

Вінницький національний технічний університет  
Факультет інтелектуальних інформаційних технологій та автоматизації  
Кафедра комп'ютерних систем управління

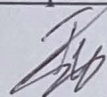
**МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА**

на тему:

**Прогнозування часових рядів з використанням масивів індикаторів у Python**

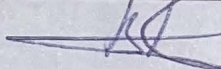
Виконав: студент 2 курсу, групи ЗАКІТ-21м  
спеціальності 151 – Автоматизація та  
комп'ютерно-інтегровані технології

11 грудня



Артур Гунько

Керівник: к.т.н., доц. каф. АІТ



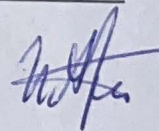
Владислав Кабачій

« 11 » грудня 2022 р.

Опонент: к.т.н., доц. каф. КН

Ігор АРСЕНЮК

« 12 » грудня 2022 р.

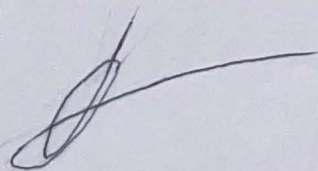


Допущено до захисту

Зав. кафедри КСУ


В'ячеслав КОВТУН

« 14 » грудня 2022



Вінницький національний технічний університет  
Факультет інтелектуальних інформаційних технологій та автоматизації  
Кафедра комп'ютерних систем управління  
Рівень вищої освіти другий (магістерський)  
Галузь знань – 15 – Автоматизація та приладобудування  
Спеціальність – 151 – Автоматизація та комп'ютерно-інтегровані технології  
Освітньо - професійна програма – Інформаційні системи і Інтернет речей

ЗАТВЕРДЖУЮ  
Завідувач кафедри КСУ

 В'ячеслав КОВТУН

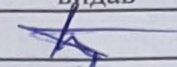
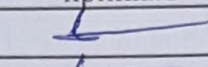
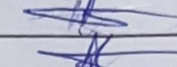
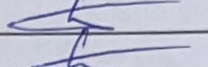
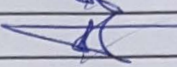
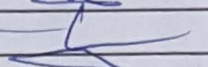
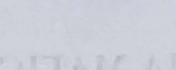
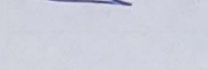
"03" жовтня 2022 року

### ЗАВДАННЯ НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ студенту Гуцько Артуру Васильовичу

1. Тема роботи Прогнозування часових рядів з використанням масивів індикаторів у Python  
керівник роботи Кабачій В.В. к.т.н., доц. каф. АІТ  
затверджені наказом ВНТУ від "14" вересня 2022 року №203
2. Термін подання студентом роботи "12" грудня 2022 року
3. Вихідні дані до роботи: Програмне забезпечення на мові Python для прогнозування часових рядів: вхідні дані представлені у вигляді сформованого дата-сету із технічними індикаторами, які проходять через модель, що на виході видає масив прогнозованих даних у вигляді сигналів продажу чи покупки позицій фінансового часового ряду на майбутній проміжок часу. Можливість задавати різні вхідні параметри, як від обраного періоду вхідних даних, так і терміном прогнозу.
4. Зміст текстової частини: Розгляд існуючих методів для прогнозування часових фінансових рядів, аналіз застосування сучасних інтелектуальних інструментів для прогнозування, створення підходу до вирішення поставленої проблеми, розгляд моделей прогнозування на основі нейронних мереж, розробка процедури для інтеграції Python з торговими терміналами, API сервісами для отримання масиву даних, формування вхідного датасету із застосуванням масивів індикаторів, розробка та тестування ПЗ.
5. Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень):  
Алгоритм роботи застосунку для прогнозування часових рядів з використанням масивів індикаторів, діаграма діяльності застосунку для прогнозування часових рядів з використанням масивів індикаторів, схема моделі LSTM, графік отриманих

даних за допомогою торгового терміналу, як результат роботи скрипта, дані, отримані за допомогою MetaTrader5 бібліотеки, вхідний датасет з використанням технічних індикаторів, шари моделі, прогноз моделі, порівняння прогнозованих даних із реальними, спрогнозовані сигнали купівлі/продажу, презентація.

6. Консультанти розділів роботи


Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	виконання прийняв
1	Кабачій В.В., к.т.н., доц.каф. АІТ		
2	Кабачій В.В., к.т.н., доц.каф. АІТ		
3	Кабачій В.В., к.т.н., доц.каф. АІТ		
4	Кабачій В.В., к.т.н., доц.каф. АІТ		

7. Дата видачі завдання “03” жовтня 2022 року

**КАЛЕНДАРНИЙ ПЛАН**

№ з/п	Назва та зміст етапу	Термін виконання		Примітка
		початок	закінчення	
1	Дослідження предметної області	5.10.2022	31.10.2022	
2	Дослідження існуючих моделей нейронних мереж та вибір потрібної	31.10.2022	14.11.2022	
3	Програмна реалізація додатку	15.11.2022	30.11.2022	
4	Попередній захист	13.12.2022	14.12.2022	
5	Остаточний захист	22.12.2022	22.12.2022	

Студент

  
( підпис )

**Артур ГУНЬКО**

(Ім'я ПРІЗВИЩЕ)

Керівник роботи

  
( підпис )

**Владислав КАБАЧІЙ**

(Ім'я ПРІЗВИЩЕ)

## АНОТАЦІЯ

УДК 025.514

Гулько А.В. Прогнозування часових рядів з використанням масивів індикаторів у Python. Магістерська кваліфікаційна робота за спеціальністю 151 – Автоматизація та комп'ютерно-інтегровані технології, освітня програма – Інформаційні системи та Інтернет речей. Вінниця: ВНТУ, 2022. 124 с.

На укр.мові. Бібліогр.: 52 назв; рис.: 21; табл. 1.

У магістерській кваліфікаційній роботі розроблено програмне забезпечення для прогнозування фінансових часових рядів. У оглядово-аналітичній частині роботи досліджено існуючі методи та підходи для прогнозування часових рядів, проаналізовано існуючі моделі нейронних мереж. Запропоновано власну модель для прогнозування часових рядів з використанням масивів індикаторів у Python. У теоретично-методичній частині обрано модель для розробки. У практичній частині розроблено програмне забезпечення для прогнозування фінансових часових рядів. Ілюстративна частина складається з 20 плакатів із результатами роботи.

Ключові слова: прогнозування, фінансовий часовий ряд, інтелектуальні технології, моделі прогнозування, системи прийняття рішень.

## ABSTRACT

UDC 004.514

Gunko A.V. Time Series Forecasting Using Indicator Arrays in Python. Master's qualification work on specialty 151 – Automation and computer-integrated technologies, educational program – Information systems and the Internet of Things. Vinnytsia: VNTU, 2022. 124 p.

In the Ukrainian language. Bibliography: 52 titles; Fig.: 21; table 1.

In the master's thesis, software for forecasting financial time series was developed. In the review and analytical part of the work, the existing methods and approaches for forecasting time series were investigated, and the existing models of neural networks were analyzed. A proprietary model for forecasting time series using arrays of indicators in Python is proposed. In the theoretical and methodological part, a model for development is selected. In the practical part, software for forecasting financial time series has been developed. The illustrative part consists of 20 posters with the results of the work.

Keywords: forecasting, financial time series, intelligent technologies, forecasting models, decision-making systems.

## Відгук керівника магістерської кваліфікаційної роботи

студента Гунько Артура Васильовича група ЗАКІТ-21м  
на тему: Прогнозування часових рядів з використанням масивів індикаторів у Python.

Актуальність роботи в контексті спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» доведена результатами інформаційного пошуку та аналізу літературних джерел. Підтвердженням актуальності роботи є опубліковані тези на міжнародній науковій конференції КУСС-2022.

Рішення, запропоноване в магістерській роботі, базується на безперервному ряді дій, включаючи пошук інформації про проблему, узагальнення, постановку прикладної задачі для розв'язку, проектування відповідних програмних засобів для вирішення проблеми, її тестування та формулювання висновків. Результати випробувань доводять раціональність та ефективність прийнятого рішення.

Дипломник показав хороший рівень спеціальних знань і «м'яких» навичок. Дипломник продемонстрував вміння: - вирішувати поставлені керівником завдання самостійно, згідно власноруч розробленої схеми заходів; - здійснювати пошук і узагальнення інформації; - комунікативні навички. Доведенням ерудиції та креативності дипломника є вчасно представлена магістерська кваліфікаційна робота.

Основні результати, представлені в роботі отримані дипломником самостійно. Матеріалу роботи властивий високий ступінь оригінальності, що доведено результатами перевірки на наявність запозичень.

Дипломник працював ритмічно, без суттєвих відхилень від затвердженого графіку. Втрати зв'язку з керівником не було.

**Недоліки:** Бажано було поглиблено проаналізувати поставлену проблему. Не всі рисунки достатньо якісні. В тексті дипломної роботи зустрічаються поодинокі вади форматування. Автор навів лише необхідні і достатні діаграми, що описують процес проектування створеної системи. Результати багатоваріантного аналізу бажано було звести в таблицю.

Загалом магістерська кваліфікаційна робота відповідає спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології», заслуговує на оцінку ▲, а її автор заслужовує присудження кваліфікації: ступінь вищої освіти магістр, спеціальність «Автоматизація та комп'ютерно-інтегровані технології», освітня програма «Інтелектуальні системи і Інтернет речей».

**Керівник магістерської кваліфікаційної роботи**

к.т.н., доц. каф. АІТ



Владислав КАБАЧІЙ

студента Гунько Артура Васильовича група ЗАКІТ-21м  
на тему: Прогнозування часових рядів з використанням масивів індикаторів у Python.

Актуальність роботи в контексті спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» доведена результатами інформаційного пошуку та аналізу літературних джерел. Додатковим підтвердженням актуальності роботи слугують опубліковані тези на міжнародній науковій конференції КУСС-2022.

Перший розділ магістерської кваліфікаційної роботи присвячений огляду літературних та інформаційних джерел за обраною темою. Проаналізовано різні підходи для вирішення проблеми, розглянуто декілька моделей прогнозування.

У другому розділі описано які бібліотека використовуватимуться при розробці моделі із застосуванням нейромережі. Створено схему функцій програмного забезпечення.

Третій розділ наводить можливі варіанти інтеграції Python торговими матеріалами. Описано їх переваги та недоліки та обґрунтовано вибір торгового терміналу MetaTrader5. Розроблено інтеграцію з даним терміналом.

Розділ 4 передбачає собою опис розробки застосунку. Представлені дані у вигляді графіків інформують про позитивні результати прогнозування.

Прийняті рішення обґрунтовані результатами огляду літератури, результатами проектування та втілені в функціонуючу програмну систему. Результати її тестування доводять логічність прийнятих рішень.

Експериментальні дослідження досить продумані. Тести охоплюють як функції інтерфейсу створеної системи, так і доводять якість та повноту виконання нею функціонального призначення, обґрунтованого на етапі проектування.

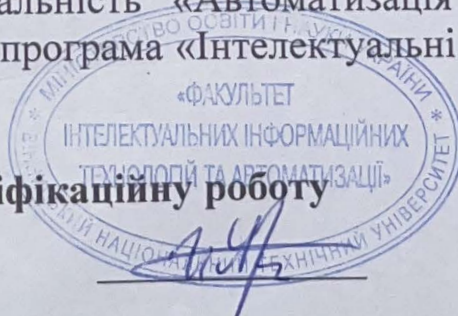
Вміст графічної частина повною мірою репрезентує всі отримані в магістерській кваліфікаційній роботі результати. Якість рисунків в графічній частині прийнятна.

**Недоліки:** На сьогодні окрім Python, існують інші ефективні інструменти для вирішення поставленої задачі, які не були проаналізовані в роботі. Не зрозуміло чому автор використовує саме модель LSTM для прогнозування часових рядів.

Загалом магістерська кваліфікаційна робота відповідає спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології», заслуговує на оцінку В, а її автор заслуговує присудження кваліфікації: ступінь вищої освіти магістр, спеціальність «Автоматизація та комп'ютерно-інтегровані технології», освітня програма «Інтелектуальні системи і Інтернет речей».

Опонент на магістерську кваліфікаційну роботу

к.т.н., доц. каф. КН



Ігор АРСЕНЮК

## ЗМІСТ

ВСТУП .....	10
1 ОГЛЯД СТАНУ ПРОБЛЕМИ ПРИЙНЯТТЯ РІШЕНЬ НА ФІНАНСОВИХ ЧАСОВИХ РЯДАХ .....	13
1.1 Часовий ряд, як об'єкт дослідження .....	13
1.2 Класифікації існуючих методів для прогнозування часових рядів .....	16
1.3 Стратегії та підходи прийняття рішень на фінансових цінових рядах...	22
1.4 Експертна система і автоматизація процесу прийняття рішень.....	24
1.5 Підходи до вирішення проблеми.....	26
2 РОЗРОБКА ПІДХОДІВ ТА МОДЕЛЕЙ ДЛЯ ПРИЙНЯТТЯ РІШЕНЬ НА ЧАСОВИХ РЯДАХ ЗА ВИКОРИСТАННЯМ PYTHON .....	28
2.1 Опис бібліотек .....	28
2.1.1 Pandas .....	28
2.1.2 NumPy .....	29
2.1.3 Scikit-learn .....	29
2.1.5 Keras.....	30
2.1.6 Pandas ТА .....	30
2.1.7 MetaTrader 5 .....	30
2.2 Вибір інструментів для подальшого моделювання .....	30
2.3 Створення моделей, схем використання із застосуванням нейромережі .....	31
2.4 Висновки .....	35
3 РОЗРОБКА ПРОЦЕДУРИ ДЛЯ ІНТЕГРАЦІЇ PYTHON З ТОРГОВИМ ТЕРМІНАЛОМ ДЛЯ ОТРИМАННЯ МАСИВУ ДАНИХ .....	36



	9
3.1 Інтеграція з різними системами постачання фінансових даних.....	36
3.1.1 MetaTrader .....	36
3.1.2 Tradermade API .....	39
3.1.3 YFinance API .....	42
3.2 Обґрунтування вибору інтеграції із MetaTrader 5 та його розробка.....	43
3.3 Висновки .....	44
4 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ .....	45
4.1 Вимоги до програмного та апаратного забезпечення.....	45
4.2. Вибір середовища для розробки .....	45
4.3 Розробка коду додатку.....	46
4.4 Висновки .....	54
ВИСНОВКИ.....	56
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	59
ДОДАТКИ.....	64
Додаток А Протокол перевірки МКР на наявність текстових запозичень .....	65
Додаток Б Технічне завдання.....	66
Додаток В Ілюстративна частина .....	70
Додаток Г Лістинг програми.....	86

## ВСТУП

**Актуальність.** Часові фінансові ряди, які використовуються на фінансових ринках, є частиною звичайних часових рядів. Здатність передбачити їх поведінку важлива для прийняття рішень щодо фінансових часових рядів. Прогнозування змін часових рядів – це процес, який вимагає багатьох наукових знань: економетрики [1], фінансової математики [2], статистики [3], фінансового планування в економіці та торгівлі [4]. Тому прогноз фінансових часових рядів є основою для побудови всієї інвестиційної галузі та її інформаційної системи підтримки. Існує багато методів, які можна використовувати для прогнозування ситуації фінансових часових рядів і прийняття рішень про відкриття угод [5-7]. Кожна з них ґрунтується на певних наукових принципах, аксіомах, припущеннях і включає широкий спектр методів та їх інструментів. Метою кожного з цих аналізів (методів) є прогнозування цінових змін або визначення ринкових умов.

Отже, питання розробки нових підходів задля вирішення задачі прийняття рішень на фінансових часових рядах є актуальними на сьогодні та набули ще більш поширеного значення із застосуванням в різних сферах. Перспективним рішенням цієї проблеми може бути використання інтелектуальної автоматизованої торгової системи, або системи підтримки прийняття рішень.

Сьогодні системи прийняття рішень відкриті до інтегрування інтелектуальних технологій. Інтеграції можуть бути як на примітивному рівні із застосуванням процедурного виду програмування, так і окремим сервісом у вигляді API інтеграцій. Такий підхід дає змогу будувати власні моделі із застосуванням поєднань методів для прогнозування цінових змін. Частка брокерів та трейдерів, що використовують подібні рішення значно зросла після початку наростання тенденцій на ринку криптовалют. Статистика демонструє 40% частки трейдерів на біржі Binance мають у використанні “ботів” для прогнозування, а з розвитком технологій штучного інтелекту, ця частка може досягнути 60% вже до кінця 2023 року.

Розробники активно підтримують створені бібліотеки, що вміщують в собі готові методи прогнозування часових рядів, а ведучі компанії пропонують потужності в оренду, що дає змогу “навчатися” штучному інтелекту значно швидше. Така стратегія із застосуванням даних технологій в першу чергу економить час на прийняття рішень, дає більш точні аналізи та прогнози, одним недоліком є правильна побудова та налаштування моделі задля отримання достовірних результатів.

**Мета роботи** полягає у підвищенні ефективності прийняття рішень на фінансових часових рядах за рахунок інтеграції інтелектуальних технологій Python та застосування масивів технічних індикаторів.

Для досягнення поставленої мети необхідно розв’язати наступні задачі:

1. Провести класифікацію існуючих методів прогнозування часових рядів та розглянути відомі підходи прийняття рішень.
2. Провести аналіз існуючих бібліотек, інструментів інтелектуальних технологій Python для прогнозування часових рядів.
3. На основі аналізу переваг відомих підходів вибрати та обґрунтувати вибір підходу, інтегрувати елементи машинного навчання задля підвищення точності прогнозів.
4. Розробити модель, схему із застосуванням нейромережі.
5. Розробити процедуру для інтеграції з існуючими системами прийняття рішень.
6. Підготувати дані у вигляді датасету.
7. Підрахувати та записати технічні індикатори у вхідний датасет.
8. Провести тестування та дослідження розробленої системи.

**Об’єкт дослідження** є процес прогнозування часових рядів фінансових ринків.

**Предметом дослідження** є прогнозування часових рядів із застосуванням нейромережі та масиву технічних індикаторів.

**Наукова новизна одержаних результатів.**

1. Запропоновано новий підхід до прийняття рішень на фінансових часових рядах, який використовує інтелектуальні технології у поєднанні з набором технічних індикаторів, що дає змогу покращити прогнозовані результати.

2. На основі запропонованого підходу розроблено застосунок, який може використовуватись як система підтримки прийняття рішень.

**Апробація результатів дисертації.** Запропонований підхід обговорювався на XVI науково-технічній конференції факультету комп'ютерних систем та автоматики 15 листопада 2022 р.

**Публікації.** За результатами виконаних досліджень підготовлено та подано до публікації статтю.

## 1 ОГЛЯД СТАНУ ПРОБЛЕМИ ПРИЙНЯТТЯ РІШЕНЬ НА ФІНАНСОВИХ ЧАСОВИХ РЯДАХ

В даному розділі розглянуто поняття часових рядів, наведено їх приклади та обґрунтовано вибір фінансових часових рядів для подальшого дослідження моделей прийняття рішень. Описана класифікація методів прогнозування часовий рядів та обґрунтовано вибір оптимальних методів аналізу для подальшої розробки моделі прийняття рішень на фінансових часових рядах.

### 1.1 Часовий ряд, як об'єкт дослідження

Ідентифікація (процесів, систем) або побудова моделей на основі спостережень посідає одне з головних місць у сучасній теорії управління та прийняття рішень у різних галузях: техніці, економіці, економетриці тощо. Розвиток різних процесів, особливо соціально-економічних, можна визначити в конкретний момент. Як правило, фіксуються лише найважливіші особливості, що визначають перехід процесу з одного стану в інший. Найбільш ефективними математичними моделями, які можна використовувати для прогнозування розвитку процесу, є ті, які використовують часові ряди [8].

Будь-який часовий ряд визначається двома обов'язковими компонентами: значенням ряду або вимірювання та моментом часу, коли це значення було зафіксовано тим чи іншим способом. Одним із типових представників часових рядів є фінансовий ряд цін (рис. 1.1). Ряд фінансових цін є результатом фінансових ринків. Фінансові ринки бувають різними:

- за терміном обігу фінансових активів розрізняють ринок грошей та ринок капіталів;

- за умовами обертання фінансових активів розрізняють первинний і вторинний ринок;
- за організаційними формами: біржовий та позабіржовий;
- за видами фінансових активів, що обертаються на ринку розрізняють товарний, валютний та фондовий ринок.

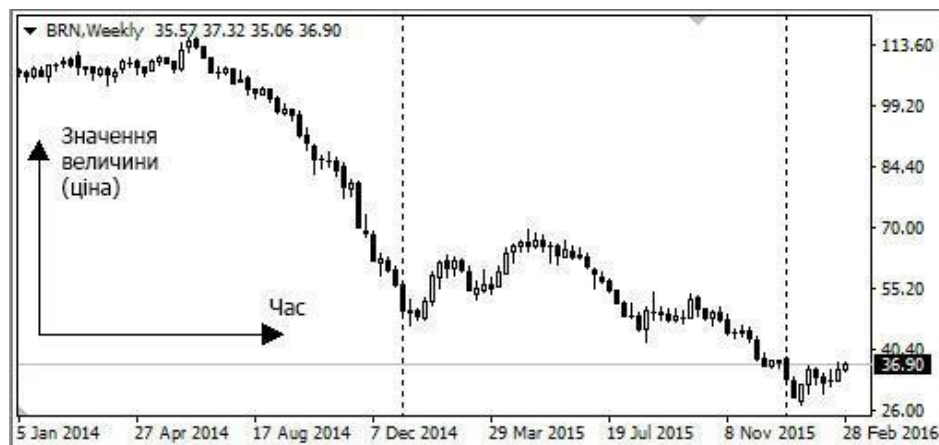


Рисунок 1.1 – Зразок фінансового цінового ряду: ціна нафти за 2014-2016 роки

При прогнозуванні цінових рядів розглянуті фондові, валютні та товарні ринки [9]. Оскільки дані про час для цих ринків є загальнодоступними, вони підходять для вирішення завдань прогнозування.

Часові ряди, що відображають історію ринку, характеризуються тим, що їхні характеристики (ціни, обсяги, індикатори тощо) формуються кількома компонентами: повільними не регулярними частинами (тенденціями), повільними сезонними частинами, циклічними або мінливими компонентами (швидкі нерегулярні малі зміни) і випадковими компонентами (описується деяким типом випадкового процесу). Важливою особливістю часових рядів з економічними властивостями є наявність періодичності зі змінними періодичними значеннями [10].

Прогнозування є одним із найскладніших, але водночас найактуальніших і популярних завдань аналізу даних. Складність процесу прогнозування пов'язана з необхідністю аналізу та оцінки великих масивів

даних, ускладненням методів, появою нових підходів до процесів прогнозування, застосування нових технологій тощо [11]. Тому сучасний стан розвитку методів прогнозування тісно пов'язаний з розвитком інформаційних технологій. Так звані прогнозні інформаційні системи, які відображають цей зв'язок у рамках економетрики, фінансової математики та статистики, проявляються в широкому діапазоні наукових застосувань, а також у таких сферах, як виробництво, фінансове планування в економіці та торгівлі. Сьогодні вони є частиною процесу управління складними системами та системами прийняття управлінських рішень і використовуються аналітиками, серед іншого, для оцінки ризиків фінансових інвестицій. Слід відзначити і наукову складову у застосуванні прогнозування часових рядів. У світі на різних рівнях зустрічаються аналітичні дані щодо руху астероїдів, як приклад, за певні проміжки часу та їх подальші прогнози траєкторій, розрахунки складних фізичних процесів із обов'язковою константою часу та у медицині, що допомагає виявляти на ранніх етапах схильність до інфаркту у пацієнтів. Останні моделі використовують за основу інтелектуальні технології та вже широко застосовуються у медичних центрах.

Основними компонентами системи прогнозування часових рядів є база даних з постійно поповнюваними ретроспективними даними, комплексна модель прогнозування та методи оцінки його якості, які згруповані відповідно до постановки завдання прогнозування. Функціонування такої системи здійснюється в режимі діалогу з особою, яка приймає рішення.

Від ефективного управління процесом прогнозування залежить якість будь-якої інформаційної системи прогнозування. Під процесом прогнозування розуміють аналіз та оцінку тенденції розвитку процесу чи явища на основі певних наукових методів з використанням наявної інформації про процес даного процесу чи явища в минулому, тобто ретроспективних даних.

Вхідні дані, які складаються з ціни Close, дати, визначеної часовим інтервалом, представленні у вигляді дата-сетів, в які входять також масиви показників індикаторів. Крім того відомі моменти за певний проміжок часу, в

які було б бажано прийняти те чи інше рішення. Необхідно створити підхід, базуючись на методах прогнозування часових рядів і за використанням бібліотек Python, та навчити нейронну мережу прогнозувати майбутні значення на основі даних в реальному часі: цінових вимірів, значень сигналів.

## 1.2 Класифікації існуючих методів для прогнозування часових рядів

Методи прогнозування розділяють на три класи [12]:

а) прогнозування на основі суб'єктивних суджень (оцінок), інтуїції, глибоких знань у конкретній області та іншої інформації, що має відношення до процесу прогнозування;

б) методи прогнозування, засновані на часових рядах з використанням однієї змінної, тобто на основі авторегресії, авторегресії з ковзним середнім і авторегресії з ковзним середнім плюс модель тенденції;

в) методи прогнозування на основі часових рядів з використанням кількох змінних [13].

Цілком логічно, що в цілому метод прогнозування може поєднувати в собі 2-3 перерахованих вище методів. У фаховій літературі описано багато методів прогнозування, заснованих на використанні часових рядів. Найпоширенішими з них є методи, які розглядають параметри в групах, регресійний аналіз (авторегресія, авторегресія з ковзним середнім, авторегресія з інтегрованим ковзним середнім, лінійна та нелінійна множинна регресія, квантильна регресія, регресійні дерева), нейронні мережі, байєсовські мережі, нечіткі множини, нечіткі нейронні мережі тощо [14].

Завдання прогнозування можуть бути поставлені по-різному в залежності від рівня прийняття рішень і конкретного завдання управління або контролю. Прогнози можуть стосуватися наступних компонентів процесу:

- детерміновані тенденції як індикатори довгострокових змін процесу;



- стохастичні (нерегулярні) тренди як індикатори коротко- та середньострокових змін;
- короткострокові зміни, тобто прогнозна волатильність (зміщення), накладена на тренд;
- сезонний вплив;
- приріст (швидкість) зміни процесу, що визначається різницею першого порядку;
- дисперсія або стандартне відхилення, як міра дисперсії процесу (наприклад, волатильність, часто використовується як міра інвестиційного ризику або міра якості виробництва);
- поєднання елементів процесу.

В залежності до того, які складові процесу необхідно прогнозувати, виникає завдання створення математичної, ймовірнісної (ймовірнісно-статистичної) або логічної моделі, що повинна забезпечити високу якість прогнозу при заданих показниках.

Найбільш поширеними методами аналізу фінансових ринків є [15]:

- а) фундаментальний аналіз;
- б) технічний аналіз;
- в) аналіз за свічками, як компонент технічного аналізу (японські свічки);
- г) фрактальний аналіз – аналогічний свічковому або як окремий метод аналізу фінансових ринків;
- д) хвильовий аналіз Еліота.

Фундаментальний аналіз вивчає рух цін під впливом макроекономічних чинників. Школа фундаментального аналізу ринку виникла з розвитком прикладної економічної науки. За свою основу вона взяла знання про макроекономічне життя суспільства і її вплив на динаміку цін конкретних товарів [16]. Основоположниками західної теорії фундаментального аналізу прийнято вважати Бенджаміна Грема і Девіда Додда, які у 1934 році в США опублікували книгу «Аналіз цінних паперів». У цій книзі вперше було введено поняття «фундаментальний аналіз» і присвоїло йому визначення як

інструменту для передбачення майбутніх біржових цін на акції. При цьому предметом дослідження фундаментального аналізу, на думку авторів, є «фінансові показники, доходи і дивіденди компанії, а також стан навколишнього економіки» [17].

Пізніші наукові видання дають визначення фундаментального аналізу як процесу «дослідження стану економіки, галузі та фінансового положення компанії з метою визначення ринкової вартості акцій компанії» [18].

Очевидно, що фундаментальний аналіз зводиться до вивчення макроекономічних новин і індикаторів [19].

Тому, фундаментальний аналіз включає дослідження та аналіз загальних економічних тенденцій товарів, цінних паперів і валютного ринків, встановлених і прихованих взаємозв'язків факторів, що впливають на їх розвиток.

Методи фундаментального аналізу не підходять для дослідження, тому що, використовуючи фундаментальний аналіз, неможливо створити математичну модель, алгоритм, та на їх основі запрограмувати експертну систему, яка здатна буде приймати рішення.

Аналіз патернів на японських свічкових графіках історично виник у Японії та є одним із найперших методів аналізу ринку. «Свічка» складається з тіла і двох «тіней»: вгору і вниз, і відображає чотири ціни. Тіло свічки являє собою прямокутник, отриманий різницею цін відкриття і закриття. Тіло біле, коли закриття вище відкриття, і чорне, коли закриття нижче відкриття. Верхня та нижня тіні представляють найвищі та найнижчі ціни відповідно за певний період часу. Велику увагу дослідженню японських свічкових комбінацій приділено в роботі Стіва Нінсона [20-21].

Метод фрактального аналізу часових рядів є одним із напрямів аналізу фінансового ринку, запропонованим Б. Мандельбротом і Р. Хадсоном [22-23], розробленим Е. Пітерсом і Е. Федером [24-25], як нелінійний динамічний Компонент дискретних методів для дослідження нелінійностей у динаміці часових рядів, включаючи фінансову динаміку. Загалом, фрактальний аналіз

спрощено до використання фракталів (сегментів), які утворюються в комбінації з п'яти або більше японських свічок або барів (барів), причому середня свічка (бар) має найбільше максимальне або мінімальне значення.

Хвильовий аналіз Еліота передбачає, що ринок рухається у дві великі фази [26]. Відповідно до хвильового принципу, рух цін на будь-якому ринку буде виражатися у вигляді п'яти хвиль в одному напрямку (тренд) і трьох хвиль в протилежному напрямку (відкат або корекція). Існує різниця між імпульсними хвилями та хвилями корекції. Імпульсний створює тренд (бичачий або ведмежий), коригувальний (відкат) рухається проти поточного тренду.

Технічний аналіз – це набір графічних, математичних, статистичних та інших методів у поєднанні зі знаннями людської (популярної) психології, які допомагають передбачити майбутні рухи цін на фінансових і товарних ринках [27-28].

Інструмент походить від теорії Чарльза Доу, яка з'явилася на початку 20 століття. Виходячи з цього, технічний аналіз вивчає характер руху ціни (ціни враховують всю відому інформацію), підтвердження та розбіжності, обсяг, який відображає зміни ціни, підтримку чи опір. Очевидно, що формування відомого індексу Dow Jones Industrial Index стало прямим продовженням застосування технічного аналізу [29].

Як і в будь-якій іншій науковій дисципліні, методи технічного аналізу базуються на наборі аксіом [30].

Аксіома 1 – Динаміка ринку враховує все. Тобто будь-які чинники, які впливають на ціни — політичні, економічні, психологічні — попередньо враховуються і відображаються в залежності ціни від часу. Тому вивчення графіка цін є обов'язковою умовою прогнозування.

Аксіома 2 – Ціна завжди рухається в одному напрямку. Періоди зростання і спаду чергуються, тому в кожному періоді існує домінуючий тренд, який триває до тих пір, поки ринок не порушиться в протилежному напрямку.

Аксиома 3 – Історія повторюється. Це тому, що психологія людей принципово не змінилася. Припущення полягає в тому, що якщо певні типи аналізу працювали в минулому, вони будуть дуже успішними в майбутньому, оскільки робота базується на стабільності людської психології. Іншими словами, з точки зору технічного аналізу, розуміння майбутнього полягає у вивченні минулого.

Загально прийнято, що методи технічного аналізу поділяються на методи графічного аналізу та аналітичні методи, як показано на рисунку 1.2. Графічні методи включають побудову графіків, виявлення закономірностей і тенденцій, які домінують на ринку. Діаграма – це певний символ, який відображає певну інформацію про ринок, характеризує основний чи допоміжний індикатор ринку та використовується для визначення напрямку руху ціни на ринку. Загалом аналітичними вважаються графічні методи, включаючи: класичні графіки, хвилі Еліота, метод Хана, графіки на діаграмах «японської свічки» та всі інші методи, що включають прості математичні розрахунки. Назва групи «Графіка» вказує на обов'язкову необхідність графічного зображення чого-небудь - будь то дані про ціни, або допоміжні індикатори, або інші аналітично розраховані інструменти. Отже, висновок полягає в тому, що майже всі методи, які використовуються в аналізі, є графічними, оскільки метрики, обчислені певним чином, обов'язково мають власні графічні правила.

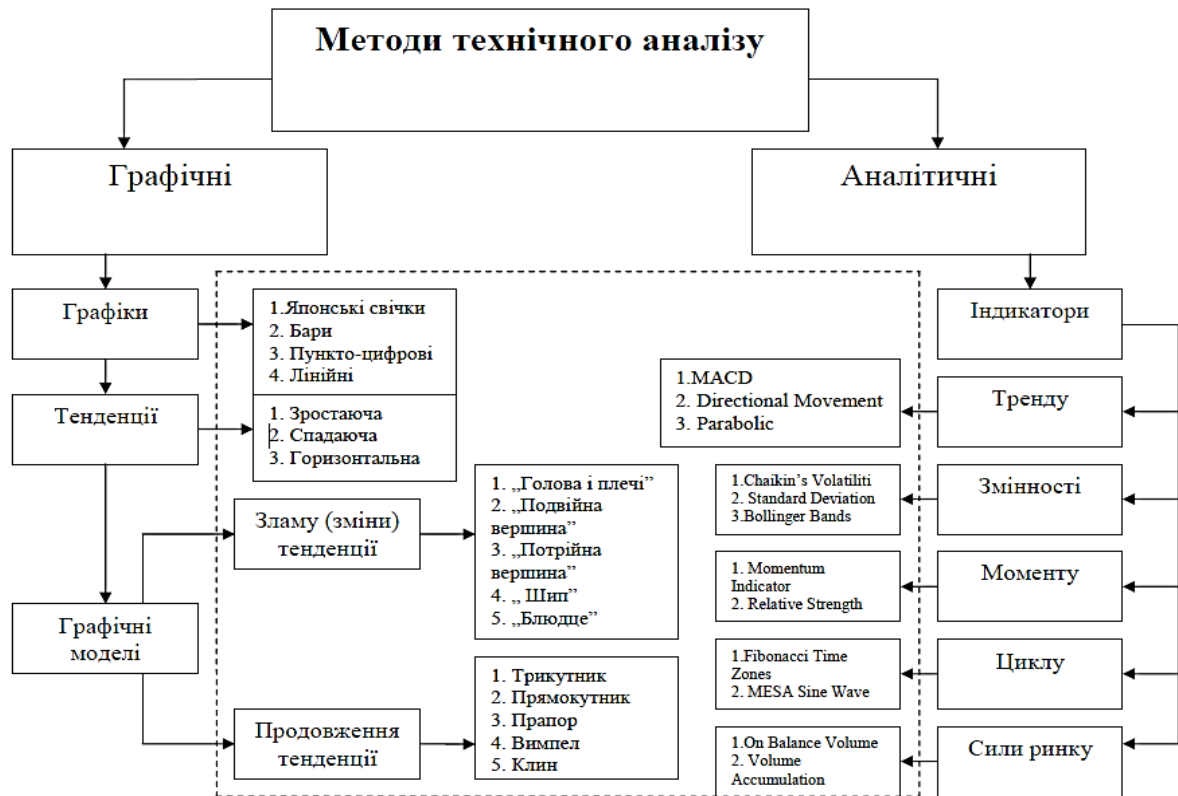


Рисунок 1.2 – Схема методів технічного аналізу [31]

Основним інструментом математичного підходу є індикатор, який є результатом математичного розрахунку на основі показника ціни або обсягу.

Індикаторів розроблено багато, але умовно їх можна розділити на кілька категорій:

а) Статистичні індикатори - фільтрує статистичний шум у цінових рухах. Ці показники будуються з використанням стандартних методів математичної статистики та теорії ймовірностей. Вони включають: Moving Average - ковзне середнє, Bollinger Band Limit - ліміт Боллінджера тощо;

б) Осцилятори є більш складними індикаторами, ніж статистичні індикатори. Вони є справжньою спробою знайти загальний індикатор, який відокремлює основні рухи від коливань цін. Основне призначення осцилятора - визначити момент розвороту тренда. Наприклад, RSI - Relative Strength Index, Moving Average Convergence/Divergence (MACD), стохастичний осцилятор [32].

### 1.3 Стратегії та підходи прийняття рішень на фінансових цінових рядах

Сьогодні існує безліч торгових стратегій, які передбачають цінові діапазони. Деякі з цих стратегій є основою механічних торгових систем або систем підтримки прийняття рішень [33-35], які використовують методи технічного та фундаментального аналізу.

Виходячи з існуючих аналітичних методів, усі стратегії можна розділити на різні групи, які орієнтовані на стилі торгівлі. Згідно з цими групами, кожен трейдер може вибрати свою психологічно сумісну торгову стратегію. Традиційно торгові стратегії можна розділити на такі типи: торгові стратегії, що працюють відповідно до трендів; торгові стратегії, які базуються на пробиванні рівнів підтримки та опору; торгові стратегії, які ефективні в межах цінових коридорів.

Перша стратегія зосереджена на висхідному або спадному тренді часового ряду (тренда), відстежуючи стабільність тренду та його тривалість. Тенденція, з якої походить назва системи, пов'язана з певним періодом часу. Прикладом такої стратегії є стратегія з використанням технічних індикаторів – ковзних середніх [36]. Перевагами цих стратегій є широкий діапазон прибутків і відносно невеликий ризик.

Друга торгова стратегія зазвичай не фокусується на напрямку тренда. Головне правило тут полягає в тому, що прорив рівнів підтримки та опору є основним сигналом про наявність тренда. Перевага цієї стратегії полягає в тому, що ви можете отримати сильніші сигнали формування тренду та можливі обмеження на його припинення. Недоліком є те, що існують помилкові прориви рівнів підтримки та опору, особливо на дуже волатильних ринках [37].

У третій торговій стратегії головне виявити формування флету (без тренду). Угода з однієї сторони квартири на іншу. Осцилятори з областями «перепроданості» і «перекупленості» широко використовуються в таких торгових системах, як Стохастичний осцилятор. Перевагою цієї стратегії є те,

що вона дозволяє отримувати прибуток, тоді як більшість торгових систем втрачає гроші. Недоліком є те, що основний діапазон руху ціни вздовж тренду пропускається, і кошти втрачаються, коли бічний тренд виходить.

За тривалістю торгівлі торгові стратегії поділяються на: внутрішньоденні, середньострокові та довгострокові торгові стратегії. У внутрішньоденній стратегії тривалість торгівлі не перевищує одного дня. Переваги їх полягають у тому, що є менший ризик, менша просадка, і вам не потрібен великий депозит, щоб почати. Недоліком є закриття угоди в кінці дня і, сподіваюся, робота в тому ж напрямку завтра.

Середньострокові стратегії передбачають торгівлю тривалістю від днів до тижнів. У таких угодах фундаментальний аналіз важливий, хоча він не завжди відіграє ключову роль. Перевага перед внутрішньоденними стратегіями полягає в тому, що робота виконується в більш спокійному режимі, з меншою залежністю від моніторів. Недоліком є те, що ви повинні стежити за новинами, оскільки успіх стратегії може бути зведений до нуля за кілька днів. Крім того, середній прибуток таких стратегій нижчий, ніж у короткострокових стратегій.

У довгостроковій торговій стратегії угоди тривають від тижнів до місяців. Перевагою цієї стратегії є час для аналізу та прийняття рішень. Недоліком порівняно з іншими є великий початковий депозит і велика просадка (втрата депозиту), необхідна під час роботи.

Інші стратегії включають управління капіталом [38]. У такій стратегії обсяг кожної угоди визначається за певними правилами. Існують стратегії для фіксованого обсягу на позицію, що визначається як певний відсоток від початкового депозиту, і стратегії для розрахунку нового обсягу на позицію. Прикладом останнього є стратегія, що включає використання підходу Мартингейла до управління капіталом.

Суть цього методу полягає в збільшенні розміру наступних угод у міру збільшення збитків (зменшення торгового балансу), тобто після програшних угод подвоєння розміру лота для покриття збитків. Теорія цього методу

базується на тому факті, що втрати рано чи пізно закінчаться, тому запис слід робити після тривалого періоду однакового результату, щоб отримати протилежний результат. Згодом або смуга програшів закінчується, або баланс рахунку закінчується.

Стратегія мартингейла виглядає привабливою для деяких трейдерів. Але ви повинні мати на увазі, що головна проблема цієї конкретної системи полягає в тому, що іноді угода може повністю скинути торговий рахунок ще до отримання будь-якого прибутку. Однак, незважаючи на такі суттєві недоліки, цей метод і методи на його основі вважаються одними з найефективніших, оскільки дозволяють створити практично беззбиткову торгову систему.

#### 1.4 Експертна система і автоматизація процесу прийняття рішень

Торгова експертна система – це спеціальна комп'ютерна програма, яка відстежує цінові діапазони і на основі заздалегідь вбудованих в програму алгоритмів дає поради користувачам купити або продати актив в певний момент або виконати дію самостійно – купити або продати актив (рис. 1.5).

Експертна система має такі структури: база даних (необов'язкова), база знань, механізм логічного висновку (розв'язувач) та інтерфейс користувача. Основою експертної системи є база знань, яка структурована так, щоб спростити процес прийняття рішень. У сфері штучного інтелекту термін «знання» стосується інформації, яка потрібна програмі для її розумного функціонування.

Характерною рисою сучасних експертних систем для операційних експериментів є їх дуже вузька спеціалізація. Ці системи використовуються для автоматизації процесів прийняття рішень на фінансових ринках, або для визначення структурної формули певного класу органічних сполук, або для пошуку найкращої конфігурації для конкретної комп'ютерної системи.



Вузька спеціалізація експериментальних експертних систем впливає з прагнення зменшити обсяг експертних знань, вбудованих у систему, щоб спростити завдання створення цих знань і збереження їх у пам'яті.

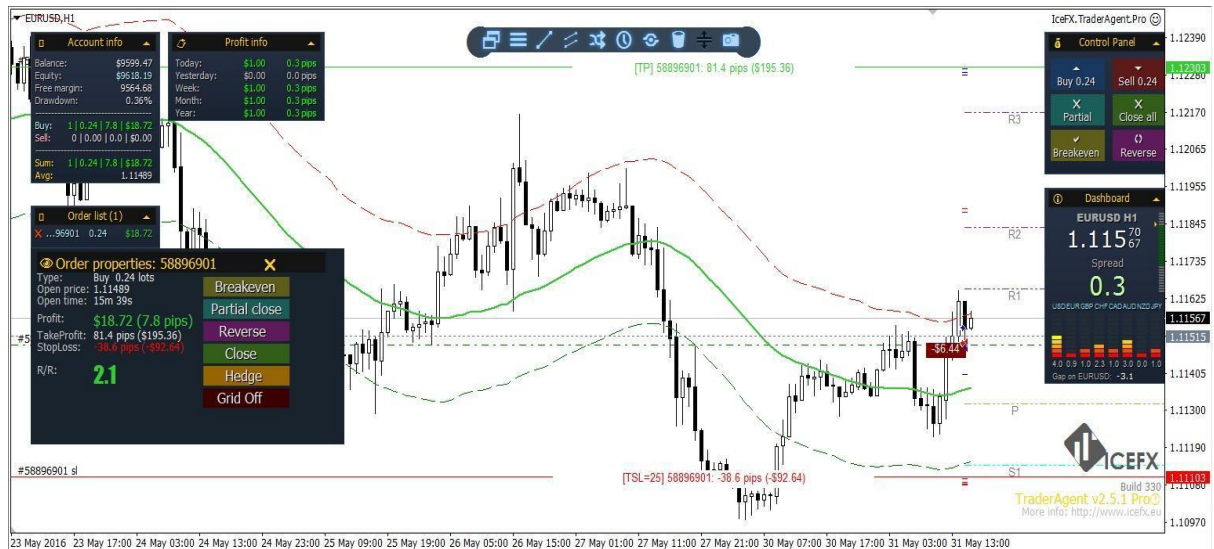


Рисунок 1.3 – Приклад роботи експертної системи

Створення торгової експертної системи проходить у сім основних етапів:

- 1) формулювання торгової стратегії;
- 2) розробка моделі прийняття рішень;
- 3) тестування торгової системи;
- 4) оптимізація торгової системи;
- 5) використання системи в торгівлі;
- 6) визначення ефективності роботи та порівняння з результатами тестування;
- 7) удосконалення даної торгової стратегії.

Експертні системи мають певні переваги при використанні. Зокрема експертні системи:

- 1) Приймаючи рішення, обов'язково виключають факти людської помилки;
- 2) Алгоритми, прописані експертами, є цілеспрямованими та працюють 24/7 для забезпечення повного та точного входу та виходу з угод;

- 3) Перевищує людські можливості при аналізі великої кількості даних;
- 4) Не робити поспішних висновків та ігнорувати певні етапи висновку;
- 5) Забезпечує розмовний режим роботи;
- 6) Надати можливість обґрунтувати рішення та відтворити його шлях;
- 7) Дозволяє самостійно налаштувати систему шляхом зміни коду.

Але навіть найкращі існуючі експертні системи мають певні обмеження порівняно з людьми-експертами, які зводяться до наступного:

- 1) Експерт – це штучний інтелект, який не може мислити про процес прийняття рішень настільки широко і різноманітно, як людина. Експерт приймає рішення виключно на основі закладених у нього математичних інструментів;
- 2) Програма може вийти з ладу. Крім того, він був створений кимось, хто міг робити помилки під час творчої фази;
- 3) Експертні системи часто не можуть отримати якісно нові знання, не передбачені в процесі розробки;
- 4) Нездатність швидко реагувати на події та новини.

Використання експертних систем у прогнозуванні цінових рядів має багато переваг і підвищує ефективність прийняття рішень у фінансових часових рядах.

### 1.5 Підходи до вирішення проблеми

Для вирішення проблеми запропоновано такий алгоритм дій:

- інтеграція з торговим терміналом MT5 для отримання масиву даних: пакет MetaTrader використовується для зручного та швидкого отримання біржової інформації в Python. Отримані таким чином дані можна використовувати для статистичних обчислень і машинного навчання.
- прогін даних, технічних індикаторів (запропоновані MT5 або створені за допомогою Python), формування дата сету для моделі та нейронної

мережі[39]: дані потрібно привести до вигляду, який “розуміє” нейронна мережа, виділити основні дані масиву, доповнити масив технічними індикаторами (пошук входів за стратегією Елдера [40]).

- навчання рекурентної нейронної мережі та її тестування, прогнозування за допомогою моделі LSTM: за допомогою цих інструментів будуємо графік, оснований на попередніх даних, таким чином визначаємо майбутню ціну валютної пари.

- аналіз результатів та тестування моделі з різними наборами даних.

## 2 РОЗРОБКА ПІДХОДІВ ТА МОДЕЛЕЙ ДЛЯ ПРИЙНЯТТЯ РІШЕНЬ НА ЧАСОВИХ РЯДАХ ЗА ВИКОРИСТАННЯМ PYTHON

### 2.1 Опис бібліотек

Python, як сучасна мова програмування, майже одразу зарекомендувала себе лідером у розробці методів машинного навчання. На сьогодні застосування різних модулів, бібліотек все більше поширюється у багатьох сферах.

Для фінансової сфери, зокрема для прогнозування фінансових часових рядів, використовується ряд бібліотек:

- Pandas;
- NumPy;
- Scikit-learn;
- Keras;
- Pandas TA;
- MetaTrader5.

В поєднанні з методами прогнозування створюються моделі для прийняття рішень, що можуть забезпечити більш точні результати.

#### 2.1.1 Pandas

Pandas – програмна бібліотека, написана для мови програмування Python для маніпулювання даними та їхнього аналізу. Вона, зокрема, пропонує структури даних та операції для маніпулювання чисельними таблицями та часовими рядами. pandas є вільним програмним забезпеченням [41].

Можливості бібліотеки:

- Об'єкт DataFrame із вбудованим індексуванням для маніпулювання даними.

- Інструменти для зчитування та записування даних між структурами даних у пам'яті та різними форматами файлів.
- Вирівнювання даних та вбудована підтримка пропущених даних.
- Переформатовування для отримання зведених наборів даних.
- Отримання зрізів за мітками, індексування з розширеними можливостями та отримання піднаборів з великих наборів даних.
- Вставлення та вилучення стовпчиків у структурах даних.
- Рушій групування, що дозволяє робити з наборами даних операції розділення-зміни-об'єднання (англ. split-apply-combine).
- Злиття та з'єднання наборів даних.
- Ієрархічне індексування осей для роботи з даними високої вимірності в структурі даних нижчої вимірності.
- Функціональність для часових рядів: породження діапазонів дат та перетворення частоти, статистики рухливого вікна, лінійні регресії рухливого вікна, зсування дат та запізнювання.

### 2.1.2 NumPy

NumPy – розширення мови Python, що додає підтримку великих багатовимірних масивів і матриць, разом з великою бібліотекою високорівневих математичних функцій для операцій з цими масивами. Попередник NumPy, Numeric, був спочатку створений Jim Hugunin. NumPy — відкрите програмне забезпечення і має багато розробників[42].

### 2.1.3 Scikit-learn

Scikit-learn – це безкоштовна програмна бібліотека машинного навчання для мови програмування Python, яка надає функціональність для створення та тренування різноманітних алгоритмів класифікації, регресії та кластеризації, таких як лінійна регресія, random forest, градієнтний бустинг, і працює у зв'язці з бібліотеками NumPy та SciPy. Scikit-learn є однією з найбільш популярних бібліотек машинного навчання[43].

### 2.1.5 Keras

Keras – це API, призначений для машинного навчання (використовує tensorflow). Keras дотримується найкращих практик для зменшення когнітивного навантаження: він пропонує узгоджені та прості API, мінімізує кількість дій користувача, необхідних для типових випадків використання, і надає чіткі та дієві повідомлення про помилки. Він також містить розширену документацію та посібники для розробників[45].

### 2.1.6 Pandas TA

Це проста у використанні бібліотека, яка використовує пакет Pandas із понад 130 індикаторами та допоміжними функціями та понад 60 шаблонами TA Lib Candlestick. Включено багато часто використовуваних індикаторів, таких як: свічковий шаблон (cdl\_pattern), просте ковзне середнє (sma), розбіжність ковзного середнього (macd), експоненціальне ковзне середнє Халла (hma), смуги Боллінджера (bbands), балансовий обсяг (obv. ), Aroon & Aroon Oscillator (aroon), Squeeze (стисніть) та багато іншого[46].

### 2.1.7 MetaTrader 5

Пакет MetaTrader для Python призначений для зручного та швидкого отримання біржової інформації через міжпроцесну взаємодію прямо з терміналу MetaTrader 5. Отримані таким чином дані можна використовувати для статистичних обчислень і машинного навчання[47].

## 2.2 Вибір інструментів для подальшого моделювання

Для подальшого моделювання та роботи з часовими фінансовими рядами буде використано поєднання бібліотек Pandas, NumPy, Keras, Pandas TA що забезпечить підтримку типу даних, як DataFrame, реалізує популярні методи для прогнозування часових рядів та дозволить створити моделі із

застосуванням інтелектуальних технологій. Такий вибір обґрунтовується можливостями бібліотек та їх популярністю.

### 2.3 Створення моделей, схем використання із застосуванням нейромережі

Структуру програмного забезпечення умовно можна розділити на окремі блоки, кожен з яких реалізує певну функцію. Таку структуру зображено на рисунку 2.3.

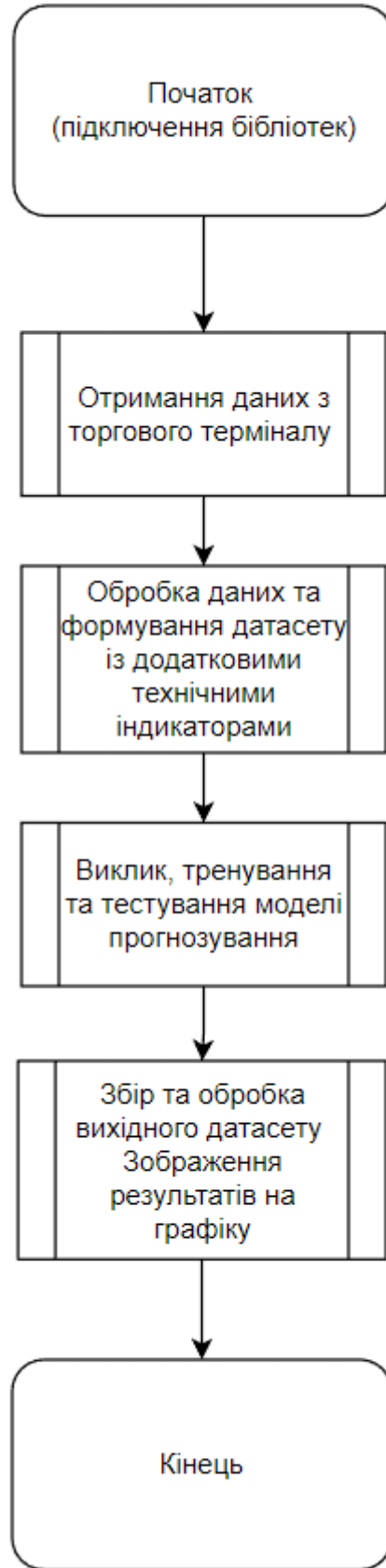


Рисунок 2.3 – Алгоритм роботи застосунку



В якості тестових даних було взято курс валютної пари EUR/USD за 03.08.19 - 03.03.20 (рис 3.3, рис 3.4). Використовується один ряд – Close. Для формування масивів індикаторів слід використовувати функції Pandas TA, що дозволяють видобути такі індикатори, як ковзке середнє. У даному випадку масив індикаторів утвориться на основі пари середніх з використанням оцінки різних часових вимірів (2MA systems)[48]. Суть методу, що створить індикатори для датасету, полягає у співставленні двох ковзких середніх на різних часових вимірах (такі, як H4, M15). В залежності від цього співставлення можна визначати сигнали відкриття/закриття позицій. Для більш точних прогнозів можливе застосування RSI та MFI індикатори. Для збору індикаторів, їх обчислення запропонована бібліотека ta-lib, що містить більше 150 індикаторів.

Після огляду існуючих методів прогнозування часових рядів за допомогою Python було обрано модель LSTM (рисунок 2.4)[49].

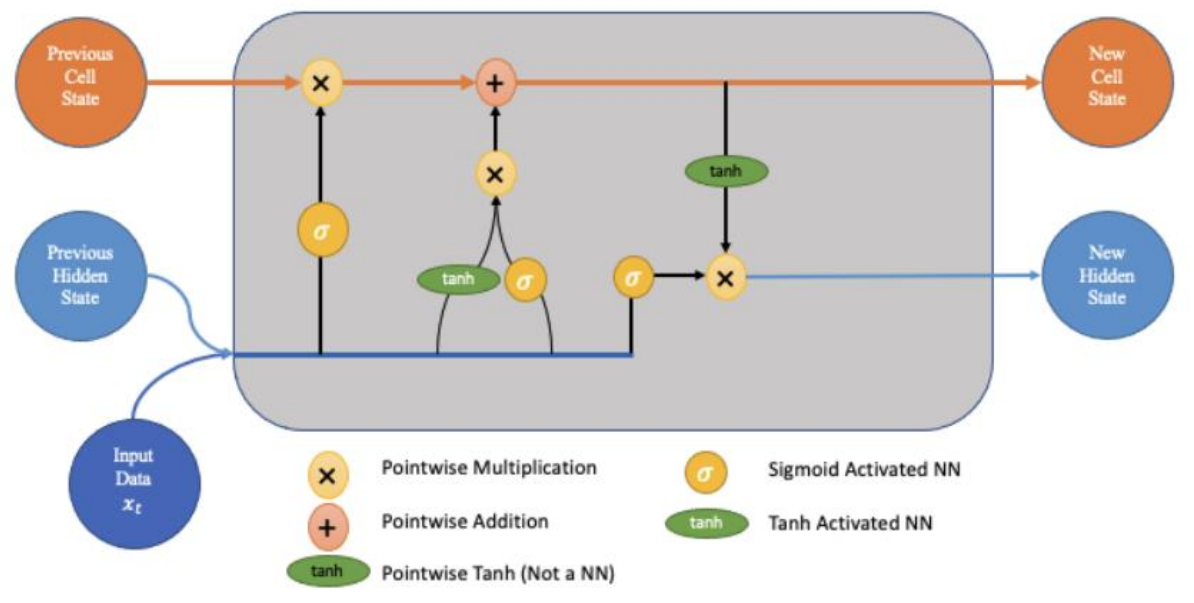


Рисунок 2.4 – Схема моделі LSTM

Використовуючи LSTM, моделі прогнозування часових рядів можуть передбачати майбутні значення на основі попередніх послідовних даних. Це забезпечує більшу точність для прогнозистів попиту, що призводить до кращого прийняття рішень для бізнесу. Її переваги[50]:

- Набори даних про попит представлені з різними сезонами. LSTM здатний фіксувати моделі як довгострокових сезонних змін, таких як річна модель, так і короткострокових сезонних змін, таких як тижневі моделі;
- Цілком природно, що події впливатимуть на попит у день, коли вони відбуваються, а також за дні до та після події. Наприклад, люди бронюють більше днів проживання, щоб відвідати спортивний захід. LSTM має можливість сортувати шаблони впливу різних категорій подій;
- LSTM може приймати вхідні дані різної довжини. Ця функція особливо корисна, коли LSTM використовується для створення загальних моделей прогнозування для конкретних клієнтів або галузей;
- Різні шари всередині LSTM підвищують його здатність фіксувати нелінійні залежності для прогнозування. Причинні фактори, як правило, мають нелінійний вплив на попит. Коли ці фактори використовуються як частина вхідної змінної, LSTM може вивчати нелінійну залежність для прогнозування;
- LSTM вимагає більше обчислень, ніж інші рекурентні нейронні мережі. Основна причина полягає в тому, що він має більше параметрів, які використовуються для прогнозів попиту. Проблему з обчисленнями вирішує cuDNN від NVIDIA. З досвіду, cuDNN може запровадити в 10+ разів більшу швидкість, ніж стандартне використання безпосередньо CUDA.

Окрім LSTM, ARIMA і Facebook Prophet є ще двома популярними моделями, які використовуються для прогнозування часових рядів:

- ARIMA – це популярний статистичний метод, який використовується в прогнозуванні часових рядів для прогнозування майбутніх тенденцій для даних часових рядів. Це клас моделей, які пояснюють дані часових рядів на основі їхніх минулих значень. Прийняття ARIMA для часових

рядів припускає, що інформація в минулому може бути використана лише для прогнозування майбутніх значень[51].

- Facebook Prophet – це процедура прогнозування даних часових рядів на основі адитивної моделі, де використовуються нелінійні тренди. Він найкраще працює з даними часових рядів, які мають сильний сезонний вплив[52].

## 2.4 Висновки

Порівнюючи методи прогнозування, постановку задачі та доступність обчислювальних ресурсів, було обрано модель LSTM для прогнозування часових рядів. Розроблено підхід та схему функцій (структури) програмного забезпечення. Наведено також опис бібліотек, які спрощують значну кількість потрібних операцій та збільшують шанси на позитивні результати прогнозування.

## 3 РОЗРОБКА ПРОЦЕДУРИ ДЛЯ ІНТЕГРАЦІЇ PYTHON З ТОРГОВИМ ТЕРМІНАЛОМ ДЛЯ ОТРИМАННЯ МАСИВУ ДАНИХ

### 3.1 Інтеграція з різними системами постачання фінансових даних

#### 3.1.1 MetaTrader

Пакет MetaTrader для Python призначений для зручного та швидкого отримання біржової інформації через міжпроцесну взаємодію прямо з терміналу MetaTrader 5. Отримані таким чином дані можна використовувати для статистичних обчислень і машинного навчання.

Переваги такої інтеграції – перспектива подальшого інтегрування Python коду у систему прийняття рішень, що дозволяє створити досить складні, як з точки архітектури ПЗ, так і функціонально, модулі прийняття рішень або ж автоматизувати процес торгівлі, у вигляді боту.

Для встановлення даного пакету, система пропонує функції, що зображені на рис 3.1

Установка пакета в командній строці:

```
pip install MetaTrader5
```

Обновление пакета в командній строці:

```
pip install --upgrade MetaTrader5
```

Рисунок 3.1 – функції встановлення пакету MetaTrader5

Функції пакету MetaTrader відображені на рис. 3.2

Функция	Действие
<a href="#">initialize</a>	Устанавливает соединение с терминалом MetaTrader 5
<a href="#">login</a>	Подключается к торговому счету с указанными параметрами
<a href="#">shutdown</a>	Закрывает ранее установленное подключение к терминалу MetaTrader 5
<a href="#">version</a>	Возвращает версию терминала MetaTrader 5
<a href="#">last_error</a>	Возвращает информацию о последней ошибке
<a href="#">account_info</a>	Получает информацию о текущем торговом счете
<a href="#">terminal_info</a>	Получает состояние и параметры подключенного терминала MetaTrader 5
<a href="#">symbols_total</a>	Получает количество всех финансовых инструментов в терминале MetaTrader 5
<a href="#">symbols_get</a>	Получает все финансовые инструменты из терминала MetaTrader 5
<a href="#">symbol_info</a>	Получает информацию по указанному финансовому инструменту
<a href="#">symbol_info_tick</a>	Получает последний тик по указанному финансовому инструменту
<a href="#">symbol_select</a>	Выбирает символ в окне <a href="#">MarketWatch</a> или убирает символ из этого окна
<a href="#">market_book_add</a>	Производит подписку терминала MetaTrader 5 на получение событий об изменениях в стакане по указанному символу
<a href="#">market_book_get</a>	Возвращает кортеж из BookInfo, содержащий записи стакана цен указанного символа
<a href="#">market_book_release</a>	Отменяет подписку терминала MetaTrader 5 на получение событий об изменениях в стакане по указанному символу
<a href="#">copy_rates_from</a>	Получает бары из терминала MetaTrader 5, начиная с указанной даты
<a href="#">copy_rates_from_pos</a>	Получает бары из терминала MetaTrader 5, начиная с указанного индекса
<a href="#">copyrates_range</a>	Получает бары в указанном диапазоне дат из терминала MetaTrader 5
<a href="#">copy_ticks_from</a>	Получает тики из терминала MetaTrader 5, начиная с указанной даты
<a href="#">copy_ticks_range</a>	Получает тики за указанный диапазон дат из терминала MetaTrader 5
<a href="#">orders_total</a>	Получает количество действующих ордеров.
<a href="#">orders_get</a>	Получает действующие ордера с возможностью фильтрации по символу или тикету
<a href="#">order_calc_margin</a>	Возвращает размер маржи в валюте счета для проведения указанной торговой операции
<a href="#">order_calc_profit</a>	Возвращает размер прибыли в валюте счета для указанной торговой операции
<a href="#">order_check</a>	Проверяет достаточность средств для совершения требуемой <a href="#">торговой операции</a>
<a href="#">order_send</a>	Отправляет из терминала на торговый сервер <a href="#">запрос</a> на совершение торговой операции.
<a href="#">positions_total</a>	Получает количество открытых позиций
<a href="#">positions_get</a>	Получает открытые позиции с возможностью фильтрации по символу или тикету
<a href="#">history_orders_total</a>	Получает количество ордеров в торговой истории в указанном интервале
<a href="#">history_orders_get</a>	Получает ордера из торговой истории с возможностью фильтрации по тикету или позиции
<a href="#">history_deals_total</a>	Получает количество сделок в торговой истории в указанном интервале
<a href="#">history_deals_get</a>	Получает сделки из торговой истории с возможностью фильтрации по тикету или позиции

Рисунок 3.2 – функції з пакету MetaTrader5

Тестовий скрипт для запуску пакету зображено на рис 3.3

```

from datetime import datetime
import matplotlib.pyplot as plt
import pandas as pd
from pandas.plotting import register_matplotlib_converters
register_matplotlib_converters()
import MetaTrader5 as mt5

# підключимося к MetaTrader 5
if not mt5.initialize():
    print("initialize() failed")
    mt5.shutdown()

# запитимо статус и параметри підключення
print(mt5.terminal_info())
# отримимо інформацію о версії MetaTrader 5
print(mt5.version())

# запитимо 1000 тиків с EURAUD
euraud_ticks = mt5.copy_ticks_from("EURAUD", datetime(2020,1,28,13), 1000, mt5.COPY_TICKS_ALL)
# запитимо тики с AUDUSD в інтервалі 2019.04.01 13:00 - 2019.04.02 13:00
audusd_ticks = mt5.copy_ticks_range("AUDUSD", datetime(2020,1,27,13), datetime(2020,1,28,13), mt5.COPY_TICKS_ALL)

# отримимо бари с різних інструментів різними способами
eurusd_rates = mt5.copy_rates_from("EURUSD", mt5.TIMEFRAME_M1, datetime(2020,1,28,13), 1000)
eurgbp_rates = mt5.copy_rates_from_pos("EURGBP", mt5.TIMEFRAME_M1, 0, 1000)
eurcad_rates = mt5.copy_rates_range("EURCAD", mt5.TIMEFRAME_M1, datetime(2020,1,27,13), datetime(2020,1,28,13))

# завершимо підключення к MetaTrader 5
mt5.shutdown()

#DATA
print('euraud_ticks(', len(euraud_ticks), ')')
for val in euraud_ticks[:10]: print(val)

print('audusd_ticks(', len(audusd_ticks), ')')
for val in audusd_ticks[:10]: print(val)

print('eurusd_rates(', len(eurusd_rates), ')')
for val in eurusrates[:10]: print(val)

print('eurgbp_rates(', len(eurgbp_rates), ')')
for val in eurgbp_rates[:10]: print(val)

print('eurcad_rates(', len(eurcad_rates), ')')
for val in eurcad_rates[:10]: print(val)

#PLOT
# створимо из отриманих даних DataFrame
ticks_frame = pd.DataFrame(euraud_ticks)
# зробимо отрисовку тиків на графіку
plt.plot(ticks_frame['time'], ticks_frame['ask'], 'r-', label='ask')
plt.plot(ticks_frame['time'], ticks_frame['bid'], 'b-', label='bid')

# виведемо легенду
plt.legend(loc='upper left')

# додати заголовок
plt.title('EURAUD ticks')

# покажемо графік
plt.show()

```

Рисунок 3.3 – Тестовий скрипт із використанням пакету MetaTrader5

Тестовий скрипт виводить графік із отриманими даними з системи MetaTrader (рис 3.4)

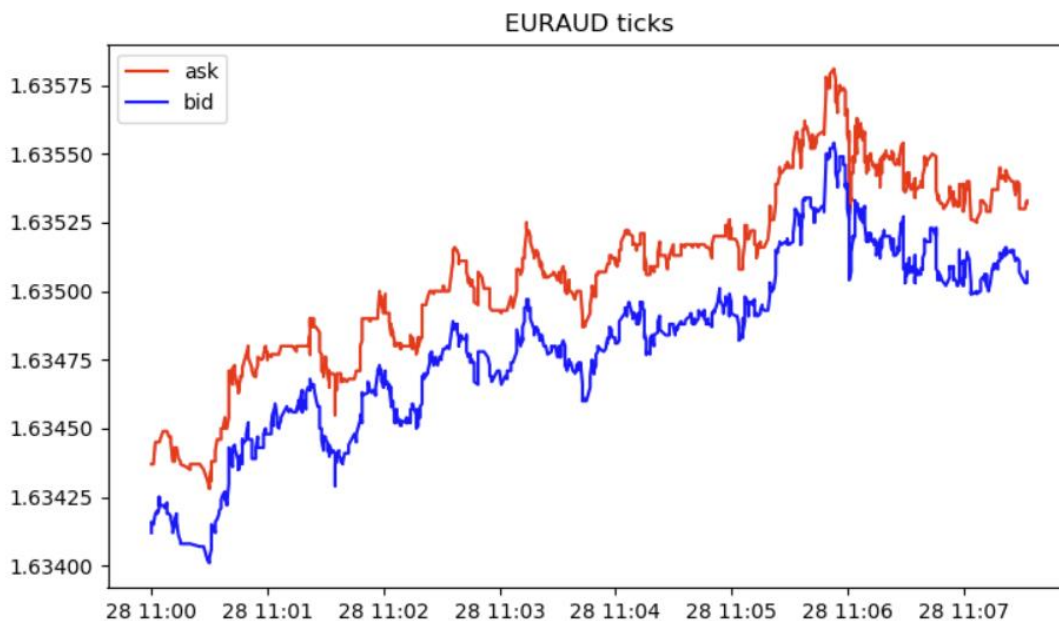


Рисунок 3.4 – Графік, як результат роботи скрипта

### 3.1.2 Tradermade API

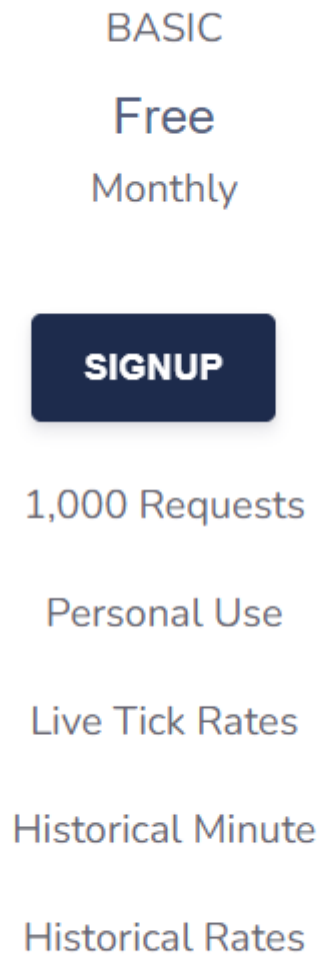
Через RESTful API Tradermade сервіс надає ціноутворення в tick, хвилинні, погодинні та щоденні ціни. Включає ціни Bid, Ask, Open, High, Low і Close для валют через кілька кінцевих точок. Хвилинні дані сягають 1990 року, а тики – 2016 року. Усі наші дані оновлюються в режимі реального часу. Курси валют перевіряються у реальному часі. На рисунку 3.1.2 зображено основні пункти даного сервісу.

RESTful API Features	
Rates	Bid Mid Ask Timestamp Open, High, Low, and Close
Instruments	4200+ Currency Pairs 40+ CFDs (Indices, Stocks, Commodities) 4100+ Cryptocurrency Pairs
Delivery	REST API
Secure	All delivery methods are SSL encrypted.
History	Minute Close from 1990 Tick History from 2016
Need to Know	Our data plan via RESTful API is a one-stop solution to all your forex and CFD data needs. You will be able to pull both real-time and historical data seamlessly.
We offer	We offer technical support for integration and can provide bulk tick and minute download otherwise not listed in our offerings. We also provide bespoke technology and data solutions to suit your needs.

Рисунок 3.5 – Характеристика Tradermade API

У сервісі є безкоштовна версія використання, в яку входять параметри, зображені на рисунку 3.1.3.





BASIC  
Free  
Monthly

**SIGNUP**

1,000 Requests

Personal Use

Live Tick Rates

Historical Minute

Historical Rates

Рисунок 3.6 – Параметри базового профіля використання

```
# # get timeseries data
# # get timeseries data
# import pandas as pd
# df = pd.DataFrame()
# from datetime import date

# today = date.today()

# # dd/mm/YY
# d1 = today.strftime("%d/%m/%Y")
# for i in range(2010, 2022):
#     x = tm.timeseries(currency='EURUSD', start=str(i)+"-06-17", fields=["open", "high", "low", "close",
#     end='{}-{}-{}'.format(i, d1.split('/')[1], d1.split('/')[0] ))
#     df = df.append(x)
# df = df.drop_duplicates()
```

Рисунок 3.7 – Приклад отримання даних за допомогою Tradersmade API

### 3.1.3 YFinance API

Yahoo Finance, безумовно, є одним із найпопулярніших джерел даних про фондовий ринок, доступних для роздрібних трейдерів, і, таким чином, є першим джерелом даних, яким більшість із трейдерів користувалися, коли починали. Варто також зазначити, що його простий у використанні та масив наборів даних дозволяє швидко створювати прототипи, що робить його цінним джерелом для тестування дослідницьких ідей та впровадження дуже простих стратегій.

Переваги сервісу:

- Безкоштовно: той факт, що доступ до величезної кількості даних, які пропонує Yahoo Finance, є безкоштовним, безумовно, є його найбільшою перевагою.
- Безліч наборів даних: на відміну від інших джерел даних, Yahoo Finance пропонує широкий спектр наборів даних, від внутрішньоденних даних про ціни до показників ESG.
- Активно підтримується: хоча це може здатися неактуальним, той факт, що бібліотека уfinance має велику спільноту користувачів, додає бібліотеці додатковий рівень міцності та надійності. Ця надійність обмежена через неофіційний характер бібліотеки.

Недоліки:

- Неофіційна бібліотека: бібліотека розроблена спільнотою користувачів, а не самим постачальником даних. Yahoo Finance може оновлювати свій API у будь-який час без попереднього повідомлення розробників, що призводить до коротких періодів часу, коли дані недоступні або пошкоджені.
- Ненадійний: оскільки це неофіційна бібліотека, постачальник не гарантує ні якість даних, ні доступ до них. Yfinance ніколи не слід використовувати для живої торгівлі реальними грошима.

### 3.2 Обґрунтування вибору інтеграції із MetaTrader 5 та його розробка

Оглядаючись на доступність різних сервісів, їх переваг та недоліків а також можливості подальшого масштабування проекту було обрано саме інтеграцію з торговим терміналом MetaTrader 5. Таким чином було проведено ознайомлення із базовими функціями користування бібліотекою MetaTrader5 у середовищі розробки Python та на рисунку 3.8, рисунку 3.9 наведено приклад інтеграції у програмі.

```
#register_matplotlib_converters()
import MetaTrader5 as mt5

# підключимся к MetaTrader 5
if not mt5.initialize():
    print("initialize() failed")
    mt5.shutdown()

# запросим статус и параметры подключения
print(mt5.terminal_info())
# получим информацию о версии MetaTrader 5
print(mt5.version())

# запросим 1000 тиков с EURAUD

symbol_info = mt5.symbol_info("EURUSD")
#actual data for future predict
#rates = mt5.copy_rates_from_pos("EURUSD", mt5.TIMEFRAME_D1, 0, 1000)

#test data
timezone = pytz.timezone("Etc/UTC")
# создадим объект datetime в таймзоне UTC, чтобы не применялось смещение локальной таймзоны
utc_from = datetime(2020, 3, 3, tzinfo=timezone)
# получим 10 баров с EURUSD H4 начиная с 01.10.2017 в таймзоне UTC
rates = mt5.copy_rates_from("EURUSD", mt5.TIMEFRAME_H4, utc_from, 1000)
```

Рисунок 3.8 – Фрагмент коду інтеграції з торговим терміналом MetaTrader5 та отримання даних валютної пари EURUSD у кількості 1000 тиків за період до 03.03.20

	time	open	high	low	close	tick_volume	spread	real_volume
100	1564689600	1.10682	1.10957	1.10640	1.10847	27761	2	0
101	1564704000	1.10849	1.10965	1.10743	1.10889	9146	2	0
102	1564718400	1.10890	1.10947	1.10699	1.10822	13528	2	0
103	1564732800	1.10822	1.11150	1.10770	1.10959	18876	2	0
104	1564747200	1.10959	1.11069	1.10783	1.10850	18423	2	0
...	...	...	...	...	...	...	...	...
994	1583121600	1.10566	1.10634	1.10418	1.10432	7949	1	0
995	1583136000	1.10432	1.10932	1.10382	1.10795	17475	1	0
996	1583150400	1.10795	1.11642	1.10775	1.11218	18713	1	0
997	1583164800	1.11219	1.11845	1.11079	1.11796	23128	1	0
998	1583179200	1.11796	1.11832	1.11235	1.11332	9742	1	0

Рисунок 3.9 – Дані, отримані за допомогою MetaTrader5 бібліотеки

### 3.3 Висновки

Для отримання актуальних даних, системи прийняття рішень пропонують пакети, що включають у себе вже готові функції або використовують технології API, що у наш час зустрічається більш розповсюдженою практикою. На основі API, або ж пакетів є перспективи створення власного програмного забезпечення для аналізу, прогнозування часових рядів. За допомогою даних інструментів є можливість створення датасетів з великими обсягами даних та проводити різні маніпуляції, використовуючи технології Python.

## 4 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 4.1 Вимоги до програмного та апаратного забезпечення

- Microsoft Windows 10/11 (32, 64 bit);
- Мінімум 8 Гб оперативної пам'яті;
- Рекомендовано 16 Гб оперативної пам'яті;
- Мінімальне підключення до мережі Інтернет зі швидкістю 1МБ\с
- Python 3.8+;
- Бібліотека Tensorflow;
- Anaconda останньої версії;
- Jupyter Notebook.

### 4.2. Вибір середовища для розробки

Jupyter – це безкоштовний інтерактивний веб-інструмент із відкритим вихідним кодом, відомий як обчислювальний блокнот, який дослідники можуть використовувати для об'єднання програмного коду, результатів обчислень, пояснювального тексту та мультимедійних ресурсів в одному документі. Обчислювальні ноутбуки існують десятиліттями, але популярність Jupyter особливо зросла за останні пару років. Цьому швидкому поширенню сприяла захоплена спільнота користувачів-розробників і перероблена архітектура, яка дозволяє ноутбуку розмовляти десятками мов програмування — факт, що відображено в його назві, яка, за словами співзасновника Фернандо Переса, була натхненна мови програмування Julia (Ju), Python (Py) і R.

Згідно з аналізом сайту GitHub, який обмінювався кодами, у вересні 2018 року було нараховано понад 2,5 мільйона загальнодоступних ноутбуків Jupyter, порівняно з приблизно 200 000 у 2015 році. Частково, це зростання

відбувається завдяки вдосконаленню веб-програмного забезпечення, яке керує такими програмами, як Gmail і Google Docs; розвиток наукового Python і науки про дані; і, особливо, легкість, з якою ноутбуки полегшують доступ до віддалених даних, які інакше було б непрактично завантажити — наприклад, із LSST.

Блокнот (Notebook – спеціальний формат середовища) Jupyter складається з двох компонентів. Користувачі вводять програмний код або текст у прямокутні комірки зовнішньої веб-сторінки. Потім веб-переглядач передає цей код до внутрішнього «ядра», яке запускає код і повертає результати. За підрахунками, було створено понад 100 ядер Jupyter, які підтримують десятки мов програмування. Зазвичай кожен ноутбук може запускати лише одне ядро та одну мову, але існують обхідні шляхи. Один демонстраційний блокнот, наприклад, розмовляє Python, Julia, R і Fortran.

Важливо, що ядра не обов'язково повинні знаходитися на комп'ютері користувача. Коли майбутні користувачі LSST використовуватимуть ноутбуки Jupyter для аналізу своїх даних, код працюватиме на суперкомп'ютері в Іллінойсі, забезпечуючи обчислювальну потужність, з якою не зрівняється жодний настільний ПК. Ноутбуки також можуть працювати в хмарі. Наприклад, проект Google Colaboratory надає інтерфейс на тему Google для ноутбука Jupyter. Це дозволяє користувачам співпрацювати та запускати код, який використовує хмарні ресурси Google, такі як графічні процесори, і зберігати свої документи на Диску Google.

### 4.3 Розробка коду додатку

На основі проаналізованих та розглянутих бібліотек, методів і моделей необхідно розробити систему прогнозування часових рядів із використанням масивів індикаторів.

Перш за все, необхідно отримати дані часового ряду. Задля цього застосовується інтеграція MetaTrader5 із Python. Оптимальним часовим інтервалом було обрано відрізок H4, при використанні якого є можливість отримати прогноз на недалеке майбутнє (саме 2-3 дні).

```
symbol_info = mt5.symbol_info("EURUSD")
timezone = pytz.timezone("Etc/UTC")
utc_from = datetime(2020, 3, 3, tzinfo=timezone)
rates = mt5.copy_rates_from("EURUSD", mt5.TIMEFRAME_H4, utc_from,
1000)
```

За допомогою використання бібліотек Pandas та Pandas TA стає можливим не тільки створити DataFrame, а й наповнити його технічними індикаторами. Таким чином, не потрібно підраховувати та вивантажувати індикатори, використовуючи функціонал MT5, достатньо викликати відповідні методи з бібліотеки Pandas TA (рисунок 4.3):

```
df = pd.DataFrame(rates)
sma10 = df.ta.sma(10)
sma30 = df.ta.sma(30)
df = pd.concat([df, sma10, sma30], axis=1)
```



Рисунок 4.3 – Датасет з використанням технічних індикаторів

При вході даних у модель прогнозування, важливо провести їх масштабування. Так як модель прогнозування найкраще за все розуміє саме значення від 0 до 1, за допомогою класу `MinMaxScaler` нормовано вхідний датасет:

```

scaler = MinMaxScaler(feature_range=(0, 1))
scaler = scaler.fit(y)
y = scaler.transform(y)
print(y.shape)

n_lookback = 800 # length of input sequences (lookback period)
n_forecast = 12 # length of output sequences (forecast period)
X = []
Y = []
for i in range(n_lookback, len(y) - n_forecast + 1):
    X.append(y[i - n_lookback: i])

```



```

Y.append(y[i: i + n_forecast])
X = np.array(X)
Y = np.array(Y)

```

Параметри *n\_lookback* та *n\_forecast* визначають на скільки кроків назад потрібно “запам’ятовувати” значення моделі та на скільки кроків вперед потрібно зробити прогнозування.

Під час тренування моделі було підібрано оптимальні параметри. На рисунку 4.4 зображено шари моделі:

```

# fit the model
model = Sequential()
model.add(LSTM(units=128, return_sequences=True,
input_shape=(n_lookback, 4)))
model.add(LSTM(units=64))
model.add(Dense(n_forecast))
model.compile(loss='mean_squared_error', optimizer='adam')

model.summary()

history = model.fit(X, Y[:, :, 0], epochs=40, batch_size=32, verbose=1)

```

```

Model: "sequential"
-----
Layer (type)                Output Shape         Param #
-----
lstm (LSTM)                  (None, 800, 128)    68096
lstm_1 (LSTM)                (None, 64)          49408
dense (Dense)                (None, 12)          780
-----
Total params: 118,284
Trainable params: 118,284
Non-trainable params: 0

```

Рисунок 4.4 – Шари моделі

При тренуванні спостерігалось зменшенні втрат впродовж кожної Епохи (рисунок 4.5) – такий результат досягнуто в першу чергу великою вибіркою даних.

```

Epoch 1/40
3/3 [=====] - 4s 592ms/step - loss: 0.0511
Epoch 2/40
3/3 [=====] - 2s 578ms/step - loss: 0.0314
Epoch 3/40
3/3 [=====] - 2s 547ms/step - loss: 0.0281
Epoch 4/40
3/3 [=====] - 2s 569ms/step - loss: 0.0218
Epoch 5/40
3/3 [=====] - 2s 577ms/step - loss: 0.0183
Epoch 6/40
3/3 [=====] - 2s 538ms/step - loss: 0.0146
Epoch 7/40
3/3 [=====] - 2s 567ms/step - loss: 0.0104
Epoch 8/40
3/3 [=====] - 2s 544ms/step - loss: 0.0072
Epoch 9/40
3/3 [=====] - 2s 577ms/step - loss: 0.0064
Epoch 10/40
3/3 [=====] - 2s 648ms/step - loss: 0.0041
Epoch 11/40
3/3 [=====] - 2s 598ms/step - loss: 0.0039
Epoch 12/40
3/3 [=====] - 2s 570ms/step - loss: 0.0036
Epoch 13/40
3/3 [=====] - 2s 598ms/step - loss: 0.0032
Epoch 14/40
3/3 [=====] - 2s 602ms/step - loss: 0.0027
Epoch 15/40
3/3 [=====] - 2s 552ms/step - loss: 0.0027
Epoch 16/40
3/3 [=====] - 2s 618ms/step - loss: 0.0027
Epoch 17/40
3/3 [=====] - 2s 593ms/step - loss: 0.0022
Epoch 18/40
3/3 [=====] - 2s 576ms/step - loss: 0.0022
Epoch 19/40
3/3 [=====] - 2s 592ms/step - loss: 0.0023
Epoch 20/40
3/3 [=====] - 2s 560ms/step - loss: 0.0022
Epoch 21/40
3/3 [=====] - 2s 612ms/step - loss: 0.0020
Epoch 22/40
3/3 [=====] - 2s 589ms/step - loss: 0.0020
Epoch 23/40
3/3 [=====] - 2s 577ms/step - loss: 0.0021
Epoch 24/40
3/3 [=====] - 2s 567ms/step - loss: 0.0020
Epoch 25/40
3/3 [=====] - 2s 569ms/step - loss: 0.0021
Epoch 26/40
3/3 [=====] - 2s 565ms/step - loss: 0.0021
Epoch 27/40
3/3 [=====] - 2s 561ms/step - loss: 0.0021
Epoch 28/40
3/3 [=====] - 2s 581ms/step - loss: 0.0020
Epoch 29/40
3/3 [=====] - 2s 607ms/step - loss: 0.0020
Epoch 30/40
3/3 [=====] - 2s 596ms/step - loss: 0.0019
Epoch 31/40
3/3 [=====] - 2s 580ms/step - loss: 0.0019
Epoch 32/40
3/3 [=====] - 2s 572ms/step - loss: 0.0019
Epoch 33/40
3/3 [=====] - 2s 557ms/step - loss: 0.0020
Epoch 34/40
3/3 [=====] - 2s 604ms/step - loss: 0.0019
Epoch 35/40
3/3 [=====] - 2s 570ms/step - loss: 0.0019
Epoch 36/40
3/3 [=====] - 2s 566ms/step - loss: 0.0020
Epoch 37/40
3/3 [=====] - 2s 554ms/step - loss: 0.0019
Epoch 38/40
3/3 [=====] - 2s 557ms/step - loss: 0.0019
Epoch 39/40
3/3 [=====] - 2s 552ms/step - loss: 0.0019
Epoch 40/40
3/3 [=====] - 2s 556ms/step - loss: 0.0019

```

Рисунок 4.5 – Епохи тренування моделі

Наступним кроком є саме передбачення моделі та приведення вихідних даних до початкового вигляду:

```
# generate the forecasts
X_ = y[- n_lookback:] # last available input sequence

X_ = X_.reshape(-1, n_lookback, 4)
print(X_.shape)
Y_ = model.predict(X_)
Y_ = Y_.reshape(-1,4)
print(Y_.shape)
Y_ = scaler.inverse_transform(Y_)
```

Щоб зобразити результати на графіку, використовується функція `plot` з бібліотеки `matplotlib`. Передбачення складає собою ціну `Close` на 12 проміжків уперед. На рисунку 4.6 зображено історичні дані та прогнозовані-на усьому проміжку даних. На рисунку 4.7 – більш детальне порівняння прогнозованих даних (синій колір) з реальними даними (червоний).

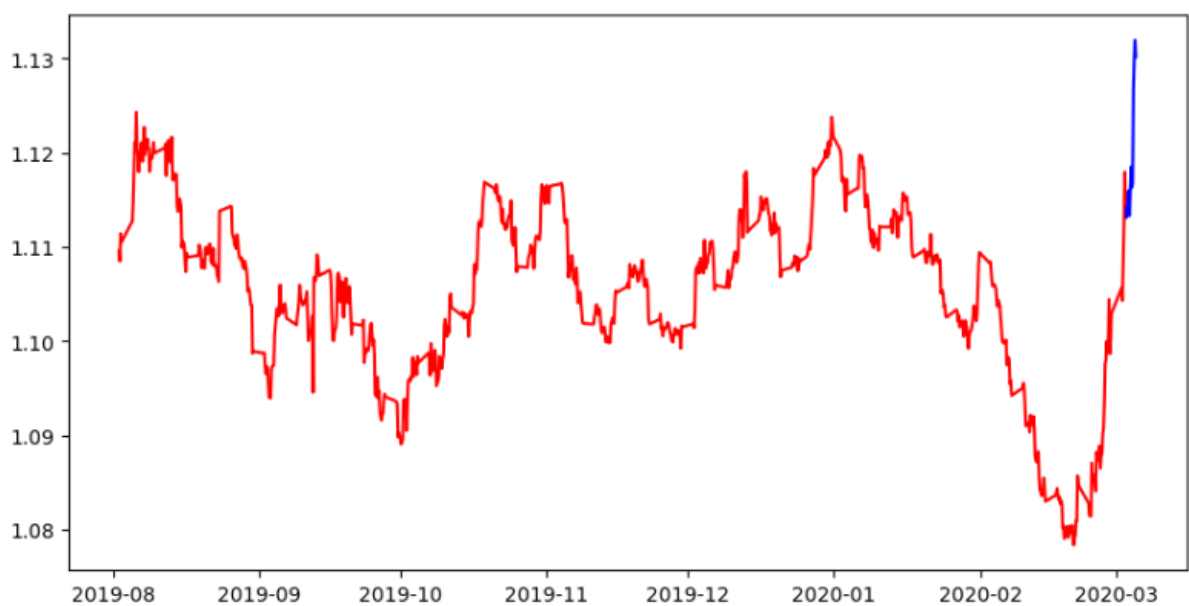


Рисунок 4.6 – Графік із зображенням історичних та прогнозованих даних

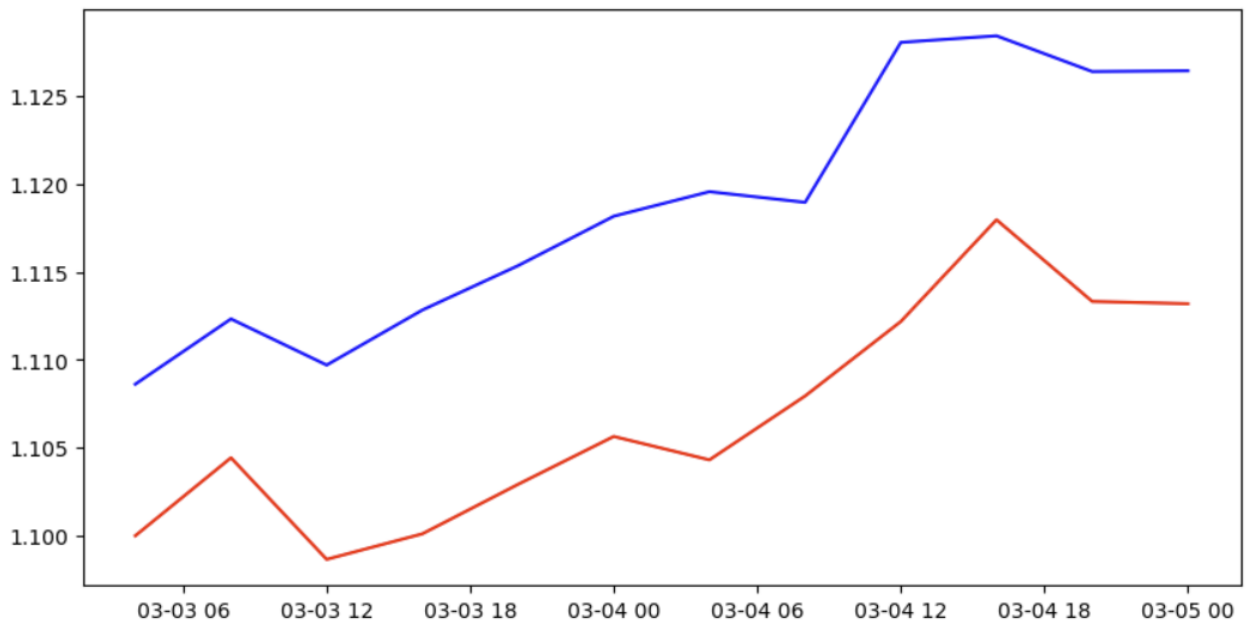


Рисунок 4.7 – Порівняння прогнозованих даних із реальними

Слід зауважити що розбіжність результатів сягає у середньому 150 пункта, для даного часового проміжку досить багато. При цьому можна спостерігати збереження тренду, що є позитивним результатом.

Виходячи з того, що передбачені значення ціни Close не є досить точними, було спроектовано сигнали для рекомендацій покупки/продажу позицій. Даний сигнал являє собою значення від 0 до 1, де 0 є купівля, а 1 – продаж. Формується цей сигнал на основі вхідних даних, а саме зміни ціни трьох попередніх значень:

```
series_shifted = df['close'].shift()
```

```
series_shifted = series_shifted/100
```

```
df['per'] = (df['close']/series_shifted - 100) * 100
```

```
df['signal'] = 0
```

```
for i in range(102,len(df)+100):
```

```
    if i>2:
```

```
        df['signal'].loc[i] = df.per.loc[i-2]+df.per.loc[i-1]+df.per.loc[i]
```

```
pd.set_option('display.max_rows', None)
```

```
df = df.iloc[3:]
```

```
df.signal -= df.signal.min()
```

```
df.signal /= df.signal.max()
```

Змінивши вихідну змінну моделі прогнозування та додавши даних сигнал у датасет було отримано графік, зображений на рисунку 4.8, на рисунку 4.9 та 4.10 зображено збільшені фрагменти графіку з рисунку 4.8. Сигнал купівлі/продажу реалізований таким чином, що дає значення у проміжку від 0 до 1, де значення, наближені до нуля – є сигналом купівлі, а наближені значення до одиниці – сигнал продажу.

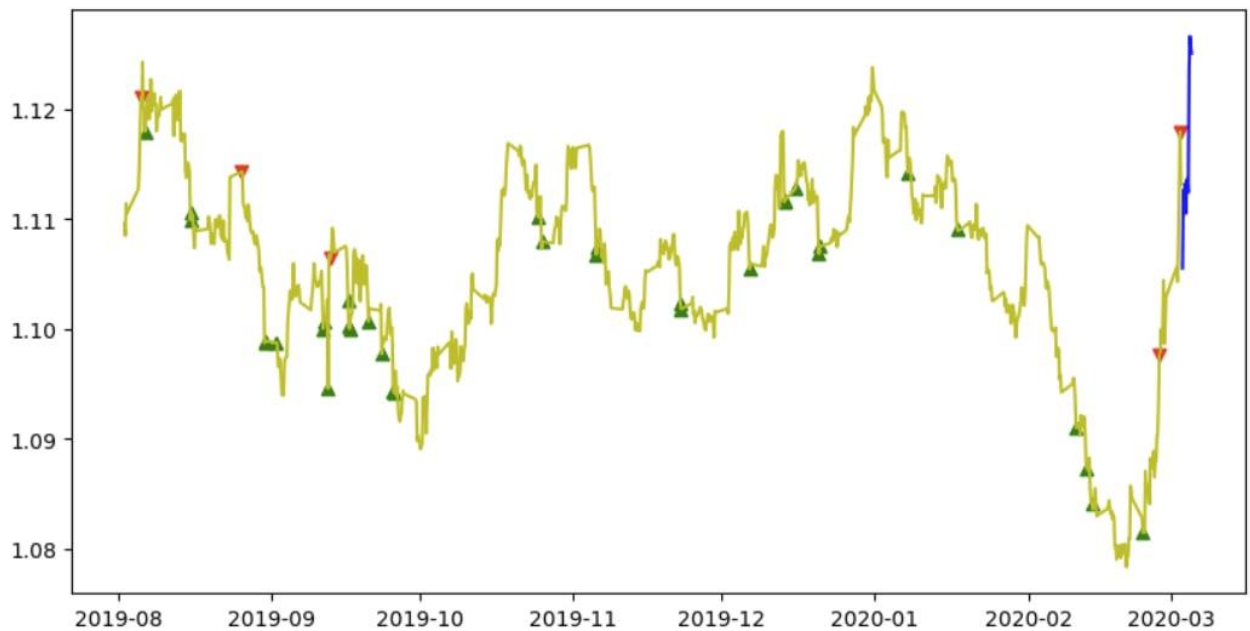


Рисунок 4.8 – Спроектвані сигнали купівлі/продажу

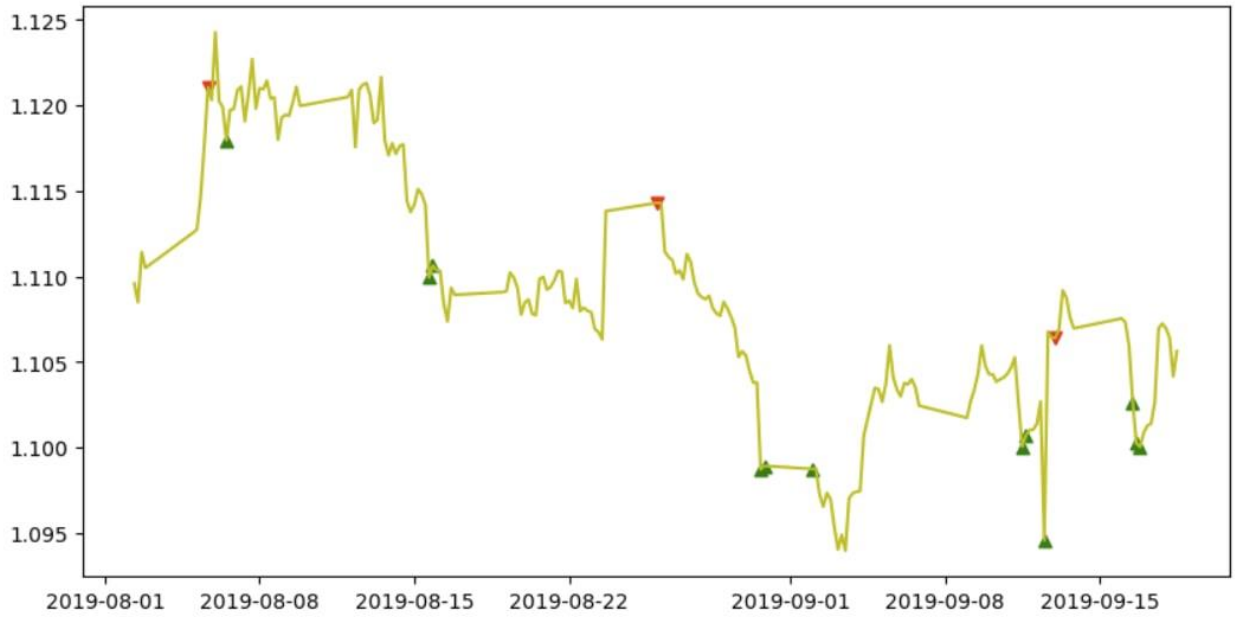


Рисунок 4.9 – Спроектовані сигнали купівлі/продажу (перші 100 значень)

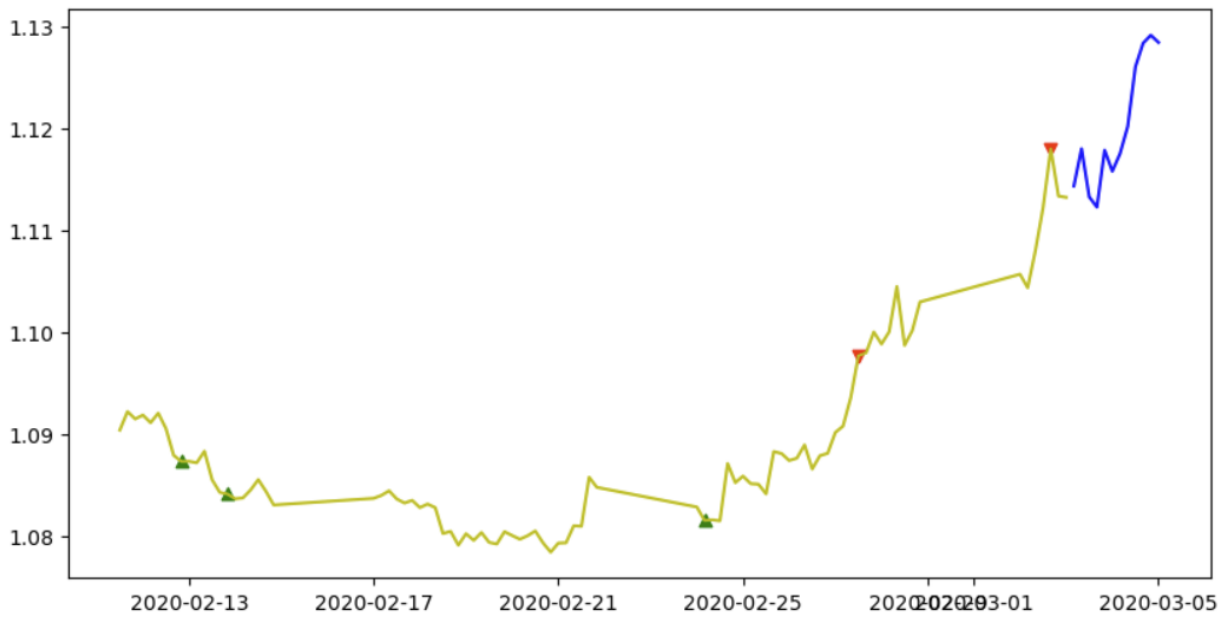


Рисунок 4.9 – Спроектовані сигнали купівлі/продажу (останні 100 значень та прогноз моделі)

#### 4.4 Висновки

Проаналізувавши отримані результати, отриманий прогноз сигналів можна використовувати як систему підтримки рішень. Точність

прогнозування насамперед досягнута за рахунок використання LSTM моделі, нормуванню вхідного датасету, використанням допоміжних технічних індикаторів та підбором параметрів для моделі прогнозування.

## ВИСНОВКИ

За час виконання магістерської роботи були вирішені такі завдання:

- Огляд існуючих методів для прогнозування часових фінансових рядів
- Огляд, аналіз застосування сучасних інтелектуальних інструментів для прогнозування
- Створено підхід до вирішення поставленої проблеми
- Розглянуто моделі прогнозування на основі нейронних мереж
- Проаналізовано, обгрунтовано та обрано LSTM модель для подальшого прогнозування
- Створено схему функцій програмного забезпечення
- Розроблено процедури для інтеграції Python з торговими терміналами, API сервісами для отримання масиву даних
- Сформовано вхідний датасет із застосуванням масивів індикаторів
- Обрано кращі часові інтервали для часового ряду
- Обрано середовище для розробки для обгрунтовано його вибір
- Розроблено програмне забезпечення у середовищі Jupyter
- Протестовано дане ПЗ з різними вхідними даними
- Розроблено допоміжний сигнал на покупку/продаж позицій для створення концепції системи підтримки прийняття рішень

У роботі представлено дослідження існуючих на сьогодні методів, моделей прогнозування із застосуванням сучасних інтелектуальних інструментів на основі Python. Дана сфера активно розвивається та представлено достатньо цікаві методи на різних фінансових ринках.

Прогнозування часових рядів охоплює не тільки фінансову складову. Практичні задачі зустрічаються майже у всіх галузях, де у кожній з них є свої тонкощі використання. Тому, методи прогнозування, в більшості випадків, що дають досить позитивні результати направлені на вирішення конкретної задачі у конкретній галузі та не може вважатись універсальним рішенням.



Розділ 1 аналізує проблему в галузі досліджень, розглядає стратегії та підходи прийняття рішень на фінансових цінових рядах та концепції автоматизації процесу прийняття рішень. Проведено аналіз застосування сучасних інтелектуальних інструментів для прогнозування. Висунуто власний підхід для вирішення проблеми.

У другому розділі описано які бібліотека використовуватимуться при розробці моделі із застосуванням нейромережі. Обгрунтовано вибір конкретної моделі (LSTM) та наведено переваги та недоліки інших. Створено схему функцій програмного забезпечення.

Третій розділ описує розробку інтеграції Python з торговими терміналами, сервісами API для отримання масиву даних. Наведено декілька систем постачання фінансових даних та описано їх переваги та недоліки. Обгрунтовано вибір торгового терміналу MetaTrader5. Розроблено інтеграцію з даним терміналом та пройдено практику у його використанні.

Розділ 4 передбачає собою опис розробки програмного забезпечення:

- Формування дата-сету
- Застосування технічних індикаторів
- Вибір кращих часових інтервалів
- Тренування та тестування моделі прогнозування
- Підбір параметрів, оптимізація вхідного дата-сету
- Аналіз отриманих даних
- Створення допоміжного сигналу для прогнозування (сигнал покупки/продажі позицій)

Отже, результатами роботи ПЗ є вихідний масив даних, який складається з сигналів рекомендацій на продаж чи покупку позицій. Такі результати обумовлені тим, що значення прогнозу ціни Close з похибкою приблизно в 150 пункта, при різних періодах призводило б до хибних прогнозів. Швидкість тренування моделі досить велика, тому підбір параметрів моделі виявився вдалим. Нормування даних дало змогу отримувати точні прогнози із

збереженням трендів на відрізках, а вихідні сигнали у перспективі можна використовувати у рамках системи підтримки прийняття рішень.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Андерсон Т. В. Статистический анализ временных рядов / Т. В. Андерсон. – М. : Мир, 1976. – 756 с.
2. Медведев Г. А. Математические основы финансовой экономики: учебник / Г. А. Медведев. – Минск : БГУ, 2011. – 303 с.
3. Елисеева И. И. Общая теория статистики: учебник / И. И. Елисеева, М. М. Юзбашев; под ред. И. И. Елисеевой, – 5-е изд., перераб. и доп. – М. : Финансы и статистика, 2004. – 656 с.
4. Ефимова О. В. Финансовый анализ: современный инструментарий для принятия экономических решений / О. В. Ефимова – М. : Омега-Л, 2009. – 350 с.
5. Кислиця Л. М. Оптимізація моделі для прийняття рішень з використанням GARCH-методів / Р. Н. Кветний, В. В. Кабачій, Л. М. Кислиця // Інформаційні технології та комп'ютерна інженерія. – 2007. – № 6. – С.122–127.
6. Кветний Р. Н. Імовірнісні нейронні мережі в задачах ідентифікації часових рядів [Електронний ресурс] / Р. Н. Кветний, В. В. Кабачій, О. О. Чумаченко // Наукові праці Вінницького національного технічного університету. – 2010. – № 3. – 6 с. – Режим доступу до журн: <http://www.nbu.gov.ua/e-journals/VNTU/2010-3/2010-3.htm>.
7. Минаев Ю. Н. Методы и алгоритмы решения задач идентификации и прогнозирования в условиях неопределенности в нейросетевом логическом базисе / Ю. Н. Минаев, О. Ю. Филимонова, Бенамеур Лиес. – М. : Горячая линия – Телеком, 2003. – 205 с.
8. Якимкин В. Н. Финансовый дилинг: технический анализ / В. Н. Якимкин. – М. : Омега-Л, 2005. – 480 с.
9. Лиховидов В. Н. Фундаментальный анализ мировых валютных рынков / В. Н. Лиховидов. – Владивосток: Мир, 1999. – 234 с.
10. Кветний Р. Н. Ідентифікація ринкових залежностей на основі явища дивергенції між ціною та індикатором технічного аналізу / Р. Н. Кветний, В.

- Ю. Коцюбинський А. В. Шкарпета // Вісник Вінницького політехнічного інституту. – 2006. – №1. – С. 23–34.
11. Шелудько В. М. Фінансовий ринок / В. М. Шелудько. – К. : Знання, 2008. – 535 с.
12. Берзлев О. Ю. Сучасний стан інформаційних систем прогнозування часових рядів / О. Ю. Берзлев // ДВНЗ «Ужгородський національний університет». – 2013. – №13. – С. 78–82.
13. Берзлев О. Ю. Адаптивні комбіновані моделі прогнозування біржових показників / О. Ю. Берзлев, М. М. Маляр, В. В. Ніколенко // Вісник Черкаського держ. технолог. ун-ту. Серія: технічні науки. – 2011. – № 1. – С. 50-54.
14. Бідюк П. І. Системний підхід до побудови математичних моделей на основі часових рядів / П. І. Бідюк, І. В. Баклан, В. М. Рифа // Системні дослідження та інформаційні технології. – 2002. – №3. – С. 114-131.
15. Бідюк П. І. Часові ряди: моделювання та прогнозування / П. І. Бідюк, О. І. Савенко, І. В. Баклан. – К. : ЕКМО, 2004. – 114 с.
16. Кендалл М. Многомерный статистический анализ и временные ряды / М. Кендалл, А. Стьюарт. – М. : Наука, 1976. – 375 с.
17. Методы анализа рынка Форекс [Електронний ресурс] – Режим доступу: <http://forex-invest.tv/stati-foreks/metodi-analiza-rinka-foreks.html> – [25.03.2017].
18. Найман Э. Л. Малая энциклопедия трейдера / Э. Л. Найман. – М. : Альпина Паблишер, 2014. – 458 с.
19. Коттл С. «Анализ ценных бумаг» Грэма и Додда / С. Коттл, Р. Ф. Мюррей, Ф. Е. Блок. – М. : Олимп-Бизнес, 2000. – 704 с.
20. Лефевр Э. Воспоминания биржевого спекулянта / Эдвин Лефевр. – М. : Олимп-Бизнес, 2008. – 416 с.
21. Колмыкова Л. И. Фундаментальный анализ финансовых рынков / Л. И. Колмыкова. – М. : Академия биржевой торговли, 2008. – 276 с.

22. Фундаментальный анализ рынка и стратегия торговли на новостях [Электронный ресурс] – Режим доступа: <http://tempofox.com/fundamentalnyj-analiz-rynka-i-strategiya-torgovli-na-novostyax/> – .
23. Нисон С. Японские свечи: графический анализ финансовых рынков. Перевод с англ. Дозорова Т. А, Волкльва М. А. / С. Нисон. – М. : Диаграмма, 1998. – 336 с.
24. Нисон С. За гранью японских свечей. Перевод с англ. Волкльва М. А., Волков А. А. / С. Нисон. – М. : Диаграмма, 2001. – 304 с.
25. Mandelbrot B. B. Statistical Methodology for NonPeriodic Cycles: From the Covariance to R/S Analysis / B. B. Mandelbrot // Annals of Economic and Social Measurement. – 1972. – N 1. – P. 259-290.
26. Mandelbrot B. B. The (mis)behavior of markets / B. B. Mandelbrot, R. L. Hudson. – New York, N.Y. : Basic Books, 2004. – 328 p.
27. Peters E. E. Fractal market analysis: applying chaos theory to investment and economics / E. E. Peters. – New York: John Wiley & Sons, Inc, 1994. – 336 p.
28. Федер Е. Фракталы. / Е. Федер. – М. : Мир, 1991. – 254 с.
29. Теория Эллиота [Электронный ресурс] – Режим доступа: <http://fxmarketwave.com/Theory.aspx>
30. Сохацька О. М. Біржова справа: Підручник / Сохацька О. М. – 2-ге вид. змін. й доп. – Тернопіль: Карт-бланш, К.: Кондор, 2008. – 632 с.
31. Демарк Т. Технический анализ - новая наука / Томас Демарк. – М.: Евро, 2012. – 286 с.
32. Эрлих А. Технический анализ товарных и фондовых рынков / А. Эрлих. – М. : ИНФРА-М, 1996. – 200 с.
33. Аксиоми технічного аналізу діяльності [Електронний ресурс] – Режим доступу: <http://skeptik.dp.ua/aksiomy-technichnogo-analizu.html> – .
34. Сохацька, О. М. Фундаментальний та технічний аналіз цін товарних та фінансових ринків: навч. посіб. / О. М. Сохацька, І. В. Роговська-Іщук, С. І. Вінницький. – К. : Кондор, 2012. – 305 с.

35. Лиховидов В. Н. Системы на основе скользящих средних / В. Н. Лиховидов // Валютный спекулянт. – 2004. – № 6. – С. 34–38.
36. Стратегії [Електронний ресурс] – Режим доступу: <http://tradelikeapro.ru/category/torgovyie-strategii/> .
37. Рэдхэд К. Управление финансовыми рисками. Пер. с англ. / К. Рэдхэд, С. Хьюс. – М. : ИНФРА-М, 1996. – 288 с.
38. Золотой крест [Електронний ресурс] – Режим доступу: <http://vpluse.net/patterny-foreks/347-zolotoj-krest> .
39. Кветний, Р. Н., Кабачій, В. В., Чумаченко, О. О. // Імовірнісні нейронні мережі в задачах ідентифікації часових рядів - 2010. - 2-3 с. / [Електронний ресурс] - Режим доступу: <http://ir.lib.vntu.edu.ua/handle/123456789/4609>
40. Pandas Library [Електронний ресурс] – Режим доступу: <https://uk.wikipedia.org/wiki/Pandas>.
42. NumPy [Електронний ресурс] – Режим доступу: <https://uk.wikipedia.org/wiki/NumPy>
43. Scikit-learn [Електронний ресурс] – Режим доступу: <https://uk.wikipedia.org/wiki/Scikit-learn>
44. XGBoost [Електронний ресурс] – Режим доступу: <https://uk.wikipedia.org/wiki/XGBoost>
- 45 Keras [Електронний ресурс] – Режим доступу: <https://keras.io>
- 46 Pandas TA [Електронний ресурс] – Режим доступу: <https://pypi.org/project/pandas-ta/>
- 47 MT5 Python [Електронний ресурс] – Режим доступу: [https://www.mql5.com/en/docs/integration/python\\_metatrader5](https://www.mql5.com/en/docs/integration/python_metatrader5)
- 48 Є.І. Бакай, В.В. Кабачій // МОДЕЛЬ ПРИЙНЯТТЯ РІШЕНЬ НА ОСНОВІ ПАРИ СЕРЕДНІХ З ВИКОРИСТАННЯМ ОЦІНКИ РІЗНИХ ЧАСОВИХ ВИМІРІВ [Електронний ресурс] – Режим доступу: <http://ir.lib.vntu.edu.ua/bitstream/handle/123456789/13106/Buckeye%20%D0%9A%D0%B0bachiye.pdf?sequence=1&isAllowed=y>

49 LSTM Model [Электронный ресурс] – Режим доступа:  
<https://towardsdatascience.com/lstm-for-time-series-prediction-de8aeb26f2ca>

50 LSTM NN [Электронный ресурс] – Режим доступа:  
<https://datascience.eu/ru/машинное-обучение/понимание-сетей-lstm/>

51 ARIMA [Электронный ресурс] – Режим доступа:  
<https://cyberleninka.ru/article/n/metodika-postroeniya-modeli-arima-dlya-prognozirovaniya-dinamiki-vremennyh-ryadov>

52 Facebook Prophet [Электронный ресурс] – Режим доступа:  
[https://facebook.github.io/prophet/docs/quick\\_start.html](https://facebook.github.io/prophet/docs/quick_start.html)

## ДОДАТКИ



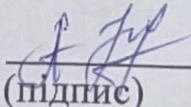
## Додаток А

(обов'язковий)

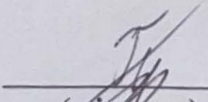

Протокол перевірки МКР на наявність текстових  
запозичень**ПРОТОКОЛ ПЕРЕВІРКИ  
КВАЛІФІКАЦІЙНОЇ РОБОТИ НА  
НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ**Назва роботи: «Прогнозування часових рядів з використанням масивів  
індикаторів у Python»Тип роботи: Магістерська кваліфікаційна робота  
(БДР, МКР)Підрозділ КСУ, ФІТА  
(кафедра, факультет)**Показники звіту подібності Unicheck**Оригінальність 87,8% Схожість 12,2%

Аналіз звіту подібності (відмітити потрібне)

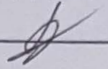
- Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
- Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її автора. Роботу направити на розгляд експертної комісії кафедри.
- Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Особа, відповідальна за перевірку  Галушак А.В.  
(підпис) (прізвище, ініціали)

Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck щодо роботи.

Автор роботи  Гулько А.В.  
(підпис) (прізвище, ініціали)Керівник роботи  Кабачій В.В.  
(підпис) (прізвище, ініціали)

**Додаток Б**  
(обов'язковий)  
Технічне завдання  
ВНТУ

ЗАТВЕРДЖЕНО  
Зав. кафедри КСУ ВНТУ,  
д.т.н., доцент  
 В'ячеслав КОВТУН

“ 03 ” исовтне 2022 р.


**ТЕХНІЧНЕ ЗАВДАННЯ**  
на виконання магістерської кваліфікаційної роботи  
**Прогнозування часових рядів з використанням масивів  
індикаторів у Python**

---

(тема)

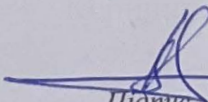
08-33.МКР.025.00.000 ТЗ  
номер

Студент групи ЗАКІТ-21м

  
Підпис

Артур ГУНЬКО

Керівник к.т.н., доц. каф. АІТ

  
Підпис

Владислав КАБАЧІЙ

1. Назва та галузь застосування

1.1. Назва – Прогнозування часових рядів з використанням масивів індикаторів у Python.

1.2. Галузь застосування – прогнозування фінансових часових рядів.

2. Підстава для проведення розробки.

Тема магістерської кваліфікаційної роботи затверджена наказом по ВНТУ від “14” вересня 2022 року №203

3. Мета та призначення розробки.

Метою магістерської кваліфікаційної роботи є підвищення ефективності прикладної задачі штучного інтелекту, яка полягає в прогнозування часових рядів.

4. Джерела розробки.

Магістерська кваліфікаційна робота виконується вперше. В ході проведення розробки повинні використовуватись такі документи:

1. Кветний, Р. Н., Кабачій, В. В., Чумаченко, О. О. // Імовірнісні нейронні мережі в задачах ідентифікації часових рядів – 2010. – 2-3 с. / [Електронний ресурс] – Режим доступу: <http://ir.lib.vntu.edu.ua/handle/123456789/4609>

2. Є.І. Бакай, В.В. Кабачій // МОДЕЛЬ ПРИЙНЯТТЯ РІШЕНЬ НА ОСНОВІ ПАРИ СЕРЕДНІХ З ВИКОРИСТАННЯМ ОЦІНКИ РІЗНИХ ЧАСОВИХ ВИМІРІВ [Електронний ресурс] – Режим доступу:

<http://ir.lib.vntu.edu.ua/bitstream/handle/123456789/13106/Buckeye%20%D0%9A%D0%B0bachiye.pdf?sequence=1&isAllowed=y>

3. Кветний Р. Н. Ідентифікація ринкових залежностей на основі явища дивергенції між ціною та індикатором технічного аналізу / Р. Н. Кветний, В. Ю. Коцюбинський А. В. Шкарпета // Вісник Вінницького політехнічного інституту. – 2006. – №1. – С. 23–34.

5. Вимоги до розробки.

5.1. Перелік головних функцій:

- отримання даних з торгового терміналу.

- формування датасету з технічними індикаторами.
- прогнозування натренованою моделлю вхідних даних.
- вивід даних у вигляді масиву та графіку, їх аналіз.

## 5.2. Основні технічні вимоги до розробки.

### 5.2.1. Вимоги до програмної платформи:

- WINDOWS 10/11;
- Jupyter;
- Бібліотека Tensorflow;
- Python 3.8+.

### 5.2.2. Умови експлуатації системи:

- Можливість запустити застосунок з командного рядку.

## 6. Стадії та етапи розробки.

### 6.1 Пояснювальна записка:

1. Аналіз методів, принципів, підходів і засобів реалізації задачі автоматизації процесами в об'єкті управління відповідно до теми кваліфікаційної роботи.  
Постановка задач дослідження «5» жовтня 2022 р.
2. Дослідження предметної області «5» жовтня 2022 р.
3. Дослідження існуючих моделей нейронних мереж «31» жовтня 2022 р.
4. Розробка застосунку для прогнозування часових рядів з використанням масивів індикаторів «15» листопада 2022 р.

### 6.2 Графічні матеріали:

1. Тестування програмного забезпечення «18» листопада 2022 р.

## 7. Порядок контролю і приймання.

- 7.1. Хід виконання роботи контролюється керівником роботи. Рубіжний контроль провести до «14» грудня 2022 р.
- 7.2. Атестація МКР здійснюється на попередньому захисті. Попередній захист магістерської кваліфікаційної роботи провести до «14» грудня 2022 р.
- 7.3. Підсумкове рішення щодо оцінки якості виконання роботи приймається на засіданні ЕК. Захист магістерської кваліфікаційної роботи провести до «22» грудня 2022 р.

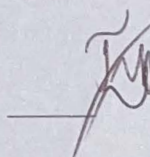

**Додаток В**

(обов'язковий)

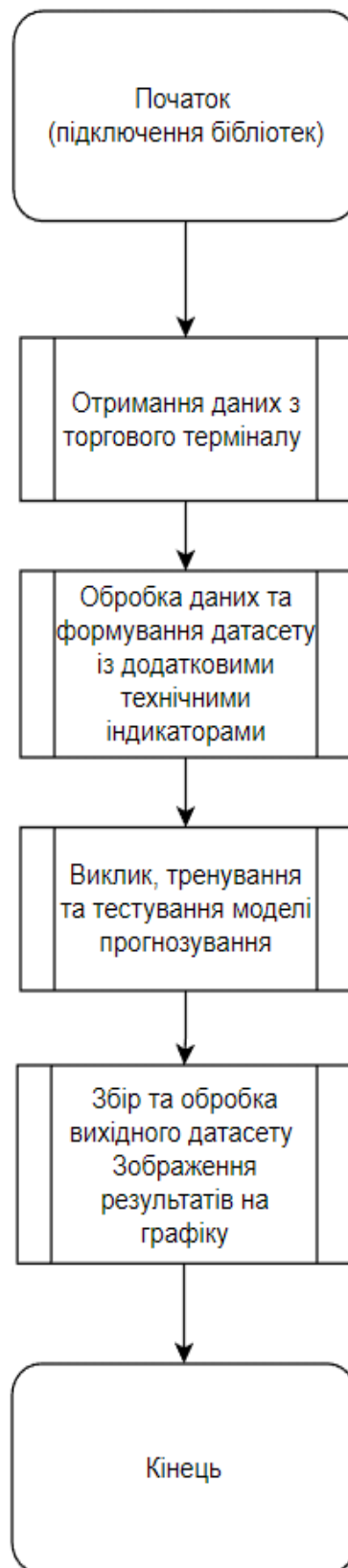
Ілюстративна частина

**ІЛЮСТРАТИВНА ЧАСТИНА****Прогнозування часових рядів з використанням масивів індикаторів у Python**

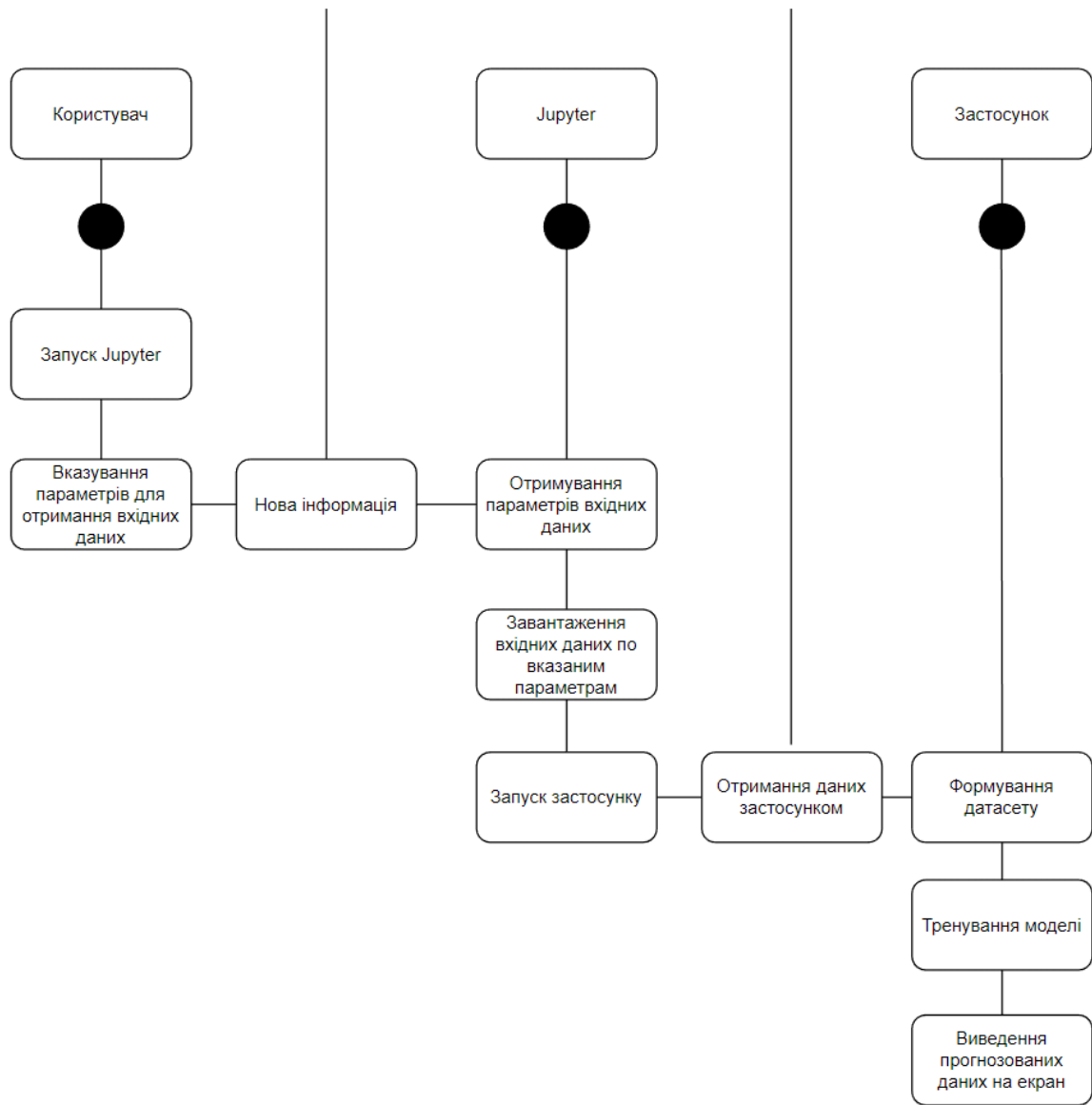
Студент групи ЗАКІТ-21м

Артур ГунькоКерівник к.т.н., доц. каф. АІТВладислав Кабачій

Алгоритм роботи застосунку для прогнозування часових рядів з використанням масивів індикаторів

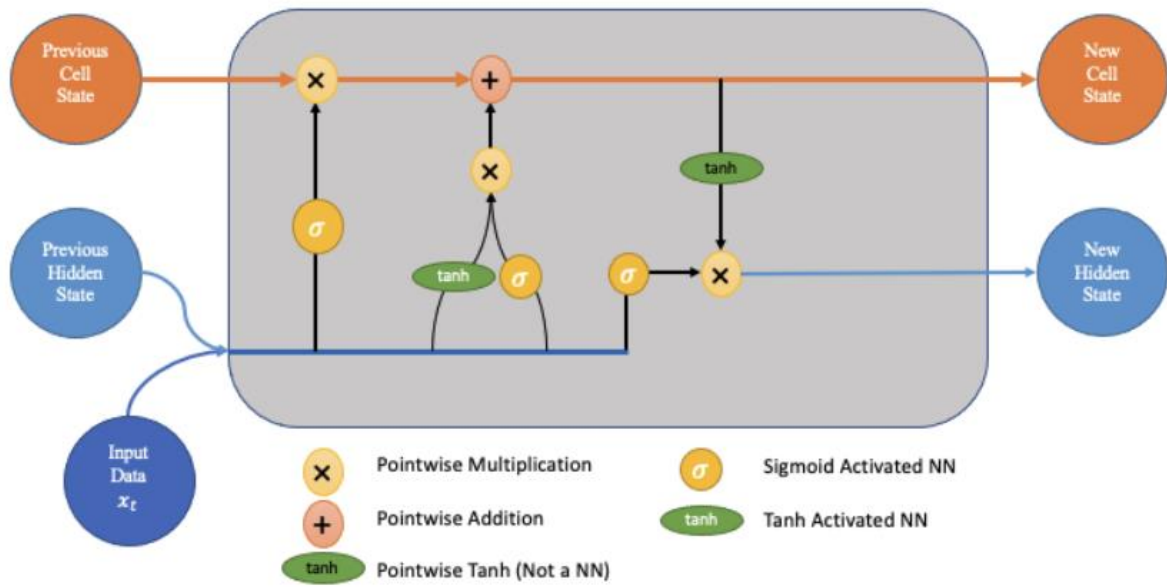


Діаграма діяльності застосунку для прогнозування часових рядів з використанням масивів індикаторів

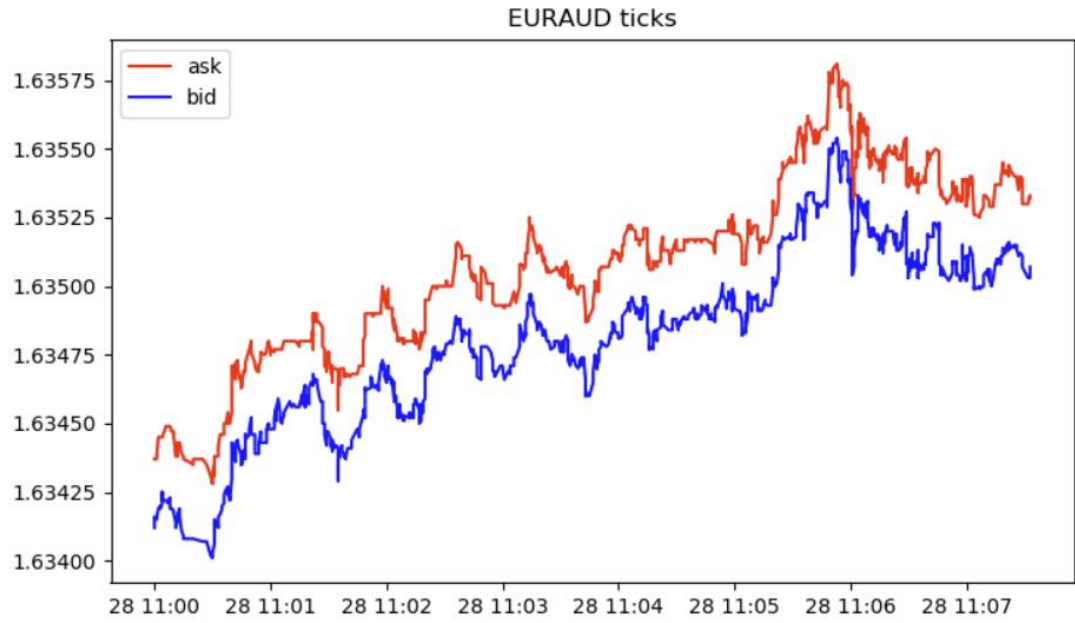




## Схема моделі LSTM



Графік отриманих даних за допомогою торгового терміналу, як результат роботи скрипта



## Дані, отримані за допомогою MetaTrader5 бібліотеки

	time	open	high	low	close	tick_volume	spread	real_volume
100	1564689600	1.10682	1.10957	1.10640	1.10847	27761	2	0
101	1564704000	1.10849	1.10965	1.10743	1.10889	9146	2	0
102	1564718400	1.10890	1.10947	1.10699	1.10822	13528	2	0
103	1564732800	1.10822	1.11150	1.10770	1.10959	18876	2	0
104	1564747200	1.10959	1.11069	1.10783	1.10850	18423	2	0
...	...	...	...	...	...	...	...	...
994	1583121600	1.10566	1.10634	1.10418	1.10432	7949	1	0
995	1583136000	1.10432	1.10932	1.10382	1.10795	17475	1	0
996	1583150400	1.10795	1.11642	1.10775	1.11218	18713	1	0
997	1583164800	1.11219	1.11845	1.11079	1.11796	23128	1	0
998	1583179200	1.11796	1.11832	1.11235	1.11332	9742	1	0

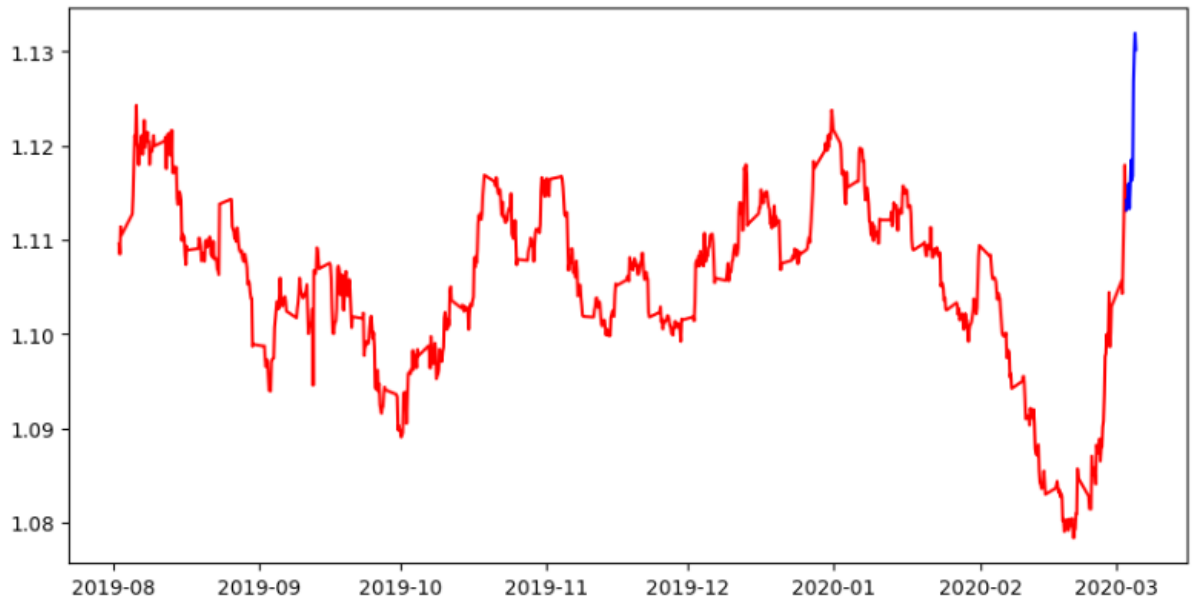
## Вхідний датасет з використанням технічних індикаторів



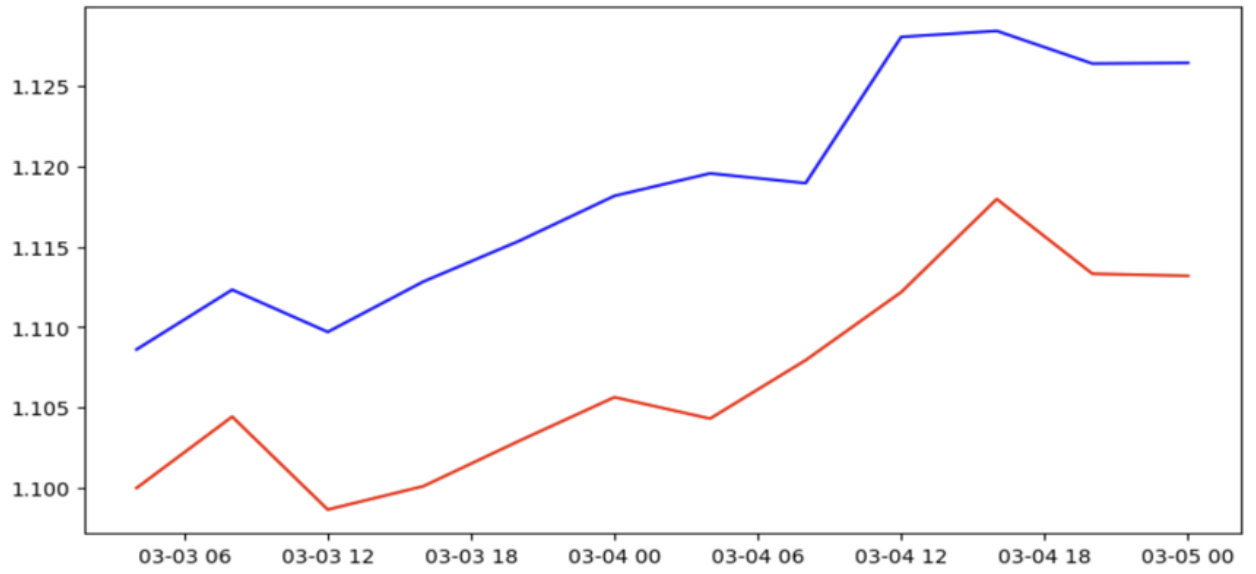
## Шари моделі

```
Model: "sequential"
-----
Layer (type)                Output Shape                Param #
-----
lstm (LSTM)                  (None, 800, 128)           68096
lstm_1 (LSTM)                (None, 64)                  49408
dense (Dense)                (None, 12)                  780
-----
Total params: 118,284
Trainable params: 118,284
Non-trainable params: 0
```

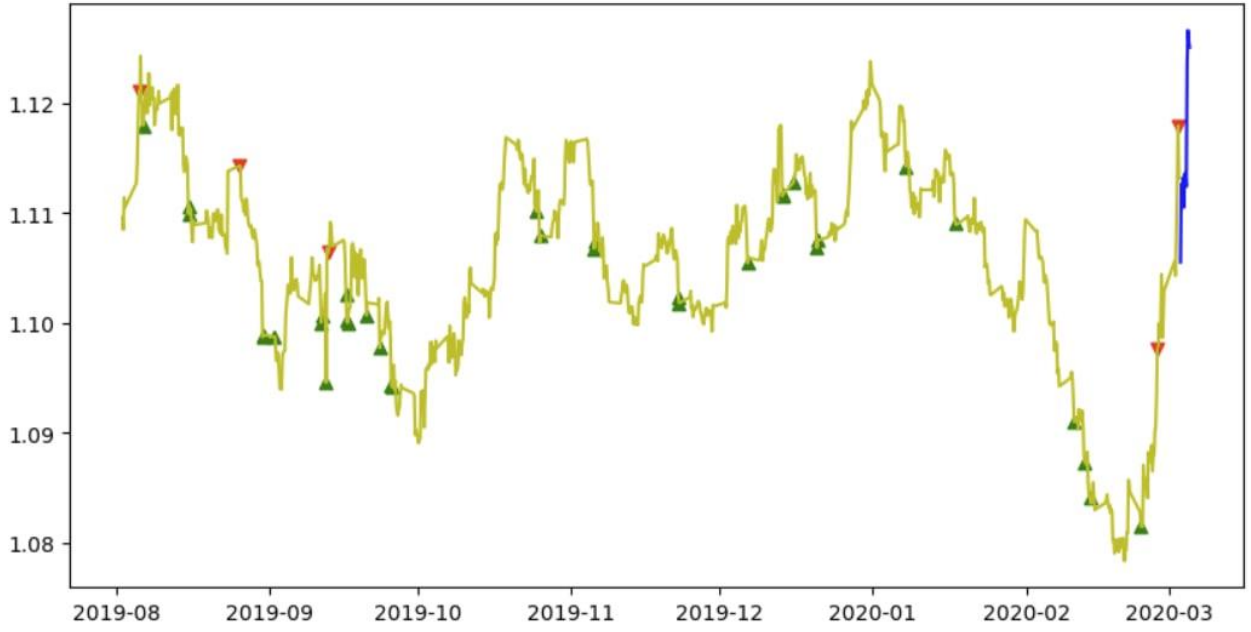
## Прогноз моделі



## Порівняння прогнозованих даних із реальними



Спроектовані сигнали купівлі/продажу та прогноз моделі



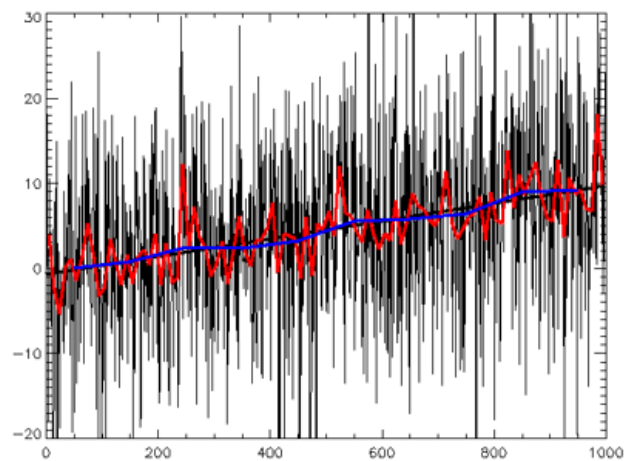


# ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ З ВИКОРИСТАННЯМ МАСИВІВ ІНДИКАТОРІВ У PYTHON

Гулько А.В.

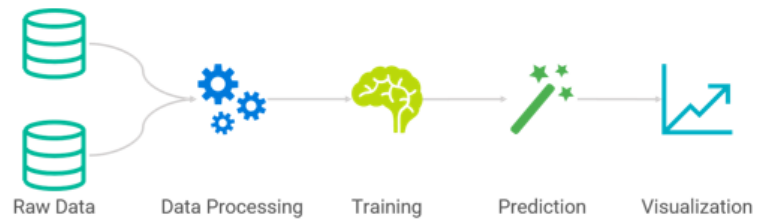
## Часовий ряд

- Часовий ряд – це серія точок даних, проіндексованих у хронологічному порядку. У більшості випадків часовий ряд – це послідовність, взята в рівновіддалені моменти часу, які слідують одна за одною. Таким чином, це послідовність даних у дискретному часі.



## Огляд проблеми

- Вхідні дані, представлені у вигляді дата-сетів, в які входять також масиви показників індикаторів. Крім того відомі моменти за певний проміжок часу, в які було б бажано прийняти те чи інше рішення. Необхідно створити підхід, базуючись на методах прогнозування часових рядів і за використанням бібліотек Python, та навчити нейронну мережу прогнозувати майбутні значення на основі даних в реальному часі: цінових вимірів та набору значень індикаторів.

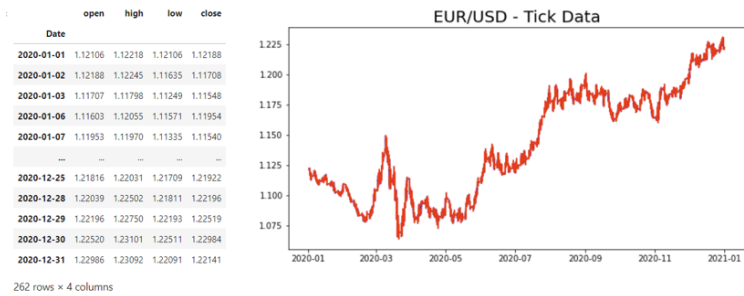


## Підхід до вирішення проблеми

- Для вирішення проблеми запропоновано даний алгоритм дій:
- інтеграція з торговим терміналом MT5 для отримання масиву даних; пакет MetaTrader використовується для зручного та швидкого отримання біржової інформації в Python. Отримані таким чином дані можна використовувати для статистичних обчислень і машинного навчання.
- прогін даних, технічних індикаторів (запропоновані MT5 або створені за допомогою Python), формування дата сету для моделі та нейронної мережі[39]: дані потрібно привести до вигляду, який "розуміє" нейронна мережа, виділити основні дані масиву, доповнити масив технічними індикаторами (пошук входів за стратегією Еллера [40]).
- навчання рекурентної нейронної мережі та її тестування, прогнозування за допомогою моделі LSTM: за допомогою цих інструментів будемо графік, оснований на попередніх даних, таким чином визначаємо майбутню ціну валютної пари.
- аналіз результатів та тестування моделі з різними наборами даних.

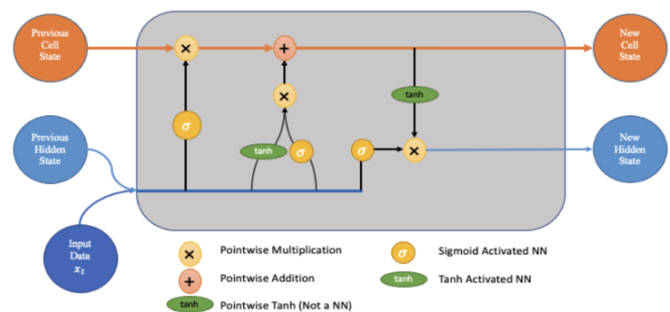
## Отримання даних з торгового терміналу

- Пакет MetaTrader для Python призначений для зручного та швидкого отримання біржової інформації через міжпроцесну взаємодію прямо з терміналу MetaTrader 5. Отримані таким чином дані використано для статистичних обчислень і машинного навчання. За тестові дані було взято курс валютної пари EUR/USD за 01.01.20 - 31.12.20



## Опис LSTM моделі

- Використовуючи LSTM, моделі прогнозування часових рядів можуть передбачати майбутні значення на основі попередніх послідовних даних. Це забезпечує більшу точність для прогнозистів попиту, що призводить до кращого прийняття рішень для бізнесу.



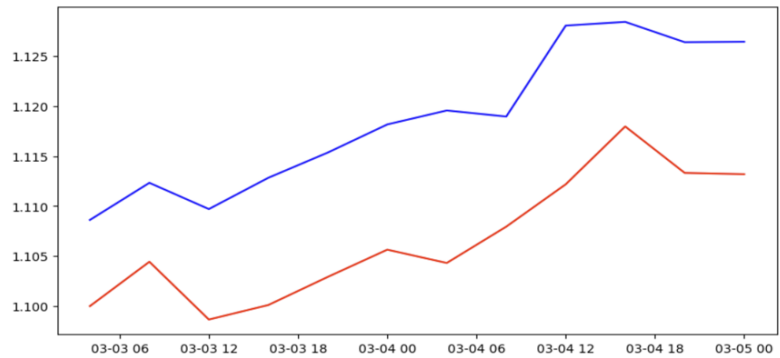
## Формування вхідного дата-сету

- Дата-сет було сформовано із використанням технічних індикаторів SMA\_10 та SMA\_30, посилаючись на стратегію Елдера



## Результати моделювання

- Після тренування нейромережі можна спостерігати графік, де є розходження з реальними даними. Середня похибка передбачених позицій ціни Close складає 150 пункти, що є дуже суттєвою саме при передбаченні майбутньої ціни. При цьому спостерігається збереження тренду, що приводить використання моделі у якості прогнозування сигналів на продаж/покупку.



## Результати моделювання

- Спроектвані сигнали модель віддає на основі сформованої допоміжної колонки із Signal, який обчислюється за формулою в залежності від трьох попередніх позицій ціни.  
Даний сигнал виступає в межах 0 та 1, де 1 – Продаж, А 0 - Купівля



## Додаток Г

### Лістинг програми

```

import numpy as np
import pandas as pd
import pandas_ta as ta
import pytz
!pip install MetaTrader5
import MetaTrader5 as mt
import matplotlib.pyplot as plt
import tensorflow as tf
print("Num GPUs Available: ", len(tf.config.experimental.list_physical_devices('GPU')))
tf.test.is_gpu_available()
from tensorflow.keras.layers import Dense, LSTM
from tensorflow.keras.models import Sequential
from sklearn.preprocessing import MinMaxScaler
from datetime import datetime
pd.options.mode.chained_assignment = None

#register_matplotlib_converters()
import MetaTrader5 as mt5

# підключимся к MetaTrader 5
if not mt5.initialize():
    print("initialize() failed")
    mt5.shutdown()

# запросим статус и параметры подключения
print(mt5.terminal_info())
# получим информацию о версии MetaTrader 5
print(mt5.version())

# запросим 1000 тиков с EURAUD

symbol_info = mt5.symbol_info("EURUSD")
#actual data for future predict
#rates = mt5.copy_rates_from_pos("EURUSD", mt5.TIMEFRAME_D1, 0, 1000)

#test data
timezone = pytz.timezone("Etc/UTC")
# создадим объект datetime в таймзоне UTC, чтобы не применялось смещение локальной таймзоны
utc_from = datetime(2020, 3, 3, tzinfo=timezone)
# получим 10 баров с EURUSD H4 начиная с 01.10.2017 в таймзоне UTC
rates = mt5.copy_rates_from("EURUSD", mt5.TIMEFRAME_H4, utc_from, 1000)

df = pd.DataFrame(rates)
sma10 = df.ta.sma(10)
sma30 = df.ta.sma(30)

df = pd.concat([df, sma10, sma30], axis=1)
df.head(-1)
df = df.iloc[100:]
df.head(-1)
plt.figure(figsize=(15,10))
plt.plot(range(len(df)),df['close'], '-r', label="Real")
# plot 'buy' signals

plt.plot(range(len(df)),df['SMA_10'], '-y', label="SMA_10")
plt.plot(range(len(df)),df['SMA_30'], '-g', label="SMA_30")

```

```

plt.legend(loc="upper right")
plt.title('Real Data with TA')
plt.show()
series_shifted = df['close'].shift()

series_shifted = series_shifted/100

df['per'] = (df['close']/series_shifted - 100) * 100
df['signal'] = 0

for i in range(102,len(df)+100):
    if i>2:
        df['signal'].loc[i] = df.per.loc[i-2]+df.per.loc[i-1]+df.per.loc[i]
pd.set_option('display.max_rows', None)
df = df.iloc[3:]

df.signal -= df.signal.min()
df.signal /= df.signal.max()
print(df.signal)
y = df[['close','low','SMA_10','SMA_30']].fillna(method='ffill')
y_1 = df[['signal','close','SMA_10','SMA_30']].fillna(method='ffill')
print(y)
# scale the data
scaler = MinMaxScaler(feature_range=(0, 1))
scaler = scaler.fit(y)
y = scaler.transform(y)
print(y.shape)
# generate the input and output sequences
n_lookback = 800 # length of input sequences (lookback period)
n_forecast = 12 # length of output sequences (forecast period)

X = []
Y = []

for i in range(n_lookback, len(y) - n_forecast + 1):
    X.append(y[i - n_lookback: i])
    Y.append(y[i: i + n_forecast])

X = np.array(X)
Y = np.array(Y)

print(X.shape, Y.shape)
# fit the model
model = Sequential()
model.add(LSTM(units=128, return_sequences=True, input_shape=(n_lookback, 4)))
model.add(LSTM(units=64))
model.add(Dense(n_forecast))

model.compile(loss='mean_squared_error', optimizer='adam')

model.summary()

history = model.fit(X,Y[:, :,0], epochs=40, batch_size=32, verbose=1)

# generate the forecasts
X_ = y[- n_lookback:] # last available input sequence

X_ = X_.reshape(-1, n_lookback, 4)
print(X_.shape)

```

```

Y_ = model.predict(X_)
Y_ = Y_.reshape(-1,4)
print(Y_.shape)
Y_ = scaler.inverse_transform(Y_)
print(Y_)

print(Y_.flatten())
# organize the results in a data frame
df_past = df[['close', 'time']].reset_index()

df_past.rename(columns={'close': 'Actual'}, inplace=True)

df_past['Date'] = pd.to_datetime(df_past['time'], unit='s')

df_past['Forecast'] = np.nan
df_past['Forecast'].iloc[-1] = df_past['Actual'].iloc[-1]

df_future = pd.DataFrame(columns=['Date', 'Actual', 'Forecast'])

df_future['Date'] = pd.date_range(start=df_past['Date'].iloc[-1] + pd.Timedelta(hours=4), freq="4H",
periods=n_forecast)
df_future['Forecast'] = Y_.flatten()

results = df_past.append(df_future).set_index('Date')

plt.figure(figsize=(10,5))
plt.plot(results.index,results['Actual'], '-r', label="Real")
plt.plot(results.index,results['Forecast'], '-b', label="Forecast")
# plot the results
#results.plot(title='AAPL')

#test data
timezone = pytz.timezone("Etc/UTC")
# создадим объект datetime в таймзоне UTC, чтобы не применялось смещение локальной таймзоны
utc_from = datetime(2020, 3, 3, tzinfo=timezone)
# получим 10 баров с EURUSD H4 начиная с 01.10.2017 в таймзоне UTC
rates = mt5.copy_rates_from("EURUSD", mt5.TIMEFRAME_H4, utc_from, 1012)

df_1 = pd.DataFrame(rates)

plt.figure(figsize=(10,5))
plt.plot(df_1.index,df_1['close'], '-b', label="real")
df_10 = df_1.tail(12)
plt.figure(figsize=(10,5))
plt.plot(df_future.index,df_future['Forecast'], '-b', label="Forecast")

plt.plot(df_future.index,df_10['close'], '-r', label="real")

results.shape
print(results)

```