи прогнозування курсу валют	36
3.3 Керівництво користувача	38
3.4 Тестування та аналіз результатів	40
3.5 Висновки до третього розділу	58
ВИСНОВКИ	59

ДЖЕРЕЛА

5

60

ДОДАТОК

63

## ВСТУП

Через появу великої кількості криптовалют та нестабільну ситуацію на міжнародних фондових біржах через мінливу геополітичну ситуацію, гостро стає питання про коливання ціни тієї чи іншої валюти навіть в межах наступної доби. Тому, було прийнято рішення про розробку інформаційної системи, яка б могла передбачати курс валют за інформацією, яка наявна в глобальній мережі Інтернет та допомагати людям, які певним чином пов'язані з міжнародною фінансовою діяльністю.

Оскільки курс будь-якої валюти переважно залежить від зацікавленості людей у цій валюті, було б доцільно використовувати саме реакцію людей на події в світі, які мають безпосередній вплив на коливання курсу валют. Було б доцільно спробувати аналізувати не тільки фінансову інформацію, але і думки та настрої людей стосовно цих подій. Для цього нам знадобиться нерафіноване та неупереджене джерело інформації. Найкраще для цього підходять повідомлення користувачів соціальної мережі Twitter, адже там кожен може написати те, що він думає про будь-яку ситуацію та це повідомлення не буде змінено редактором.

Метою цієї випускної кваліфікаційної роботи бакалавра є розробка системи для підвищення точності прогнозування курсу валют за допомогою застосування фінансових показників та семантичного аналізу повідомлень з соціальних мереж.

Об'єктом дослідження є безпосередній вплив фінансових показників та інформації з відкритих джерел на коливання курсу валют.

Предметом дослідження є методи прогнозування курсу валют на основі економічних показників та інформації з соціальних мереж.

Для досягнення мети було сформовано наступні завдання:

- проаналізувати існуючі методи інтелектуального прогнозування курсу валют;

- проаналізувати підходи до покращення точності прогнозування курсу валют;
- обрати інструменти штучного інтелекту для передбачення курсу валют, а саме: обрати архітектуру нейронної мережі, набори даних, з якими будемо працювати та фінансові показники, які будуть безпосередньо впливати на результати прогнозування;
- дослідити рівень кореляції різних параметрів з різних наборів даних для отримання найбільш точного результату прогнозування;
- реалізувати алгоритм попередньої обробки даних для коректного відпрацювання алгоритму передбачення, проаналізувати ефективність системи у конкретній задачі та результати згенеровані програмою;
- розробити веб-застосунок для прогнозування курсу та довести його працездатність.

Перший розділ роботи описує: аналіз проблематики предметної області, постановку задачі на випуск кваліфікаційну роботу, аналіз інструментів, які будуть використані під час роботи, аналіз існуючих методів прогнозування курсу валют, алгоритм програми та вимоги до інформаційної системи.

В другому розділі роботи проведено функціональний аналіз системи, розроблено архітектуру системи, спроектовано веб-застосунок, розроблено дизайн для нього та наведено детальний алгоритм прогнозування курсу валют на основі аналізу інформації з соціальних мереж.

В третьому розділі проведено вибір засобів розробки системи, описано структуру програмного забезпечення, розроблено керівництво користувача, проведено тестування додатку та проаналізовано результати прогнозування.

В результаті виконання роботи було спроектовано та розроблено покращену систему для прогнозування курсу валют за допомогою інформації, отриманої з соціальних мереж, спроектовано та розроблено веб-застосунок, а також було проаналізовано результати і доведено підвищену ефективність системи, порівняно з стандартними програмами прогнозування курсу валют. Роботу буде продовжено у напрямку покращення точності роботи програми.



## РОЗДІЛ 1. АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ ТА МОДЕЛЕЙ ПРОГНОЗУВАННЯ КУРСУ ВАЛЮТ ЗА ДОПОМОГОЮ ІНФОРМАЦІЇ, НАЯВНОЇ У ВІДКРИТОМУ ДОСТУПІ

### 1.1 Аналіз проблематики та актуальність

У світовій фінансовій економіці ринок обміну іноземної валюти є одним із найбільших світових ринків із середнім обсягом торгів понад 7,5 трильйона доларів США щодня. Метою обміну валюти є інвестиції, торгівля, спекуляція та хеджування. Обмін валюти є однією з найголовніших складових міжнародних відносин. Він впливає не тільки на великі компанії та держави, але і на звичайних споживачів. Світ стоїть на шляху глобалізації, через це багато товарів та послуг, якими ми користуємось щодня, можуть бути вироблені не тільки в нашій країні, таким чином курс валют безпосередньо впливає на ціноутворення в кожній державі світу. Тому людям, які планують економічні стратегії, вкрай необхідно мати інструменти, які б могли допомогти їм в цьому. Людський мозок побудований так, що він може доходити складних дедуктивних та індуктивних висновків, але при цьому, об'єм інформації, який він може опрацювати досить обмежений, тим більше за таких мінливих умов. Саме тут людині може допомогти комп'ютер, через свою здатність обраховувати велетенську кількість даних. Приймаючи до уваги це, постало завдання створити інформаційну систему, яка б могла поєднати людську реакцію на різні події у світі з фінансовою інформацією, яка з'являється щосекунди у різних куточках світу.

### 1.2 Аналіз існуючих алгоритмів передбачення курсу валют на основі аналізу інформації з соціальних мереж

Для передбачення курсу валют застосовується багато різних автоматизованих систем. Звернувшись до роботи [1], було з'ясовано, які

підходи найчастіше використовуються авторами наукових робіт для прогнозування курсу валют (рис 1.1).

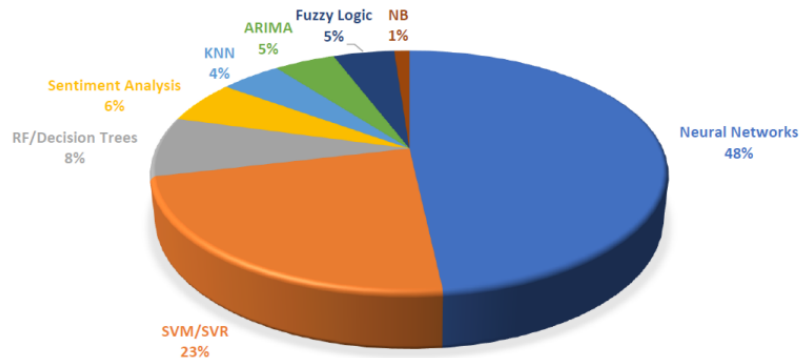


Рисунок 1.1 - Використання підходів для задачі прогнозування курсу валют

На рис. 1.1 можна побачити частоту використання тих чи інших підходів у задачі прогнозування курсу валют:

- Neural Networks - Нейронні мережі (48%);
- SVM/SVR (Support Vector Machine/Support Vector Regression) - Метод опорних векторів (23%);
- RF (Random Forest)/Decision Trees - Випадкові ліси/Дерева рішень (8%);
- Sentiment Analysis - Аналіз тональності тексту (6%);
- KNN (k-nearest neighbors) - Метод k-найближчих сусідів (4%);
- ARIMA (autoregressive integrated moving average) - модель авторегресійної інтегрованої ковзної середньої (5%);
- Fuzzy Logic - Нечітка логіка (5%);
- NB (naive Bayes) - Наївний Байєсів класифікатор (1%).

Можна зробити висновок, що найчастіше, для задачі прогнозування курсу валют, використовуються нейронні мережі, отже застосуємо саме їх.

Для необхідно визначити, яку модель нейронних мереж буде використано у якості основи у цій роботі. Посилаючись на роботу [2], було

з'ясовано, які моделі нейронних мереж найчастіше використовуються для задачі цієї роботи (рис 1.2).

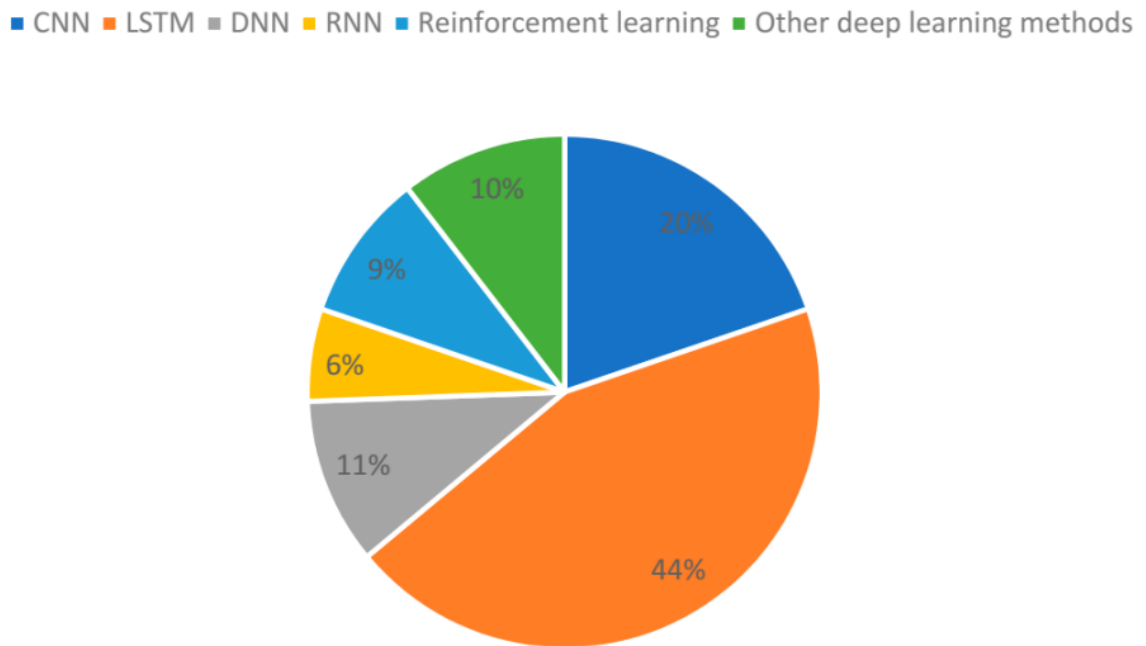


Рисунок 1.2 - Використання різних моделей нейронних мереж для задачі прогнозування курсу валют

На рис. 1.2 можна побачити частоту використання тих чи інших моделей нейронних мереж у задачі прогнозування курсу валют:

- LSTM (Long short-term memory) - Довга короткочасна пам'ять (44%);
- CNN (Convolutional neural network) - Згорткові нейронні мережі (20%);
- DNN (Deep neural network) - Глибинні нейронні мережі (11%);
- Other deep learning methods - Інші методи глибинного навчання (10%);
- Reinforcement learning - Навчання з підкріпленням (9%)
- RNN (Recurrent neural network) - Рекурентні нейронні мережі (6%).

Отже, найчастіше використовується модель LSTM та згорткові нейронні мережі. Також необхідно визначити точність різних моделей у задачі прогнозування курсу валют. Для цього звернемось до інших робіт.

У роботі [3] порівнюється декілька архітектур нейронних мереж: RNN, LSTM, GRU. Було обрано 4 комбінації курсів валют для більш незалежного дослідження. Вчені дійшли висновку, що моделі LSTM та GRU показали кращі результати ніж звичайна RNN. Як результат було отримано 4 таблиці з похибками по кожній комбінації курсів валют:

Таблиця 1.1 - Результат експерименту для GBP/USD [3]

Model	Timesteps	MSE ( $10^{-3}$ )	RMSE ( $10^{-3}$ )	MAE ( $10^{-3}$ )	MAPE (%)
RNN	30	1.846	42.960	35.336	2.774
LSTM	30	2.435	49.351	48.357	3.749
BiLSTM	30	1.520	38.984	37.456	2.903
GRU	30	6.311	79.441	78.185	6.067
RNN	60	<b>0.710</b>	<b>26.645</b>	<b>25.118</b>	<b>1.945</b>
LSTM	60	0.853	29.200	26.617	2.075
BiLSTM	60	1.489	38.593	37.098	2.873
GRU	60	3.324	57.658	56.723	4.401

Таблиця 1.2 - Результат експерименту для EUR/GBP [3]

Model	Timesteps	MSE ( $10^{-3}$ )	RMSE ( $10^{-3}$ )	MAE ( $10^{-3}$ )	MAPE (%)
RNN	30	0.032	5.628	4.456	0.503
LSTM	30	<b>0.006</b>	<b>2.407</b>	<b>1.708</b>	<b>0.194</b>
BiLSTM	30	0.168	12.962	12.701	1.440
GRU	30	0.151	12.273	11.343	1.277
RNN	60	0.075	8.684	7.270	0.817
LSTM	60	0.006	2.536	1.957	0.222
BiLSTM	60	0.076	8.705	8.434	0.957
GRU	60	0.135	11.612	10.548	1.187

Таблиця 1.3 - Результат експерименту для AUD/USD [3]

Model	Timesteps	MSE ( $10^{-3}$ )	RMSE ( $10^{-3}$ )	MAE ( $10^{-3}$ )	MAPE (%)
RNN	30	0.530	23.030	20.406	2.875
LSTM	30	0.568	23.835	22.481	3.149
BiLSTM	30	4.821	69.433	68.646	9.566
GRU	30	<b>0.201</b>	<b>14.162</b>	<b>13.588</b>	<b>1.929</b>
RNN	60	1.061	32.578	31.106	4.377
LSTM	60	0.345	18.585	16.365	2.291
BiLSTM	60	6.603	81.263	80.897	11.320
GRU	60	0.298	17.270	16.626	2.351

Таблиця 1.4 - Результат експерименту для CAD/CHF [3]

Model	Timesteps	MSE ( $10^{-3}$ )	RMSE ( $10^{-3}$ )	MAE ( $10^{-3}$ )	MAPE (%)
RNN	30	0.089	9.450	7.388	1.023
LSTM	30	0.092	9.570	8.537	1.167
BiLSTM	30	0.225	15.011	14.312	1.931
GRU	30	<b>0.027</b>	<b>5.247</b>	<b>4.264</b>	<b>0.583</b>
RNN	60	0.183	13.543	12.030	1.641
LSTM	60	0.154	12.396	11.088	1.514
BiLSTM	60	0.032	5.674	4.858	0.651
GRU	60	0.032	5.661	4.340	0.603

Можна побачити, що найкращий результат вдалось досягти саме за допомогою LSTM мережі.

У роботі [4] досліджується вплив додавання проаналізованих заголовків новин до набору даних з курсом біткоїна. Також досліджувалась ефективність різних моделей для передбачення курсу валют: ARIMA, SARIMA, регресійні моделі: лінійна регресія, регресор SGD, регресор посилення XG і регресор опорних векторів; а також архітектури нейронних мереж: RNN, LSTM, GRU. Результатом цієї роботи стало незначне покращення результатів передбачення. Дослідники пов'язують це з тим, що

новини не були попередньо відібрані та були з різних сфер. Отже ми будемо використовувати тільки інформацію, що безпосередньо пов'язана з біткоїном.

У роботі [5] було також досліджено вплив заголовків новин, але дослідники використали попередню обробку новин та вибрали лише ті, де є згадування валюти, що аналізується. Результатом роботи було покращення спроможності нейронної мережі передбачати курс валют, а також показано досить непогані результати роботи мережі LSTM. Дослідники не використовували жодних додаткових фінансових показників у своїй роботі.

Отже зважаючи на вищесказане, ми будемо обробляти інформацію з відкритих джерел тільки безпосередньо пов'язану з біткоїном та використовувати архітектуру LSTM для передбачення, а також застосуємо фінансові показники та спробуємо покращити результати передбачення.

Оскільки ми проводимо аналіз даних у конкретних моментах часу, нам необхідна мережа, яка показує гарні результати в обробці часових рядів. Саме архітектура рекурентних мереж налаштована на обробку таких даних. Існує ще один фактор за яким ми обираємо мережу, що підходить нам – наявність різких стрибків, або різкої зміни параметру, що виникає через довільні проміжки часу. Такі умови значно скорочують кількість можливих архітектур нейронних мереж для вибору. Однією з відповідних нейронних мереж є LSTM (long short-term memory (укр. довга короткочасна пам'ять)).

### 1.3 Архітектура LSTM

**Довга короткочасна пам'ять** (ДКЧП, *long short-term memory, LSTM*) – це архітектура рекурентних нейронних мереж, яку запропонували автори однойменної наукової праці, Зепп Хохрайтер та Юрген Шмідгубер у 1997 році. Вона призначена для подолання проблем зі зворотним потоком похибок, а також може обробляти часові інтервали, що перевищують 1000 кроків, навіть у випадку шумних, нестисливих вхідних послідовностей. Це досягається за допомогою ефективного алгоритму на основі градієнта, який

забезпечує постійний (тобто ні вибухаючий, ні зникаючий) потік помилок через внутрішні стани спеціальних комірок [6].

Розглянемо основні принципи роботи LSTM мережі [7].

Основною відмінністю LSTM мережі від інших рекурентних мереж є наявність стану, який проходить через всю комірку у вигляді горизонтальної лінії (рис. 1.3).

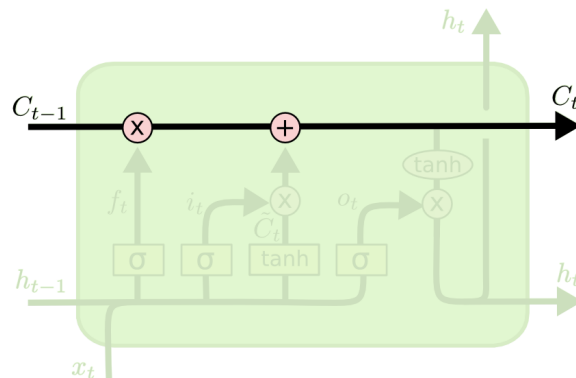


Рисунок 1.3 - Стан комірки [7]

У комірці існує механізм «забування» а також додавання інформації, який відбувається за допомогою так званих «вентилів» (англ. gates) (рис. 1.4).

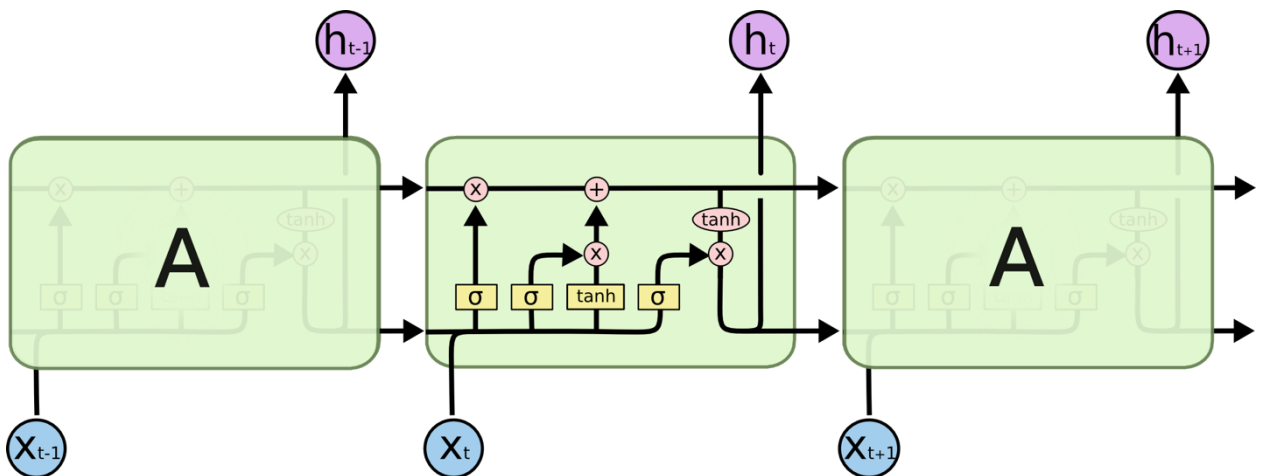
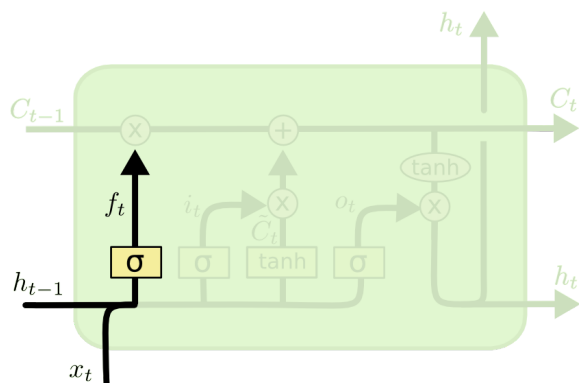


Рисунок 1.4 - Мережа LSTM [7]

Для того, щоб зрозуміти, яким чином працює мережа, та навіщо потрібні вентиля, розглянемо процес обробки інформації в комірці покроково.

На першому кроці вирішується, від якої інформації нам треба позбутися у цій комірці. Це відбувається за допомогою сигмоїдального шару «забуття». Параметри  $h_{t-1}$  та  $x_t$  впливають на сигмоїдальний шар і на виході ми

отримуємо число від 0 до 1: 0 – повністю стерти інформацію, а 1 – повністю зберегти (рис. 1.5).

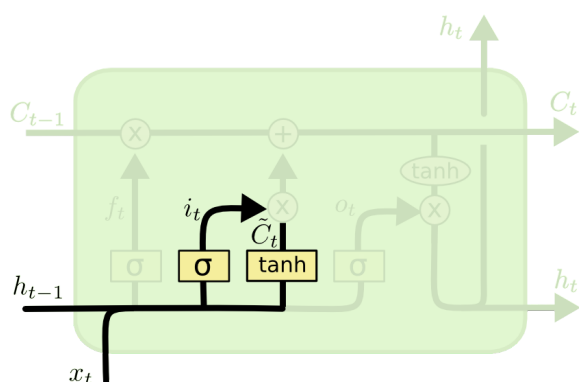


$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

Рисунок 1.5 - Шар «забуття» [7]

Якщо розглядати цю мережу для нашої задачі передбачення курсу валют, цей шар може значно покращити результати. Наприклад, коли курс зазнає сильних змін, нам потрібно кардинально змінити нашу схему прогнозування. Саме для таких випадків використовується цей шар і саме тому, ця мережа нам підходить.

На наступному кроці вирішується, яку інформацію треба зберегти у стані комірки. Цей крок складається з двох частин. По-перше, сигмоїдальний шар визначає, які дані ми будемо оновлювати. Наступний шар  $\tanh$  створює вектор нових значень, які можна додати до стану комірки (рис. 1.6).

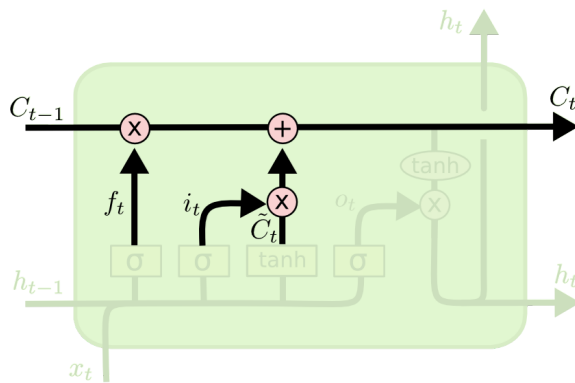


$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

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

Рисунок 1.6 - Шар оновлення [7]

В результаті виконання даного кроку було визначено зміни, які необхідно внести до стану комірки.

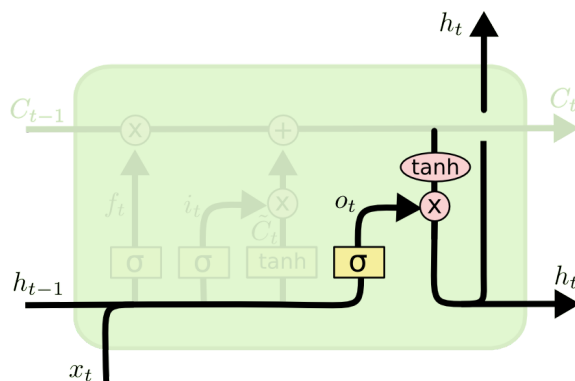


$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Рисунок 1.7 - Застосування змін [7]

На рисунку 1.7 видно, що для того, щоб застосувати зміни нам потрібно помножити значення старого стану на  $f_t$  (результат шару «забуття»), а потім додати значення  $i_t * \tilde{C}_t$  (результат шару оновлення).

Нарешті на останньому кроці нам потрібно визначити, які дані ми будемо «випускати» з комірки. Цей крок не впливає на  $C_t$ , але впливає на  $h_t$ . Для цього нам потрібно пропустити  $h_{t-1}$  через сигмоїдальний шар, а потім помножити на стан комірки, пропущений через  $\tanh$  (для нормалізації у межах  $[-1; 1]$ ) (рис. 1.8).



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

$$h_t = o_t * \tanh(C_t)$$

Рисунок 1.8 - Шар виводу [7]

Велика кількість шарів та дій може трохи налякати, але після покрокового пояснення виявляється, що це не так складно, а також, що саме така архітектура найліпше підходить для вирішення завдання цієї курсової роботи.

## 1.4 Фінансові показники для аналізу динаміки зміни курсу валют

Оскільки нам потрібно розуміти загальну тенденцію ціни на валютному ринку, а також швидкість зміни цієї тенденції було обрано два фінансових показники для подальшого аналізу [8]:

- 1 **SMA** - ковзне середнє або рухоме середнє;
- 2 **RSI** - індекс відносної сили.

### 1.4.1 Показник SMA

Ковзне середнє або рухоме середнє – один із інструментів аналізу випадкових процесів та часових рядів, він показує середню ціну за певний проміжок часу (рис. 1.9) [9].

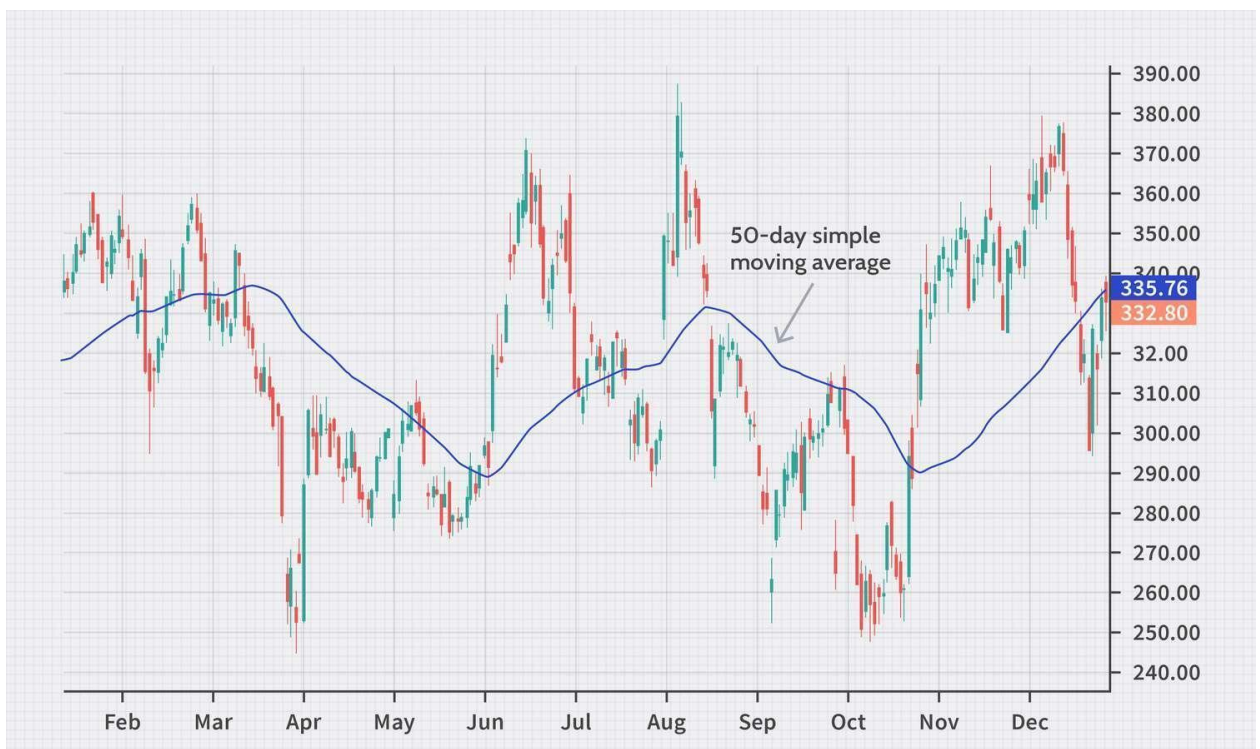


Рисунок 1.9 - Показник SMA [9]

Цей показник обчислюється за формулою:

$$SMA = \frac{\sum_{x=1}^n A_x}{n}, \quad (1.1)$$

де  $A_x$  – це значення ціни у період  $x$ , а  $n$  – це кількість таких періодів.

Отже цей показник допоможе знайти загальний тренд поведінки курсу валюти, а також згладити різкі зміни курсу.

#### 1.4.2 Показник RSI

Індекс відносної сили – це індикатор, який можна використовувати для аналізу поведінки курсу валют, він вимірює силу тренду та ймовірність його зміни (рис. 1.10) [10].



Рисунок 1.10 - Показник RSI [10]

Він обчислюється за формулою:

$$RSI = 100 - \frac{100}{RS}, \quad (1.2)$$

де

$$RS = \frac{EMA[N] \text{ of } U}{EMA[n] \text{ of } D}, \quad (1.3)$$

де  $N$  – це період за який розраховується показник,  $U$  та  $D$  – позитивні та негативні цінові зміни.  $EMA$  – це ще один фінансовий показник. Він не буде розглядатися в цій роботі.

$RSI$  – це нормований показник цінової тенденції, він лежить в діапазоні  $[0, 100]$ , а отже нам буде зручно використовувати його для передбачення ціни.

### 1.5 Вимоги до інформаційної системи

Функціональні вимоги:

- 1) інформаційна система повинна отримувати актуальні дані з відкритих джерел;
- 2) інформаційна система повинна коректно обробляти отримані набори даних;
- 3) інформаційна система повинна правильно обраховувати задані фінансові показники за формулами 1.1 та 1.2;
- 4) інформаційна система повинна навчати нейронну мережу без перенавчання та з заданою похибкою
- 5) інформаційна система повинна надавати актуальні прогнозовані дані, щодо курсу валют.

Нефункціональні вимоги:

- 1) інформаційна система повинна мати зрозумілий та ергономічний користувацький інтерфейс;
- 2) інформаційна система не повинна перевантажувати комп'ютер користувача (тобто обрахунки повинні проводитись на серверній частині застосунку);

### 1.6 Висновки до першого розділу

Проаналізувавши існуючі алгоритми передбачення курсу валют за допомогою інформації з відкритих джерел, стало зрозуміло, що існують інформаційні системи спрямовані на вирішення цієї задачі, але у них дослідники не звертають увагу на дослідження в області фінансів.

Отже було вирішено використати фінансові показники, які можуть спростити навчання нейронної мережі та покращити результати передбачення.

Була обрана архітектура нейронної мережі LSTM, через її гарні показники у передбаченні часових рядів з різкої зміною параметрів в умовах невизначених проміжків часу між цими змінами. Був обраний базовий алгоритм для навчання нейронної мережі та сформовані вимоги до інформаційної системи.

## РОЗДІЛ 2. ПРОЕКТУВАННЯ СИСТЕМИ ПРОГНОЗУВАННЯ КУРСУ ВАЛЮТ

### 2.1 Функціональний аналіз системи прогнозування курсу валют

Для початкового проектування системи необхідно представити системи у вигляді «чорної скрині» (рис 2.1). Таким чином буде зрозуміло, які входи, виходи системи, а також які механізми використовує система при роботі і чим вона керується.

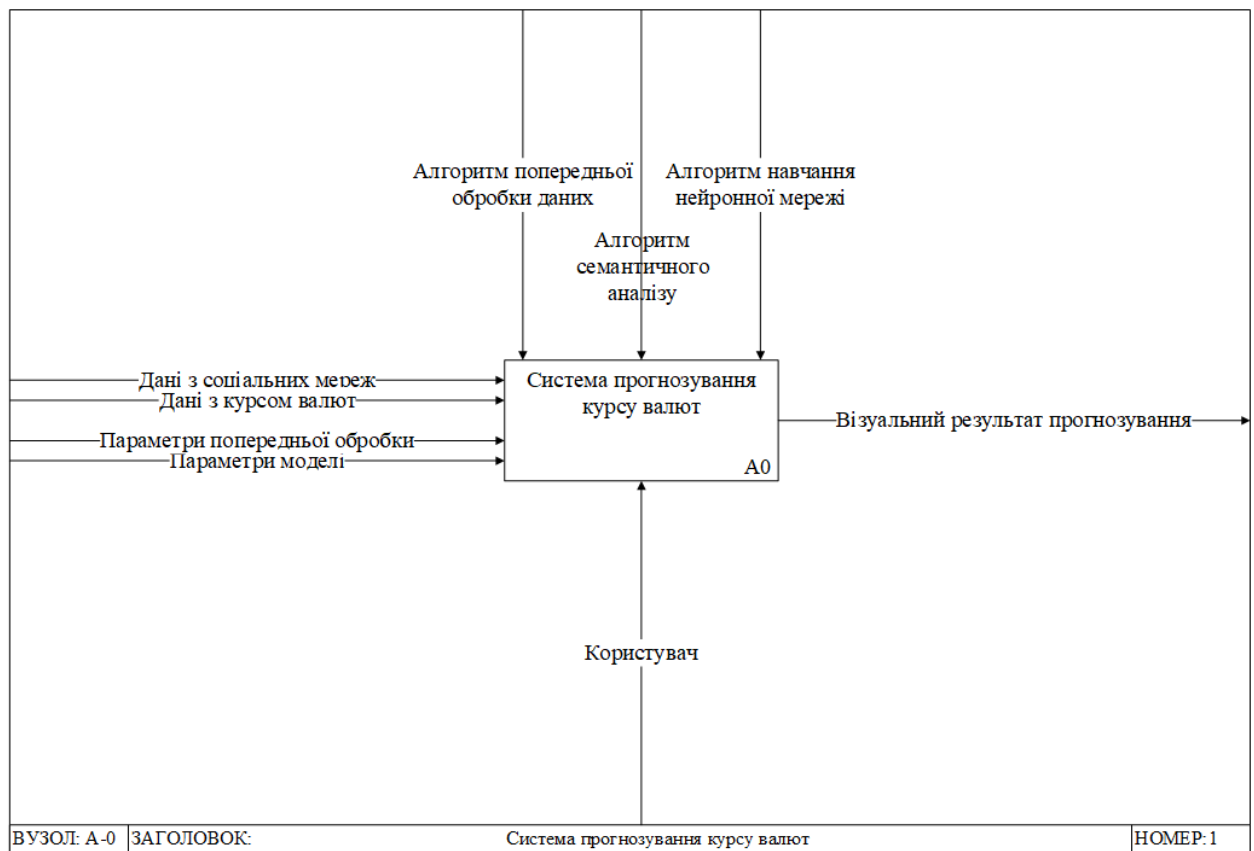


Рисунок 2.1 - Контекстна діаграма системи прогнозування курсу валют

Опис основних елементів контекстної діаграми, що представлена на рис. 2.1, наведено в таблиці 2.1.

Таблиця 2.1 - Опис чорної скрині

<b>Вид активності</b>	<b>Активність</b>	<b>Опис</b>
Чорна скриня	Інтелектуальний модуль прогнозування курсу валют	Позначення модуля у вигляді чорної скрині
Вхід	Дані з відкритих джерел	Дані, що надходять в модуль з відкритих джерел
	Дані з курсом валют	Дані, що надходять в модуль з курсом валют
	Параметри попередньої обробки	Параметри для попередньої обробки даних
	Параметри моделі	Гіперпараметри для навчання моделі
Управління	Алгоритм попередньої обробки даних	Алгоритм, яким керується система для валідної попередньої обробки даних
	Алгоритм навчання нейронної мережі	Алгоритм, яким керується система для ефективного навчання нейронної мережі
	Алгоритм семантичного аналізу	Алгоритм, яким керується система для проведення семантичного аналізу повідомлень
Механізм	Користувач	Механізм для введення даних в систему
Вихід	Результати прогнозування	Вихідні результати

Далі необхідно розкрити чорну скриню та визначити всі функції, що відбуваються всередині, для цього побудуємо дерево функцій (рис. 2.2):

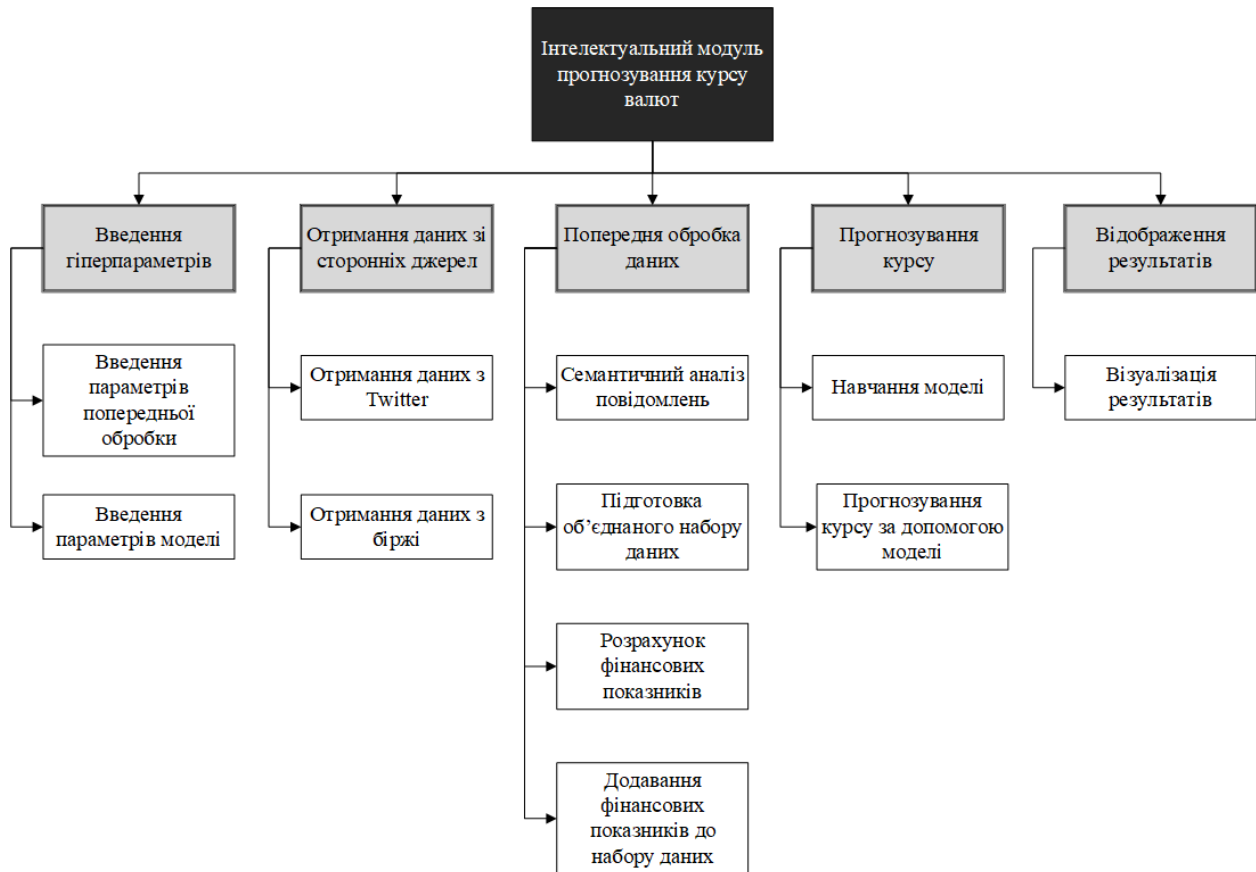


Рисунок 2.2 - Дерево функцій системи

В дереві функцій (рис. 2.2) схематично зображено:

- чорним кольором - систему;
- сірим кольором - основні блоки функцій системи;
- білим кольором - функції системи.

Опис всіх функцій:

- введення параметрів моделі - функція приймає параметри введені користувачем, де зазначені всі налаштування моделі;
- введення параметрів попередньої обробки даних - функція приймає параметри введені користувачем, де зазначено, які стовпці треба залишити в об'єднаному масиві, а які видалити;
- отримання даних з Twitter - функція відправляє запит до соціальної мережі Twitter та отримує об'єкт в форматі ключ-значення;

— семантичний аналіз повідомлень - функція проводить семантичний аналіз кожного повідомлення, на виході отримуємо значення Polarity (полярність тексту) та Sensitivity (значущість тексту);

— отримання даних з біржі - функція відправляє запит до біржі та отримує об'єкт в форматі ключ-значення;

— розрахунок фінансових показників - функція приймає масив з даними з біржі та на їх основі розраховує фінансові показники RSI та SMA, на виході отримуємо масив з фінансовими показниками для кожного рядка масиву з даними з біржі;

— додавання фінансових показників до набору даних - функція приймає масив з даними з біржі та додає до них масив з фінансовими показниками, на виході отримуємо об'єднаний набір даних;

— підготовка об'єданого набору даних - функція приймає об'єднаний масив з даними з біржі та фінансовими показниками і оброблений масив з даними з Twitter, об'єднує їх та групує всі рядки по годинам;

— навчання моделі - функція приймає параметри та набір даних і навчає моделі згідно цих параметрів;

— прогнозування курсу валют за допомогою моделі - функція прогнозує курс валют за навченою моделлю, обраховує похибку та будує графіки похибки та прогнозованого курсу;

— візуалізація результатів - функція будує графік похибки та графік передбаченого курсу валют, а також виводить загальну похибку навчання.

Після визначення всіх функцій, треба визначити порядок виконання цих функцій (рис 2.3).

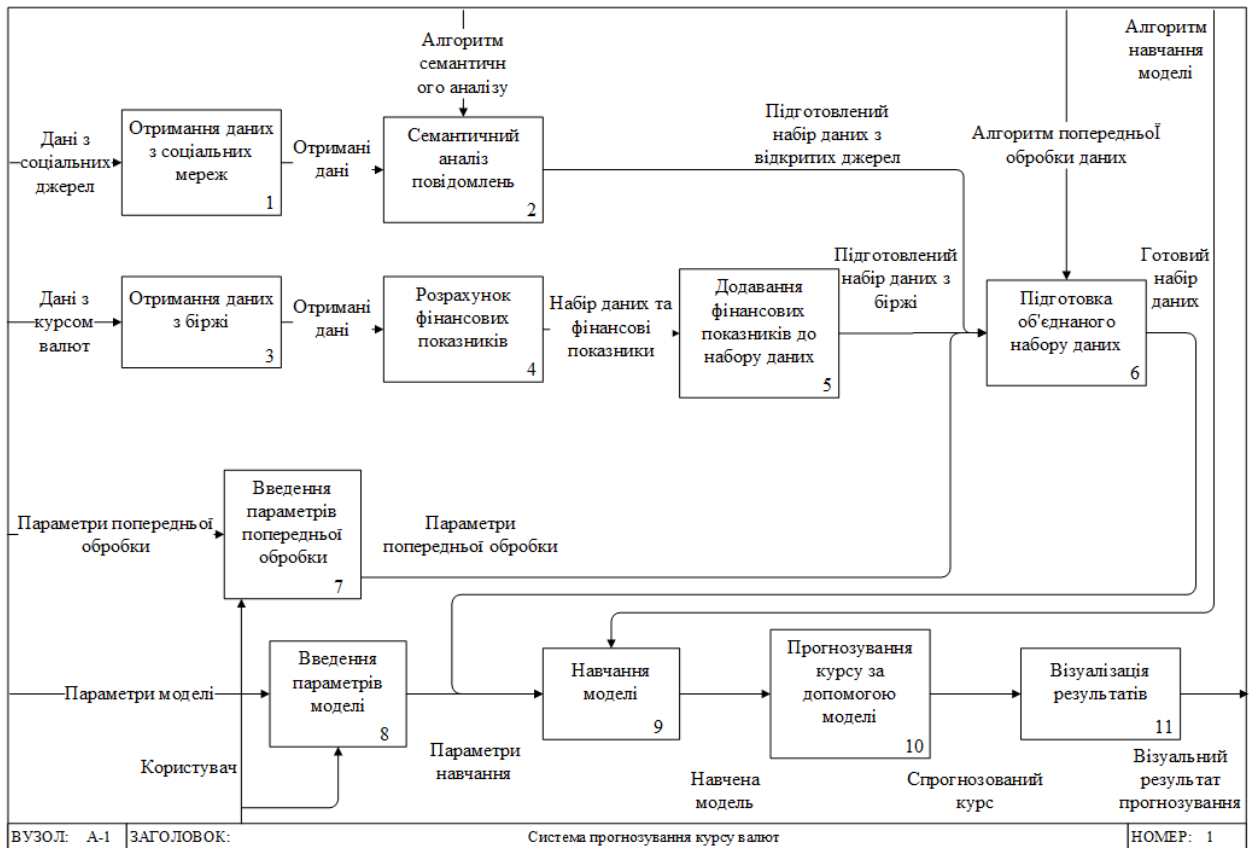


Рисунок 2.3 - Технологія прогнозування курсу валют на основі фінансових показників та даних з соціальних мереж

Опис технології прогнозування курсу валют на основі фінансових показників та даних з відкритих мереж:

Входи: дані з відкритих джерел, дані з курсом валют.

Виходи: візуальний результат прогнозування.

Управління: алгоритм семантичного аналізу, алгоритм навчання моделі, алгоритм попередньої обробки даних.

Механізми: користувач.

Алгоритм прогнозування курсу валют потребує попередньої підготовки даних, процес підготовки можна розділити на два паралельні потоки:

Потік 1. Попередня обробка даних з відкритих джерел:

1) отримання даних з відкритих джерел - на вхід надходять дані з відкритих джерел, які приймає блок отримання даних з відкритих джерел, виходять отриманні дані;

2) семантичний аналіз повідомлень - отримані дані проходять семантичний аналіз даних, який в свою чергу керується алгоритмом семантичного аналізу, виходять проаналізовані дані;

Потік 2. Попередня обробка даних з біржі:

3) отримання даних з біржі - на вхід надходять дані з біржі, які приймає блок отримання даних з біржі, виходять отриманні дані;

4) розрахунок фінансових показників - основі отриманих даних розраховуються фінансові показники, на виході - набір даних та фінансові показники;

5) додавання фінансових показників до набору даних - відбувається об'єднання набору даних з біржі та фінансових показників, виходить набір даних з фінансовими показниками;

6) Введення параметрів попередньої обробки - користувач вводить параметри попередньої обробки даних;

7) Підготовка об'єднаного набору даних - знаючи параметри попередньої обробки, керуючись алгоритмом попередньої обробки даних проводиться підготовка об'єднаного набору даних, на виході отримуємо готовий набір даних;

8) Введення параметрів моделі - користувач вводить параметри навчання моделі;

9) Навчання моделі - на основі введених параметрів та алгоритму навчання моделі, модель навчається, отримуємо навчену модель;

10) Прогнозування курсу за допомогою моделі - відбувається прогнозування курсу за допомогою моделі і отримуємо прогнозований курс;

11) Візуалізація результатів - за допомогою графіків відбувається візуалізація результатів, на виході отримуємо візуальний результат прогнозування.

## 2.2 Архітектура інформаційної системи

Система повинна бути реалізована в форматі веб-застосунку, всі обчислювальні дії повинні відбуватися на боці серверу.

Отримувати дані застосунок повинен з різних ресурсів та від користувача. Передача даних зі сторонніх ресурсів відбувається за допомогою API (Application Programming Interface).

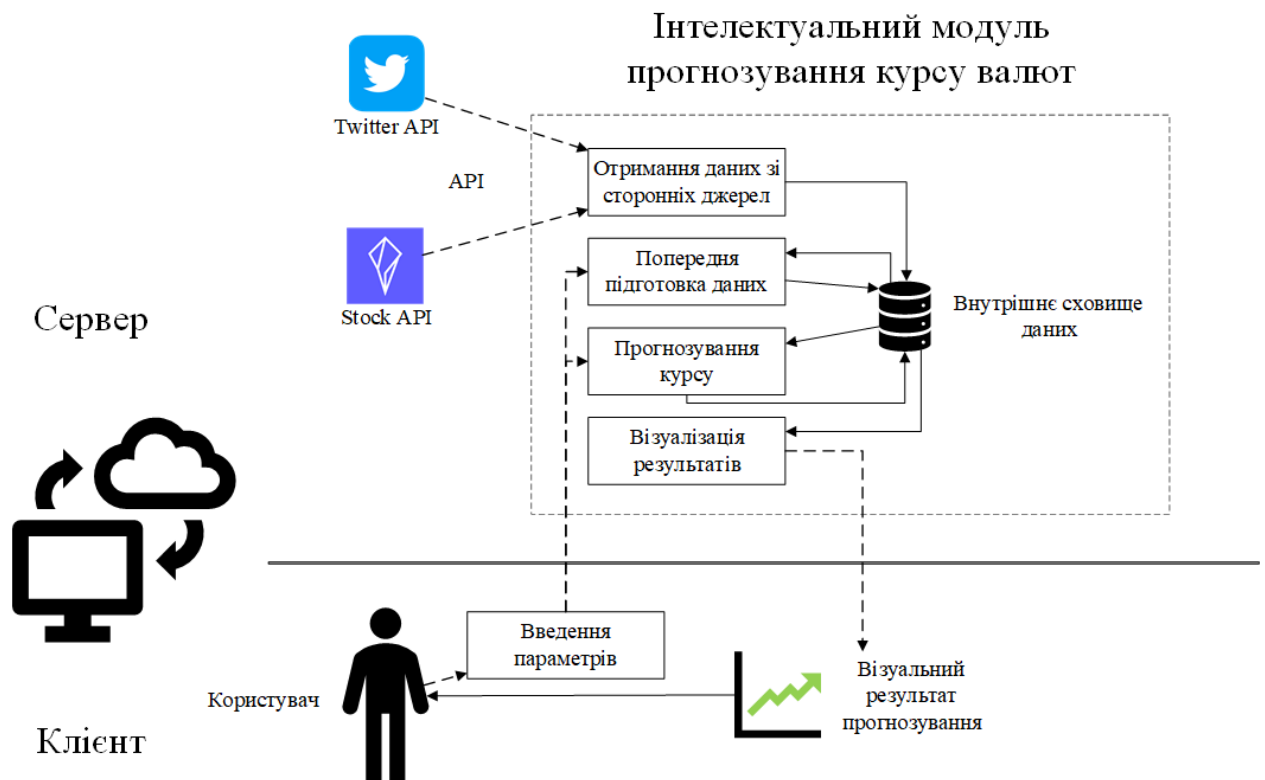


Рисунок 2.4 - Архітектура системи

Опис архітектури системи:

Всі наступні дії відбуваються послідовно:

Після та перед діями 2 - 5, дані зберігаються та видобуваються з внутрішнього сховища даних:

- 1) **Введення параметрів** - користувач вводить параметри попередньої обробки даних та параметри моделі;
- 2) **Отримання даних зі сторонніх джерел** - за допомогою API відбувається отримання даних зі сторонніх ресурсів (Twitter API, Stock API);

3) Попередня підготовка даних - відбувається попередня підготовка даних, отриманих зі сторонніх джерел, на основі параметрів, які ввів користувач (пункт 1);

4) Прогнозування курсу - за допомогою нейронної мережі відбувається прогнозування курсу валют, на основі параметрів, які ввів користувач (пункт 1);

5) Візуалізація результатів - на основі результатів отриманих з попереднього пункту, відбувається побудова графіків похибки та спрогнозованих даних.

На стороні клієнта відбувається введення параметрів та відображення результатів, всі інші дії відбуваються на боці серверу.

Така архітектура дозволить знизити навантаження на персональний комп'ютер користувача, що дозволить не перевантажувати його.

### 2.3 Проектування програмного користувацького інтерфейсу

Дизайн користувацького інтерфейсу буде розроблений у програмі Figma, оскільки це зручний інструмент, який дозволяє реалізувати дизайн будь-якої складності.

Сайт складається з двох сторінок. На сторінці з налаштуваннями користувач має змогу налаштувати попередню обробку даних, а також параметри навчання нейронної мережі. Для зображення процесу обробки даних системою буде використано анімацію процесу завантаження.

Forex Prediction : Home Home About App

**Close Price**   
Ціна закриття торгів

**Polarity**   
Настрій тексту повідомлення

**Sensitivity**   
Чутливість тексту повідомлення

**Tweet Volume**   
Кількість твітів за годину

**BTC volume**   
Обсяг торгів

**SMA**   
Показник SMA

**RSI**   
Показник RSI

Train set: 50% | Test set: 50%

---

Epochs:

Train Hours:

[Обрахувати](#)

Рисунок 2.5 - Дизайн сторінки налаштувань

Сторінка з результатами має містити: графік похибки нейронної мережі на стадії навчання, графік порівняння реального курсу валюти та передбаченого курсу валюти, та загальну похибку моделі.

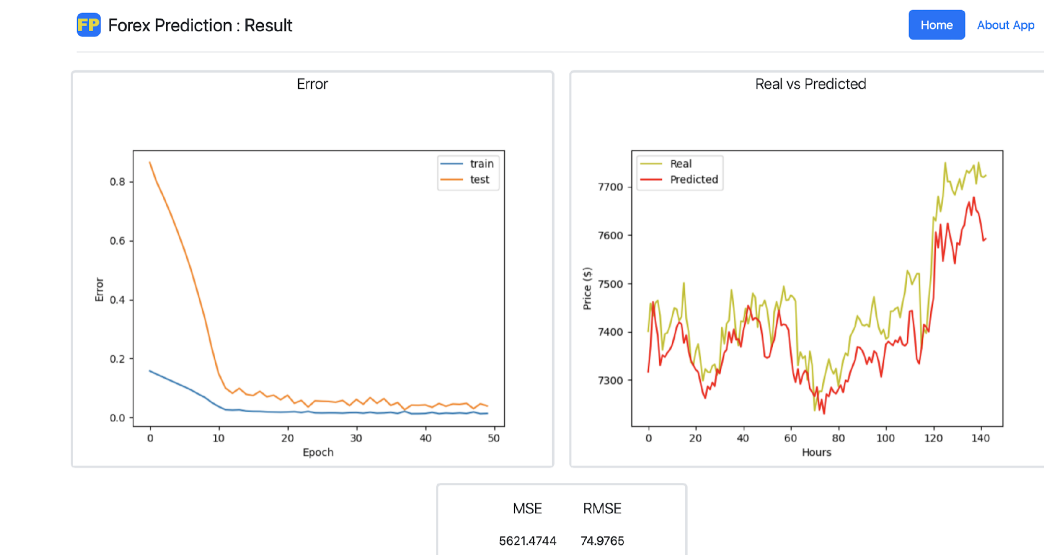


Рисунок 2.6 - Дизайн сторінки результатів

## 2.4 Алгоритм прогнозування курсу валют

Дані від сторонніх ресурсів приходять в форматі ключ-значення.  
Приклад таких об'єктів з Twitter:

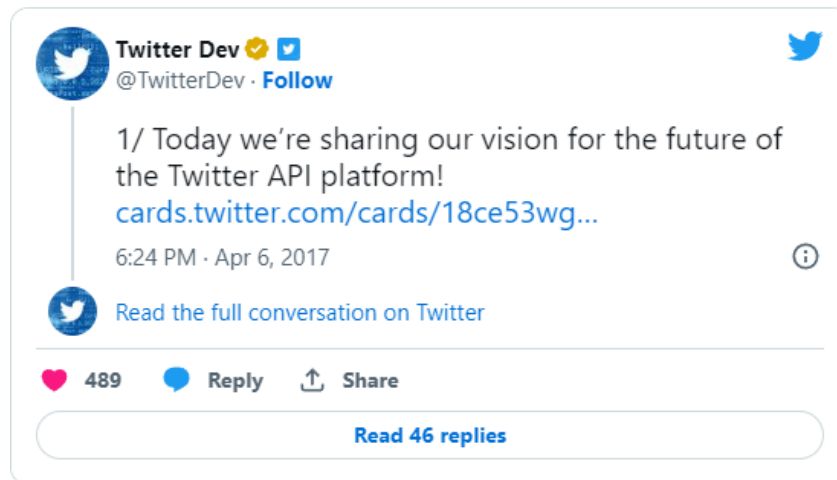


Рисунок 2.7 - Приклад Твіту [11]

```
{
  "created_at": "Thu Apr 06 15:24:15 +0000 2017",
  "id_str": "850006245121695744",
  "text": "1/ Today we're sharing our vision for the future of the Twitter",
  "user": {
    "id": 2244994945,
    "name": "Twitter Dev",
    "screen_name": "TwitterDev",
    "location": "Internet",
    "url": "https://dev.twitter.com/",
    "description": "Your official source for Twitter Platform news, updates & ev
  },
  "place": {
  },
  "entities": {
    "hashtags": [
    ],
    "urls": [
      {
        "url": "https://t.co/XweGngmx1P",
        "unwound": {
          "url": "https://cards.twitter.com/cards/18ce53wgo4h/3xo1c",
          "title": "Building the Future of the Twitter API Platform"
        }
      }
    ],
    "user_mentions": [
    ]
  }
}
```

Рисунок 2.8 - Приклад об'єкту ключ-значення з Twitter API [11]

З усіх цих даних, нас цікавить тільки ключі: `created_at` та `text`:

- `created_at` - дата та час створення твіту;
- `text` - текст твіту.

Приклад об'єкту ключ-значення з біржі:

```
{
  "id": "2",
  "name": "Nasdaq 100",
  "o": "8447.32",
  "c": "8450.54",
  "h": "8480.24",
  "l": "8445.74",
  "ch": "+5.61",
  "cp": "+3.06%",
  "cty": "united-states",
  "t": "1599249599",
  "dateTime": "2019-10-15 12:30:00"
}
```

Рисунок 2.9 - Приклад об'єкту ключ-значення з Stock API [12]

Нас цікавлять ключі:

- o - ціна відкриття години торгів;
- c - ціна закриття години торгів;
- h - максимальна ціна за цю годину;
- l - мінімальна ціна за цю годину;
- dateTime - часова мітка.

Необхідно перетворити дані отримані зі сторонніх ресурсів в:

- 1) набір даних з твітами, з ключовим словом «біткоїн» (рис. 2.10)

[13];

0	2018-07-11 19:35:15.363270	b'tj'	b"Next two weeks prob v boring (climb up to 9k...	0.007273	0.590909
1	2018-07-11 19:35:15.736769	b'Kool_Heart'	b'@Miss_rinola But you\Xe2\X80\X99ve heard abo...	0.000000	0.000000
2	2018-07-11 19:35:15.744769	b'Gary Lang'	b'Duplicate skilled traders automatically with...	0.625000	0.500000
3	2018-07-11 19:35:15.867339	b'Jobs in Fintech'	b'Project Manager - Technical - FinTech - Cent...	0.000000	0.175000
4	2018-07-11 19:35:16.021448	b'ERC20'	b'Coinbase App Downloads Drop, Crypto Hype Fad...	0.333333	0.500000

Рисунок 2.10 - Набір даних з соціальної мережі Twitter

- 2) набір даних з курсом відповідної валюти (рис. 2.11) [14].

Timestamp	Open	High	Low	Close	Volume (BTC)	Volume (Currency)	Weighted Price
2018-07-10 01:00:00	6666.75	6683.90	6635.59	6669.73	281.73	1875693.72	6657.70
2018-07-10 02:00:00	6662.44	6674.60	6647.00	6647.00	174.10	1160103.29	6663.38
2018-07-10 03:00:00	6652.52	6662.82	6621.99	6632.53	231.41	1536936.22	6641.70
2018-07-10 04:00:00	6631.17	6655.48	6625.54	6635.92	120.38	799154.77	6638.52
2018-07-10 05:00:00	6632.81	6651.06	6627.64	6640.57	94.00	624289.31	6641.32

Рисунок 2.11 - Набір даних з показниками курсу валюти

Стовпцями набору даних а) є: dt – часова мітка, name – ім'я автора, text – текст твіту, polarity – полярність тексту [-1; 1], sensitivity – чутливість (значущість) тексту. Отже кожен твіт описується кортежем:

`<dt, name, text, polarity, sensitivity>`.

Стовпцями набору даних б) є: dt – часова мітка, open – ціна відкриття години торгів, high – максимальна ціна за цю годину, low – мінімальна ціна за цю годину, close\_price – ціна закриття години торгів, volume\_btc – об'єм торгів в еквіваленті біткоїну, volume\_dollar – об'єм торгів в еквіваленті долару, weighted\_price – середня ціна. Кортеж має вигляд:

`<dt, open, high, low, close_price, volume_btc, volume_dollar, weighted_price>`.

Оскільки новини з'являються в мережі кожної секунди, а ціну валюти на біржі ми отримуємо погодинно, нам потрібно синхронізувати всі показники. Буде доречно використовувати якийсь принцип зведення до погодинного аналізу, наприклад підрахунок середнього значення емоційних показників. Далі нам треба обрати, які саме показники ми будемо використовувати для подальшої обробки. Одним з найголовніших етапів є нормалізація даних, і нормалізувати дані ми будемо по стовпцям таблиці (тобто по змінним).

Алгоритм складається з таких кроків:

крок 1: попередня обробка даних;

крок 2: застосування фінансових показників;

крок 3: навчання нейронної мережі;

крок 4: передбачення курсу.

**Крок 1.** Попередня обробка даних відбувається у три етапи:

а) спочатку нам необхідно підготувати набір даних з інформацією з відкритих джерел. Для цього нам потрібно видалити стовпці «ім'я автора повідомлення» та «текст повідомлення» і залишити стовпці з часом написання та заздалегідь проведеним емоційним аналізом повідомлень. Як було зазначено вище, нам треба синхронізувати час обох наборів даних, отже округлюємо час вниз, далі рахуємо кількість повідомлень (tweet\_vol) за кожну годину та групуємо ці дані. Отримуємо кортеж вигляду:

`<dt, polarity, sensitivity, tweet_vol>;`

б) готуємо набір даних з різними параметрами валюти. Видаляємо непотрібні стовпці (пункт 2.3) і отримуємо кортеж:

`<dt, close_price, volume_btc>.`

Синхронізуємо час з попередніми датасетом;

с) об'єднуємо набори даних, отримуємо кортеж:

`<dt, polarity, sensitivity, tweet_vol, close_price, volume_btc>.`

Стовпцями об'єднаного набору даних є: часова мітка, полярність тексту, чутливість тексту, кількість повідомлень за кожну годину, ціна закриття години торгів, об'єм торгів в еквіваленті біткоїну.

**Крок 2.** Розраховуємо фінансові показники RSI та SMA для об'єднаного набору даних за формулами:

$$SMA_t = \frac{\sum_{x=1}^5 ClosePrice^{t+x}}{5}, \quad (2.1)$$

$$RS_t = \frac{\sum_{x=1}^5 ClosePrice^{t+x}/5 \text{ if } ClosePrice^{t+x+1} - ClosePrice^{t+x} \geq 0}{\sum_{x=1}^5 ClosePrice^{t+x}/5 \text{ if } ClosePrice^{t+x+1} - ClosePrice^{t+x} < 0}, \quad (2.2)$$

$$RSI_t = 100 - \frac{100}{RS_t}, \quad (2.3)$$

де t – кортеж з відповідною часовою міткою.

Додається два стовпця до об'єднаного набору даних. Отримуємо кортеж:

$\langle dt, \text{polarity}, \text{sensitivity}, \text{tweet\_vol}, \text{close\_price}, \text{volume\_btc}, \text{rsi}, \text{sma} \rangle$ .

Ініціалізуємо змінні  $k$  – кількість годин за якими буде проведено передбачення,  $n$  – кількість змінних в одному кортежі.

Отриманий набір даних розбиваємо на піднабори, де  $k * n$  змінних будуть використані для передбачення.

Отримуємо:

$[\langle dt, \text{polarity}, \text{sensitivity}, \text{tweet\_vol}, \text{close\_price}, \text{volume\_btc}, \text{rsi}, \text{sma} \rangle^{t-k},$   
 $\langle dt, \text{polarity}, \text{sensitivity}, \text{tweet\_vol}, \text{close\_price}, \text{volume\_btc}, \text{rsi}, \text{sma} \rangle^{t-k+1},$

...

$\langle dt, \text{polarity}, \text{sensitivity}, \text{tweet\_vol}, \text{close\_price}, \text{volume\_btc}, \text{rsi}, \text{sma} \rangle^{t-1},$

$\langle dt, \text{polarity}, \text{sensitivity}, \text{tweet\_vol}, \text{close\_price}, \text{volume\_btc}, \text{rsi}, \text{sma} \rangle^t]$ ,

де елементи  $[t-k; t-1]$  будуть використані для передбачення, а елемент  $t$  буде спрогнозовано.

**Крок 3.** Навчаємо нейронну мережу на цих наборах даних. Отримуємо значення помилки.

**Крок 4.** Застосовуємо нейронну мережу для передбачення та отримуємо набір передбачених даних.

## 2.5 Висновки до другого розділу

У другому розділі роботи було представлено систему у вигляді чорної скрині, що допомогло зрозуміти вхідні та вихідні дані, а також які механізми використовує система при роботі і чим вона керується. Було розроблено дерево функцій та описано всі функції, які виконує система.

Було вирішено розробити веб-застосунок, який буде працювати на клієнт-серверній архітектурі для того, щоб не перевантажувати персональний комп'ютер користувача.

Було спроектовано дизайн веб-застосунку з двома сторінками: сторінкою налаштувань та сторінкою результатів.

Було детально описано алгоритм роботи системи, вхідні дані, перетворення та підготовку цих даних.

## РОЗДІЛ 3. РЕАЛІЗАЦІЯ СИСТЕМИ ПРОГНОЗУВАННЯ КУРСУ ВАЛЮТ

### 3.1 Вибір засобів розробки інформаційної системи

Для написання основного модулю інформаційної системи було обрано мову програмування Python [15], оскільки вона дозволяє використовувати багато зручних бібліотек для налаштування та навчання нейронних мереж.

Для написання веб-застосунку було обрано фреймворк Django [16], оскільки він має широкий спектр інструментів для контролювання, налагодження, тестування та адміністрування системи, а також має можливість простого додавання сторонніх модулів.

Для обробки даних було обрано ряд бібліотек:

- 1) `pandas` – для зчитування інформації з файлів формату `.csv`, попередньої обробки даних (обробка дати, видалення непотрібних стовпців, групування інформації, додавання інформації до набору даних, об'єднання наборів, тощо);
- 2) `keras` – для моделювання нейронної мережі та передбачення курсу валют;
- 3) `matplotlib` – для візуалізації результатів;
- 4) `sklearn` – для попередньої обробки даних та оцінки похибки;
- 5) `numpy` та `math` – допоміжні бібліотеки, для полегшення підрахунків.

### 3.2 Опис структури програмного забезпечення системи прогнозування курсу валют

Опис структури програмного модулю є невід'ємною частиною роботи, оскільки вона пов'язує проект системи з безпосередньою реалізацією програми на обраній мові програмування. Оскільки для написання

веб-застосунку було обрано фреймворк Django, на рис. 3.1 наведено структуру програмного забезпечення.

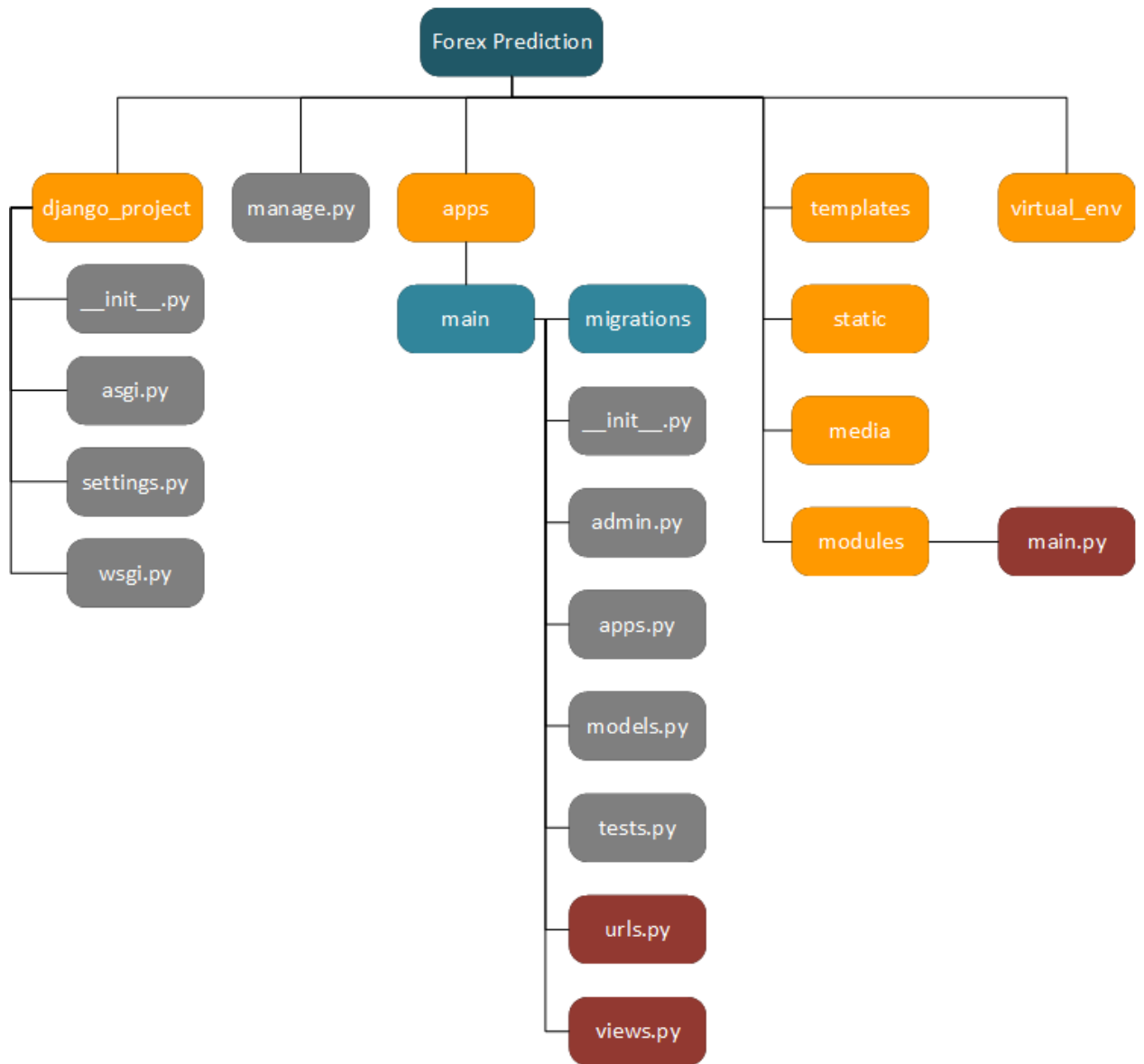


Рисунок 3.1 - Структура програмних модулів

Склад програмного забезпечення, що представлений на рисунку 3.1, включає такі елементи:

- помаранчеві елементи – основні каталоги Django проекту;
- зелені елементи – каталоги застосунків;
- червоні елементи – основні файли Django проекту;
- сірі елементи – додаткові файли Django проекту.

Файл `main.py` у каталозі `modules` це основний файл у якому відбувається налаштування нейронної мережі та передбачення курсу валют.

В каталозі `templates` зберігаються шаблони HTML файлів; в каталозі `static` - файли для генерація HTML сторінок, такі як каскадні таблиці стилів (CSS), Java Script файли та інше; в каталозі `media` - файли з сирими даними для роботи системи.

Специфікація основних файлів програмного забезпечення:

1) За допомогою файлу `manage.py` відбувається ініціалізація і налаштування веб-застосунку, а також викликається файл `urls.py`, який вказує, яка веб-сторінка має бути завантажена першою;

2) Викликається файл `views.py`, який вказує, який шаблон веб-сторінки необхідно завантажити (усі шаблони зберігаються у папці `templates`);

3) Завантажується головна сторінка сайту, де користувачу необхідно ввести параметри роботи програми;

4) Після натискання на кнопку «Обрахувати», всі параметри, введені користувачем, відправляються у файл `main.py`, де відбувається попередня обробка даних і навчання нейронної мережі.

5) Після цього, результати повертаються у файл `views.py`, де викликається завантаження сторінки з результатами.

### 3.3 Керівництво користувача

Заходячи на головну сторінку, користувач має змогу налаштувати параметри попередньої обробки даних, налаштувати параметри навчання моделі, а також запустити процес навчання моделі (рис. 3.2).

The screenshot shows the 'Forex Prediction : Home' interface. It is divided into two main sections, 'БЛОК 1' and 'БЛОК 2'.  
**БЛОК 1** (Data Source Selection): A vertical list of seven items, each with a title and a checked checkbox:  
 - **Close Price**: Ціна закриття торгів  
 - **Polarity**: Настрій тексту повідомлення  
 - **Sensitivity**: Чутливість тексту повідомлення  
 - **Tweet Volume**: Кількість твітів за годину  
 - **BTC volume**: Обсяг торгів  
 - **SMA**: Показник SMA  
 - **RSI**: Показник RSI  
**БЛОК 2** (Training Parameters): A horizontal panel with a slider for 'Train set: 50%' and 'Test set: 50%'. Below the slider are two dropdown menus: 'Epochs' set to 50 and 'Train Hours' set to 3. A blue button labeled 'Обрахувати' is positioned at the bottom left of this panel.

Рисунок 3.2 - Головна сторінка веб-додатку

Нижче наведені можливості при налаштуванні параметрів попередньої обробки даних (блок 1):

- 1) Close Price (Check Box) - можливість додати в набір даних, або прибрати з набору даних ціну закриття торгів;
- 2) Polarity (Check Box) - можливість додати в набір даних, або прибрати з набору даних настрої тексту твіту;
- 3) Sensitivity (Check Box) - можливість додати в набір даних, або прибрати з набору даних чутливість тексту твіту;
- 4) Tweet Volume (Check Box) - можливість додати в набір даних, або прибрати з набору даних кількість твітів за годину;
- 5) BTC volume (Check Box) - можливість додати в набір даних, або прибрати з набору даних обсяг торгів в біткоїні;
- 6) SMA (Check Box) - можливість додати в набір даних, або прибрати з набору даних фінансовий показник SMA;
- 7) RSI (Check Box) - можливість додати в набір даних, або прибрати з набору даних фінансовий показник RSI.

Нижче наведені можливості при налаштуванні параметрів навчання моделі (блок 2):

- 1) Train/Test set (Range) - можливість налаштування відсоткового відношення навчального і тестового набору даних;
- 2) Epochs (Number) - можливість налаштування кількості епох для навчання моделі;
- 3) Train Hours (Number) - можливість налаштування параметру k (див. пункт 2.4).

Послідовність дій, при використанні додатку:

- 1) Налаштувати параметри попередньої обробки даних;
- 2) налаштувати параметри навчання моделі;
- 3) натиснути кнопку «Обрахувати»;
- 4) чекати на результат.

### 3.4 Тестування та аналіз результатів

Вирішивши об'єднати три набори даних ми ризикуємо не отримати покращених результатів. Для полегшення роботи програми необхідно виключити з набору даних ті змінні, які слабо впливають на роботу програми. Виявимо такі змінні за допомогою кореляційного аналізу.

Проведемо дослідження рівня кореляції усіх змінних набору даних:

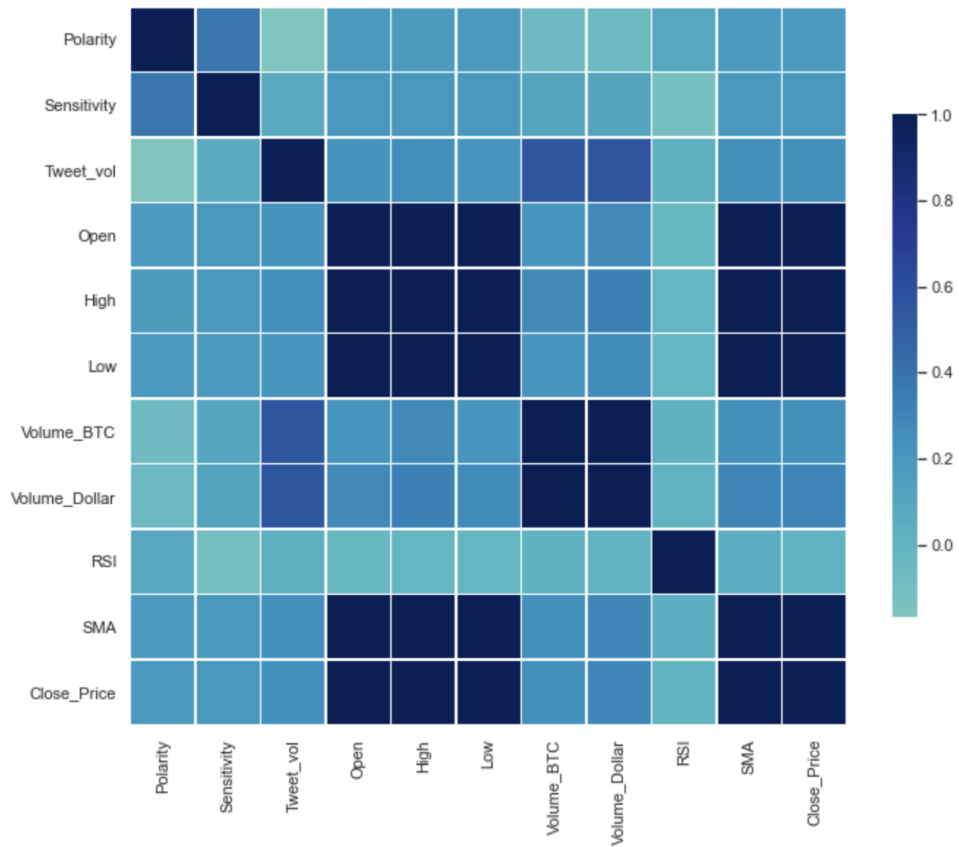


Рисунок 3.3 - Коефіцієнти кореляції для набору даних

З рисунку 3.3 можемо побачити, що показники Tweet\_vol та Volume\_BTC, мають досить високий рівень кореляції, також цікавими є показники Polarity, Sensitivity та RSI. Проведемо додаткові дослідження цих параметрів за допомогою теплових карт.

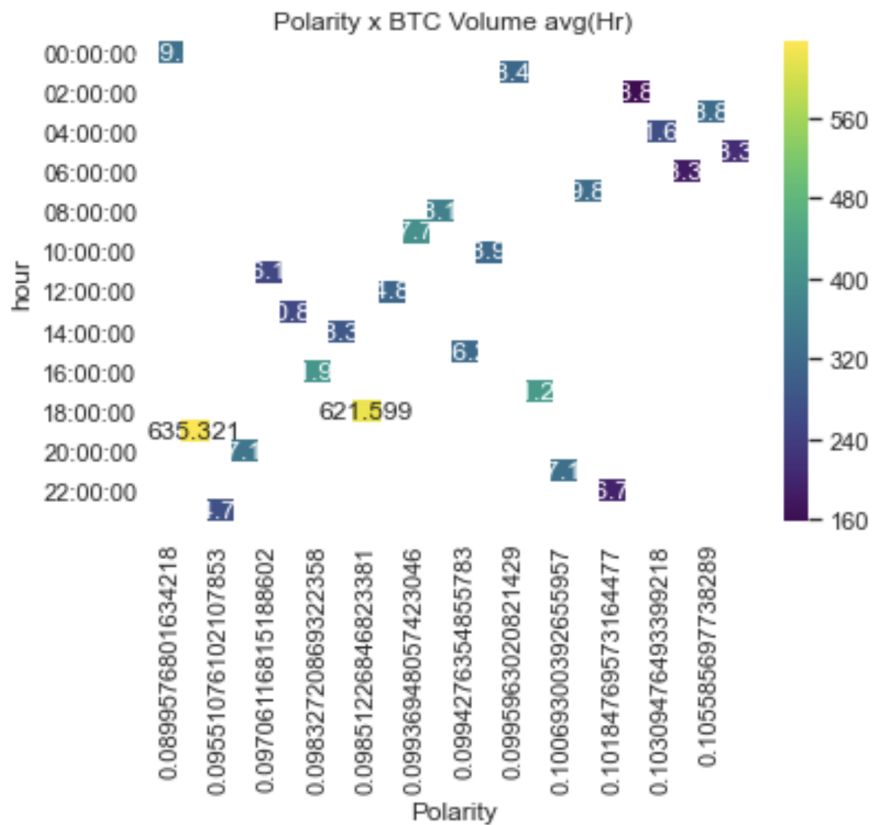


Рисунок 3.4 - Взаємозв'язок параметрів Polarity та BTC\_volume

На рисунку 3.4 ми можемо побачити, що присутні ознаки зворотної пропорційності – чим більше Polarity, тим менше BTC\_volume. При чому, між 18:00 та 20:00 год. спостерігається найбільший об'єм торгівлі BTC.

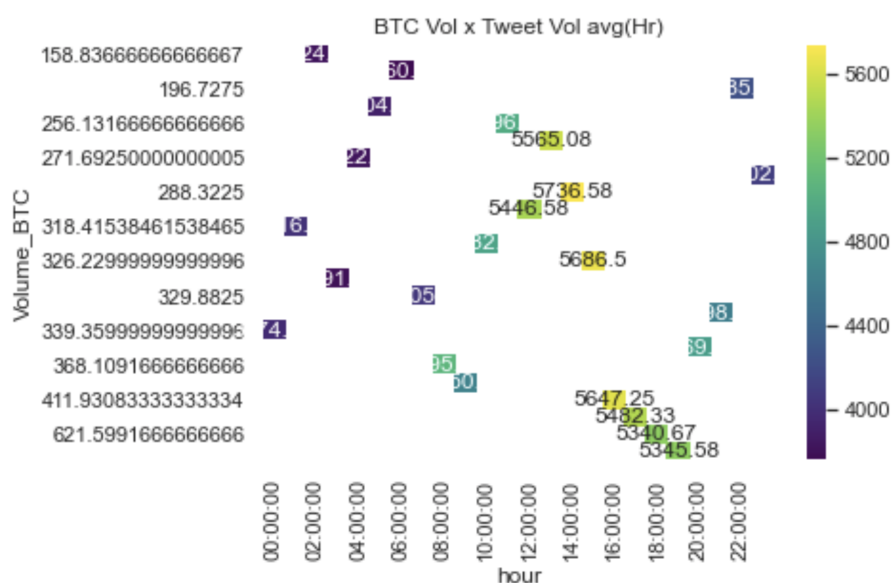


Рисунок 3.5 - Взаємозв'язок параметрів BTC\_volume та Tweet\_volume

На рисунку 3.5 видно, що між параметрами BTC\_volume та Tweet\_volume є зв'язок. Можна сказати, що найбільший об'єм торгів спостерігається між 15:00 та 18:00 год., тобто ближче до закінчення робочого дня.

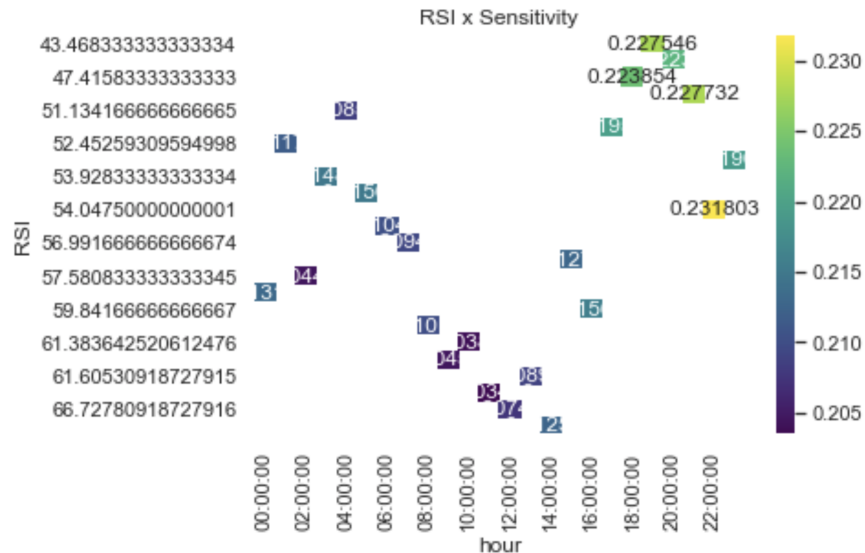


Рисунок 3.6 - Взаємозв'язок між параметрами RSI та Sensitivity

На рисунку 3.6 ми також бачимо зв'язок між параметрами різних наборів даних.

Отже, маючи мінімум 3 пари змінних в 2-х різних наборах даних з гарними показниками кореляції між ними можна припустити, що в нас вийде отримати гарні результати прогнозування курсу валют.

Оскільки параметри Open, High, Low та Dollar\_volume не показали свого зв'язку з іншими параметрами, доцільно видалити їх з набору даних та працювати далі з рештою змінних.

Будемо проводити експерименти з нейронною мережею, подаючи на вхід інформацію за k годин з n змінними та будемо передбачати ці змінні на m годин вперед. Потім відбувається зсув на одну годину вправо, тобто змінні, які були у таблиці під індексом k-1, перемістяться на позицію k-2 (рис. 3.7).

var1(t-3)	var2(t-3)	var3(t-3)	var4(t-3)	var5(t-3)	var6(t-3)	var7(t-3)	var1(t-2)	var2(t-2)	var3(t-2)	...	var5(t-1)	var6(t-1)	var7(t-1)	var1(t)	var2(t)	var3(t)
0.135117	0.589271	0.500135	0.095922	0.045637	0.108712	0.158793	0.111700	0.722717	0.212514	...	0.053607	0.090026	0.075562	0.118921	0.676582	0.345485
0.111700	0.722717	0.212514	0.113362	0.045561	0.104128	0.191207	0.111101	0.649855	0.365349	...	0.013985	0.071296	0.055624	0.114892	0.659195	0.440157
0.111101	0.649855	0.365349	0.111752	0.053607	0.090026	0.075562	0.118921	0.676582	0.345485	...	0.020145	0.053563	0.096830	0.112718	0.557567	0.327165
0.118921	0.676582	0.345485	0.099946	0.013985	0.071296	0.055624	0.114892	0.659195	0.440157	...	0.041566	0.034744	0.091616	0.042799	0.348710	0.169128
0.114892	0.659195	0.440157	0.064663	0.020145	0.053563	0.096830	0.112718	0.557567	0.327165	...	0.454323	0.015003	0.087526	0.019607	0.444672	0.106583

Рисунок 3.7 - Дані, що подаються для навчання в нейронну мережу

Визначивши змінні, що подаються на вхід, необхідно провести ряд експериментів, план для яких відображений в таблиці 3.1

Таблиця 3.1 - План експериментів

№	k	m	Розмір навчальної вибірки	Розмір тестової вибірки	Кількість епох
1	2	1	80%	20%	50
2	3	1	80%	20%	50
3	4	1	80%	20%	50
4	3	1	60%	40%	50
5	3	1	70%	30%	50
6	3	1	60%	40%	100
7	Експеримент без застосування фінансових показників та без застосування набору даних з емоційним аналізом твітів				
8	Експеримент з застосуванням фінансових показників та без застосування набору даних з емоційним аналізом твітів				
9	Експеримент без застосування фінансових показників та з застосуванням набору даних з емоційним аналізом твітів				

Проведемо експерименти з таб. 3.1:

Експеримент 1

Таблиця 3.2 - Параметри експерименту 1

k	Розмір навчальної вибірки	Розмір тестової вибірки	Кількість епох

2	80%	20%	50
---	-----	-----	----

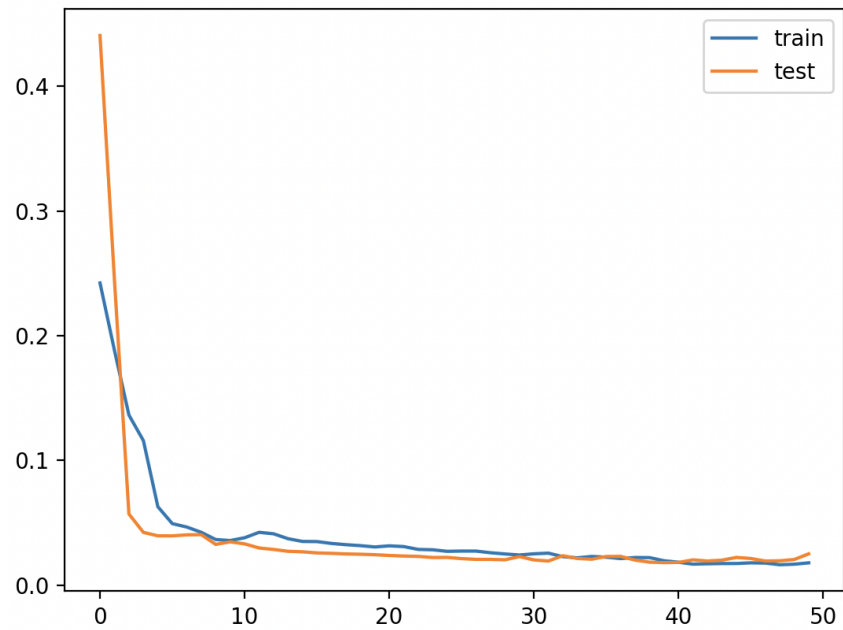


Рисунок 3.8 - Помилка на навчальному та тестовому набору даних

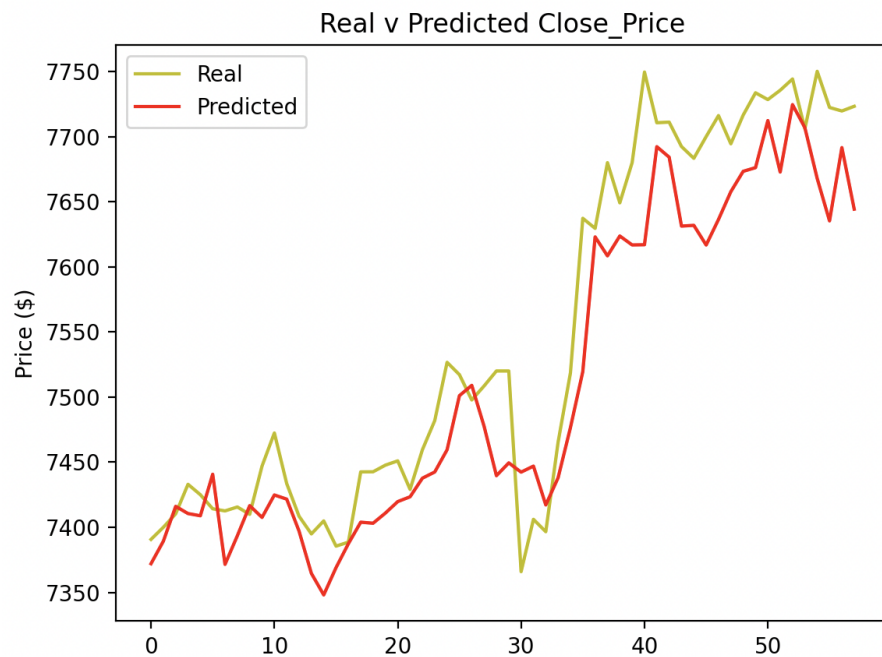


Рисунок 3.9 - Реальна та передбачена ціна

Таблиця 3.3 - Помилки експерименту 1

<b>MSE</b>	<b>RMSE</b>
------------	-------------

2454.452	49.542
----------	--------

## Експеримент 2

Таблиця 3.4 - Параметри експерименту 2

к	Розмір навчальної вибірки	Розмір тестової вибірки	Кількість епох
3	80%	20%	50

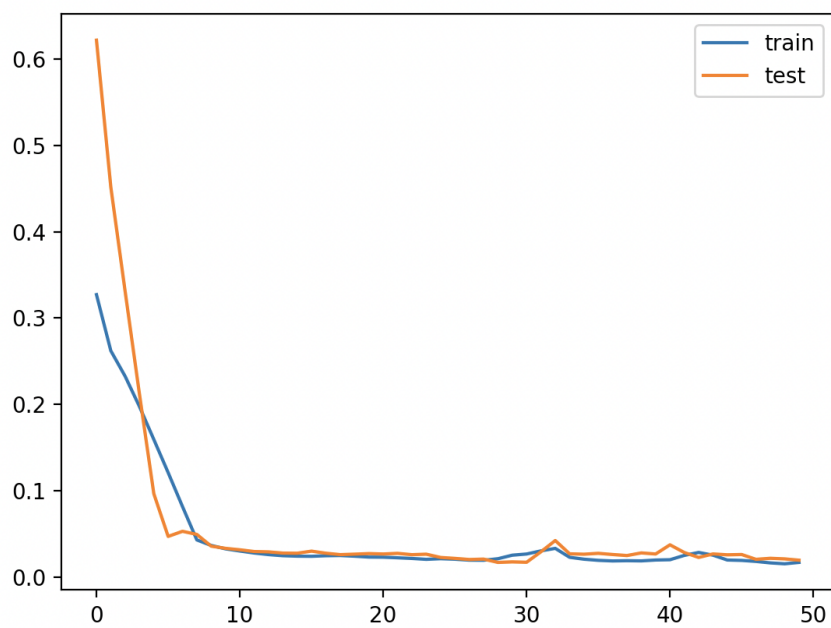


Рисунок 3.10 - Помилка на навчальному та тестовому набору даних

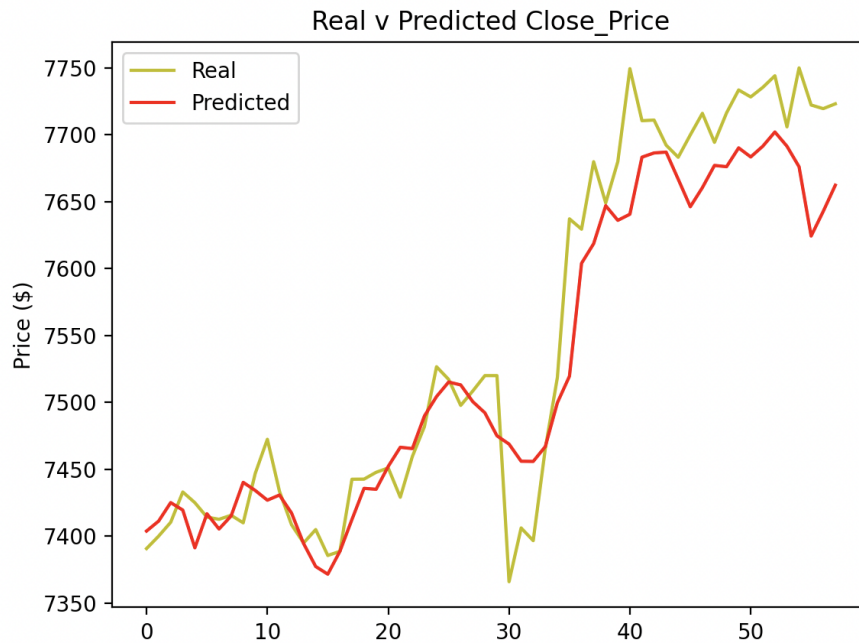


Рисунок 3.11 - Реальна та передбачена ціна

Таблиця 3.5 - Помилки експерименту 2

<b>MSE</b>	<b>RMSE</b>
1780.665	42.198

## Експеримент 3

Таблиця 3.6 - Параметри експерименту 3

<b>k</b>	<b>Розмір навчальної вибірки</b>	<b>Розмір тестової вибірки</b>	<b>Кількість епох</b>
4	80%	20%	50

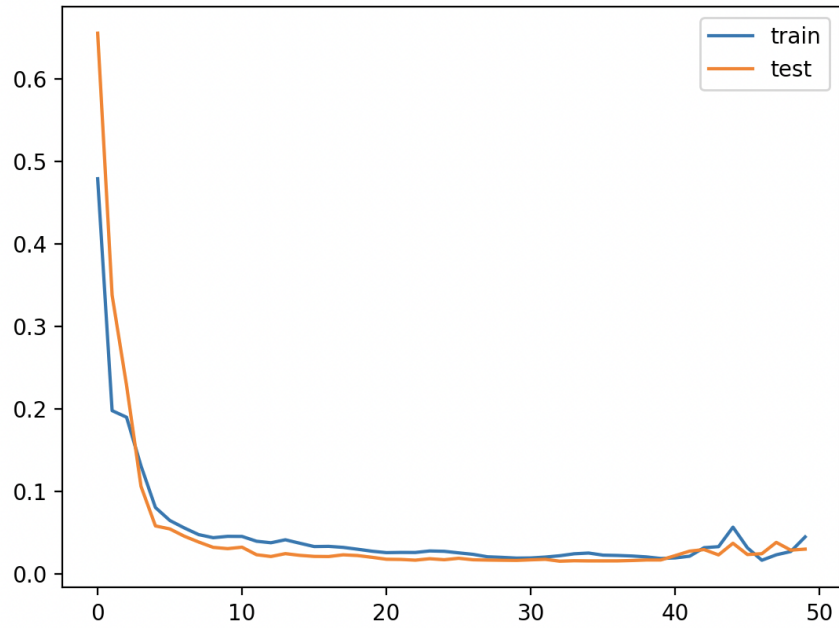


Рисунок 3.12 - Помилка на навчальному та тестовому набору даних

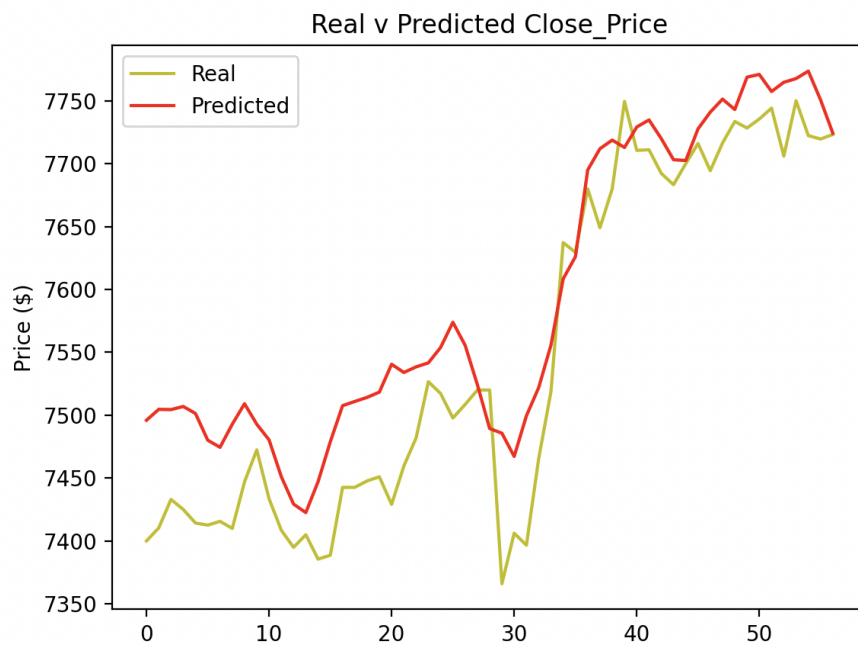


Рисунок 3.13 - Реальна та передбачена ціна

Таблиця 3.7 - Помилки експерименту 3

<b>MSE</b>	<b>RMSE</b>
3272.824	56.328

## Експеримент 4

Таблиця 3.8 - Параметри експерименту 4

<b>k</b>	<b>Розмір навчальної вибірки</b>	<b>Розмір тестової вибірки</b>	<b>Кількість епох</b>
3	60%	40%	50

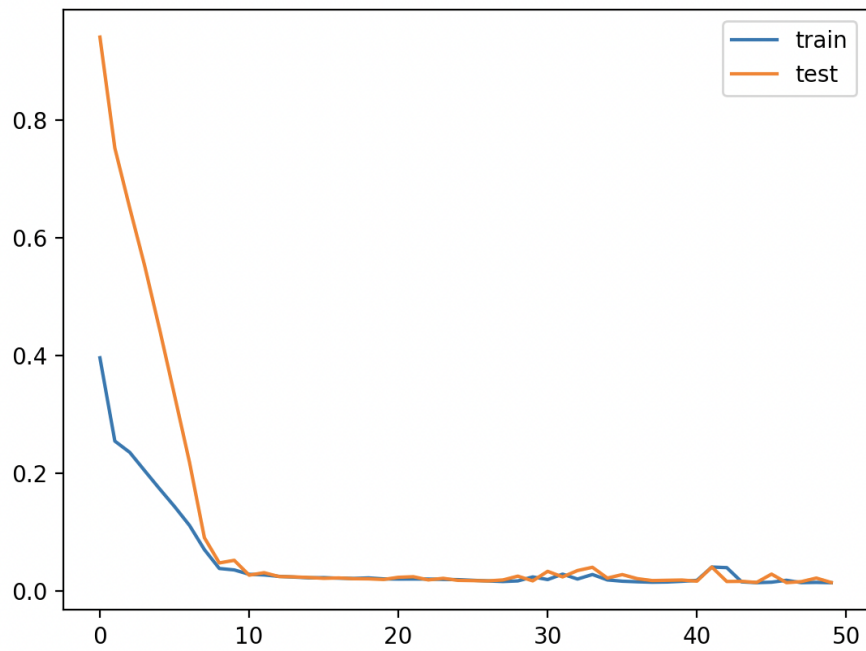


Рисунок 3.14 - Помилка на навчальному та тестовому набору даних

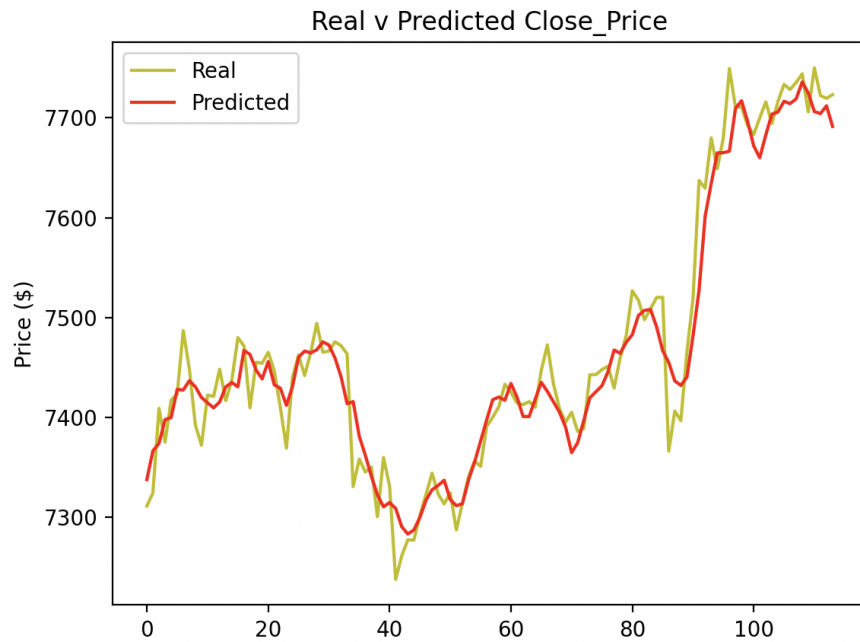


Рисунок 3.15 - Реальна та передбачена ціна

Таблиця 3.9 - Помилки експерименту 4

<b>MSE</b>	<b>RMSE</b>
902.540	30.042

## Експеримент 5

Таблиця 3.10 - Параметри експерименту 5

<b>k</b>	<b>Розмір навчальної вибірки</b>	<b>Розмір тестової вибірки</b>	<b>Кількість епох</b>
3	70%	30%	50

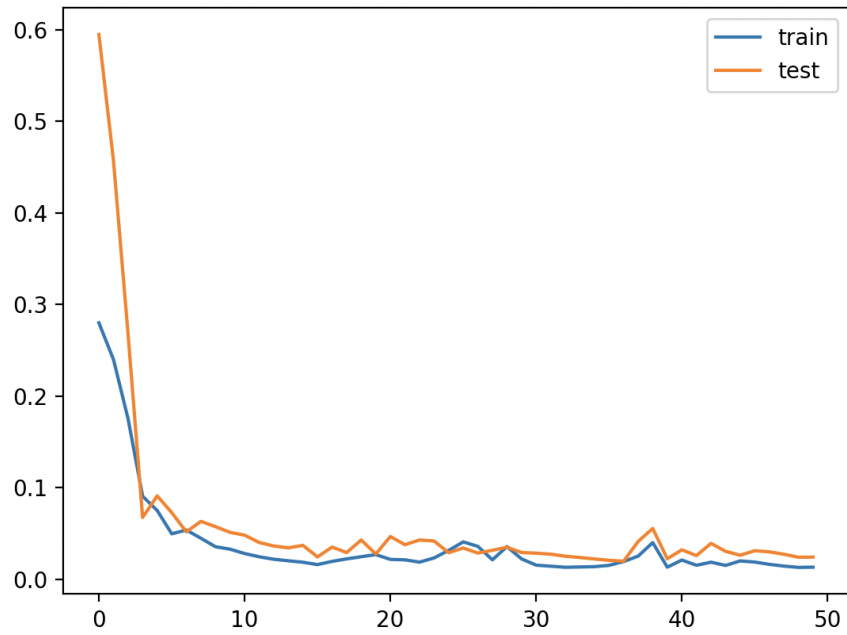


Рисунок 3.16 - Помилка на навчальному та тестовому набору даних

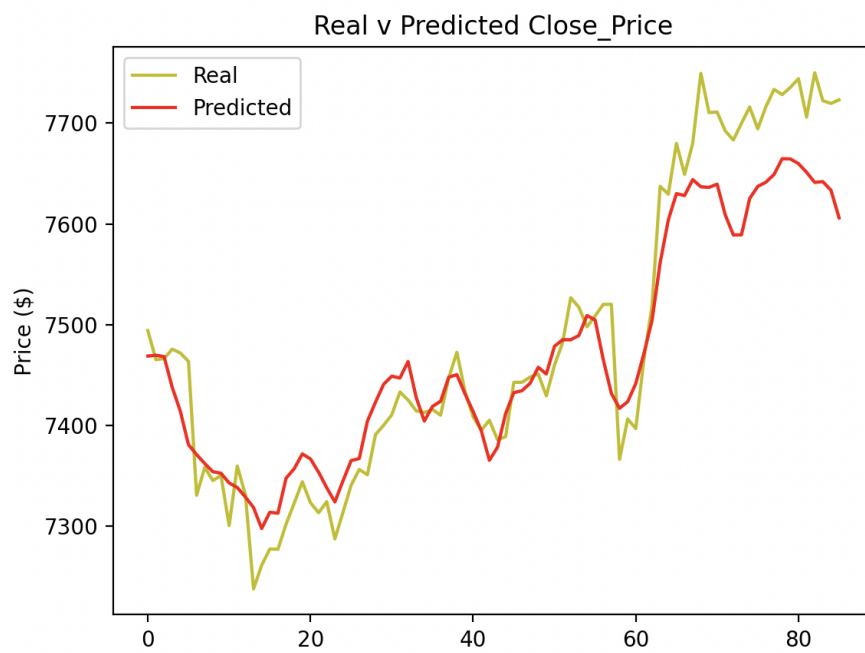


Рисунок 3.17 - Реальна та передбачена ціна

Таблиця 3.11 - Помилки експерименту 5

<b>MSE</b>	<b>RMSE</b>
2491.556	49.915

Експеримент 6

Таблиця 3.12 - Параметри експерименту 6

<b>k</b>	<b>Розмір навчальної вибірки</b>	<b>Розмір тестової вибірки</b>	<b>Кількість епох</b>
3	60%	40%	100

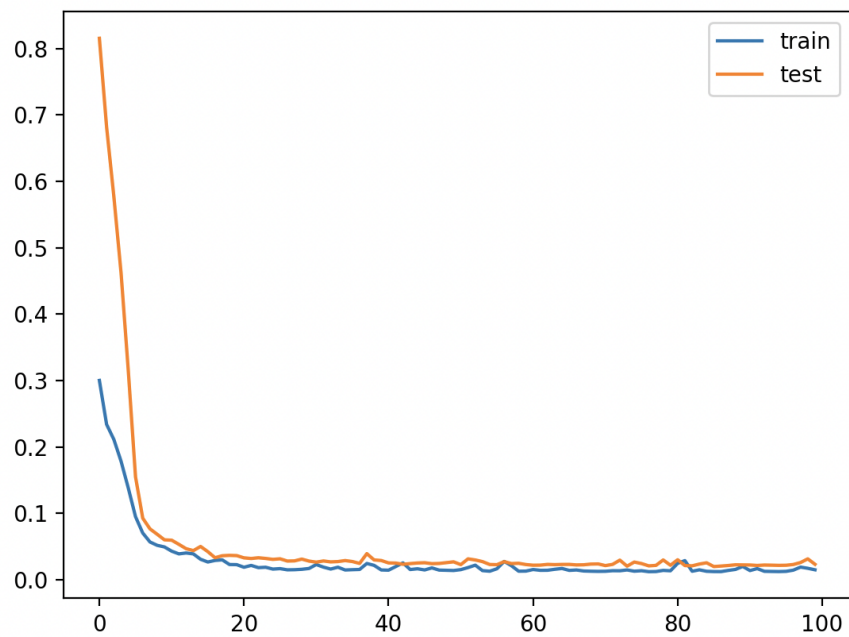


Рисунок 3.18 - Помилка на навчальному та тестовому набору даних

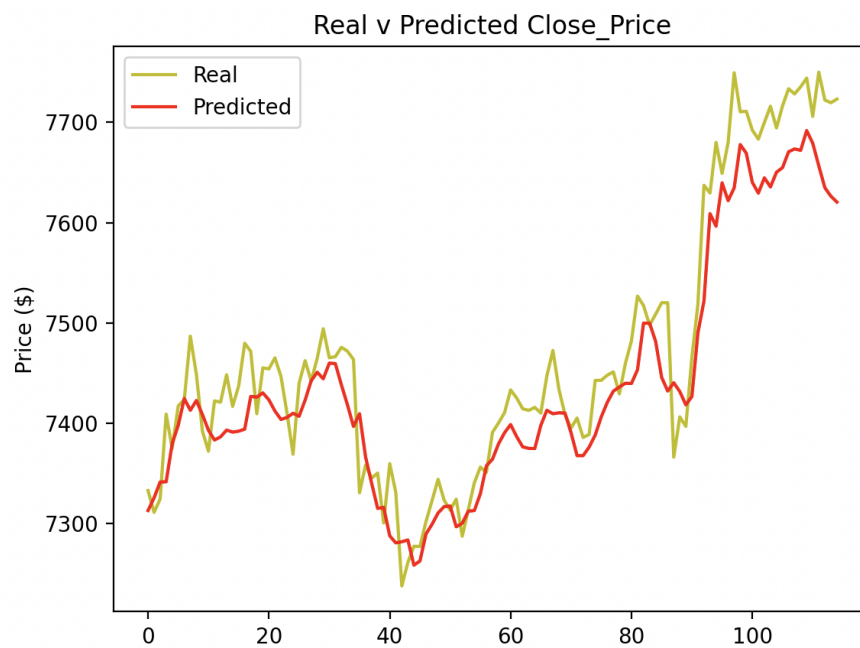


Рисунок 3.19 - Реальна та передбачена ціна

Таблиця 3.13 - Помилки експерименту 6

<b>MSE</b>	<b>RMSE</b>
2087.104	45.685

Експеримент 7 – без фінансових показників та без емоційного аналізу твітів

Таблиця 3.14 - Параметри експерименту 7

<b>k</b>	<b>Розмір навчальної вибірки</b>	<b>Розмір тестової вибірки</b>	<b>Кількість епох</b>
3	60%	40%	50

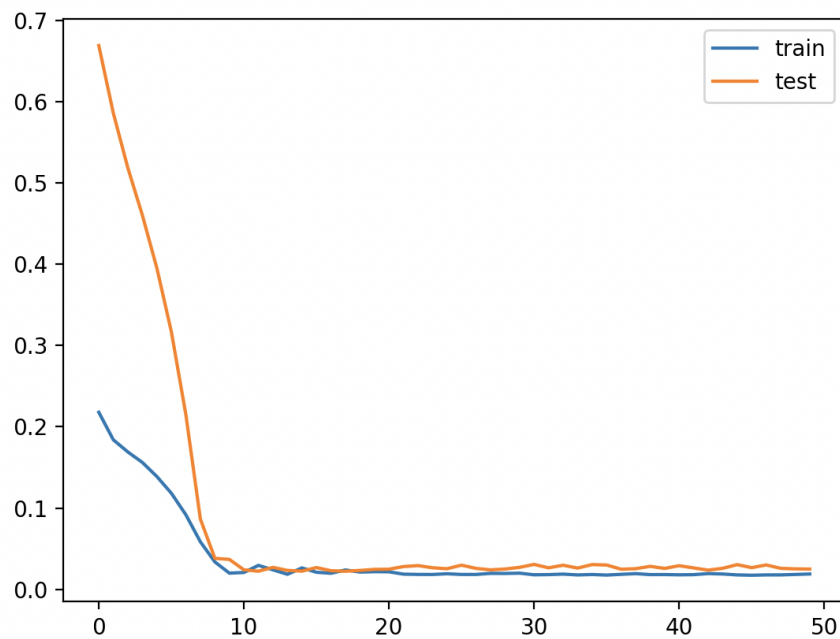


Рисунок 3.20 - Помилка на навчальному та тестовому набору даних

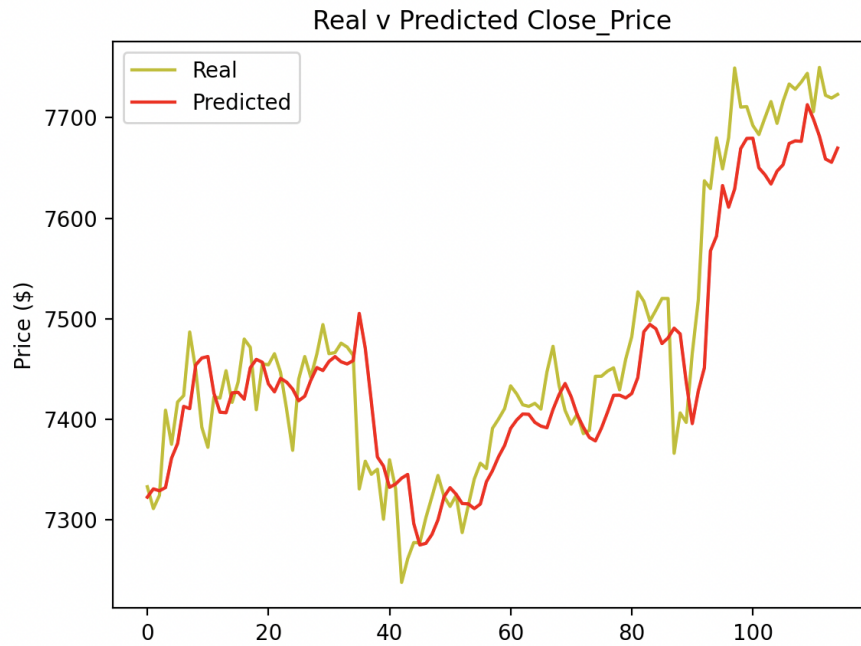


Рисунок 3.21 - Реальна та передбачена ціна

Таблиця 3.15 - Помилки експерименту 7

<b>MSE</b>	<b>RMSE</b>
2769.729	52.628

Експеримент 8 – з фінансовими показниками та без емоційного аналізу твітів

Таблиця 3.16 - Параметри експерименту 8

<b>k</b>	<b>Розмір навчальної вибірки</b>	<b>Розмір тестової вибірки</b>	<b>Кількість епох</b>
3	60%	40%	50

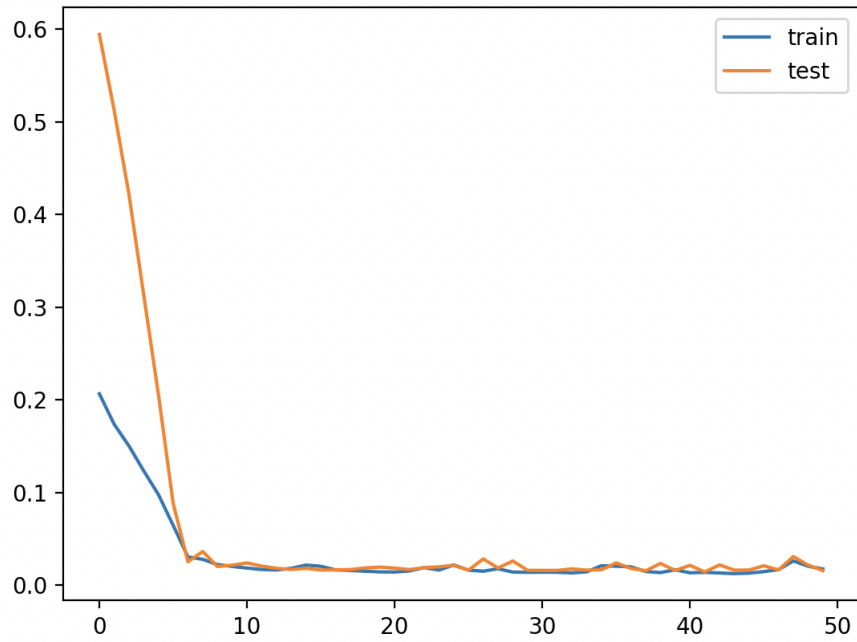


Рисунок 3.22 - Помилка на навчальному та тестовому набору даних

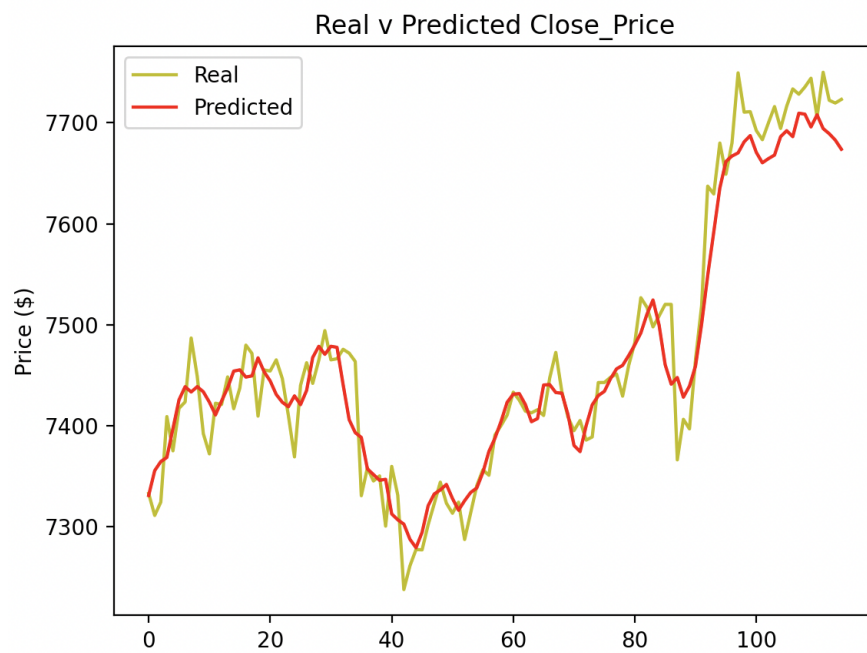


Рисунок 3.23 - Реальна та передбачена ціна

Таблиця 3.17 - Помилки експерименту 8

<b>MSE</b>	<b>RMSE</b>
1063.068	32.605

Експеримент 9 – без фінансових показників та з емоційним аналізом твітів

Таблиця 3.18 - Параметри експерименту 9

<b>k</b>	<b>Розмір навчальної вибірки</b>	<b>Розмір тестової вибірки</b>	<b>Кількість епох</b>
3	60%	40%	50

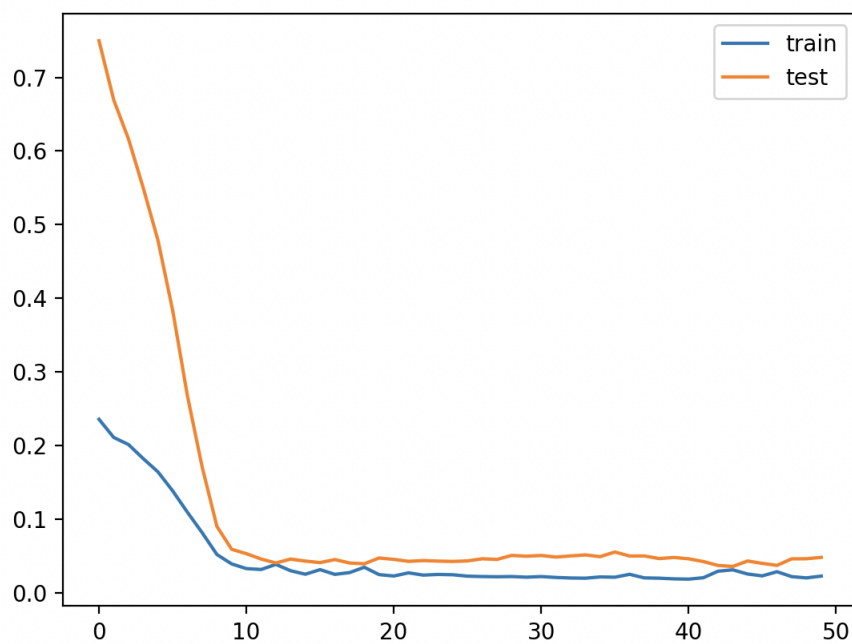


Рисунок 3.24 - Помилка на навчальному та тестовому набору даних

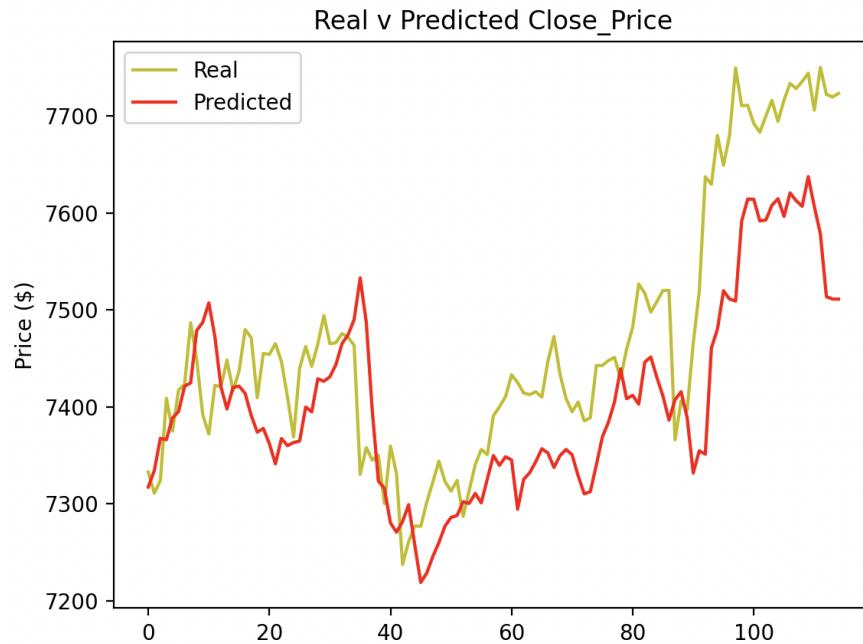


Рисунок 3.25 - Реальна та передбачена ціна

Таблиця 3.19 - Помилки експерименту 9

<b>MSE</b>	<b>RMSE</b>
8937.341	94.538

Для проведення експериментів було обрано 4 змінні:

- 1)  $k$  – кількість годин, за якою формується навчальний набір даних;
- 2) розмір навчальної вибірки;
- 3) розмір тестової вибірки;
- 4) кількість епох.

Результативність тієї чи іншої змінної оцінювалась за допомогою метрики MSE або середньоквадратичного відхилення реальних та передбачених значень ціни закриття торгів.

При збільшенні параметра  $k$ , точність прогнозування спочатку зростає, але при  $k=4$  вона значно падає. Найкращий результат було отримано при  $k=3$ .

При проведенні експериментів над розмірами навчальної та тестової вибірок найліпший результат показало співвідношення 60/40, що насправді дивно, але через неоднорідність та специфічність даних таке можливо.

Необхідні додаткові дослідження на інших наборах даних та інших проміжках часу.

При збільшенні кількості епох до 100, одразу почалося перенавчання нейронної мережі, що спричинило погіршення результатів передбачення.

Порівнюючи результати роботи програми при різних комбінаціях подачі наборів даних на вхід, виявилось, що найкращий результат отримується при застосуванні як фінансових показників, так і набору даних з емоційним аналізом твітів.

### 3.5 Висновки до третього розділу

Провівши ряд досліджень та експериментів, було обрано змінні, які будуть використовуватись під час навчання нейронної мережі, а також отримано результати навчання нейронної мережі.

Незважаючи на випадковість та непередбачуваність показників, результати виявились досить точними, а розбіжності у реальних та передбачуваних значеннях були очікуваними.

## ВИСНОВКИ

Під час виконання випускної кваліфікаційної роботи було: досліджено та доведено актуальність роботи, виявлено проблематику задачі прогнозування курсу валют та сформовано завдання дослідження в рамках даної кваліфікаційної роботи, було проаналізовано існуючі інформаційні системи, проаналізовано підходи до покращення точності прогнозування курсу валют, порівняно різні архітектури нейронних мереж, спроектовано та розроблено веб-застосунок для комунікації з користувачем, запропоновано покращений алгоритм системи, проведено аналіз наборів даних та реалізовано системи прогнозування курсу валют, спираючись на дані з соціальних мереж.

Для розуміння основної структури веб-застосунку було розроблено архітектуру Django проекту та дизайн веб-застосунку за допомогою програми Figma.

Було обрано економічні показники для збільшення ефективності роботи програми, спираючись на потреби в даній задачі.

Були проведені експерименти з різними значеннями параметрів та сформовано висновки стосовно цих експериментів. Вдалося досягнути значного покращення точності програми, змінюючи ці параметри.

Оскільки програма працює з великою кількістю випадкової інформації, більш точних результатів отримати не вдалось, але такого результату цілком достатньо для розуміння загальної тенденції руху курсу. Це може значно полегшити роботу людям у міжнародній фінансовій сфері.

Зважаючи на вищесказане, було сформовано список завдань для покращення існуючої інформаційної системи: додатково дослідити ефективність системи на інших наборах даних та реалізувати функцію налаштування джерел інформації.

## ДЖЕРЕЛА

1. A Review on Recent Advancements in FOREX Currency Prediction [Електронний ресурс] / [S. Islam, E. Hossain, A. Rahman та ін.] // MDPI. – 2020. – Режим доступу до ресурсу: <https://www.mdpi.com/1999-4893/13/8/186/pdf>.
2. Hu Z. A Survey of Forex and Stock Price Prediction Using Deep Learning [Електронний ресурс] / Z. Hu, Y. Zhao, M. Khushi // MDPI. – 2021. – Режим доступу до ресурсу: <https://www.mdpi.com/2571-5577/4/1/9/pdf>.
3. Qi L. Event-Driven LSTM For Forex Price Prediction [Електронний ресурс] / L. Qi, M. Khushi, J. Poon // IEEE Asia-Pacific Conference. – 2022. – Режим доступу до ресурсу: <https://arxiv.org/pdf/2102.01499.pdf>.
4. Sucharita Atha. Do Deep Learning Models and News Headlines Outperform Conventional Prediction Techniques on Forex Data? [Електронний ресурс] / Sucharita Atha, Bharath Kumar Bolla. – 2019. – Режим доступу до ресурсу: <https://arxiv.org/pdf/2205.10743.pdf>.
5. Group, Extract and Aggregate: Summarizing a Large Amount of Finance News for Forex Movement Prediction [Електронний ресурс] / Deli Chen, Shuming Ma, Keiko Harimoto та ін.] // Peking University. – 2017. – Режим доступу до ресурсу: <https://arxiv.org/pdf/1910.05032.pdf>.
6. Hochreiter S. Long Short-Term Memory [Електронний ресурс] / S. Hochreiter, J. Schmidhuber. – 1997. – Режим доступу до ресурсу: <http://www.bioinf.jku.at/publications/older/2604.pdf>.
7. Understanding LSTM Networks [Електронний ресурс] – режим доступу до ресурсу: <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>.
8. Financial Stock Market Forecast using Data Mining Techniques [Електронний ресурс] / K.Senthamarai Kannan, P. Sailapathi Sekar, M. Mohamed Sathik, P. Arumugam // IMECS. – 2010. – Режим доступу до ресурсу: [https://www.researchgate.net/profile/Paliah-Arumugam/publication/44260645\\_Fin](https://www.researchgate.net/profile/Paliah-Arumugam/publication/44260645_Fin)

ancial\_Stock\_Market\_Forecast\_using\_Data\_Mining\_Techniques/links/5779fb7b08ae4645d611f534/Financial-Stock-Market-Forecast-using-Data-Mining-Techniques.pdf.

9. Adam Hayes. Simple Moving Average (SMA) [Електронний ресурс] / Adam Hayes // Investopedia. – 2022. – Режим доступу до ресурсу: <https://www.investopedia.com/terms/s/sma.asp>.

10. Relative Strength Index (RSI) [Електронний ресурс] // StockCharts. – 2021. – Режим доступу до ресурсу: [https://school.stockcharts.com/doku.php?id=technical\\_indicators:relative\\_strength\\_index\\_rsi](https://school.stockcharts.com/doku.php?id=technical_indicators:relative_strength_index_rsi).

11. Twitter API Developer Platform [Електронний ресурс] - Режим доступу: <https://developer.twitter.com/en/docs/twitter-api/v1/data-dictionary/overview>.

12. Stock API Documentation [Електронний ресурс] - Режим доступу: <https://fcsapi.com/document/stock-api>.

13. Датасет з твітами, пов'язаними з біткоїном [Електронний ресурс] – Режим доступу до ресурсу: <https://data.world/mercal/btc-tweets-sentiment>.

14. Датасет з цінами на біткоїн [Електронний ресурс] – Режим доступу до ресурсу: <https://www.kaggle.com/datasets/mczielinski/bitcoin-historical-data>.

15. Документація мови програмування Python [Електронний ресурс] – Режим доступу до ресурсу: <https://www.python.org/doc/>.

16. Документація фреймворку Django [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.djangoproject.com/en/4.2/>.

17. Документація бібліотеки pandas [Електронний ресурс] – Режим доступу до ресурсу: <https://pandas.pydata.org/docs/>.

18. Документація бібліотеки keras [Електронний ресурс] – Режим доступу до ресурсу: <https://keras.io/api/>.

19. Документація бібліотеки matplotlib [Електронний ресурс] – Режим доступу до ресурсу: <https://matplotlib.org/stable/users/index.html>.

20. Документація бібліотеки sklearn [Електронний ресурс] – Режим доступу до ресурсу: [https://scikit-learn.org/stable/user\\_guide.html](https://scikit-learn.org/stable/user_guide.html).
21. Документація бібліотеки numpy [Електронний ресурс] – Режим доступу до ресурсу: <https://numpy.org/doc/>.

## ДОДАТОК

```
import base64
from math import sqrt

import matplotlib
from numpy import concatenate
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_squared_error
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from pandas import DataFrame
from pandas import concat
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import LSTM
from io import BytesIO

np.random.seed(1234675)

def preprocessing():
    # Готуємо датасет твітер для обробки
    notclean = pd.read_csv("main/media/cleanprep.csv", delimiter=',', engine='python',
header=None)
    notclean.columns = ['dt', 'name', 'text', 'polarity', 'sensitivity']

    notclean = notclean.drop(['name', 'text'], axis=1)

    notclean['dt'] = pd.to_datetime(notclean['dt'])

    notclean['DateTime'] = notclean['dt'].dt.floor('h')
    # Групуємо дата сет по годинам
```

```

vdf = notclean.groupby(pd.Grouper(key='dt', freq='H')).size().reset_index(name='tweet_vol')
vdf.index = pd.to_datetime(vdf.index)
vdf = vdf.set_index('dt')
notclean.index = pd.to_datetime(notclean.index)
vdf['tweet_vol'] = vdf['tweet_vol'].astype(float)
df = notclean.groupby('DateTime').agg(lambda x: x.mean())
df['Tweet_vol'] = vdf['tweet_vol']
df = df.drop(df.index[0])
df = df.drop(['dt'], axis=1)
# Готуємо датасет біткоіна
btcDF = pd.read_csv("main/media/btcSave2.csv", engine='python')
    btcDF.columns = ['Timestamp', 'Open', 'High', 'Low', 'Close', 'Volume (BTC)', 'Volume
(Currency)', 'Weighted Price']

btcDF['Timestamp'] = pd.to_datetime(btcDF['Timestamp'])
btcDF = btcDF.set_index(pd.DatetimeIndex(btcDF['Timestamp']))
btcDF = btcDF.drop(['Timestamp'], axis=1)

# З'єднуємо датасети
Final_df = pd.merge(df, btcDF, how='inner', left_index=True, right_index=True)

Final_df['RSI'] = rsi(Final_df['Close'])
Final_df['SMA'] = sma(Final_df['Close'])
Final_df = Final_df[5:]

Final_df[Final_df['RSI'] == np.nan]['RSI'] = Final_df['RSI'].mean()

Final_df['RSI'][-1] = Final_df['RSI'].mean()

RSI = Final_df['RSI'].mean()
for i in range(len(Final_df)):
    if np.isnan(Final_df['RSI'][i]):
        Final_df['RSI'][i] = RSI

Final_df = Final_df.drop(['Weighted Price'], axis=1)

```

```

Final_df.columns = ['Polarity', 'Sensitivity', 'Tweet_vol', 'Open', 'High', 'Low', 'Close_Price',
'Volume_BTC',
                    'Volume_Dollar', 'RSI', 'SMA']
Final_df = Final_df[
    ['Polarity', 'Sensitivity', 'Tweet_vol', 'Open', 'High', 'Low', 'Volume_BTC', 'Volume_Dollar',
'RSI', 'SMA',
    'Close_Price']]
return Final_df

```

# Фінансовий аналіз (Просте рухоме середнє)

```

def sma(df, periods=5):
    lst = []
    for i in range(len(df)):
        if i < periods:
            lst.append(np.nan)
        else:
            lst.append(round(np.mean(df[i:periods + i]), 2))
            #print(df[i:periods + i])

    return lst

```

# Фінансовий аналіз (Індекс відносної сили)

```

def rsi(df, periods=5):
    df = df.diff()
    lst = []

    for i in range(len(df)):
        if i < periods or i == len(df) - 1:
            lst.append(np.nan)
        else:
            avg_gain = (sum([x for x in df[i:periods + i] if x >= 0]) / periods)
            avg_loss = (sum([abs(x) for x in df[i:periods + i] if x <= 0]) / periods)

```

```

if avg_loss == 0:
    lst.append(np.nan)
else:
    rs = avg_gain / avg_loss

    rsi = 100 - (100 / (1 + rs))

    lst.append(round(rsi, 2))
return lst

# Розбиваємо дані на проміжки k + m
def series_to_supervised(data, n_in=1, n_out=1, dropnan=True):
    n_vars = 1 if type(data) is list else data.shape[1]
    df = DataFrame(data)
    cols, names = list(), list()

    for i in range(n_in, 0, -1):
        cols.append(df.shift(i))
        names += [('var%d(t-%d)' % (j + 1, i)) for j in range(n_vars)]

    for i in range(0, n_out):
        cols.append(df.shift(-i))
        if i == 0:
            names += [('var%d(t)' % (j + 1)) for j in range(n_vars)]
        else:
            names += [('var%d(t+%d)' % (j + 1, i)) for j in range(n_vars)]

    agg = concat(cols, axis=1)
    agg.columns = names

    if dropnan:
        agg.dropna(inplace=True)
    return agg

```

```

def processingDataSet(execDict):

    Final_df = preprocessing()

    values = Final_df.values
    groups = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
    i = 1

    cor = Final_df.corr()

    Final_df['time'] = Final_df.index.to_series().apply(lambda x: x.strftime("%X"))

    hour_df = Final_df

    hour_df = hour_df.groupby('time').agg(lambda x: x.mean())

    df = Final_df
    df = df.drop(['Open', 'High', 'Low', 'Volume_Dollar'], axis=1)

    values = df.values
    cols = df.columns.tolist()
    cols = cols[-1:] + cols[:-1]
    df = df[cols]
    df = df[['Close_Price', 'Polarity', 'Sensitivity', 'Tweet_vol', 'Volume_BTC', 'SMA', 'RSI']]

    if execDict['enable_close_price'] == 0:
        df = df.drop(['Close Price'], axis=1)
    if execDict['enable_polarity'] == 0:
        df = df.drop(['Polarity'], axis=1)
    if execDict['enable_sensitivity'] == 0:
        df = df.drop(['Sensitivity'], axis=1)
    if execDict['enable_tweet_vol'] == 0:
        df = df.drop(['Tweet_vol'], axis=1)
    if execDict['enable_volume_BTC'] == 0:

```

```

df = df.drop(['Volume_BTC'], axis=1)
if execDict['enable_SMA'] == 0:
    df = df.drop(['SMA'], axis=1)
if execDict['enable_RSI'] == 0:
    df = df.drop(['RSI'], axis=1)

cols = df.columns.tolist()
scaler = MinMaxScaler(feature_range=(0, 1))
scaled = scaler.fit_transform(df.values)

n_hours = int(execDict['train_hours'])
n_features = len(cols)
n_obs = n_hours * n_features

reframed = series_to_supervised(scaled, n_hours, 1)

values = reframed.values

n_train_hours = int(reframed.size / (n_hours + 1) / n_features * float(execDict['range']) / 100)

train = values[:n_train_hours, :]
test = values[n_train_hours:, :]
return {"train": train, "test": test, "n_hours": n_hours, "n_features": n_features, "n_obs": n_obs,
"scaler": scaler}

def lstm(execDict):
    matplotlib.pyplot.switch_backend('Agg')
    diction = processingDataSet(execDict)

    train = diction["train"]
    test = diction["test"]
    n_hours = diction["n_hours"]
    n_features = diction["n_features"]

```

```

n_obs = diction["n_obs"]
scaler = diction["scaler"]

train_X, train_y = train[:, :n_obs], train[:, -n_features]
test_X, test_y = test[:, :n_obs], test[:, -n_features]

train_X = train_X.reshape((train_X.shape[0], n_hours, n_features))
test_X = test_X.reshape((test_X.shape[0], n_hours, n_features))

model = Sequential()
model.add(LSTM(10, input_shape=(train_X.shape[1], train_X.shape[2])))
model.add(Dense(1))
model.compile(loss='mae', optimizer='adam')

history = model.fit(train_X, train_y, epochs=int(execDict['epoch']), batch_size=6,
validation_data=(test_X, test_y), verbose=2,
shuffle=False, validation_split=0.2)

plt.plot(history.history['loss'], label='train')
plt.plot(history.history['val_loss'], label='test')
plt.legend()
plt.ylabel('Error')
plt.xlabel('Epoch')
#plt.show()

buffer = BytesIO()
plt.savefig(buffer, format='png')
buffer.seek(0)
img_png = buffer.getvalue()
graph1 = base64.b64encode(img_png)
graph1 = graph1.decode('utf-8')
buffer.close()

```

```

yhat = model.predict(test_X)
test_X = test_X.reshape((test_X.shape[0], n_hours * n_features,))

# yhat=yhat.reshape((yhat.shape[0]))

inv_yhat = concatenate((yhat, test_X[:, -n_features + 1:]), axis=1)
inv_yhat = scaler.inverse_transform(inv_yhat)
inv_yhat = inv_yhat[:, 0]

test_y = test_y.reshape((len(test_y), 1))
inv_y = concatenate((test_y, test_X[:, -n_features + 1:]), axis=1)
inv_y = scaler.inverse_transform(inv_y)
inv_y = inv_y[:, 0]

mse = (mean_squared_error(inv_y, inv_yhat))
#print('Test MSE: %.3f % mse)
rmse = sqrt(mean_squared_error(inv_y, inv_yhat))
#print('Test RMSE: %.3f % rmse)
plt.clf()
#plt.title('Real v Predicted Close_Price')
plt.ylabel('Price ($)')
plt.xlabel('Hours')
plt.plot(inv_y, label='Real', color='y')
plt.plot(inv_yhat, label='Predicted', color='r')
plt.legend(loc="upper left")

buffer = BytesIO()
plt.savefig(buffer, format='png')
buffer.seek(0)
img_png = buffer.getvalue()
graph2 = base64.b64encode(img_png)
graph2 = graph2.decode('utf-8')
buffer.close()

percent_errors = abs((inv_y - inv_yhat) * 100 / inv_y)

```

```
return {'graph1': graph1, 'graph2': graph2, 'MSE': format(mse, '.4f'), 'RMSE': format(rmse, '.4f')}
```