

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

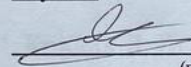
МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Інформаційна технологія прогнозування курсу криптовалюти «Біткоїн»»

Виконав: студент 2-го курсу, групи
2КН-22м,

спеціальності 122 – «Комп'ютерні
науки»



Герус Я. К.

(прізвище та ініціали)

Керівник: к.т.н., доцент каф. КН



Крилик Л. В.

(прізвище та ініціали)

« 07 » 12 2023 р.

Опонент: к.т.н., доцент каф. АІТ



Богач І. В.

(прізвище та ініціали)

« 07 » 12 2023 р.

Допущено до захисту

Завідувач кафедри КН

д.т.н., проф. Яровий А. А.

(прізвище та ініціали)

« 08 » 12 2023 р.

Вінниця - 2023 року

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра комп'ютерних наук
Рівень вищої освіти II-й (магістерський)
Галузь знань – 12 Інформаційні технології
Спеціальність – 122 Комп'ютерні науки
Освітньо-професійна програма – Системи штучного інтелекту

ЗАТВЕРДЖУЮ

Завідувач кафедри КН
д.т.н., проф. Яровий А. А.

(підпис)

« 29 » 08 2023 року

ЗАВДАННЯ

НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Герусу Ярославу Костянтиновичу

(прізвище, ім'я, по батькові)

1. Тема роботи: «Інформаційна технологія прогнозування курсу криптовалюти «Біткоїн»»

Керівник роботи к.т.н., доц., доц. каф. КН, Крилик Л. В.

затверджені наказом вищого навчального закладу від «18» 03. 2023 року №247

2. Строк подання студентом роботи 13 листопада 2023 року


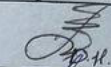
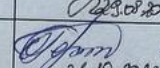
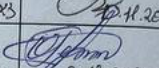
3. Вихідні дані до роботи: мова програмування – об'єктно-орієнтована; мінімальний період прогнозування – 1 година; час від подачі заявки до отримання результату прогнозування не перевищує 2 хвилини; інтерфейс користувача має бути інтуїтивно зрозумілим.

4. Зміст текстової частини: вступ, обґрунтування доцільності розробки інформаційної технології прогнозування курсу криптовалюти «Біткоїн»; моделювання інформаційної технології прогнозування курсу криптовалюти «Біткоїн»; програмна реалізація інформаційної технології прогнозування курсу криптовалюти «Біткоїн»; економічна частина, висновки, список використаних джерел, додатки.

5. Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень): діаграма послідовності взаємодії модулів інформаційної технології прогнозування курсу криптовалюти «Біткоїн»; загальна структурна модель програмного модуля; схема алгоритму побудови математичної моделі прогнозування курсу криптовалюти «Біткоїн»; схема загального алгоритму функціонування програмного модуля прогнозування курсу криптовалюти

«Біткоїн»; діаграма діяльності серверу під час процесу автентифікації користувача; діаграма діяльності серверу під час процесу перевірки на стаціонарність; загальний вигляд інтерфейсного вікна роботи програми.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	виконання прийняв
1-3	Крилик Л.В., к.т.н., доц. каф. КН	 29.08.2023	 29.08.2023
4	Ратушняк О. Г., к.т.н, доц. каф. ЕПВМ	 01.10.2023	 08.10.2023

7. Дата видачі завдання 29.08, 2023 року

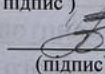
КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Обґрунтування доцільності розробки	01.09.23 - 05.09.23	Розділ 1
2	Моделювання інформаційної технології прогнозування курсу криптовалюти «Біткоїн»	06.09.23 - 16.09.23	Розділ 2
3	Програмна реалізація інформаційної технології прогнозування курсу криптовалюти «Біткоїн»	17.09.23 - 07.10.23	Розділ 3
4	Підготовка економічної частини	08.10.23 - 23.10.23	Розділ 4
5	Апробація результатів дослідження	24.10.23 - 01.11.23	тези доповіді, авторське право на твір
6	Оформлення пояснювальної записки, графічного матеріалу та презентації	02.11.23 - 10.11.23	Пояснювальна записка, графічний матеріал, презентація

Студент


(підпис)

Керівник роботи


(підпис)

Герус Я. К.
(прізвище та ініціали)

Крилик Л. В.
(прізвище та ініціали)

АНОТАЦІЯ

УДК 004.42

Герус Я. К. Інформаційна технологія прогнозування курсу криптовалюти «Біткоїн». Магістерська кваліфікаційна робота зі спеціальності 122 Комп'ютерні науки, освітня програма – Системи штучного інтелекту. Вінниця: ВНТУ, 2023. 127 с.

Укр. мовою. Бібліогр.: 40 назв; рис.: 22; табл.: 13.

У магістерській кваліфікаційній роботі проведено аналіз предметної галузі криптовалют. Досліджено основні підходи щодо прогнозування курсу криптовалюти «Біткоїн». Виконано аналіз аналогів, розглянуто методи та технології, які використовуються для вирішення подібних задач та обґрунтована доцільність розробки. Розроблено структурну модель інформаційної технології та математичну авторегресійну модель інтегрованого ковзного середнього для прогнозування курсу криптовалюти «Біткоїн», що сприяє розширенню функціональних можливостей інформаційної технології, а також збільшує точність прогнозування.

Ця робота дає можливість користувачу отримати новий інструмент для аналізу криптовалютного ринку та інструмент для перевірки своїх стратегій торгівлі зі зручним та інтуїтивно зрозумілим дизайном.

В роботі приведені результати досліджень та аналіз роботи розробленої інформаційної технології прогнозування курсу криптовалюти «Біткоїн». Доведена економічна доцільність розробки.

Ілюстративна частина складається з 7 плакатів із результатами моделювання.

Ключові слова: криптовалюти, Біткоїн, прогнозування, інформаційна технологія, сервіси, часовий ряд.

ABSTRACT

Gerus Y. K. Information technology of forecasting the exchange rate of the cryptocurrency «Bitcoin». Master's thesis in the specialty 122 Computer science, educational program «Artificial intelligence systems». Vinnytsia: VNTU, 2023. 127 p.

In Ukrainian language. Bibliogr.: 40 titles; Fig.: 22; 13 tables.

In the master's qualification thesis, an analysis of the subject area of cryptocurrencies was carried out. The main approaches of forecasting the exchange rate of the cryptocurrency "Bitcoin" have been studied. An analysis of analogues was carried out, methods and technologies used to solve similar problems were considered. A structural model of information technology and a mathematical autoregressive model of an integrated moving average have been developed for forecasting the exchange rate of the cryptocurrency "Bitcoin", which contributes to the expansion of the functional capabilities of information technology, and also increases the accuracy of forecasting.

This work allows the user to get a new tool for analyzing the cryptocurrency market and a tool for checking their trading strategies with a convenient and intuitive design.

The work presents the results of research and analysis of the information technology of forecasting the exchange rate of the cryptocurrency "Bitcoin". The economic feasibility of the development has been proven.

The illustrative part consists of 7 posters with simulation results.

Keywords: cryptocurrencies, Bitcoin, forecasting, information technology, services, time series.

ЗМІСТ

ВСТУП	4
1 ОБҐРУНТУВАННЯ ДОЦІЛЬНОСТІ РОЗРОБКИ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ПРОГНОЗУВАННЯ КУРСУ КРИПТОВАЛЮТИ «БІТКОЇН»	8
1.1 Аналіз предметної галузі криптовалют	8
1.2 Огляд відомих програмних систем-аналогів.....	11
1.3 Огляд методів та програмних засобів для прогнозування курсу криптовалют.....	15
1.4 Постановка задачі дослідження.....	21
1.5 Висновок до розділу 1	22
2 МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ПРОГНОЗУВАННЯ КУРСУ КРИПТОВАЛЮТИ «БІТКОЇН».....	24
2.1 Обґрунтування вибору методу прогнозування курсу криптовалюти «Біткоїн».....	24
2.2 Математична модель прогнозування курсу криптовалюти «Біткоїн»	27
2.3 Розробка UML-діаграми послідовності інформаційної технології прогнозування курсу криптовалюти «Біткоїн».....	31
2.4 Висновок до розділу 2	32
3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ПРОГРАМУВАННЯ КУРСУ КРИПТОВАЛЮТИ «БІТКОЇН».....	33
3.1 Обґрунтування вибору архітектури інформаційної технології прогнозування курсів криптовалюти «Біткоїн»	33
3.2 Проектування структури та розробка алгоритму функціонування інформаційної технології прогнозування курсу криптовалюти «Біткоїн»	35
3.3 Обґрунтування вибору мови програмування	40
3.4 Обґрунтування вибору фреймворків та бібліотек для розробки.....	48
3.5 Реалізація програмного модуля інформаційної технології прогнозування курсів криптовалюти «Біткоїн»	57

3.6 Тестування інформаційної технології прогнозування курсу криптовалюти «Біткоїн» та аналіз отриманих результатів	63
3.7 Висновок до розділу 3	70
4 ЕКОНОМІЧНА ЧАСТИНА	72
4.1 Проведення комерційного та технологічного аудиту науково-технічної розробки	72
4.2 Визначення рівня конкурентоспроможності розробки	76
4.3 Розрахунок витрат на проведення науково-дослідної роботи.....	79
4.3.1 Витрати на оплату праці.....	79
4.3.2 Відрахування на соціальні заходи	82
4.3.3 Сировина та матеріали.....	82
4.3.4 Спецустаткування для наукових (експериментальних) робіт	83
4.3.5 Програмне забезпечення для наукових (експериментальних) робіт	84
4.3.6 Амортизація обладнання, програмних засобів та приміщень	85
4.3.7 Паливо та енергія для науково-виробничих цілей	86
4.3.8 Службові відрядження.....	87
4.3.9 Інші витрати.....	87
4.3.10 Накладні (загальновиробничі) витрати.....	88
4.4 Розрахунок економічної ефективності науково-технічної розробки при її можливій комерціалізації потенційним інвестором	89
4.5 Висновок до розділу 4	94
ВИСНОВКИ.....	95
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	97
Додаток А (обов'язковий) Протокол перевірки кваліфікаційної роботи на наявність текстових запозичень.....	101
Додаток Б (обов'язковий) Лістинг програми	102
Додаток В (обов'язковий) Ілюстративна частина.....	117
Додаток Г (довідниковий) Інструкція користувача.....	123

ВСТУП

Актуальність теми. Сьогодні спостерігається трансформація традиційного світу грошей та фінансів. Інновації у фінансовому секторі, нові технології, інструменти та системи «тягнуть» за собою глибокі зміни звичних нам фінансових інститутів. Одним із найважливіших етапів цих трансформаційних процесів стала поява криптовалют, що супроводжується стрімким розвитком пов'язаних з ними технологій та відсутністю чіткої картини майбутнього виду екосистеми. Нині держави демонструють неоднозначне ставлення до криптовалют. Аналіз підходів країн до регулювання криптовалютних ринків показав, що єдиної позиції щодо правового статусу розглянутого виду цифрових грошей натепер у світовому співтоваристві не вироблено. Немає також і єдиного визначення самого поняття цифрових грошей, отже, і інструменти його регулювання змінюються у різних країнах та інститутах у межах одних територіальних кордонів. Єдиним питанням, стосовно якого у позиціях країн простежується спільна думка, це необхідність створення збалансованої нормативно-правової бази, визначення заходів та інструментів для запобігання використанню криптовалют у злочинних цілях [1].

Правовий статус криптовалют істотно відрізняється у різних країнах. Багато хто з них розглядає цифрову валюту як товар або інвестиційний актив. Ряд країн визнав криптовалюту як розрахункову грошову одиницю (наприклад, Японія). При цьому Комісія з цінних паперів Сполучених Штатів Америки (SEC) поки не визначила правовий статус криптовалют та застерігає своїх громадян, що не зможе захистити їх в разі потрапляння коштів в руки зловмисників [2]. В Україні як, і в багатьох країнах, відсутній законодавчий статус криптовалюти. Але попри те, що відсутнє правове визначення криптовалюти, вона почала набирати значну популярність саме як форма розрахунку.

Зв'язок роботи з науковими програмами, планами, темами. Магістерська кваліфікаційна робота виконана відповідно до напрямку наукових

досліджень кафедри комп'ютерних наук Вінницького національного технічного університету 22 К1 «Розробка прикладних інтелектуальних інформаційних технологій та систем».

Мета і завдання дослідження. Метою магістерської кваліфікаційної роботи є розширення функціональних можливостей інформаційної технології прогнозування курсу криптовалюти «Біткоїн» за рахунок використання нових часових проміжків горизонту прогнозування. Це дозволить розробити зручний клієнтський додаток для інформаційної технології прогнозування курсу криптовалюти «Біткоїн», який відповідатиме всім запитам користувача, буде інтуїтивно зрозумілим та зможе конкурувати з іншими системами-аналогами на ринку.

Для досягнення поставленої мети потрібно розв'язати такі **завдання**:

- Провести аналіз технічного рівня існуючих систем прогнозування та обґрунтувати доцільність розробки інформаційної технології прогнозування курсу криптовалюти «Біткоїн».
- Розробити математичну модель інформаційної технології прогнозування курсу криптовалюти «Біткоїн».
- Розглянути методи та технології прогнозування курсу криптовалюти «Біткоїн».
- Спроекувати архітектуру та структуру програмного модуля прогнозування курсу криптовалюти «Біткоїн».
- Проаналізувати та обґрунтувати вибір інструментів для розробки.
- Виконати програмну реалізацію програмного модуля інформаційної технології прогнозування курсу криптовалюти «Біткоїн».
- Виконати тестування програми та провести аналіз отриманих результатів.
- Економічно обґрунтувати доцільність розробки інформаційної технології прогнозування курсу криптовалюти «Біткоїн».

Об'єктом дослідження є процес прогнозування курсу криптовалюти «Біткоїн».

Предметом дослідження є програмні засоби проектування та реалізації інформаційної технології прогнозування курсу криптовалюти «Біткоїн».

Методи дослідження. У процесі дослідження використовувались: дослідження сучасних методів прогнозування, дослідження методів обробки масивів інформації, дослідження методів аналізу отриманих даних, методи та підходи до розробки веб-орієнтованих програмних додатків, методи об'єктно-орієнтованого програмування.

Наукова новизна одержаних результатів полягає в наступному: вдосконалено інформаційну технологію прогнозування курсу криптовалюти «Біткоїн», яка відрізняється від існуючих використанням математичної авторегресійної моделі інтегрованого ковзного середнього для прогнозування курсу криптовалюти. Використання цієї моделі дозволило розширити функціональні можливості інформаційної технології, а також забезпечило збільшення точності прогнозування.

Практичне значення одержаних результатів полягає у наступному:

1. Розроблено алгоритм функціонування інформаційної технології прогнозування курсу криптовалюти «Біткоїн».
2. Розроблено діаграму послідовності взаємодії модулів інформаційної технології прогнозування курсу криптовалюти «Біткоїн».
3. Здійснено програмну реалізацію інформаційної технології прогнозування курсу криптовалюти «Біткоїн».

Достовірність теоретичних положень магістерської кваліфікаційної роботи підтверджується строгістю постановки задач, коректним застосуванням методів під час доведення наукових положень, порівнянням результатів із

відомими та збіжністю результатів моделювання з результатами, що отримані під час впровадження розроблених програмних засобів.

Особистий внесок магістранта. Результати магістерської кваліфікаційної роботи отримані самостійно. В публікації у співавторстві здобувачу належить дослідження особливостей розробки інформаційної технології прогнозування курсу криптовалюти «Біткоїн» [1, 2].

Апробація результатів роботи. Результати досліджень було апробовано на ЛП науково-технічній конференції підрозділів Вінницького національного технічного університету (НТКП ВНТУ–2023) м. Вінниці у 2023 р.

Публікації. За результатами магістерської кваліфікаційної роботи опубліковано тези доповідей на науково-технічних конференціях [1, 2] та подано заявку на реєстрацію авторського права на твір у формі комп'ютерної програми – «Інформаційна технологія прогнозування курсу криптовалюти «Біткоїн»», номер заявки с202307799 від 09.11.2023 р.

1 ОБҐРУНТУВАННЯ ДОЦІЛЬНОСТІ РОЗРОБКИ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ПРОГНОЗУВАННЯ КУРСУ КРИПТОВАЛЮТИ «БІТКОЇН»

1.1 Аналіз предметної галузі криптовалют

Криптовалюта – технологія, яка сприяє обміну коштами в електронній формі. Основна ідея цієї технології полягає у тому, що вона дає можливість проводити обміни без участі регулюючої сторони, яка підтримувала би інфраструктуру (наприклад, банк). Водночас криптовалюта дає змогу безпечно проводити обмін ресурсами між користувачами, які не довіряють один одному. Така поведінка досягається тим, що роль контролю за операціями всередині криптовалютної мережі перекладена з централізованих фінансових інститутів на мережу автономних комп'ютерів, які утворюють децентралізовану систему, що працює поза впливом будь-якого інституту управління. За своєю суттю криптовалюти побудовані навколо ідеї універсального, незмінного обліку, який є повністю публічним і постійно перевіряється мережею потужних комп'ютерів, що працюють незалежно один від одного [3]. Частіше за все криптовалюти мають плаваючий курс до національних валют та між собою.

Перевагами криптовалют є:

- Відкритість коду. Завдяки тому, що основні криптовалюти мають відкритий код, який може переглянути будь хто забезпечується однаковою функціональна можливість для всіх користувачів мережі, система є прозорою і відкритою для всіх і кожен може користуватись мережею.
- Анонімність. Порівнюючи зі звичайними електронними фіатними операціями, які можуть відслідковуватись керуючими органами, дізнатись інформацію про власника криптовалютного гаманця складніше – у відкритому доступі є лише адреса гаманця і дані про транзакції.

- Децентралізація. Криптовалюта є незалежною системою без єдиного центру управління. Її емісія і правила проведення транзакцій регулюється відкритим програмний кодом, встановленим на комп'ютерах мережі.
- Обмеженість. Зазвичай, емісія основних криптовалют обмежена набором деяких, вказаних в відкритому коді, математичних правил. Це дає можливість криптовалютам рости в майбутній перспективі і позитивно впливає на безпеку.
- Надійність. Натепер зламати чи підробити систему, створену з тисяч комп'ютерів користувачів вважається неможливим.

Недоліками є:

- Відсутність гарантій. Оскільки керуючого органу просто не існує і користувачі персонально несуть відповідальність за свої кошти, вся відповідальність лягає на плечі користувача. Такі притаманні звичайним фінансовим інститутам інструменти як повернення коштів у криптовалютній мережі просто неможливі.
- Волатильність. Курс криптовалюти є досить нестабільним. Він залежить від поточного попиту користувачів, легко маніпулюється лідерами думок і гостро реагує на зовнішні чинники: політичні, економічні, інформаційні.
- Ризик заборони. Нині правовий статус криптовалют є досить хитким. Не має однозначної чіткої позиції фінансових інститутів багатьох держав щодо використання криптовалют [4].
- Небезпека втрати. Доступ до електронного гаманця здійснюється за допомогою спеціально згенерованого ключа, у разі його втрати доступ до коштів буде втрачено назавжди.

Криптовалюта – один з найновіших, найсучасніших і найменш вивчених способів для інвестування у фінансовий ринок, оскільки дозволяє людям з обмеженими знаннями в технології блокчейну та області інформаційних технологій проводити фінансові операції без участі брокерів, банків чи інших фінансових установ. Проводити транзакції у криптовалютній мережі може будь-яка людина з доступом до комп'ютеру та інтернету. Не дивлячись на високі

ризика інвестування, з кожним днем велика кількість людей вкладає свої кошти в цей ринок, тим самим підтримує роботу системи і підвищує популярність криптовалютної сфери.

Більш формально, криптовалюта – це система, яка відповідає таким умовам:

- Система не потребує головного контролюючого органу, її роботу забезпечують самі користувачі системи.
- Система веде облік криптовалютних одиниць та надає доступ до історії транзакцій всередині системи.
- Система визначає, чи можна створювати нові криптовалютні одиниці. При цьому вона визначає умови їх виникнення та спосіб визначення власності на ці нові одиниці.
- Право власності на криптовалютні одиниці можна довести лише криптографічно.
- Система дозволяє здійснювати транзакції, в яких змінюється право власності на криптографічні одиниці. Транзакція може бути записана лише суб'єктом, який підтверджує поточне право власності на ці одиниці.
- Якщо одночасно вводяться дві різні інструкції щодо зміни власності на одні і ті ж криптографічні одиниці, система виконує щонайбільше одну з них [5].

Bitcoin є першою і найбільш популярною криптовалютою. Вперше концепт біткоїну був описаний в 2008 році Сатоші Накамото. Достеменно невідомо чи справді існує така людина, чи можливо це група людей під вигаданим псевдонімом. Сатоші описував Біткоїн як валюту, якою могла володіти будь-яка людина в світі і використовувати її для будь-яких трат. Більше всього такі гроші нагадували цифровий аналог золота. Система була спроектована так, що як і золото біткоїни були в невеликій кількості і їх неможливо підробити [6].

Найвідоміша криптовалюта «Біткоїн» в листопаді 2023 перевищила значення в \$36 000 з сумарною капіталізацією в більш ніж \$717 000 000. Цікавою

особливістю емісії біткоїна є закладене в відкритий код обмеження на загальну емісію кількості криптовалюти, тобто з кожною новою згенерованою одиницею біткоїну, кількість монет, які можна добути зменшується. Такі правила дозволяють рости курсу біткоїна. Через активне інвестування коштів в біткоїн, починають набирати популярність й інші криптовалюти, які мають інші правила видобутку, стратегію розвитку тощо. Оскільки біткоїн був першою криптовалютою, на основі якої почали розвиватись усі інші, то такі криптовалюти називають альткоїнами. Зараз на ринку існує велика кількість токенів різних альткоїнів. Однак, більшість криптовалют не користуються таким попитом, який є у біткоїна, оскільки вони не надають нових технологічних рішень і мають низьку капіталізацію. Курс багатьох нових криптовалют залежить від маркетингу, а не технологій, що дає можливість прогнозувати їх курс за допомогою аналізу різних зовнішніх чинників.

1.2 Огляд відомих програмних систем-аналогів

Першим аналогом для прогнозування курсу криптовалюти «Біткоїн» є «Walletinvestor» [7]. Компанія надає можливість отримати короткий або довготривалий прогноз курсу на своєму сайті. Вона спеціалізується на прогнозуванні активів фондового ринку, тому точність прогнозування нової для них криптовалютної сфери може бути невисокою. Також система дає можливість прогнозувати курс лише на довгі горизонти прогнозування (більше місяця), короткострокові прогнозування, наприклад на одну годину, є платною функцією. Методи, використані для прогнозу не розголошуються і є комерційними. Загальний вигляд інтерфейсного вікна сайту продемонстровано на рисунку 1.1 та рисунку 1.2.

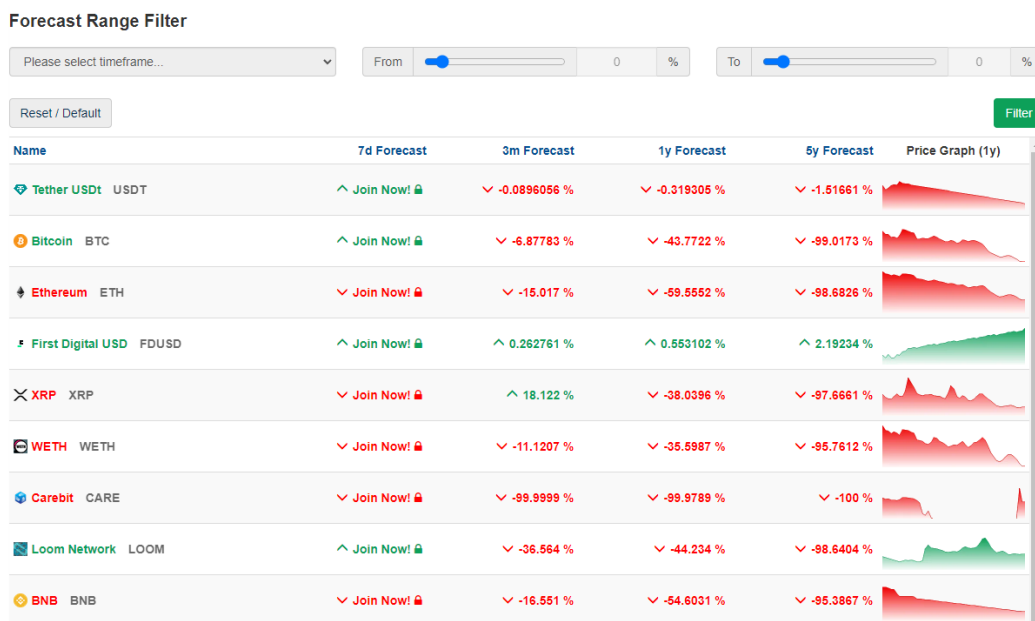


Рисунок 1.1 – Загальний вигляд інтерфейсного вікна сайту з прогнозами
Walletinvestor

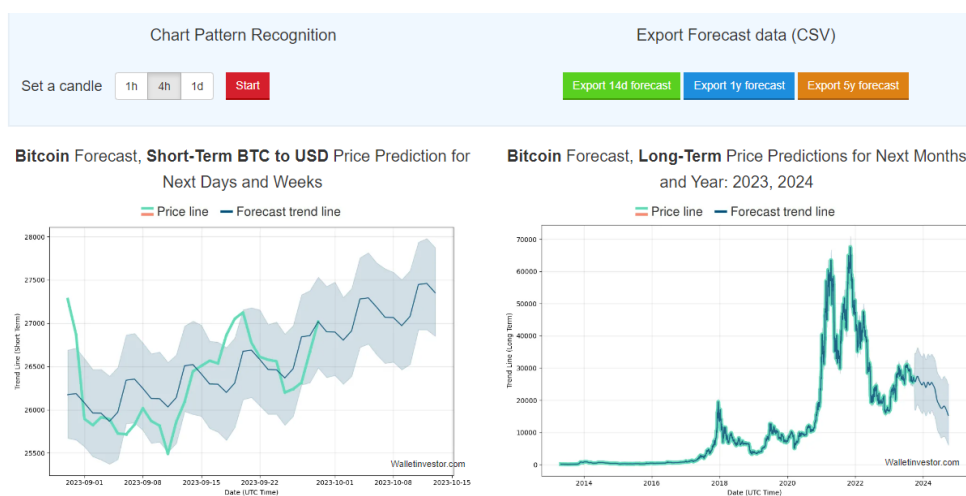


Рисунок 1.2 – Загальний вигляд інтерфейсного вікна сайту Walletinvestor з прогнозами щодо курсу біткоїна

Ще один сайт для прогнозування курсу криптовалют – «Belinvestor» [8]. Компанія пропонує лише прогнози на короткий горизонт прогнозування криптовалют. Сайт belinvestor надає можливість аналізу криптовалютної сфери використовуючи лише методи технічного аналізу, ігноруючи інші методи.

Загальний вигляд інтерфейсного вікна сайту продемонстровано на рисунку 1.3 та рисунку 1.4.

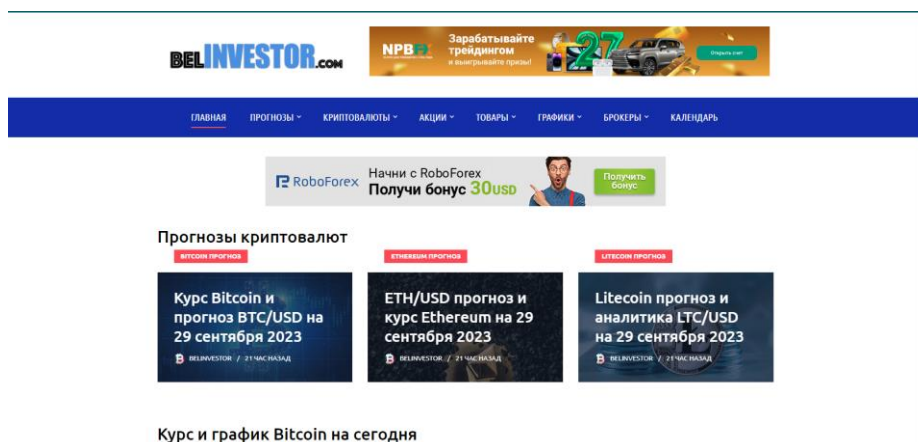


Рисунок 1.3 – Загальний вигляд інтерфейсного вікна сайту з прогнозами Belinvestor

Курс Bitcoin и прогноз BTC/USD на 29 сентября 2023

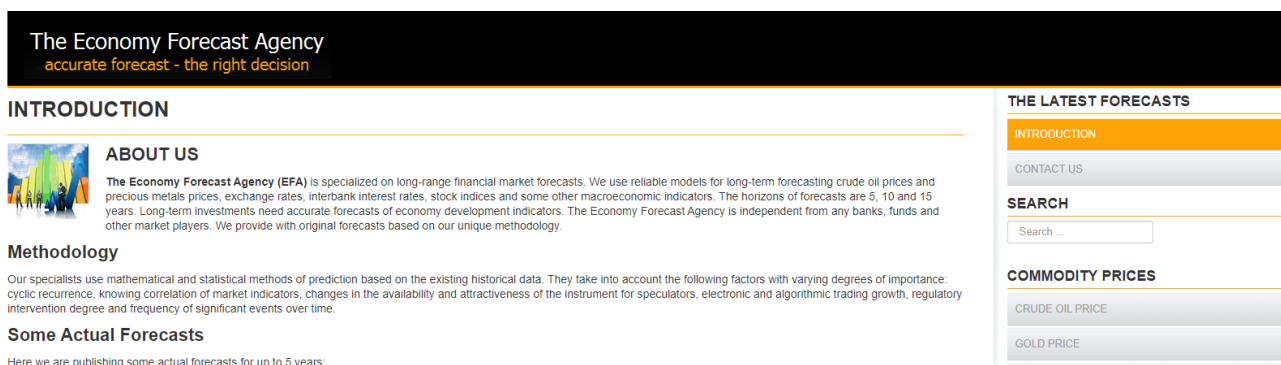
Дополнительным сигналом в пользу снижения криптовалюты BTC/USD выступит тест области сопротивления на графике цены монеты. Вторым сигналом станет отбой от верхней границы нисходящего канала. Ранее был получен слабый сигнал для покупок криптовалюты Bitcoin. Сигнал сформировался за счет пересечения сигнальных линий на уровне 26345. Ближайшая область сопротивления по BTC/USD на графике располагается на уровне 26985. Область поддержки по Bitcoin располагается на уровне 26085.



Рисунок 1.4 – Загальний вигляд інтерфейсного вікна сайту Belinvestor з прогнозами щодо курсу біткоїна

Третім сайтом для прогнозування курсу криптовалют є сайт від Economy Forecast Agency – longforecast [9]. На сайті стверджується, що для прогнозування

використовуються різні математичні та статистичні методи прогнозування на основі історичних даних (рис. 1.5, 1.6). Вони використовують такі фактори як: циклічна повторюваність, кореляція ринкових індикаторів, зростання електронної та алгоритмічної торгівлі тощо.



The Economy Forecast Agency
accurate forecast - the right decision

INTRODUCTION

ABOUT US

The Economy Forecast Agency (EFA) is specialized on long-range financial market forecasts. We use reliable models for long-term forecasting crude oil prices and precious metals prices, exchange rates, interbank interest rates, stock indices and some other macroeconomic indicators. The horizons of forecasts are 5, 10 and 15 years. Long-term investments need accurate forecasts of economy development indicators. The Economy Forecast Agency is independent from any banks, funds and other market players. We provide with original forecasts based on our unique methodology.

Methodology

Our specialists use mathematical and statistical methods of prediction based on the existing historical data. They take into account the following factors with varying degrees of importance: cyclic recurrence, knowing correlation of market indicators, changes in the availability and attractiveness of the instrument for speculators, electronic and algorithmic trading growth, regulatory intervention degree and frequency of significant events over time.

Some Actual Forecasts

Here we are publishing some actual forecasts for up to 5 years:

THE LATEST FORECASTS

INTRODUCTION

CONTACT US

SEARCH

Search ...

COMMODITY PRICES

CRUDE OIL PRICE

GOLD PRICE

Рисунок 1.5 – Загальний вигляд інтерфейсного вікна сайту з прогнозами LongForecast

BITCOIN PRICE PREDICTION 2023, 2024, 2025, 2026

2023/09/29. Bitcoin Price Today.

Bitcoin price equal to **26997** dollars a coin. Today's range: \$26873 - \$27234. The previous day close: \$27027. The change was -30, -0.11%.

↓ 26997 -0.11%

Recent changes:

Period	2 Days	3 Days	1 Week	2 Weeks	1 Month
Chng. %	+2.4%	+3.0%	+2.8%	-0.8%	+2.8%
Price	26360	26213	26262	27219	26267

Bitcoin Converter

Bitcoin To Dollar

0.5 equal to **13,498.48**

Bitcoin Price Prediction For 2023, 2024, 2025, 2026 And 2027

Month	Open	Min-Max	Close	Total %
2023				
Sep	25936	24919-28919	27027	4.2%
Oct	27027	24957-31060	28364	9.4%
Nov	28364	23404-30311	25166	-3.0%
Dec	25166	22452-25832	24142	-6.9%
2024				
Jan	24142	24142-28920	27028	4.2%
Feb	27028	23401-27028	25162	-3.0%
Mar	25162	24025-27641	25833	-0.4%
Apr	25833	25833-32064	29966	15.5%
May	29966	27890-32088	29989	15.6%
Jun	29989	29989-37222	34787	34.1%
Jul	34787	31150-35840	33495	29.1%
Aug	33495	28658-33495	30815	18.8%
Sep	30815	30232-34782	32507	25.3%

Рисунок 1.6 – Загальний вигляд інтерфейсного вікна сайту LongForecast з прогнозами щодо курсу біткоїна

Недоліком є великий горизонт прогнозування – на сайті доступно прогнозування мінімум на місяць вперед і максимум на 5 років. Ще одним недоліком є нерозголошення методів прогнозування, відповідно перевірити систему на ефективність можна лише шляхом довготривалих спостережень.

1.3 Огляд методів та програмних засобів для прогнозування курсу криптовалют

За останнє десятиліття увага, яку приділяють науковці цій тематиці зростає в рази. Незважаючи на зростаючу кількість нових досліджень, головною ціллю дослідників на сьогодні є визначення критеріїв та показників, які чинять вплив на криптовалюту, загальна кластеризація періоду всього існування будь-якої криптовалюти, знаходження індикаторів та кореляцію між різними криптовалютами тощо.

Натепер не існує чіткої методології, яка б в повній мірі описала цілу сферу криптовалют в розрізі моделювання та прогнозування. Основним завданням науковців, які займаються дослідженнями в цій галузі – це визначення системи критеріїв та показників, які чинять вплив, проте це не дає повного розуміння роботи усієї сфери.

1) Розглянемо фундаментальний метод моделювання та прогнозування криптовалюти. Фундаментальний аналіз – це метод, який використовується інвесторами та аналітиками для визначення внутрішньої вартості активів або бізнесу. Для точної оцінки фундаментальні аналітики будуть ретельно вивчати внутрішні та зовнішні чинники, щоб визначити переоцінений актив або бізнес, про які йде мова. Їх висновки можуть потім допомогти краще сформулювати стратегію, яка з більшою ймовірністю принесе гарний прибуток. Кінцева мета фундаментального аналізу – отримати очікувану ціну акції та порівняти її з поточною ціною. Якщо число вище поточної ціни, можна зробити висновок, що компанія недооцінена. Якщо вона нижча за ринкову, то можна припустити, що компанія зараз переоцінена [10].

Вартість віртуальних монет, так само як і у випадку акцій на фінансових ринках, встановлюється ринковим шляхом. Чим більше попит на певну криптовалюту, тим менше кількість монет на ринку і вище їх курс. Попит в свою чергу залежить від різних чинників. Наприклад технологічні рішення та інновації, які привносить дана криптовалюта буде збільшувати її популярність. Проте криптовалюти залежать й від політичних або медійних чинників. Якщо завтра якась країна, з високим ВВП зробить якусь криптовалюту офіційним способом розрахунку, то її ціна значно виросте. Попит формується на тлі новин, нових розробок, анонсів компаній.

2) Оскільки криптовалюти мають певну залежність від людського попиту, то одним із методів прогнозування курсу є кількісна оцінка цього попиту та відповідно до її величини аналіз впливу на вартість криптовалюти. Для того, щоб проаналізувати попит і популярність тієї чи іншої криптовалюти можна отримати дані про те, як часто ця тема обговорюється у інтернеті, наприклад, один з таких показників – кількість постів зі згадуванням відповідної криптовалюти у соціальній мережі Twitter.

Перевагою такого підходу є:

- Кількісна інтерпретація.
- Відносна простота отримання чи обчислення.
- Легка інтеграція у математичні моделі.

Основні недоліки такі:

- Потрібно мати досвід для проведення кластеризації та виявлення справді тих чинників, які впливають на курс, а не є новинним шумом.
- Обмеження по сумарній кількості отриманої інформації та швидкості отримання (кількість запитів у хвилину).

3) Наступний метод дослідження криптовалюти – системний підхід. Такий метод пропонує визначити сферу криптовалюти як єдину систему, яка може впливати на розвиток інших суб'єктів економіки, як частина фінансової системи. Криптовалютний ринок – це відкрита, динамічна, складна, стохастична система, що перебуває у процесі формування. У системі є суб'єкт та об'єкт управління, а

розвиток системи на умовах децентралізації здійснюється набором елементів, що утворюють керуючий об'єкт у вигляді саморегульованої організації. Сукупність елементів, що гарантують випуск та обіг криптовалют, а також їх використання утворює об'єкт керування – сам криптовалютний ринок [11].

4) Наступним способом прогнозування криптовалют є розгляд прогнозування криптовалют за допомогою нейронних мереж. Штучні нейронні мережі (ARN) як математичні моделі та їх програмні чи апаратні реалізації засновані за принципом організації та функціонування природніх нейронних мереж – мереж нервових клітин живого організму. Ця концепція виникла під час вивчення процесів, що відбуваються в мозку та спроб моделювати ці процеси. Першою такою спробою стали нейронні мережі Маккалоха та Піттса. А після розробки алгоритмів навчання отримані моделі використовувались у практичних цілях: для прогнозування завдань, для розпізнавання шаблонів, для контрольних завдань тощо [12].

ANN – це система пов'язаних та взаємодіючих простих процесорів (штучних нейронів). Такі процесори зазвичай досить прості, особливо в порівнянні з процесорами, що використовуються в персональних комп'ютерах. Кожен процесор у такій мережі обробляє лише періодично одержувані ним сигнали та періодично надсилає сигнали іншим процесорам. Однак, будучи підключеними до відносно великої мережі з керованою взаємодією, такі локально прості процесори можуть колективно виконувати досить складні завдання.

4) Існують і традиційні методи для аналізу та прогнозування з використанням математичних моделей. Такі методи використовуються наприклад у прогнозуванні погоди, продажів, захворюваності тощо. Математичні моделі дозволяють підійти до процесу прогнозування більш об'єктивно, відкинувши емоції чи особисті переконання. Ще однією перевагою є точність, оскільки математичні моделі використовують точні дані для навчання і виявлення закономірностей, в той час як фундаментальний аналіз чи аналіз попиту використовують дані, які важко систематизувати.

Проте старі класичні методи, які базуються на припущеннях стаціонарності математичних моделей не відображають швидко змінні, нелінійні процеси. На сучасному етапі з'являється все більше якісно нових моделей та методів їхнього дослідження. Одними з таких методів є прогнозування на основі часових рядів.

Часовий ряд це послідовність даних, які проіндексовані в хронологічному порядку (зазвичай від початку до кінця). Зазвичай часовий ряд є послідовністю, взятою на однакових проміжках в часі, які йдуть одна за одною. Таким чином, він є послідовністю даних дискретного часу. Основними завданнями аналізу часових рядів є визначення характерних особливостей, підбір їх статистичних моделей, прогнозування показників. Часові ряди можуть бути як стаціонарними, так і нестаціонарними. Стаціонарний часовий ряд не має тенденції до зміни середнього значення, дисперсії або автокореляції з часом. Нестационарний часовий ряд має тенденцію до зміни одного або декількох з цих параметрів з часом. Часовому ряду притаманні такі ознаки:

- Послідовні в часі показники часових рядів є взаємозалежними, особливо це відноситься до близько розташованих спостережень.

- Залежно від моменту спостереження показники часового ряду мають різну інформативність, інформаційна цінність спостережень зменшується за ступенем їх віддалення від поточного моменту часу.

- Зі збільшенням кількості показників часового ряду точність статистичних характеристик не збільшується пропорційно числу спостережень, а за появи нових закономірностей розвитку вона може навіть зменшуватися [13].

Ще один метод – індуктивний підхід на основі моделювання часового тренду середньорічних значень криптовалюти біткоїн. Результати дослідження [14] представлені у вигляді графіку, зображеного на рисунку 1.7.

Отже, як висновок до теми прогнозування курсу біткоїна, то очевидно те, що ця тема є досить нова і актуальна. Вона дуже схожа до задачі прогнозування звичайного курсу валют, проте має і деякі власні відмінності.

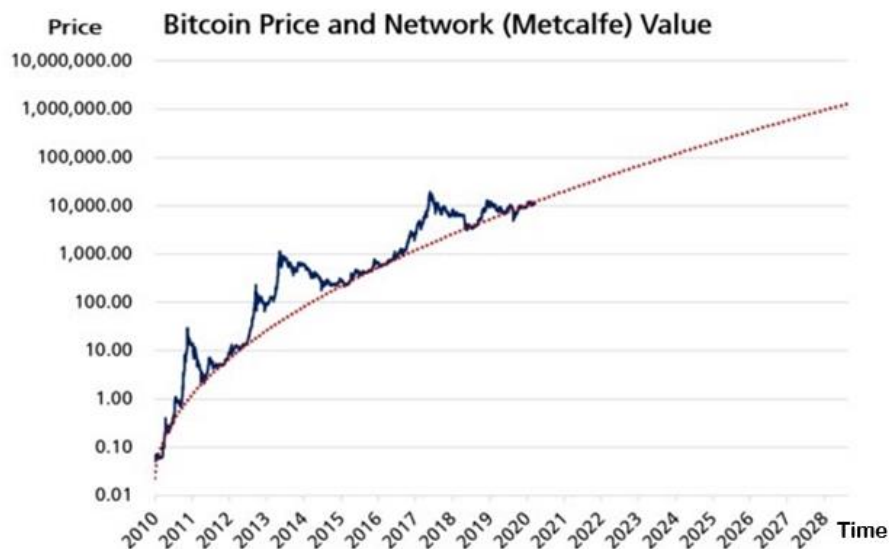


Рисунок 1.7 – Тренд середніх значень біткоїну за весь період існування на 2022 рік

Зараз існує декілька способів вирішення такої задачі, але, на жаль, вони представлені у вигляді комерційного рішення і недостатньо повно розкривають деталі її реалізації. Також, було виявлено, що соціум має вагомий вплив на курс криптовалют, проте слабо проаналізований математично. Тобто проблема прогнозування криптовалют залишається відкритою та актуальною для вивчення.

Переваги та недоліки розглянутих методів можна представити за допомогою таблиці. 1.1.

Таблиця 1.1 – Переваги та недоліки методологій дослідження криптовалют

Метод	Переваги	Недоліки
Фундаментальний аналіз	Криптовалютна сфера розглядається з точки зору інвестицій. Надає кількісну оцінку доцільності інвестування в той чи інший актив.	Дає лише оцінку інвестицій в булевій формі.

Продовження таблиці 1.1

Метод	Переваги	Недоліки
Фундаментальний аналіз	Враховує різні фактори, основні з яких є макроекономічні показники, які добре підходять для оцінки інвестиційного потенціалу країни.	В основі лежать конкретні макроекономічні показники певної країни, які не підходять при дослідженні для сфери криптовалют, оскільки вона не має прив'язки до конкретної країни.
Системний підхід	Надає структурний опис заданої системи. Розкриває взаємозв'язки з іншими системами. Описує процеси всередині системи. Надає рекомендації щодо покращення процесів. Визначає основні фактори, які впливають на систему.	Не надає кількісного показнику аналізу системи. Суттєво залежить від сталості компонентів системи. Важко застосувати при описі нових структур.
Нейронні мережі	Дає хорошу модель після проведення багатьох ітерацій дослідження. Досить проста модель в реалізації.	Не враховуються фактори, які впливають на показник.
Математичні моделі часових рядів	Модель має високу точність. Враховує вплив інших факторів.	Необхідність чітко структурованих, систематизованих даних. потребує постійної підтримки і вдосконалення системи
Індуктивний підхід та моделювання часового тренду	Підходить для опису динаміки показників. Корисний на початку роботи, коли потрібно підтвердити/спростувати припущення.	При прогнозуванні модель дає нижчу точність, ніж інші методи. Не є самодостатнім. Потребує роботи в поєднанні з іншими методологіями.

1.4 Постановка задачі дослідження

Поєднання різноманітних видів діяльності з ІТ-технологіями все частіше набувають широкого застосування та удосконалюють сфери нашого життя. Гарною колаборацією ІТ та фінансів в наш час є криптовалюти, які з кожним роком набувають актуальності та стрімкого використання.

Стрімкий розвиток інформаційних технологій трансформує сучасну економічну систему. За підрахунками фахівців обсяг інтернет-торгівлі у 2023 році перевищив позначку в 2,5 трлн. дол. США.

Проблема полягає у відсутності простого, безкоштовного і водночас багатофункціонального та продуктивного інструменту для прогнозування курсу біткоїна.

Загальний опис функціоналу інформаційної технології для прогнозування курсу криптовалюти «Біткоїн»:

- Реалізація входу до інформаційної системи за допомогою графічного інтерфейсу.
- Реалізація вибору періоду прогнозування.
- Реалізація виведення результуючого графіку за допомогою графічного інтерфейсу.
- Реалізація виведення історії прогнозів користувача.

У цій роботі потрібно реалізувати інформаційну технологію для прогнозування курсу криптовалюти «Біткоїн». Інформаційна технологія буде використовуватись користувачами для отримання графіку прогнозованої ціни криптовалюти «Біткоїн». Таким чином, використання такої розробки дасть можливість користувачу отримати новий інструмент для аналізу криптовалютного ринку та інструмент для перевірки своїх стратегій торгівлі.

Інформаційна технологія, що розроблюється, перш за все орієнтована на точність та зручність прогнозування. Точність полягає у можливості отримати більший результат прогнозування ніж в існуючих програм-аналогів. Зручність полягає у можливості вибрати необхідні для користувача періоди прогнозування,

це є платною функцією у програм-аналогів та можливістю перегляду власної історії прогнозів для аналізу отриманих результатів.

Програмний модуль буде здійснювати підключення до бази даних постачальника даних, в нашому випадку за допомогою Binance API, підключення до розробленої бази даних з використанням сервісу Firebase та обробляти дані, отримані з вище вказаних технологій.

Тому, для вирішення проблеми прогнозування курсу біткоїна потрібно вирішити такі основні завдання:

1. Здійснити проектування структури інформаційної технології прогнозування курсу криптовалюти «Біткоїн».
2. Розробити математичну модель прогнозування курсу криптовалюти «Біткоїн».
3. Реалізувати програмний модуль інформаційної технології.
4. Провести тестування інформаційної технології.

Отже, потрібно розробити інформаційну технологію, в якій буде реалізовано алгоритм прогнозування курсу біткоїна з достатньою точністю.

1.5 Висновок до розділу 1

В першому розділі було розглянуто основні поняття криптовалют, проведено аналіз технічних рішень програм для прогнозування курсу криптовалют та доведено актуальність розробки інформаційної технології прогнозування курсу криптовалюти «Біткоїн». Розглянуто системи-аналоги: «Walletinvestor», «Belinvestor» та «Long Forecast», наведено вікна інтерфейсу роботи програм. Виявлено, що існуючі рішення такої задачі представлені у вигляді комерційного сервісу і не розкривають усіх деталей реалізації або не мають достатнього функціоналу, обмежуючи користувача певними рамками прогнозування.

Вся описана та проаналізована вище інформація і стала передумовою для розробки інформаційної технології прогнозування курсу криптовалюти «Біткоїн», яка дозволить дати користувачам більше інформації для аналізу криптовалютної сфери. Задача прогнозування курсу криптовалюти «Біткоїн» є, дуже схожа до задачі прогнозування звичайного курсу валют, проте має і деякі власні відмінності. Вплив соціуму на курс криптовалют є достатньо суттєвим, проте слабо проаналізованим з математичної перспективи, тому створення інформаційної технології прогнозування курсу криптовалюти «Біткоїн» є доцільним.

Наведено основні методи та програмні засоби для створення інформаційної технології прогнозування курсу криптовалюти «Біткоїн». З метою розробки інформаційної технології було обрано метод прогнозування з використанням математичних моделей часових рядів, оскільки такий метод дає можливість отримати прогноз із заданою точністю при наявності актуальних, цілісних даних, що підходить для предметної галузі криптовалют.

Визначено задачі дослідження, вимоги клієнта, які допоможуть в подальшому відобразити структуру системи, підібрати доцільні технології та мови для реалізації інформаційної технології прогнозування курсу криптовалюти «Біткоїн».

2 МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ПРОГНОЗУВАННЯ КУРСУ КРИПТОВАЛЮТИ «БІТКОЇН»

2.1 Обґрунтування вибору методу прогнозування курсу криптовалюти «Біткоїн»

Значні якісні зміни на економічних ринках, поява нових технологій, інформаційна революція та відповідні зміни в поведінці людини і суспільства призвели до того, що для їх адекватного опису та прогнозування потрібні нові математичні інструменти. Старі класичні методи, засновані на припущеннях про стаціонарність, уже не можуть відображати нелінійні процеси, які швидко змінюються. На сучасному етапі з'являються нові, більш якісні моделі та методи їх дослідження. Одними з таких методів є прогнозування на основі часових рядів.

Сукупність величин $\{x_1, x_2, \dots, x_n\}$, називається часовим рядом, при цьому кожне значення відповідає значенню параметра в конкретний момент часу. Основними завданнями аналізу часових рядів є визначення характерних особливостей, підбір їх статистичних моделей, прогнозування показників.

Аналіз часового ряду починається з побудови і вивчення його графіку. Спочатку потрібно перевірити часовий ряд на стаціонарність. Для будь-якого часового ряду можна знайти деякі числові характеристики, такі як – математичне сподівання, дисперсія, коваріація. Щоб часовий ряд був стаціонарним він має мати сталі і скінченні математичні сподівання та дисперсію для кожного періоду часу, а також сталу і скінченну автоковаріацію будь-якого порядку в усі моменти часу. Якщо часовий ряд нестаціонарний, то перш за все потрібно виділити нестаціонарну складову ряду. Процес виявлення тренду та інших складових ряду, які призводять до порушення стаціонарності, може проходити в кілька етапів. На кожному з них розглядається ряд залишків, отриманий в результаті розрахунку з вихідного ряду обраної моделі тренду, або результат диференціальних та інших перетворень ряду [15].

Наступним етапом після перевірки на стаціонарність є виділення тренду. Тренд – це зміна, що визначає загальний напрям розвитку, основну тенденцію часових рядів.

Існує три основних типи трендів:

1) Тренд середнього, коли часовий ряд виглядає як коливання біля повільно зростаючої чи спадаючої величини.

2) Тренд дисперсії. В цьому випадку в часі змінюється амплітуда коливань змінної. Часто економічні процеси зі зростаючим середнім мають і зростаючу дисперсію.

3) Тренд автоковаріації і автокореляції, зміна величини кореляції між поточним і попереднім значеннями ряду.

На практиці для перевірки наявності тренду використовуються параметричні тести на сталість математичного очікування та дисперсії за критеріями Стюдента і Фішера [16]. Ці тести можна застосовувати тільки у випадку нормального розподілу даних. Для виключення тренду потрібно задати його модель. Якщо природа тренду має теоретичне пояснення, то моделювання тренду виконується з використанням цієї теорії. Зазвичай природа тренду невідома. Тоді як формальну модель використовують апроксимацію тренду за допомогою лінійної комбінації будь-яких поліномів. При цьому до складу моделі входить також вільний член. Параметри моделі тренду визначаються за методом найменших квадратів.

Часові ряди можуть мати сезонний компонент. Він показує коливання навколо трендового компонента. Наприклад, його наявність для економічних часових рядів пояснюється сезонним характером виробництва та споживання. Головною ідеєю виділення сезонного компоненту є порівняння даних за відповідні періоди, а не за минулі періоди. Загалом виділяють одразу декілька різних підходів до аналізу структурних особливостей часових рядів, які містять у собі сезонні або циклічні коливання. Найпростішим вважається підхід, при якому застосовується розрахунок значень сезонного компоненту за допомогою

методу ковзного середнього, а також будується адитивна або мультиплікативна модель даного часового ряду [17].

Авторегресія – це регресійна залежність випадкової послідовності від її попередніх значень. Відповідно, авторегресивна модель – це модель динамічної регресії, яка відображає зміни в часі досліджуваної змінної (змінних) від попередніх значень. Побудова звичайних лінійних регресій не могла дати повне уявлення про напрями розвитку таких часових рядів. Внаслідок цього були запроваджені процеси авторегресії. Вони широко використовуються для прогнозування стаціонарних та нестаціонарних рядів, тому авторегресивні моделі отримали багато модифікацій та підвидів [18].

Існує велика кількість видів авторегресійних моделей для аналізу одномірних динамічних рядів: авторегресивна модель першого порядку, моделі вищого порядку, ковзної середнього порядку, моделі з дробовим інтегруванням.

Модель ARIMA (Autoregressive Integrated Moving Average) – це лінійна модель часового ряду, в якій залежні та незалежні змінні асоціюються не тільки одночасними, але й з історичними (лаговими) значеннями. ARIMA складається з трьох компонентів:

- 1) Авторегресійний компонент (AR) описує залежність між поточним значенням часового ряду та його попередніми значеннями.
- 2) Інтеграційний компонент (I) робить часовий ряд стаціонарним, видаляючи тренд і сезонні коливання.
- 3) Пересувний середній компонент (MA) описує залежність між поточним значенням часового ряду та його випадковим компонентом.

$ARIMA(p, q, d)$ є основною моделлю для прогнозування нестаціонарних часових рядів, які можуть бути приведені до стаціонарного виду за допомогою логарифмування або різниць. Запис процесу має три параметри: p – кількість лагових змінних, q – кількість лагових збурень, d – кількість різниць. Оскільки за часовим рядом Біткоїна важко визначити тренд і він не має сезонності, тобто він є нестаціонарним то ARIMA-модель гарно підходить, оскільки має

інтеграційний компонент для перетворення такого ряду. Також на курс Біткоїна впливають його попередні значення, що підходить для роботи моделі. Ще одним важливим фактором є вплив зовнішніх факторів (таких як політичні явища, економічні впливи тощо), який може врахувати модель. Саме тому для задачі прогнозування курсу біткоїна доцільно застосувати ARIMA-модель.

2.2 Математична модель прогнозування курсу криптовалюти «Біткоїн»

Першим етапом для створення математичної моделі є отримання часового ряду та його перевірка на стаціонарність. Існує багато методів перевірки чи ряд стаціонарний:

1) Автокореляція (ACF) – це статистичний метод, який використовується для оцінки кореляції між значеннями часового ряду з різними інтервалами часу. Автокореляція j -того порядку визначається за допомогою рівності:

$$r_j = \frac{\text{cov}(y_t, y_{t-j})}{D(y_t)} = \frac{\gamma_j}{\gamma_0}. \quad (2.1)$$

Цей коефіцієнт визначає степінь залежності між спостереженнями, які знаходяться на відстані j періодів. Функція автокореляції стаціонарного часового ряду «швидко спадає» з ростом j після кількох перших значень. Якщо ж графік спадає повільно, то є підстави припустити, що ряд нестаціонарний [17].

2) PACF – часткова функція автокореляції. Замість того, щоб знаходити кореляції теперішнього значення з лагами, такими як ACF, ми знаходимо кореляцію залишків (яка зберігається після усунення ефектів, які вже були пояснені більш ранніми лагами) з наступним значенням лага. Отже, «частковим», а не «повним». Графік окремої автокореляційної функції *PACF* має швидко спадати для стаціонарного процесу.

3) Тест Дікі-Фуллера (ADF). Суть тесту Дікі-Фуллера полягає у формуванні двох гіпотез:

1) Нульова гіпотеза – одиничний корінь існує, ряд нестационарний.

2) Альтернативна гіпотеза – одиничного кореня не існує, ряд стаціонарний.

Часовий ряд має одиничний корінь якщо перші різниці цього ряду утворюють стаціонарний ряд. Дана умова записується так: $X_t \sim I(1)$, якщо $\Delta X_t = X_t - X_{t-1}$ є стаціонарним $\Delta X_t \sim I(0)$. На практиці перевіряють значення коефіцієнта, а в авторегресійному рівнянні першого порядку AR (1):

$$X_t = \rho X_{t-1} + \varepsilon_t, \quad (2.2)$$

де X_t – часовий ряд,

ε_t – похибка.

На основі отриманої оцінки та похибки будується відповідна t -статистика, значення якої порівнюється зі значеннями, наведеними у таблиці Дікі-Фуллера (для відповідного рівня значущості). Якщо p -значення статистики менше за значення із таблиці, то процес не має одиничного кореня, гіпотеза щодо нестационарності відхиляється і ряд визнається стаціонарним. У протилежному випадку отримується протилежний висновок [13].

Наступним етапом є виділення тренду та сезонності. Нехай є деякий часовий ряд

$$y_t = \alpha + \beta t + \varepsilon_t. \quad (2.3)$$

Це сума заданої складової $\alpha + \beta t$ (лінійний тренд) і випадкової складової ε_t , яка є стаціонарним часовим рядом з нульовим середнім. Вважаючи t незалежною змінною, можемо виділити тренд в моделі $y_t = \alpha + \beta t + \varepsilon_t$, за допомогою звичайної техніки оцінювання параметрів регресійного рівняння.

Після цього утворюється цілий ряд залишків, для коректного опису якого рекомендовано застосувати моделі стаціонарних часових рядів.

Часові ряди можуть мати сезонну компоненту. Загалом виділяють одразу декілька різних підходів до аналізу структурних особливостей часових рядів, які містять у собі сезонні або циклічні коливання. Найпростішим вважається підхід, при якому застосовується розрахунок значень сезонної компоненти за допомогою методу ковзного середнього [13].

Далі можна приступати до побудови ARIMA-моделі. ARIMA(p, d, q) процеси є загальною моделлю прогнозування нестаціонарних часових рядів, які можуть бути приведені до стаціонарного вигляду за допомогою логарифмування або різниць. Запис процесу має три параметри: p – кількість лагових змінних, q – кількість лагових збурень, d – кількість різниць. Під різницею розуміється трансформація часового ряду за правилом

$$\begin{aligned} \overline{y}_t &= y_t - y_{t-1}, \\ t &= \overline{2, T}. \end{aligned} \quad (2.4)$$

ARIMA-модель тісно пов'язана з ARMA-моделлю типу:

$$\begin{aligned} y_t - \varphi_1 y_{t-1} - \dots - \varphi_p y_{t-p} &= \delta + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \dots - \theta_p \varepsilon_{t-p}, \\ \varepsilon_t &\approx N(0, \sigma^2). \end{aligned} \quad (2.5)$$

Прийmemo, що p та q значення дорівнюють 1. Тоді ARMA набуде вигляду:

$$\begin{aligned} y_t - \varphi_1 y_{t-1} &= \delta + \varepsilon_t - \theta_1 \varepsilon_{t-1}, \\ \varepsilon_t &\approx N(0, \sigma^2). \end{aligned} \quad (2.6)$$

Будемо вважати, що $|\varphi_1| < 1$ і $|\theta_1| < 1$.

Тоді вирази для середнього, дисперсії та коваріації ARMA(1,1) процесу:

$$E(y_t) = \frac{\delta}{1 - \varphi_1}, \gamma_0 = V(y_1) = \sigma^2 \frac{1 + \theta_1^2 - 2\varphi_1\theta_1}{1 - \varphi_1^2},$$

$$\gamma_1 = Cov(\gamma_t, \gamma_{t-1}) = \varphi_1\gamma_0 - \theta_1\sigma^2. \quad (2.7)$$

Для автокореляцій вищих порядків отримуємо рекурентне співвідношення

$$\gamma_k = Cov(\gamma_t, \gamma_{t-1}) = \varphi_1\gamma_{k-1},$$

$$k > 1. \quad (2.8)$$

Застосовуючи це співвідношення отримаємо:

$$p_k = \varphi_1^{k-1} p_1,$$

$$k > 1. \quad (2.9)$$

З рівності видно, що ACF для ARMA(1,1) процесу проводиться так само як ACF для AR(1) процесу. Цей висновок можна узагальнити на випадок ARMA(p,q) процесу. Перші q значень ACF визначаються взаємодією AR і MA компонент, а слідує за цим поведінка розглянутої автокореляційної функції така ж, як в AR(p) процесі.

Аналогічний висновок справедливий для частинної автокореляційної функції ARMA(p,q) процесу. Вона спадає подібно PACF для MA(q) процесу. Припустимо, що часовий ряд y_t після того, як до нього застосували d разів оператор послідовної різниці, став стаціонарним рядом $\Delta^d y_t$ що задовольняє ARMA(p,q) моделі

$$y_t - \varphi_1 y_{t-1} - \dots - \varphi_p y_{t-p} = \delta + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \dots - \theta_p \varepsilon_{t-p},$$

$$\varepsilon_t \approx N(0, \sigma^2). \quad (2.10)$$

Тоді процес y_t називається інтегрованим процесом авторегресії і ковзного середнього: ARIMA(p,d,q) [17].

2.3 Розробка UML-діаграми послідовності інформаційної технології прогнозування курсу криптовалюти «Біткоїн»

Для кращого розуміння роботи програмного модуля інформаційної технології прогнозування курсу криптовалюти «Біткоїн» побудуємо UML діаграму.

UML (англ. Unified Modeling Language) – уніфікована мова моделювання. Вона є невід'ємною частиною уніфікованого процесу розробки програмного забезпечення. UML є мовою широкого профілю, це відкритий стандарт, що використовує графічні позначення для створення абстрактної моделі системи, яка називається UML-моделлю.

UML поділяють на структурні діаграми та діаграми поведінки. Оскільки ми описуємо цикл роботи програми доцільно розглянути шаблони, які належать до діаграм поведінки, а саме: діаграма прецедентів (Use-case diagram), діаграма діяльності (Activity diagram), діаграма послідовності (Sequence diagram).

Діаграма діяльності ілюструє процес використання, показує потік повідомлень від однієї дії до іншої та цілісну роботу усієї системи. Діаграма діяльності вважається розширеним варіантом звичайних блок-схем. Проте блок-схема не має стандартизованої нотації та не дає можливість візуалізації паралельного виконання дій та синхронізації.

Діаграма прецедентів дозволяє показати типи ролей та їх зв'язок із системою. Проте вона не показує порядок виконання кроків. Зображує функціональні вимоги (те, що система може зробити) з точки зору користувача. Може описуватись текстом або у вигляді діаграми.

Діаграма послідовності показує особливості відправлення і прийому повідомлень об'єктами з плином часу. Впорядкованість за часом потрібно розуміти як послідовність дій. Тому було обрано діаграму послідовностей, оскільки вона дає представлення роботи повного циклу програмного застосунку [19].

Діаграма послідовності міститиме компоненти, які описують цикл роботи програмного модуля, а саме: веб-клієнт, сервер, постачальник даних, база даних. Також додамо об'єкт користувач для відображення взаємодії користувача з іншими модулями програми. Розроблена діаграма послідовностей зображена на рисунку 2.1.

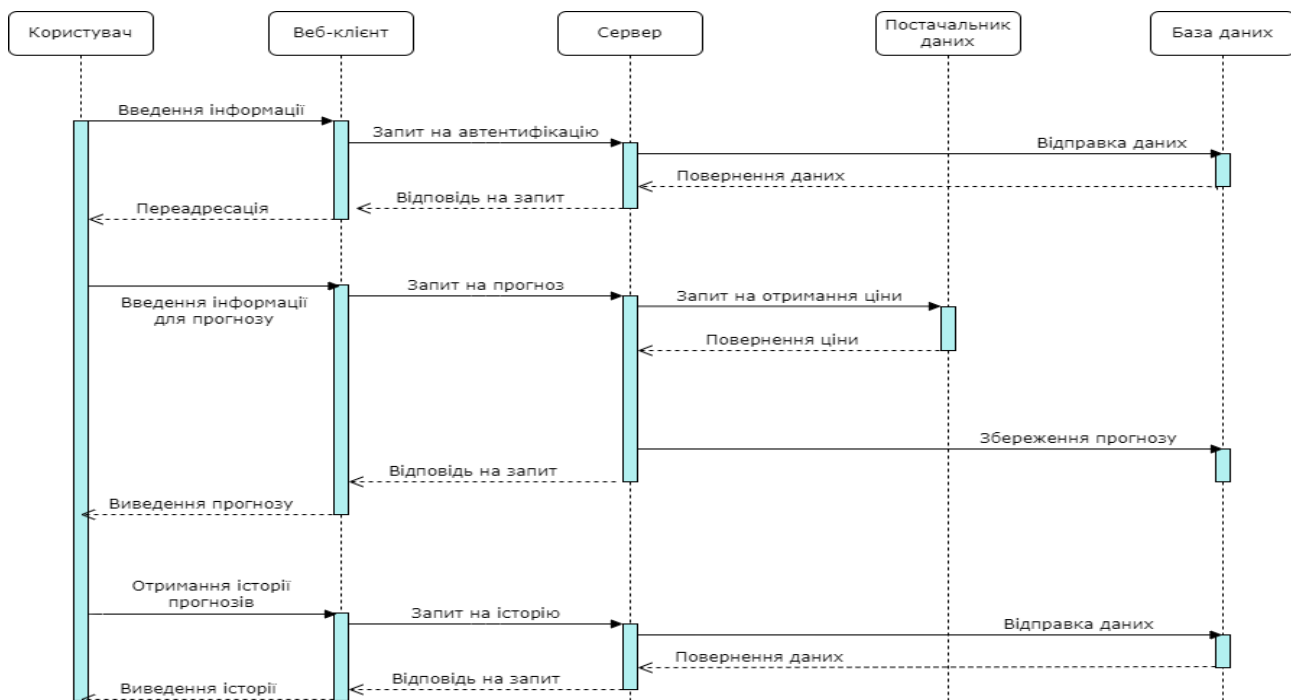


Рисунок 2.1 – UML-діаграма послідовності взаємодії модулів інформаційної технології прогнозування курсу криптовалюти «Біткоїн»

2.4 Висновок до розділу 2

У другому розділі було розглянуто методи прогнозування курсу криптовалюти Біткоїн та обґрунтовано використання методу аналізу часових рядів.

Розроблено математичну модель програмного модуля прогнозування курсу криптовалюти біткоїна на основі авторегресійної моделі інтегрованого ковзного середнього. Модель відповідає за генерування прогнозів на основі переданої користувачем інформації. Розроблено UML-діаграму послідовності для опису роботи програмного модуля інформаційної технології прогнозування курсу криптовалюти «Біткоїн».

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ПРОГРАМУВАННЯ КУРСУ КРИПТОВАЛЮТИ «БІТКОІН»

3.1 Обґрунтування вибору архітектури інформаційної технології прогнозування курсу криптовалюти «Біткоїн»

Архітектура програмного забезпечення – це структура програми, системи або модуля, яка відображає взаємодію між елементами системи, їх властивості та логічні частини за які відповідає той чи інший елемент. Побудування архітектури є першим етапом розробки ПЗ. При проектуванні модуля, орієнтуючись на поставлене технічне завдання, необхідно розробити найбільш оптимальну архітектуру, враховуючи швидкість роботи майбутнього проекту, зв'язки основних елементів системи, а також закласти можливість масштабування архітектури. Архітектура програмного забезпечення створює обмеження для написання функціонала, декларує єдиний стиль коду та закріплює ключові кроки, створені на ранніх етапах розробки.

Інформаційна технологія прогнозування курсу криптовалюти має клієнт-серверну архітектуру. Сервер – потужна обчислювальна машина, призначена для збереження та обробки інформації та її передачі між користувачами. Клієнт – приймає та обробляє введену користувачем інформацію, а також відображає існуючі дані користувача. Принцип роботи такої архітектури полягає в тому, що клієнт відповідає за пряму взаємодію з користувачем за допомогою створеного інтерфейсу. Клієнт збирає необхідну інформацію від користувача додатку, за необхідності опрацьовує та модифікує, а потім відправляє на сервер за допомогою HTTP або TCP протоколів передачі даних (найпопулярніші протоколи, які задовільняють потребу комунікації клієнта та сервера). TCP протокол відрізняється від HTTP протоколу тим, що дозволяє створити постійне підключення між двома елементами системи (у нашому випадку клієнт та сервер), саме тому зміни можуть динамічно передаватись у випадку зміни даних у будь-якому елементі системи. Сервер відповідає за обробку та збереження

надісланої інформації з клієнта. Сервер може обробляти декілька клієнтів, або система може містити декілька серверів, які виконують різні функції, залежно від побудованої архітектури, та зв'язуються між собою за допомогою HTTP або TCP протоколів. Сервер має доступ до бази даних системи, та відповідає за безпеку доступу до неї, а також за операції з інформацією. При побудові архітектури, можна вказати пріоритетність запитів до серверу, таким чином сервер може в першу чергу обробляти більш важливі для роботи системи функції [20].

Структурна схема взаємодії клієнт-серверної архітектури зображено на рисунку 3.1. Схема відображає взаємодію між сервером, базою даних, та двома клієнтами інформаційної технології безконтактного обміну персональними даними.

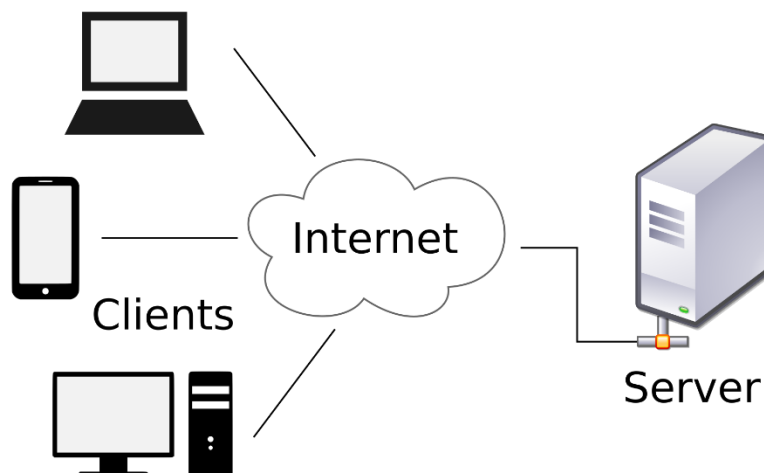


Рисунок 3.1 – Схема системи клієнт-сервер

Функції сервера:

- Зберігання, доступ, захист і резервне копіювання даних.
- Обробка клієнтського запиту.
- Відправлення результату (відповіді) клієнту.

Функції клієнта:

- Надання користувальницького інтерфейсу.

- Формулювання запиту до сервера і його відправка.
- Отримання результатів запиту і відправка додаткових команд (запитів на додавання, оновлення або видалення даних).

3.2 Проектування структури та розробка алгоритму функціонування інформаційної технології прогнозування курсу криптовалюти «Біткоїн»

Прогнозування курсу криптовалют це трудомісткий процес, який потребує великих обчислювальних потужностей. Для швидкої роботи технології та можливості подальшої її підтримки доцільно використати модульний підхід.

Модульний підхід до розробки – підхід, заснований на зменшенні складності програми та пристосованості до передачі модулів та рішень з інших додатків. Всі функціональні можливості розділені на окремі модулі та працюють через інтерфейси між ними. Модуль є логічно завершеною частиною програмного продукту. Кожен окремий модуль має надати:

- Поширення програмних елементів в окремі файли з адаптацією для інтеграції на завершальній стадії розробки продукту.
- Логічна повнота кожного компонента або модуля.
- Функціональні та розмірні обмеження для практичної організації та безперебійної передачі модулів.
- Ізоляція модулів від інших для підвищення стійкості до несправностей.
- Універсальність модулів для повторного використання в інших продуктах.
- Мінімалізм у реалізації, для ясності програмного коду та практичної роботи з ним.

Цей підхід дозволяє впровадити логіку більш прозоро, простіше писати та перевіряти, зменшити витрати на технічне обслуговування та модифікацію, підвищити стійкість до несправностей. Основний принцип – більше модулів з меншою кількістю з'єднань.

В програмі будуть представлені: модуль інтерфейсу, модуль прогнозування курсу біткоїна, модуль роботи з постачальником даних. Структура програмного модуля прогнозування курсу біткоїна наведена на рисунку 3.2



Рисунок 3.2 – Загальна структурна модель програмного модуля

Завдяки графічному інтерфейсу, який пов'язаний з модулями прогнозування та роботи з постачальником даних користувач отримує всю необхідну йому інформацію. Він відповідає за коректне відображення на екрані результатів прогнозування, а також за введення та виведення відповідної інформації, тобто введення проміжку часу, на який необхідно спрогнозувати курс біткоїна та виведення графіка.

Модуль прогнозування пов'язаний з модулем роботи з постачальником даних. Цей модуль є головним модулем програми та містить алгоритм прогнозування курсу біткоїна. Прогнозування відбувається при застосуванні авторегресійної моделі ARIMA.

Модуль роботи з постачальником даних пов'язаний з модулем прогнозування та самою базою даних постачальника. Даний модуль забезпечує швидке і безперебійне отримання актуальних даних на заданий користувачем

період. В отриманих даних має міститись вся потрібна інформація для прогнозування курсу біткоїна за попередні дні, тижні або місяці.

Отримані з модуля постачальника дані мають містити інформацію про стан курсу біткоїна за попередній період, а саме: дату, за яку надаються дані; ціну відкриття за цю дату; ціну закриття; найбільшу та найменшу ціну за дату. В майбутньому для покращення результатів прогнозування можна додавати інші показники: об'єм торгів, кількість проведених операцій тощо. Дані мають бути синхронізовані з програмним модулем прогнозування курсу біткоїна, забезпечувати швидкий імпорт та автоматичний експорт усієї інформації, необхідної для прогнозування.

Алгоритм функціонування інформаційної технології прогнозування курсу криптовалюти «Біткоїн» можна поділити на алгоритм побудови математичної моделі та алгоритм роботи програмного модуля прогнозування криптовалюти Біткоїн.

Графічне подання алгоритму аналізу часового ряду та побудови математичної моделі прогнозування курсу криптовалюти Біткоїн зображено на рисунку 3.3.

Опишемо наведений алгоритм аналізу часового ряду та побудови математичної моделі прогнозування курсу криптовалюти Біткоїн (рис. 3.3).

1. Отримання даних часового ряду, проведення їх аналізу та обробки.
2. На основі оброблених даних будується графік часового ряду.
3. На основі отриманого графіку відбувається побудова тренду.
4. На основі отриманого графіку відбувається виділення сезонного компоненту.
5. Перевірка стаціонарності часового ряду. Якщо ряд стаціонарний відбувається перехід до кроку 7.
6. Розрахунок різниць одиничного кореню і перехід до кроку 5.
7. Вибір авторегресійної моделі для прогнозування.
8. Тестування вибраної авторегресійної моделі.

9. Перевірка вибраної моделі прогнозування. Якщо модель не задовольняє потреби тестувальника то відбувається перехід до кроку 7.
10. Кінець побудови математичної моделі.

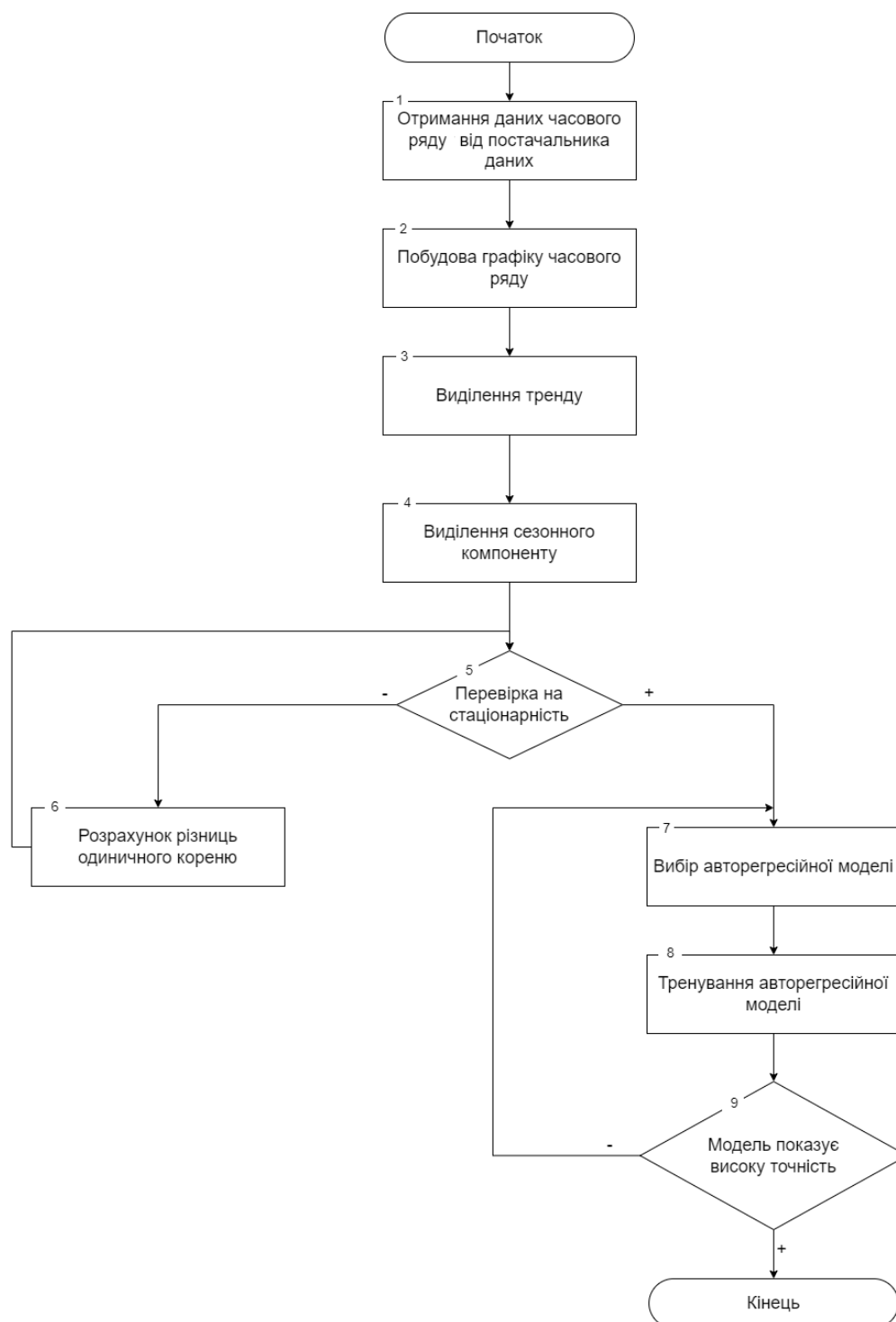


Рисунок 3.3 – Схема алгоритму побудови математичної моделі прогнозування курсу криптовалюти «Біткоїн»

Розглянемо алгоритм, який реалізує програмний модуль прогнозування курсу біткоїна (рис. 3.4).



Рисунок 3.4 – Схема загального алгоритму функціонування програмного модуля прогнозування курсу «Біткоїн»

Алгоритм складається з таких кроків:

1. Користувач входить до програмного модуля за допомогою графічного інтерфейсу, щоб отримати прогнозоване значення курсу біткоїна на період, який його цікавить він повинен авторизуватись в системі. Відбувається встановлення з'єднання з базою даних.
2. Перевірка авторизації користувача. Якщо користувач авторизований, то відбувається перехід до 4 кроку.
3. Якщо отримана помилка, то виводиться повідомлення про помилку та відбувається перехід до кроку 10.
4. З'єднання з постачальником даних.
5. Перевірка з'єднання з постачальником даних. Якщо з'єднання встановлено, то відбувається перехід до 6 кроку.
6. Виведення головного меню програмного модуля, яке містить спадаюче меню для вибору терміну прогнозування та відповідної кнопки для прогнозу.
7. Вибір терміну прогнозування за допомогою спадаючого меню.
8. Після натискання користувачем відповідної кнопки, відбувається прогнозування на основі авторегресійної моделі прогнозування.
9. Підготовка даних до побудови графіка на бажаний період.
10. Виведення графіка за допомогою графічного інтерфейсу програмного модуля.
11. Завершення роботи програмного модуля.

3.3 Обґрунтування вибору мови програмування

Мови програмування та технології забезпечують вирішення різних завдань та цілей. Деякі з них призначені лише для вирішення конкретного завдання, наприклад, для розробки сайтів або для програмних продуктів, інші – поєднують різні функції та вирішують наведені завдання. Для програмної реалізації

програмного модуля прогнозування курсу біткоїна потрібно розробити клієнтську та серверну частини.

Для розробки серверної частини розглядались мови php та Python.

PHP – це серверна скриптова мова програмування, яка використовується для створення динамічних веб-сторінок. Синтаксис мови бере початок з C, Java та Perl, і є легким для вивчення. Основною метою PHP є надання веб-розробникам можливості швидкого створення веб-сторінок, які динамічно генеруються, проте сфера застосування PHP не обмежується лише цим, а може також використовуватися для створення інших типів програм, таких як десктопні програми, мобільні додатки та ігри. PHP гарно підходить для написання скриптів та повноцінних веб-застосунків, які виконуються на серверній стороні. Для повноцінної роботи веб-програми, написаної на PHP, потрібні сервер, парсер (CGI-додаток) і клієнтське програмне забезпечення (веб-браузер), яке відображає результат виконання коду [21].

PHP – це проект відкритого програмного забезпечення, який можна безкоштовно використовувати і поширювати.

PHP є однією з найпопулярніших мов програмування для веб-розробки. PHP використовується для створення таких веб-сайтів, як WordPress, Drupal і Joomla. За 2022 рік PHP посів 5 місце у рейтингу найпоширеніших мов програмування.

Перевагами PHP є:

- Синтаксис. PHP гарно підходить для початку вивчення серверної мови програмування. Код PHP легко читається, незалежно від того, як він використовується (написання невеликих скриптів або створення потужних додатків, що реалізують програми з об'єктно-орієнтованим підходом).
- Гнучкість та кросплатформеність. PHP сумісний із усіма популярними платформами (Linux, Windows, MacOS). Програми, написані на PHP успішно працюють на найрізноманітнішому серверному забезпеченні (IIS, Apache, Nginx та інші).

- Масштабованість. Додатки, написані на PHP, можуть максимально збільшити продуктивність при збільшенні апаратних ресурсів. Веб-додатки, розподілені по декількох серверах, можуть справлятися з великим навантаженням (високим трафіком).
- Інтеграція з HTML. На звичайну HTML-сторінку у місця з динамічним контентом можна легко вставити блоки PHP коду. Вони додаються подібно до HTML-тегів, не порушуючи структуру сторінки.
- Розвиток. Спільнота розробників постійно працює над впровадженням додаткових можливостей, які розширюють функціональність мови, спрощують її синтаксис і покращують захист від можливих атак.
- Документація. На офіційному сайті проекту представлені загальні відомості про кожну функціональну одиницю мови з детальними прикладами використання.
- Вирішення проблем. В інтернеті існує безліч форумів, присвячених програмуванню на PHP.
- Перспективи. Більшість CMS були створені на чистому PHP та фреймворках. Цим зумовлені популярність та затребуваність PHP програмістів.

Недоліки PHP:

- Наявність «дір» у системі безпеки. Багато проблем було вирішено в 7-й версії, проте деякі вразливості, як і раніше, присутні.
- Націленість на веб-розробку. Незважаючи на наявність інших сфер застосування, мова програмування PHP продовжує асоціюватися з Інтернетом.
- Деякі протиріччя синтаксису. Оскільки над розвитком PHP працюють програмісти-ентузіасти, деякі їхні дії виявилися неузгодженими. Код містить елементи, характерні для різних мов, а також різні назви вбудованих функцій, у зв'язку з чим може бути плутанина.

Далі розглянемо мову Python. Python – це високорівнева, універсальна і дуже популярна мова програмування. Мова програмування Python використовується у веб-розробці, програмах машинного навчання, а також у всіх передових технологіях в індустрії програмного забезпечення. Мова програмування Python дуже добре підходить для початківців, а також досвідчених програмістів з іншими мовами програмування, такими як C++ і Java.

Нині Python-найпоширеніша мова програмування високого рівня загального призначення. Python дає змогу програмувати в об'єктно-орієнтованій і процедурній парадигмі. Програми на Python зазвичай менші, ніж на інших мовах програмування, таких як Java. Вони вимагають порівняно невеликого обсягу введення від програміста і завжди легко читаються завдяки необхідним відступам. Python використовують майже всі високотехнологічні гіганти, включно з Google, Amazon, Facebook, Instagram, Dropbox і Uber. Одна з головних переваг Python – велика колекція стандартних бібліотек [22].

Python використовується у двох основних сферах: наукових обчисленнях та науковій інформації. Він має широкий вибір бібліотек, ресурсів і ресурсів, корисних для дослідження.

Python має бути однією з найпростіших мов, оскільки він підкреслює читабельність коду. Його бібліотеки та база коду для читання допомагають програмістам оновлювати й підтримувати інформацію про програмне забезпечення без додаткових зусиль і витрат на курси розробки програмного забезпечення.

Концепція машинного навчання включає в себе класифікацію кількох речей, таких як фінансові послуги, поведінка розпізнавання голосу і навіть вибір місць на Netflix. Python використовується для машинного навчання через свої зрозумілі бібліотеки та фреймворки машинного навчання PyTorch, які включають дві з найпопулярніших – TensorFlow і scikit learn для кластеризації та вибору моделі.

Python є досконалою мовою в індустрії інвестиційного банкінгу та фінансових хедж-фондів. Він не має вбудованої підтримки алгебри лінійної

попередньої обробки, але натомість має досить простий спосіб написання коду. Аналітики хедж-фондів використовують найбільш зрозумілий спосіб представлення набору матриць і векторів у вигляді списків і вкладених списків.

Існує багато переваг використання Python як мови програмування, тому наведемо деякі з них:

- Прискорення роботи програми за допомогою спеціальних моделей виконання.
- Підтримка модулів і пакетів.
- Модулі використовують Інтернет-протоколи та надають допомогу в супутніх технологіях.
- Заохочує модульність програми та повторне використання коду.
- Забезпечує підвищення ефективності продуктів або проектів.
- Підтримує функціональну та процедурну мову програмування.
- Відкритий вихідний код та його розповсюдження можна вільно поширювати.
- Сумісний з платформами, базою даних, продуктами та програмам [23].

Отже, серверна частина інформаційної технології буде реалізована мовою програмування Python.

Для розробки клієнтської частини розглядалися мови Java, C#, JavaScript

Java – це об'єктно-орієнтована мова програмування, випущена в 1995 році Sun Microsystems як основний компонент платформи Java. Oracle працює над цією мовою з 2009 року і придбав Sun Microsystems того ж року. В офіційній реалізації програми Java компілюються в байт-код і інтерпретуються віртуальною машиною для певної платформи. Oracle надає компілятори Java та віртуальні машини Java, які відповідають специфікаціям процесів спільноти Java відповідно до публічної ліцензії GNU.

Мова запозичила багато синтаксису з C та C ++. Зокрема, за основу взята об'єктна модель C ++, однак вона модифікована. Можливість виникнення певних конфліктних ситуацій, які можуть виникнути через помилки програміста, була

полегшена, а процес розробки об'єктно-орієнтованих програм спрощений. Ряд дій, які програмісти мають виконати на C / C ++, доручаються віртуальній машині.

Java була розроблена насамперед як незалежна від платформи мова, тому вона має менші апаратні можливості, що зменшує швидкість програм у порівнянні, скажімо, з C ++. За необхідності такі дії дозволяють Java викликати підпрограми, написані іншими мовами програмування [24].

Серед переваг об'єктно-орієнтованої мови Java виділимо:

- Однією з головних переваг Java є незалежність платформи, на якій працюють програми: один і той же код може виконуватися на Windows, Solaris, Linux, Macintosh та інших. Це дійсно необхідно, коли програми завантажуються з Інтернету для подальшого запуску в різних операційних системах.
- Синтаксис мови Java подібний до синтаксису мови C ++, і для програмістів, які знайомі з C і C ++, це не складно навчитися.
- Java – повністю об'єктно-орієнтована мова, навіть більше, ніж C ++. Усі сутності в Java є об'єктами, за винятком кількох примітивних типів, таких як числа. (Об'єктно-орієнтоване програмування полегшує розробку складних проектів).
- Висока надійність. Творці надали інструменти для Java, які виключають можливість навіть створення додатків, які приховували б найпоширеніші помилки.

C # – це об'єктно-орієнтована мова програмування. Він був розроблений у 1998-2001 роках групою інженерів Microsoft під керівництвом Андерса Хейлсберга та Скотта Вілтомота як мова розробки додатків для платформи Microsoft .NET Framework [25].

C # належить до сімейства мов із C-подібним синтаксисом, синтаксис якого найближчий до C ++ та Java. Мова має статичний тип, підтримує поліморфізм, перевантаження операторів (включаючи явні та неявні оператори приведення),

делегати, атрибути, події, змінні, властивості, загальні типи та методи, ітератори, анонімні функції з тісною підтримкою, LINQ, винятки, коментарі в XML форматі.

Переїнявши багато у своїх попередників: мови C ++, Delphi, Modul, Smalltalk і, зокрема, Java, C # спираючись на практику їх використання, виключає певні моделі, які постали проблематичними в розробці програмного забезпечення системи, наприклад, C #, на відміну від C ++, не підтримує успадкування декількох класів (тим часом дозволяється впровадження декількох інтерфейсів).

Мова C # має кілька недоліків:

C # дуже легко розібрати. Це означає, що дуже ймовірно, що ваш код отримають і вивчать конкуренти. Звичайно, існують спеціальні засоби, які можуть ускладнити цей процес, але на 100% захиститися від нього практично неможливо.

.NET використовує концепцію компіляції JIT. Це означає, що програма буде компілюватися в машинний код за необхідності під час подання заявки. З одного боку, звичайно, це круто, але перший старт – це дуже серйозні «гальма».

C # – це не всюдисуша мова. Більшість програмістів зосереджуються на секторі комерційних підприємств, який накладає дуже серйозні обмеження на пошук роботи в малих містах, де вони бачили лише Delphi або PHP. Більше того, у всіх випадках C # в першу чергу асоціюється з Windows. В найближчому майбутньому нічого не зміниться, і Windows продовжуватиме домінувати на ринку, але все ще існує невеликий ризик.

JavaScript – це об'єктно-орієнтована мова, яка використовується для розробки веб-, мобільних, настільних та серверних додатків. Також мова має підтримку функціонального програмування [26]. JavaScript не компілюється, на відміну від C#, а інтерпретується інтерпретатором в реальному часі. Ця мова програмування широко використовується для створення кросплатформених додатків. JavaScript має велику кількість бібліотек для написання серверних, мобільних, настільних та веб-додатків.

JavaScript у порівнянні з C-подібними мовами має корінні відмінності: об'єкти, з можливістю інтроспекції; функції як об'єкти першого класу; автоматичне приведення типів; автоматичне прибирання сміття; анонімні функції [27].

JavaScript має велику підтримку спільноти. Існує велика кількість ресурсів з прикладами коду, вирішенням типових проблем, а також форумів, де можна знайти відповідь для рішення поставленої задачі. Низький поріг входу та популярність мови створює великий відсоток JavaScript розробників на ринку з надлишковою кількістю кадрів. Це формує умови для легкого пошуку кваліфікованих та легкозамінних працівників [27].

Зараз JavaScript популярний та стрімко розвивається. Розробники бібліотек та фреймворків стабільно випускають оновлення та реалізують новий функціонал, що спрощує написання проектів [27].

Недоліками JavaScript є:

- Відкрита архітектура. Відсутність архітектурних обмежень не створюють єдиного стилю написання коду, що провокує написання немасштабованого коду, який важко підтримувати. Цей недолік справедливий для використання JavaScript без фреймворків. Більшість фреймворків (наприклад Angular та NestJS) мають архітектурні обмеження.
- Нестрога типізація. Відсутність строгої типізації провокує написання низькоякісного коду з великою кількістю помилок та важкою підтримкою. Цей недолік є справедливим лише для JavaScript. TypeScript має строгу типізацію, що вирішує описані проблеми, але дотримання правил в TypeScript є опціональним.
- Низька швидкість роботи. Інтерпретація коду в реальному часі створює низьку потужність коду. Тому при написанні великих проектів потрібно робити оптимізацію коду. Незважаючи на недоліки JavaScript, ця мова є універсальним інструментом для написання кропслатформених додатків будь-яких розмірів з можливістю безпроблемної підтримки. Швидкість і

результативність розробки мовою JavaScript є кращою ніж у розглянутих аналогів [27].

3.4 Обґрунтування вибору фреймворків та бібліотек для розробки

Оскільки обрані мови програмування JavaScript і Python є одними з найпопулярніших на ринку на 2023 рік, для них створено величезну кількість фреймворків, які спрощують та пришвидшують процес програмування.

Розглянемо фрейворки клієнтської частини, для написання якої буде використовуватись мова JavaScript. Одразу потрібно зазначити, що для написання клієнтських застосунків можна використовувати і так званий «чистий» JavaScript, тобто без використання додаткових бібліотек або фреймворків. JavaScript має всю необхідну структуру для цього. Перевагами такого підходу є чистота коду та відносно висока швидкість роботи розробленого застосунку, оскільки завантажені модулі знижують швидкість роботи програми. Крім цього, перевагою такого підходу є відносна легкість в пошуку програміста в команду для підтримки проекту, оскільки різні програмісти на ринку можуть знати лише певний фреймворк, але всі вони знатимуть основи JavaScript. Головними недоліками є занадто низька швидкість написання клієнтських застосунків та складність в підтримці такого проекту, оскільки зазвичай програми, написані на «чистому» JavaScript досить громіздкі і в них важко розібратись. Тому такий підхід прийнятний для написання невеликих застосунків, типу сайту-визитівки.

Важливим критерієм при обранні фреймворків та бібліотек для розгляду є підтримка SPA. SPA – (з англ. single page application) односторінковий додаток. Раніше дані на веб-сторінці відображались статично і для оновлення інформації потрібно було перезавантажувати сторінку та проводити запит на отримання нових даних. Пізніше, сторінки стали більш інтерактивними, але навігація по сайту супроводжувалась завантаженням браузером нової сторінки. Сучасні додатки мають підтримку SPA, це дозволяє одноразово завантажити візуальну

частину для роботи додатку, а потім лише динамічно завантажувати або оновлювати дані для відображення без потреби в перезавантаженні сторінки. Це дозволяє покращити швидкість роботи додатку та дати користувачу новий досвід роботи з веб-додатками.

Проведемо порівняльну характеристику найпопулярніших фреймворків: Vue JS, Angular та бібліотеки React для реалізації клієнтської частини програмного модуля прогнозування курсу криптовалюти «Біткоїн».

Angular – фреймворк з відкритим кодом для мови JavaScript з модульною системою, розроблений у компанії Google. Основною ідеєю є архітектурні обмеження, які відсутні у звичайному JavaScript та строга типізація, яка реалізовується за допомогою використання мови програмування TypeScript (JavaScript з ООП надбудовою). Тобто Angular JS спонукає користувачів використовувати ООП стиль розробки, хоча також можлива розробка в функціональному стилі [28].

Angular насамперед призначений для спрощення складного процесу створення та управління JS-додатками. Він заснований на структурі MVC. Цей фреймворк особливо корисний для створення односторінкових сайтів. Оскільки бібліотека заснована на звичайному JS і HTML, Angular автоматично обробляє маніпуляції з DOM і AJAX-запити, які інакше розробникам довелося б писати самостійно. Angular надає модульні будівельні блоки JS-коду, які можна комбінувати й тестувати, а отже, їх можна швидко додати на будь-яку HTML-сторінку за допомогою простих тегів.

Angular дає дуже гарний контроль над даними на стороні клієнта – вам не потрібно думати про те, як вибрати елементи з DOM і помістити в них значення. Завдяки доступним прив'язкам даних можна оновлювати дані в JavaScript-частині та бачити зміни в HTML-частині. Це стосується і зворотної дії. Щойно користувач змінює щось у HTML-частині, JavaScript-частина отримує нові значення. Фреймворк також має сильну залежність від ін'єкцій: є зумовлені класи для виконання AJAX-запитів і управління маршрутами [29].

Переваги використання фреймворку Angular:

- Є повноцінним фреймворком, що має все необхідне для написання сучасних веб-застосунків без завантаження додаткових пакетів.

- Компонентний підхід. Цей підхід дозволяє розробку додатку у якості великої кількості структурних одиниць – компонентів. Використання компонентного підходу дозволяє створити додатковий рівень абстракції. З практичної сторони, компонентний підхід дозволяє створити універсальні компоненти, які будуть перевикористовуватись в різних місцях додатку, що пришвидшує розробку та впровадження нового функціоналу.

- Модульний підхід. Завдяки наявності модулів – чудово масштабується, якщо попередньо чітко планувати структуру застосунку й слідувати їй.

- Висока швидкість роботи при великій кількості функціоналу. Використання Angular дозволяє створювати комплексні рішення з великою кількістю функціоналу та має високу потужність при роботі під навантаженнями.

- Строга типізація. Використання TypeScript дозволяє розробнику краще описувати код та покращує підтримку і безпеку написаного коду.

Недоліки Angular:

- Високий поріг входу. Оскільки структура передбачає доволі великий обсяг знань, яким необхідно володіти щоб почати писати на Angular, крива входу є доволі крутою. Як наслідок, недоліком є висока складність, порівняно з іншими фреймворками й бібліотеками.

- Висока завантаженість. Оскільки Angular одразу після завантаження має велику кількість інструментів для розробки, деякі з них рідко використовуються на практиці, або мають кращі аналоги. Непотрібні модулі навантажують систему. При створенні невеликих проектів Angular має низьку потужність роботи.

- Низька швидкість роботи, яку дуже просто «зіпсувати». Але при цьому, якщо знати про те, як працює Angular Change Detection і як в цілому оптимізувати швидкість роботи й завантаження вебзастосунків, стає більш ніж конкурентною [30].

Vue.js – фреймворк для створення веб-додатків. На відміну від інших популярних бібліотек та фреймворків, Vue.js не був розроблений великою

компанією. Розробка починалась з етапу стартапу та переросла у самостійний проект. Vue.js дозволяє створювати SPA та може використовуватись як у якості бібліотеки (для розширення існуючого функціоналу), так і у якості фреймворку в якому буде проводитись розробка [31].

Переваги використання Vue.js:

- Можливість поступової інтеграції. Vue.js можна поступово використовувати при розробці проекту, підключаючи бібліотеку для окремих сторінок. У випадку коли потрібно написати новий проект, існує Vue CLI, який містить в собі усі необхідні інструменти для старту розробки.

- Низький поріг входу. Vue.js доволі легко вивчити без досвіду з іншими бібліотеками. Фреймворк має детальну документацію та велику кількість уроків.

- Мутабельність. Мутабельність – стиль роботи з референсним типом даних при якому для введення змін до об'єкту не створюється ідентичний об'єкт з тим самим набором полів але з новими даними, а змінюється існуючий оновлюючи дані. Мутабельність Vue.js дозволяє створювати підписки на один об'єкт та модифікувати його з будь-якого місця додатку, а зміни динамічно будуть відображатись у всіх місцях.

- Підтримка SPA. Vue.js дозволяє створювати SPA, що підвищує актуальність використання.

- Велика кількість інструментів. Vue CLI містить велику кількість інструментів для розробки додатків. Необхідний функціонал вже реалізований і не використання сторонніх бібліотек не необхідним. Також Vue.js має широкий функціонал для полегшення розробки візуальної частини додатку.

Недоліки використання:

- Маленька спільнота. Спільнота Vue.js ще формується так як фреймворк новий. Нехватка спеціалістів та досвідчених програмістів зменшує шанс отримання кваліфікованої допомоги та ускладнює пошук кадрів.

- Ризик завеликої гнучкості. У Vue.js можуть з'являтись проблеми з інтеграцією у великі проекти, а прикладів комбінування декількох бібліотек або фреймворків мало [30].

React являє собою бібліотеку для значного спрощення побудови й маніпулювання DOM елементами. Творцем цієї бібліотеки виступає компанія Facebook. Спочатку вони розробляли й використовували React безпосереднього лише для написання своєї соцмережі, і тільки згодом зробили публічною, надавши до неї доступ широкому загалу [30].

Переваги використання React:

- Компонентний підхід. Аналогічно до Angular, дана бібліотека підтримує компонентну структуру написання коду.

- Низький поріг входу. Синтаксис є максимально зрозумілим, якщо вам знайомі Javascript і HTML.

- Оптимізація SPA. React використовує власну концепцію Shadow DOM, яка зберігає декілька копій візуального інтерфейсу: поточну версію та майбутню версію зі змінами які будуть оновлені та відображені користувачу. Після змін React порівнює ці дві копії і вносить зміни лише в ті DOM елементи, які безпосередньо змінились, а не повністю переписує DOM дерево.

- Підтримка спільноти. React – стрімко ростуча бібліотека з великою підтримкою зі сторони розробників та спільноти. Для React створено велику кількість бібліотек які спрощують роботу. Публічні ресурси містять багато матеріалів для вивчення та професійного розвитку з використанням React.

- JSX. JSX – синтаксис бібліотеки React, який дозволяє писати код, використовуючи в одному файлі JavaScript, HTML та CSS. Це зменшує обсяг коду та спрощує розробку.

Недоліки React:

- Відсутність архітектурних обмежень. React не накладає на розробника архітектурних обмежень при створенні додатку, що може призвести до створення немасштабованої, незручної структури, при написанні коду недосвідченим розробником.

- Оскільки після завантаження бібліотеки React не включає практично жодних додаткових можливостей, окрім роботи з DOM, для нього створена велика кількість бібліотек та утиліт. Розробка і підтримка таких бібліотек

найчастіше виконується самими програмістами на безоплатній основі. Відповідно є ризик що якась вибрана на початку розробки бібліотека пізніше перестане підтримуватись або оновлюватись [30].

Порівнявши Angular, React та Vue.js, для створення клієнтської частини інформаційної технології прогнозування курсу криптовалюти було обрано бібліотеку React через її оптимізацію для створення SPA, широку підтримку спільноти та підхід до створення проектів з невеликою кількістю функціоналу.

Для розробки серверної частини додатку було розглянуто два фреймворки: Django та Flask.

Django – високорівневий фреймворк мови програмування Python для швидкого створення вебсистем. Архітектура Django схожа до архітектури MVC (Model-view-controller), яка використовується у багатьох інших фреймворках та мовах програмування, проте Django має деякі технічні відмінності. Те що називається «контролером» в класичній моделі MVC, в Django називається «вигляд», а те, що мало б бути «виглядом», називається «шаблон». Django є open source фреймворком, тобто його код можна вільно використовувати, читати та змінювати [32].

Перевагами Django є:

- Є повноцінним фреймворком, де одразу після завантаження доступна більшість необхідних інструментів для розробки серверних додатків. Наприклад власна ORM система, підтримка міграцій бази даних, вбудована система аутентифікації користувача, власна панель адміністратора та багато іншого.

- Структура. Django як фреймворк визначає структуру проекту. Вона допомагає розробникам зрозуміти, де і як додавати нову функціональність. Завдяки тому, що всі проекти мають однакову структуру, набагато простіше знайти готові рішення й отримати допомогу від спільноти. Величезна кількість відданих своїй справі розробників допоможе вам швидко впоратися з будь-яким завданням.

- Різносторонність. На Django можна побудувати практично будь-який тип веб-сайту, від систем управління контентом до соціальних мереж і новинних

сайтів. Він працює в будь-якому клієнтському середовищі і може надавати контент практично в будь-якому форматі (HTML, RSS-канали, JSON, XML тощо).

- Безпека. Одразу після завантаження фреймворк Django включає механізми запобігання поширених атак на зразок SQL-ін'єкцій (XSS) і підробки міжсайтових запитів (CSRF). Також Django надає безпечний спосіб керування обліковими записами користувачів та паролями, уникаючи поширених помилок, таких як розміщення інформації про сеанс у файли cookie (замість цього файли cookie містять тільки ключ, а фактичні дані зберігаються в базі даних) або безпосереднє зберігання паролів замість хеша пароля.

- Масштабованість. У Django використовується компонентна архітектура «share-nothing» (кожна частина не залежить від інших і може бути замінена або змінена в міру необхідності). Чіткий поділ компонентів означає, що Django може масштабуватися в міру зростання трафіку шляхом додавання обладнання будь-якого рівня – кеш-сервера, сервера баз даних, сервера додатків тощо[33].

Недоліки фреймворку Django:

- Django ORM. Django ORM сьогодні значно поступається конкуренту – останній SQLAlchemy. Django ORM заснований на шаблоні Active Record, який гірше, ніж шаблон Unit of Work, який використовується в SQLAlchemy. На практиці це виражається в тому, що в Django моделі можуть «зберігати» себе за бажанням, а транзакції відключені за замовчуванням.

- Повільний розвиток. Django – великий, монолітний фреймворк. Це дає змогу спільноті розробляти сотні багатofункціональних модулів і застосунків, але уповільнює розвиток самого Django. Крім того, фреймворк має підтримувати зворотну сумісність, тому розробка йде відносно повільно [34].

Flask – мікрофреймворк для вебдодатків, створений для мови програмування Python. Його основу складає інструментарій Werkzeug та рушій Jinja2. Він називається мікрофреймворком, оскільки не вимагає спеціальних засобів чи бібліотек. У Flask відсутній функціонал для роботи з базою даних, перевірки форм або інші компоненти, які надають широковживані функції за

допомогою сторонніх бібліотек. Він надає лише основні інструменти для створення веб-застосунків, такі як роутинг, шаблони та запити HTTP. Але при цьому залишає багато місця для гнучкості та розширення. Це дозволяє розробникам легко налаштовувати і використовувати тільки необхідні компоненти, що робить Flask придатним для малих і середніх проектів [35].

Переваги Flask:

- Продуктивність. Оскільки Flask – мікрофреймворк, де одразу після завантаження доступна невелика частина інструментів, швидкість роботи програмного продукту значно пришвидшується.

- Гнучкість. Ще однією перевагою того що, Flask є мікрофреймворком є велика кількість різних плагінів, розроблених ентузіастами для найрізноманітніших задач. Це дозволяє налаштувати проект під себе, підбравши максимально зручні інструменти для розробки.

- Низький поріг входу. Початковому програмісту легше розібратись з невеликим фреймворком, де не потрібно слідувати чіткій структурі продукту.

- Підтримка REST API запитів. Дозволяє швидко впровадити і легко обмінюватись даними з клієнтською частиною застосунку без додаткових налаштувань системи.

Недоліки Flask:

- Слабка масштабованість. Оскільки Flask одразу після завантаження не надає чіткої архітектури для розбудови застосунку, з його ростом в майбутньому можуть виникнути проблеми з масштабуванням.

- Робота з базами даних. Flask не пропонує жодних рішень для управління базами даних. Це дає можливість вибрати ОРМ на власний розсуд проте час розробки збільшується і обробка можливих помилок стане складнішою.

- Безпека. Чим далі розвиватиметься проект тим більше сторонніх модулів потрібно буде встановлювати. Відповідно збільшується ризик завантаження неякісних або зловмисних бібліотек. Також ускладниться і підтримка розробленого продукту, оскільки після встановлення певного модуля, він залежний від команди, яка його розробила. Такі команди мають різну кількість

учасників, свою документацію та цикл випуску оновлень, що може також вплинути на швидкість розробки.

Порівнявши Django та Flask для створення серверної частини інформаційної технології прогнозування курсу криптовалюти було обрано фреймворк Flask, оскільки він більше підходить для розробки невеликих застосунків та дає можливість самому налаштувати структуру програмного модуля.

Оскільки Flask не має своїх модулів для використання бази даних та автентифікації користувача потрібно додатково встановити сервіс Firebase, який надасть потрібний функціонал.

Firebase – платформа, створена компанією Google, яка представляє собою комплексне рішення для розробки мобільних, веб та серверних додатків. Система містить в собі сервіси для виконання різноманітних задач.

Для розробки серверної частини інформаційної технології прогнозування курсу криптовалюти необхідно буде використовувати сервіси Firebase RealTime Database та Firebase Authentication.

Firebase RealTime Database – база даних, яка розміщена в хмарному сховищі. Вона дозволяє зберігати дані в форматі JSON і синхронізувати їх в реальному часі без необхідності робити додаткові запити до бази. Для цього використовуються хмарна база даних NoSQL. Це надає змогу залишити дані доступними навіть коли застосунок буде відключеним [36].

Firebase Authentication надає серверні служби і готові бібліотеки для автентифікації користувачів. Автентифікація потрібна для безпечного збереження в сховищі і відображення інформації, притаманної лише певному користувачу. З допомогою Firebase Authentication можна налаштувати автентифікацію з використанням паролю, номеру телефону, електронної пошти, постачальників посвідчень таких як Google, Facebook, Twitter тощо [37].

Недоліком Firebase є ціна використання, яка зростає в залежності від кількості запитів на сервер та збережених даних в хмарному сховищі.

3.5 Реалізація програмного модуля інформаційної технології прогнозування курсу криптовалюти «Біткоїн»

Для початку роботи програми потрібно провести автентифікацію користувача. Це потрібно для майбутнього збереження результатів роботи програми та отримання історії прогнозів саме цього користувача. Для автентифікації було обрано сервіс Firebase Authentication. Було створено шлях для REST API запити, який приймає електронну пошту та пароль. Якщо такий користувач є в системі Firebase Authentication, у якості відповіді на API запит повертається унікальний id автентифікованого користувача.

Діаграма діяльності серверу під час процесу автентифікації користувача зображена на рисунку 3.5



Рисунок 3.5 – Діаграма діяльності серверу під час процесу автентифікації користувача

Зображений вище алгоритм (рис. 3.5) можна записати у вигляді коду:

```
@app.route("/login", methods=["POST"])
def login():
    try:
        data = request.get_json()
        email = str(data["email"])
        password = str(data["password"])
        user = auth.sign_in_with_email_and_password(email, password)
        print(user)
        user_id = user["localId"]
        id_token = user["idToken"]

        response = {
            "message": "User authenticated successfully.",
            "user_id": user_id,
            "id_token": id_token,
        }
        return jsonify(response)

    except Exception as e:
        return jsonify({"error": str(e)})
```

Після автентифікації користувач отримує доступ до основного функціоналу програмного модуля. Після цього у нього є можливість або отримати прогноз на вибраний період або отримати історію прогнозів, якщо вона є.

Якщо користувач вибере на який термін він хоче отримати прогноз, відбудеться запит на сервер для отримання прогнозу. На сервері після отримання даних від клієнта відбудеться побудова математичної моделі для прогнозування курсу криптовалюти «Біткоїн». Алгоритм побудови цієї моделі, а також теоретичні відомості для його реалізації були описані у попередньому розділі. Розглянемо кроки з цього алгоритму.

Першим кроком є отримання актуальних даних з постачальника даних. В якості постачальника даних було обрано Binance API. Binance – найпопулярніша криптовалютна біржа на 2023 рік з добовим об’ємом торгівлі більше ніж 8 млрд \$. Binance представляє свій Restful сервіс для отримання найактуальніших

даних криптовалютного ринку, а саме: час відкриття і закриття, ціну відкриття і закриття в цей час, найнижчу та найвижчу ціну за цей час, об'єм торгівлі за цей час, кількість операцій. Усі ці дані можна використовувати як параметри для навчання авторегресійної моделі.

Запишемо код, для отримання даних з Binance API та перетворення цих даних у спеціальний тип DataFrame, потрібний для навчання авторегресійної моделі.

```
def get_bitcoin_prices(start_time, end_time, interval):
    url = "https://data-api.binance.vision/api/v3/klines"
    params = {
        "symbol": "BTCUSDT",
        "interval": interval,
        "startTime": start_time,
        "endTime": end_time,
    }
    response = requests.get(url, params=params)
    data = response.json()
    bitcoin_data_float = {"date": [], "close": []}
    for row in data:
        bitcoin_data_float["date"].append(float(row[0]))
        bitcoin_data_float["close"].append(float(row[4]))
    df = pd.DataFrame(bitcoin_data_float)
    df["date"] = pd.to_datetime(df["date"], unit="ms", origin="unix")
    df.set_index(["date"], inplace=True)
    return df
```

Наступним кроком є перевірка на стаціонарність отриманого часового ряду. Для цього використовується тест Дікі-Фулера з бібліотеки statsmodels. StatsModels – це пакет для Python, який дозволяє користувачам досліджувати дані, оцінювати статистичні моделі та виконувати статистичні тести [38].

Діаграма діяльності сервера під час процесу перевірки на стаціонарність зображена на рисунку 3.6.

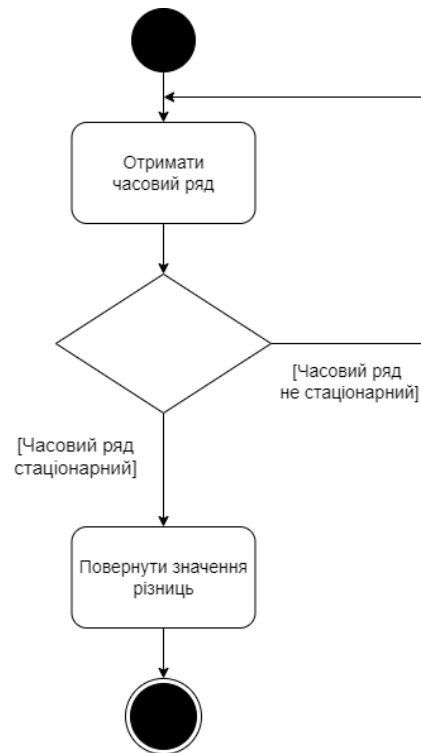


Рисунок 3.6 – Діаграма діяльності серверу під час процесу перевірки на стаціонарність

Зображений вище алгоритм (рис. 3.6) можна записати у вигляді коду:

```

def stationarity_check(data, max_diff=5):
    for i in range(max_diff + 1):
        stationarity_result = adfuller(data)

        print("ADF value:", stationarity_result[0])
        print("p-value:", stationarity_result[1])
        print("Critical values:")

        if (stationarity_result[1] <= 0.05) and (
            stationarity_result[4]["5%"] > stationarity_result[0]
        ):
            print("\u001b[32mStationary\u001b[0m")
            return i

        data = np.diff(data)
        print("\x1b[31mNon-stationary\x1b[0m")

    return max_diff
  
```


Було обрано математичну модель інтегрованого ковзного середнього (ARIMA) для прогнозування курсу криптовалюти «Біткоїн». ARIMA приймає три параметри p, d, q . Параметр d – це значення різниць, яке ми отримуємо як результат роботи тесту Дікі-Фулера. Для знаходження параметрів q та p скористаємось функціями автокореляції та часткової автокореляції з бібліотеки `statsModels` відповідно:

```
def qvalue_acf(data, max_qvalue=0.07, min_qvalue=-0.07, max_lag=5):
    acf_values = acf(data, nlags=max_lag)
    for lag in range(1, max_lag + 1):
        if acf_values[lag] > max_qvalue or acf_values[lag] < min_qvalue:
            return lag

    return 0
```

```
def pvalue_pacf(data, max_pvalue=0.07, min_pvalue=-0.07, max_lag=5):
    pacf_values = pacf(data, nlags=max_lag)
    for lag in range(1, max_lag + 1):
        if pacf_values[lag] > max_pvalue or pacf_values[lag] < min_pvalue:
            return lag

    return 0
```

Після того як визначено всі три параметри можна приступати до тренування моделі. Створимо шлях для REST API запити, який прийматиме дату початку і кінця часового ряду, потрібного для прогнозу та айді користувача. Цей шлях має зберігати результати прогнозу у `FireBase RealTime Database` та повертати результат прогнозу. Код, який реалізує таку поведінку наведений нижче.

```
@app.route("/forecast", methods=["POST"])
def forecast():
    try:
        data = request.get_json()
        start_time = int(data["start_time"])
        end_time = int(data["end_time"])
        interval = str(data["interval"])
        user_id = str(data["user_id"])
        bitcoin_data = get_bitcoin_prices(start_time, end_time, interval)
        rows = int(len(bitcoin_data) * 0.9)
        training_df = list(bitcoin_data[0:rows][["close"]])
```

```

testing_df = list(bitcoin_data[rows:]["close"])
d_value = stationarity_check(training_df)
diff = training_df
for i in range(d_value):
    diff = np.diff(diff)

p_value = pvalue_pacf(diff)
q_value = qvalue_acf(diff)
seasonal = seasonal_check(bitcoin_data, p_value, d_value, q_value)
model_predictions = []
target = len(testing_df)
for i in range(target):
    model = ARIMA(
        training_df,
        order=(p_value, d_value, q_value),
        trend=None,
        seasonal_order=seasonal,
    )
    model_fit = model.fit()
    output = model_fit.forecast()
    result = list(output)[0]
    model_predictions.append(result)
    actual_test_value = testing_df[i]
    training_df.append(actual_test_value)

def forecast_direction(forecast, testing_df):
    last_testing_price = testing_df[-1]
    forecast_price = forecast[0]

    return bool(forecast_price > last_testing_price)

forecast = list(model_fit.forecast())
automape = mean_absolute_percentage_error(testing_df, model_predictions)
forecast_data = {
    "date": datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S"),
    "forecast": forecast,
    "mape": automape,
    "direction": forecast_direction(forecast, testing_df),
    "interval": interval,
}
ref.child(user_id).push(forecast_data)

response = {
    "date": datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S"),
    "forecast": forecast,
    "mape": automape,
    "direction": forecast_direction(forecast, testing_df),
    "interval": interval,
}

```

```

    }
    return jsonify(response)

except Exception as e:
    return jsonify({"error": str(e)})

```

Створимо останній шлях, який потрібний для отримання історії прогнозів користувача. Він має приймати айді користувача, який хоче отримати свою історію прогнозів та повернути з бази FireBase RealTime Database ці дані. Код, який реалізовує таку поведінку наведений нижче:

```

@app.route("/user_forecasts", methods=["POST"])
def get_forecasts():
    try:
        data = request.get_json()
        user_id = str(data["user_id"])
        forecasts_ref = ref.child(user_id)
        forecasts = forecasts_ref.get()

        if forecasts is None:
            forecasts = []

        response = {
            "message": "User forecasts retrieved successfully.",
            "forecasts": forecasts,
        }
        return jsonify(response)

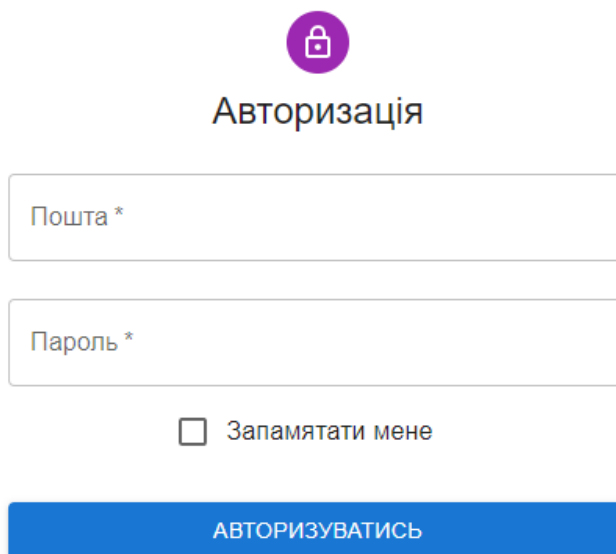
    except Exception as e:
        return jsonify({"error": str(e)})

```

3.6 Тестування інформаційної технології прогнозування курсу криптовалюти «Біткоїн» та аналіз отриманих результатів

Для того щоб почати роботу з інформаційною технологією, потрібно мати персональний комп'ютер. Головною умовою є стабільне підключення до мобільної мережі інтернету або Wifi.

На рисунку 3.7 зображено вікно авторизації користувача.



Авторизація

Пошта *

Пароль *

Запам'ятати мене

АВТОРИЗУВАТИСЬ

Рисунок 3.7 – Загальний вигляд інтерфейсного вікна авторизації користувача

Після успішної авторизації користувач переадресовується на вікно головного меню програми. Це вікно містить всю необхідну користувачу інформацію. Наприклад у цьому вікні відображається графік криптовалюти «Біткоїн» за останні 30 днів, деяка інформація про ціну за останній день, а саме: остання ціна, найбільшу та найменшу ціну, а також обсяг торгівлі. Також у цьому вікні знаходиться коротка таблиця історії останніх прогнозів користувача. Ця таблиця містить колонки: дата прогнозу, тип прогнозу, числове значення прогнозу, приблизна точність прогнозу та напрямок руху ціни. Біля графіку знаходиться кнопка вибору горизонту прогнозування, яка запускає початок прогнозу. У лівій частині вікна знаходиться випадаюче меню, яке надає навігацію по програмному модулю. Звідти можна перейти в головне вікно програми, в особистий кабінет або в вікно, де відобразатимуться прогнози за останній день, тиждень або місяць.

На рисунку 3.8 подано «Головне вікно програми», де відображається графік ціни за останні 30 днів та навігаційне меню.

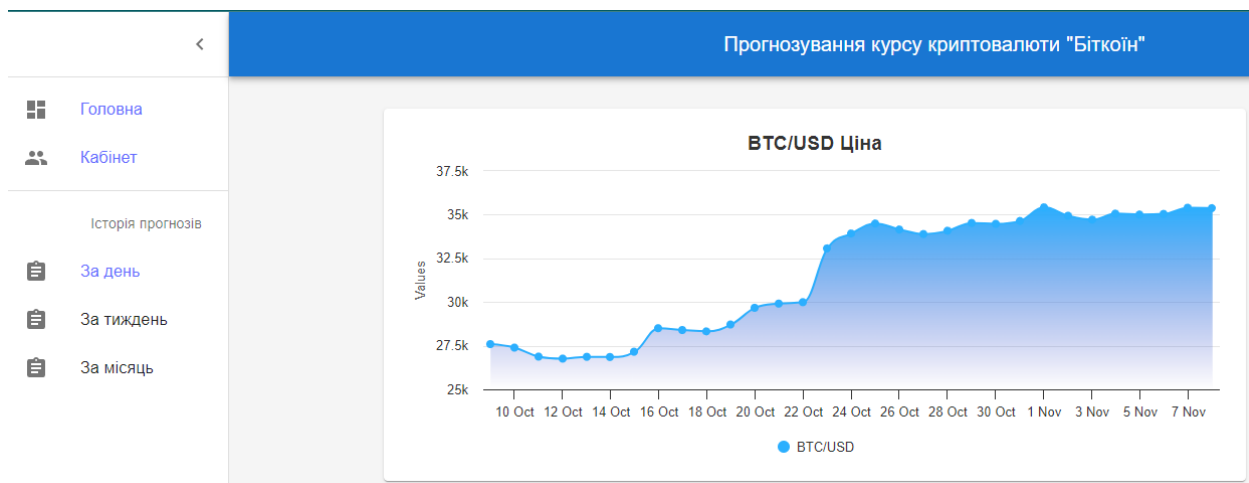


Рисунок 3.8 – Загальний вигляд інтерфейсного вікна головного меню з графіком

На рисунку 3.9 подано «Головне вікно програми», де відображається інформація про останню ціну «Біткоїна» та кнопка вибору горизонту прогнозування.

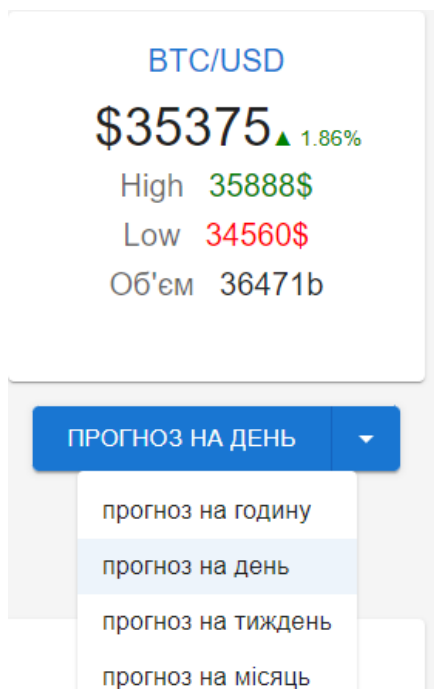


Рисунок 3.9 – Загальний вигляд інтерфейсного вікна головного меню з ціною та кнопкою

На рисунку 3.10 подано «Головне вікно програми», де відображається таблиця з останніми прогнозами користувача.


Останні прогнози				
Дата	Прогноз	Тип прогнозу	Точність	Напрямок
2023-11-08 17:03:21	35318	На годину	77%	
2023-11-08 16:58:23	35549	На день	71%	
2023-11-04 14:40:24	35910	На місяць	69%	
2023-11-04 13:40:36	34728	На годину	77%	

Рисунок 3.10 – Загальний вигляд інтерфейсного вікна головного меню з прогнозами

Для отримання прогнозу на бажаний період потрібно зі списку заданих періодів прогнозування вибрати потрібний і натиснути на кнопку «Прогноз на:». Оскільки процес тренування і прогнозування моделі займає проміжок часу, більший ніж звичайний http запит, який користувачі виконують щодня, спочатку відкриється модальне вікно з повідомленням про те, що відбувається прогноз. На рисунку 3.11 зображено це вікно очікування. Після того як запит виконається, виведеться графік прогнозу, де міститиметься інформація про дані ціни біткоіна, які були використані для прогнозу і сам прогноз. Такий графік можна приближувати і віддаляти для детальнішого огляду отриманих результатів.

Ваш прогноз на день

×



Зачекайте будь
ласка!

Відбувається
прогноз

Рисунок 3.11 – Загальний вигляд інтерфейсного вікна очікування прогнозу

На рисунку 3.12 та рисунку 3.13 відображено вікно отриманого прогнозу та вікно масштабованого графіку відповідно.



Рисунок 3.12 – Загальний вигляд інтерфейсного вікна отриманого прогнозу



Рисунок 3.13 – Загальний вигляд інтерфейсного масштабованого вікна прогнозу

Для того щоб переглянути історію прогнозів, потрібно натиснути на одну з 3 кнопок «За день», «За тиждень», «За місяць» у випадяючому вікні в секції історія прогнозів. Після натиснення відбудеться переадресація на сторінку, де виведуться усі прогнози за вибраний період часу. При натисканні на будь-який із прогнозів відкриється модальне вікно, де як і в випадку отриманого прогнозу можна буде переглянути графік.

На рисунку 3.14 подано вікно історії прогнозів.

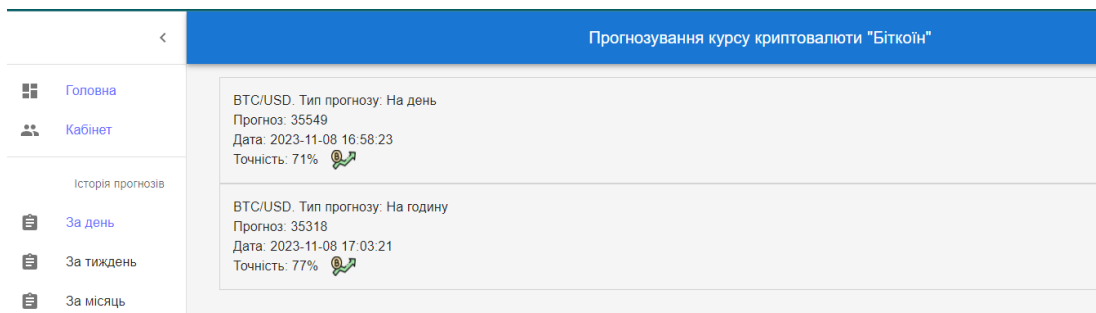


Рисунок 3.14 – Загальний вигляд інтерфейсного вікна історії прогнозів

Ще одним важливим фактором для тестування інформаційної технології прогнозування курсу криптовалюти «Біткоїн» є показник точності прогнозування. Про точність прогнозу зазвичай судять за величиною помилки прогнозу, тобто за різницею між прогнозом і фактичним значенням змінної, що цікавить. Однак такий підхід до оцінки точності можливий тільки у двох випадках. Перший – коли період прогнозування вже закінчився і дослідник має у своєму розпорядженні фактичні значення змінних. Для короткострокових прогнозів це цілком реально; другий випадок – коли прогнози перебувають у стадії формулювання, тобто прогнозуються на якийсь момент у минулому, для якого вже є фактичні дані. У цьому разі наявну інформацію можна розділити на дві частини. Перша частина містить 90% часового ряду – це тренувальний датасет. Він потрібен для тренування авторегресійної моделі ARIMA. Інші 10% – валідаційний датасет, на якому буде відбуватись перевірка роботи моделі.

Проте перевірка точності одного прогнозу мало що може сказати досліднику. Гарний одиничний прогноз може бути отриманий і по поганій моделі, і навпаки. Звідси випливає, що про якість прогнозів застосовуваних методик і моделей можна судити лише по сукупності зіставлень прогнозів і їхньої реалізації.

Існують такі метрики для визначення точності прогнозування моделі: MSE – середнє квадратів похибок прогнозу, RMSE – середньоквадратична похибка прогнозу, MAD – середня абсолютна похибка, MAPE – середнє значення абсолютних величин відносних похибок. Перші три критерії виражають похибку в одиницях вимірювання і тому їх величина залежить від специфіки часового ряду. Останній критерій вимірюється у відносних одиницях, тому можна говорити про деякий загальний рівень адекватності моделі на основі його порівняння. До того ж він більш зрозумілий простому користувачу. Значення MAPE для оцінки точності прогнозування наведено в таблиці 3.1. Проте потрібно зазначити, що наведені значення не є постійними і змінюються в залежності від мети прогнозування. Дослідник сам встановлює межі показника точності, які його задовольняють [17].

В випадку криптовалютної сфери, яка більш волатильна і відповідно складніше піддається точному прогнозуванню прийнято зменшувати значення критерію MAPE на 10% при оцінці точності прогнозування.

Таблиця 3.1 Значення MAPE для оцінки точності прогнозування

MAPE	Точність прогнозу
> 90%	Висока
80 – 90%	Добра
60 – 80%	Задовільна
50 – 60%	Погана
<50%	Незадовільна

Провівши 100 ітерацій роботи програми було встановлено, що середнє значення критерію MAPE розробленої інформаційної технології дорівнює 72%. Аналогічні розрахунки було проведено з програмами-аналогами WalletInvestor,

BellInvestor, LongForecast. У таблиці 3.2 наведено результат порівняння точності прогнозу розробленого додатку з аналогами.

Таблиця 3.2 Порівняння результату роботи програм

Програмний продукт	Точність прогнозування
Розроблена інформаційна технологія	73%
WalletInvestor	55%
BellInvestor	68%
LongForecast	63%

Проаналізувавши можливості та переваги розробленого програмного модуля інформаційної технології прогнозування курсу криптовалюти «Біткоїн», можна зробити висновок, що така інформаційна технологія є ефективним засобом для прогнозування курсу біткоіна. З її допомогою можна отримати прогноз ціни на визначений період часу, отримати ілюстративний графік та перевірити історію прогнозування. Реалізація цієї інформаційної технології зможе надати користувачам новий інструмент для аналізу криптовалютного ринку та перевірки своїх стратегій торгівлі.

3.7 Висновок до розділу 3

У третьому розділі обґрунтовано вибір архітектури інформаційної технології прогнозування курсу криптовалюти «Біткоїн». Для реалізації інформаційної технології прогнозування курсу криптовалюти «Біткоїн» було обрано клієнт-серверний тип архітектури, оскільки така архітектура підтримує багатокористувацьку роботу, гарантує цілісність і безпеку даних, має гарну оптимізацію та можливість розподілу функцій між вузлами мережі. Розроблено схему алгоритму роботи програмного модуля та описано детально його кроки, щоб мати набір послідовних інструкцій для реалізації програмного забезпечення.

Проаналізовано мови програмування для створення як клієнтських частин: Java, C#, JavaScript, так і серверних частин: PHP та Python. Визначено їх особливості, переваги та недоліки, а також доцільність у використанні для поставленого завдання. В результаті аналізу для реалізації клієнтської частини програмного модуля для інформаційної технології було обрано мову JavaScript, а для серверної частини – Python. Переваги мови програмування JavaScript: наявність великої кількості бібліотек та фреймворків, динамічність та гнучкість. Перевагами Python є підтримка модулів та оптимізація ресурсозатратних обчислень.

Проаналізовано такі фреймворки як React, Vue js, Angular. Для реалізації клієнтської частини інформаційної технології прогнозування курсу криптовалюти «Біткоїн» обрано React, оскільки він має гарну оптимізацію SPA та використовує концепцію Shadow DOM. Для реалізації серверної розглядалися фреймворки Django та Flask. Було обрано Flask, оскільки він гарно підходить для створення невеликих застосунків через малу вагу та швидкість роботи.

Описано процес реалізації авторизації користувача та створення авторегресійної моделі інтегрованого ковзкого середнього (ARIMA), наведені їх діаграми діяльності сервера. Проведено розрахунок p , d , q значень для моделі ARIMA. Також продемонстровано створення Rest шляхів, для можливості обміну інформацією між клієнтською та серверною частинами.

Під час досліджень програма показала себе як якісний продукт, що відповідає усім поставленим вимогам. Проведено тестування реалізованого програмного модуля інформаційної технології прогнозування курсу криптовалюти «Біткоїн», яке підтвердило правильність його функціонування.

Порівнюючи результати роботи розробленого програмного продукту з програмами-аналогами WalletInvestor, BellInvestor та LongForecast, за критерієм точності MAPE, розробка забезпечує точність прогнозування на 5% вище ніж у аналога BellInvestor, на 10% вище ніж у аналога WalletInvestor та на 18% вище ніж у аналога LongForecast. Тестування інформаційної технології показало повну відповідність до поставлених задач.

4 ЕКОНОМІЧНА ЧАСТИНА

Ефективне впровадження науково-технічної розробки стає можливим, якщо вона відповідає поточним вимогам науково-технічного прогресу та враховує економічні аспекти. Надання оцінки економічної ефективності отриманих результатів науково-дослідної роботи є важливою частиною цього процесу.

Магістерська робота, що присвячена розробці та дослідженню «Інформаційна технологія прогнозування курсу криптовалюти «Біткоїн»», віднесена до науково-технічних робіт, спрямованих на введення на ринок. Рішення про комерціалізацію розробки може бути прийняте протягом самої роботи, дозволяючи реалізувати можливість виведення її на ринок. Цей напрямок розглядається як пріоритетний, оскільки розроблені результати можуть бути корисними для різних зацікавлених сторін і приносити економічні вигоди. Однак для успішного втілення цього процесу важливо знайти зацікавленого інвестора, який був би зацікавлений у реалізації цього проекту, і переконати його в обґрунтованості таких інвестицій.

Для цього визначені наступні етапи виконання робіт:

1. Проведено комерційний аудит науково-технічної розробки, що включає в себе визначення науково-технічного рівня та комерційного потенціалу.
2. Розраховані витрати на реалізацію науково-технічної розробки.
3. Проведено розрахунок економічної ефективності науково-технічної розробки в разі її впровадження та комерціалізації потенційним інвестором, а також обґрунтовано економічну доцільність комерціалізації для інвестора.

4.1 Проведення комерційного та технологічного аудиту науково-технічної розробки

Метою проведення комерційного і технологічного аудиту дослідження за темою «Інформаційна технологія прогнозування курсу криптовалюти «Біткоїн»»

є оцінювання науково-технічного рівня та рівня комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності.

Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням 5-ти бальної системи оцінювання за 12-ма критеріями, наведеними в таблиці 4.1

Таблиця 4.1 – Рекомендовані критерії оцінювання науково-технічного рівня і комерційного потенціалу розробки та бальна оцінка

Бали (за 5-ти бальною шкалою)					
	0	1	2	3	4
Технічна здійсненність концепції					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено працездатність продукту в реальних умовах
Ринкові переваги (недоліки)					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою

Продовження таблиці 4.1

Бали (за 5-ти бальною шкалою)					
	0	1	2	3	4
Ринкові перспективи					
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування	Потрібні незначні фінансові ресурси. Джерела фінансування	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Результати оцінювання науково-технічного рівня та комерційного потенціалу науково-технічної розробки потрібно зведені до таблиці 4.2. Для опитування було залучено експертів з ВНТУ, кафедри комп'ютерних наук: к.т.н., доцент Крилик Людмила Вікторівна, к.т.н., доцент Озеранський Володимир Сергійович, к.т.н., проф. Колесницький Олег Костянтинович.

Таблиця 4.2 – Результати оцінювання науково-технічного рівня і комерційного потенціалу розробки експертами

Критерії	Експерт (ПІБ, посада)		
	Крилик Л.В.	Озеранський В.С.	Колесницький О. К.
	Бали:		
1. Технічна здійсненність концепції	4	4	4
2. Ринкові переваги (наявність аналогів)	3	3	4
3. Ринкові переваги (ціна продукту)	4	4	4
4. Ринкові переваги (технічні властивості)	4	4	4
5. Ринкові переваги (експлуатаційні витрати)	4	3	3
6. Ринкові перспективи (розмір ринку)	3	3	3
7. Ринкові перспективи (конкуренція)	3	3	4
8. Практична здійсненність (наявність фахівців)	4	4	4
9. Практична здійсненність (наявність фінансів)	3	3	3
10. Практична здійсненність (необхідність нових матеріалів)	4	4	4
11. Практична здійсненність (термін реалізації)	4	4	4
12. Практична здійсненність (розробка документів)	3	3	3
Сума балів	СБ ₁ =43	СБ ₂ =42	СБ ₃ =44
Середньоарифметична сума балів СБ _с	$(43+42+44)/3 = 43$		

За результатами розрахунків, наведених в таблиці 4.2, зробимо висновок щодо науково-технічного рівня і рівня комерційного потенціалу розробки. При цьому використаємо рекомендації, наведені в таблиці 4.3 [39].

Таблиця 4.3 – Науково-технічні рівні та комерційні потенціали розробки

Середньоарифметична сума балів СБ, розрахована на основі висновків експертів	Науково-технічний рівень та комерційний потенціал розробки
41...48	Високий
31...40	Вище середнього
21...30	Середній
11...20	Нижче середнього
0...10	Низький

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Інформаційна технологія прогнозування курсу криптовалюти «Біткоїн»» становить 43 бали, що, відповідно до таблиці 4.3, свідчить про комерційну важливість проведення даних досліджень оскільки рівень комерційного потенціалу розробки високий.

Дана розробка призначена для користувачів, які цікавляться криптовалютами та мають доступ до мережі інтернет. Для використання технології їм знадобиться перейти на сайт ресурсу та купити продукт за фіксованою ціною.

4.2 Визначення рівня конкурентоспроможності розробки

В процесі визначення економічної ефективності науково-технічної розробки також доцільно провести прогноз рівня її конкурентоспроможності за сукупністю параметрів, що підлягають оцінюванню.

Одиничний параметричний індекс розраховуємо за формулою [40]:

$$q_i = \frac{P_i}{P_{базі}} \quad (4.1)$$

де q_i – одиничний параметричний індекс, розрахований за i -м параметром;

P_i – значення i -го параметра виробу;

$P_{базі}$ – аналогічний параметр базового виробу-аналога, з яким проводиться порівняння.

Загальні технічні та економічні характеристики розробки представлено в таблиці 4.4.

Таблиця 4.4 – Основні техніко-економічні показники аналога та розробки, що проектується

Показник	Варіанти		Відносний показник якості	Коефіцієнт вагомості параметра
	Базовий (товар-конкурент)	Новий (інноваційне рішення)		
Точність	65	>70	1,08	70%
Горизонт прогнозування	1	4	4	30%

Нормативні параметри оцінюємо показником, який отримує одне з двох значень: 1 – пристрій відповідає нормам і стандартам; 0 – не відповідає.

Груповий показник конкурентоспроможності за нормативними параметрами розраховуємо як добуток частинних показників за кожним параметром за формулою [40]:

$$I_{нп} = \prod_{i=1}^n q_i, \quad (4.2)$$

де $I_{нп}$ – загальний показник конкурентоспроможності за нормативними параметрами;

q_i – одиничний (частинний) показник за i -м нормативним параметром;

n – кількість нормативних параметрів, які підлягають оцінюванню.

За нормативними параметрами розроблюваний пристрій відповідає вимогам ДСТУ, тому $I_{nn} = 1$.

Значення групового параметричного індексу за технічними параметрами визначаємо з урахуванням вагомості (частки) кожного параметра [40]:

$$I_{ТП} = \sum_{i=1}^n q_i \cdot \alpha_i, \quad (4.3)$$

де $I_{ТП}$ – груповий параметричний індекс за технічними показниками (порівняно з виробом-аналогом);

q_i – одиничний параметричний показник i -го параметра;

α_i – вагомість i -го параметричного показника, $\sum_{i=1}^n \alpha_i = 1$;

n – кількість технічних параметрів, за якими оцінюється конкурентоспроможність.

Проведемо аналіз параметрів згідно даних таблиці 4.4.

$$I_{mn} = 1,08 \cdot 0,7 + 4 \cdot 0,3 = 1,96.$$

Груповий параметричний індекс за економічними параметрами розраховуємо за формулою [40]:

$$I_{ЕП} = \sum_{i=1}^m q_i \cdot \beta_i, \quad (4.4)$$

де $I_{ЕП}$ – груповий параметричний індекс за економічними показниками;

q_i – економічний параметр i -го виду;

β_i – частка i -го економічного параметра, $\sum_{i=1}^m \beta_i = 1$;

m – кількість економічних параметрів, за якими здійснюється оцінювання.

Проведемо аналіз параметрів згідно даних таблиці.

$$I_{EP} = 0,76 \cdot 0,5 + 0,84 \cdot 0,5 = 0,80.$$

На основі групових параметричних індексів за нормативними, технічними та економічними показниками розрахуємо інтегральний показник конкурентоспроможності за формулою [40]:

$$K_{INT} = I_{HP} \cdot \frac{I_{TP}}{I_{EP}}, \quad (4.5)$$

$$K_{INT} = 1 \cdot 1,96 / 0,80 = 2,45.$$

Інтегральний показник конкурентоспроможності $K_{INT} > 1$, отже розробка переважає відомі аналоги за своїми техніко-економічними показниками.

4.3 Розрахунок витрат на проведення науково-дослідної роботи

Витрати, пов'язані з проведенням науково-дослідної роботи на тему «Інформаційна технологія прогнозування курсу криптовалюти «Біткоїн»», під час планування, обліку і калькулювання собівартості науково-дослідної роботи групуємо за відповідними статтями.

4.3.1 Витрати на оплату праці

До статті «Витрати на оплату праці» належать витрати на виплату основної та додаткової заробітної плати керівникам відділів, лабораторій, секторів і груп, науковим, інженерно-технічним працівникам, конструкторам, технологам, креслярам, копіювальникам, лаборантам, робітникам, студентам, аспірантам та іншим працівникам, безпосередньо зайнятим виконанням конкретної теми, обчисленої за посадовими окладами, відрядними розцінками, тарифними ставками згідно з чинними в організаціях системами оплати праці.

Основна заробітна плата дослідників. Витрати на основну заробітну плату дослідників (Z_o) розраховуємо у відповідності до посадових окладів працівників, за формулою [40]:

$$Z_o = \sum_{i=1}^k \frac{M_{ni} \cdot t_i}{T_p}, \quad (4.6)$$

де k – кількість посад дослідників залучених до процесу досліджень;

M_{ni} – місячний посадовий оклад конкретного дослідника, грн;

t_i – число днів роботи конкретного дослідника, дн.;

T_p – середнє число робочих днів в місяці, $T_p=21$ дні.

$$Z_o = 42000 \cdot 10 / 21 = 19091 \text{ (грн.)}$$

Проведені розрахунки зведемо до таблиці 4.5.

Таблиця 4.5 – Витрати на заробітну плату дослідників

Найменування посади	Місячний посадовий оклад, грн	Оплата за робочий день, грн	Число днів роботи	Витрати на заробітну плату, грн
Керівник проекту	42000	1909,1	10	19091
Інженер-програміст	39000	1772,7	55	97500
Всього				116591

Основна заробітна плата робітників. Витрати на основну заробітну плату робітників (Z_p) за відповідними найменуваннями робіт НДР на тему «Інформаційна технологія прогнозування курсу криптовалюти «Біткоїн»» розраховуємо за формулою:

$$Z_p = \sum_{i=1}^n C_i \cdot t_i, \quad (4.7)$$

де C_i – погодинна тарифна ставка робітника відповідного розряду, за виконану відповідну роботу, грн/год;

t_i – час роботи робітника при виконанні визначеної роботи, год.

Погодинну тарифну ставку робітника відповідного розряду C_i можна визначити за формулою:

$$C_i = \frac{M_M \cdot K_i \cdot K_c}{T_p \cdot t_{зм}}, \quad (4.8)$$

де M_M – розмір прожиткового мінімуму працездатної особи, або мінімальної місячної заробітної плати (в залежності від діючого законодавства), прийmemo $M_M=6500$ грн;

K_i – коефіцієнт міжкваліфікаційного співвідношення для встановлення тарифної ставки робітнику відповідного розряду [40]:

K_c – мінімальний коефіцієнт співвідношень місячних тарифних ставок робітників першого розряду з нормальними умовами праці виробничих об'єднань і підприємств до законодавчо встановленого розміру мінімальної заробітної плати.

T_p – середнє число робочих днів в місяці, приблизно $T_p = 21$ дн;

$t_{зм}$ – тривалість зміни, год.

$$C_1 = 6500,00 \cdot 1 \cdot 1,65 / (21 \cdot 8) = 65,8 \text{ (грн.)}$$

$$З_{р1} = 65,8 \cdot 2 = 131,6 \text{ (грн.)}$$

Проведені розрахунки зведемо до таблиці 4.6.

Таблиця 4.6 – Величина витрат на основну заробітну плату робітників

Найменування робіт	Тривалість роботи, год	Розряд роботи	Погодинна тарифна ставка, грн	Величина оплати на робітника грн
1.Підготовчі	2	1	65,8	131,6
2.Налагоджувальні	10	2	72,4	723,8
3.Випробувальні	2	4	98,7	197,4
Всього				1052,9

Додаткова заробітна плата дослідників та робітників. Додаткову заробітну плату розраховуємо як 10 ... 12% від суми основної заробітної плати дослідників та робітників за формулою:

$$Z_{\text{дод}} = (Z_o + Z_p) \cdot \frac{H_{\text{дод}}}{100\%}, \quad (4.9)$$

де $H_{\text{дод}}$ – норма нарахування додаткової заробітної плати. Прийmemo 11%.

$$Z_{\text{дод}} = (116591 + 1052,9) \cdot 11 / 100\% = 12940,81 \text{ (грн.)}$$

4.3.2 Відрахування на соціальні заходи

Нарахування на заробітну плату дослідників та робітників розраховуємо як 22% від суми основної та додаткової заробітної плати дослідників і робітників за формулою:

$$Z_n = (Z_o + Z_p + Z_{\text{дод}}) \cdot \frac{H_{\text{зн}}}{100\%}, \quad (4.10)$$

де $H_{\text{зн}}$ – норма нарахування на заробітну плату. Приймаємо 22%.

$$Z_n = (116591 + 1052,9 + 12940,81) \cdot 22 / 100\% = 28728,61 \text{ (грн.)}$$

4.3.3 Сировина та матеріали

До статті «Сировина та матеріали» належать витрати на сировину, основні та допоміжні матеріали, інструменти, пристрої та інші засоби і предмети праці, які придбані у сторонніх підприємств, установ і організацій та витрачені на проведення досліджень за темою «Інформаційна технологія прогнозування курсу криптовалюти «Біткоїн»».

Витрати на матеріали (M), у вартісному вираженні розраховуються окремо по кожному виду матеріалів за формулою:

$$M = \sum_{j=1}^n H_j \cdot C_j \cdot K_j - \sum_{j=1}^n B_j \cdot C_{ej}, \quad (4.11)$$

де H_j – норма витрат матеріалу j -го найменування, кг;

n – кількість видів матеріалів;

C_j – вартість матеріалу j -го найменування, грн/кг;

K_j – коефіцієнт транспортних витрат, ($K_j = 1,1 \dots 1,15$);

B_j – маса відходів j -го найменування, кг;

C_{ej} – вартість відходів j -го найменування, грн/кг.

Проведені розрахунки зведемо до таблиці 4.7.

Таблиця 4.7 – Витрати на матеріали

Найменування матеріалу, марка, тип, сорт	Ціна за 1 кг, грн	Норма витрат, кг	Вартість витраченого матеріалу, грн
Папір А 4	146	1	146
Ручка	14	1	14
Диск оптичний OPTIMA CD	15	1	15
Flesh-пам'ять GOODRAM 64 С10А	410	1	410
Всього			585
З врахуванням коефіцієнта транспортування			643,5

4.3.4 Спецустаткування для наукових (експериментальних) робіт

До статті «Спецустаткування для наукових (експериментальних) робіт» належать витрати на виготовлення та придбання спецустаткування необхідного для проведення досліджень, також витрати на їх проектування, виготовлення, транспортування, монтаж та встановлення.

Балансову вартість спецустаткування розраховуємо за формулою:

$$B_{\text{спец}} = \sum_{i=1}^k C_i \cdot C_{\text{нр.і}} \cdot K_i, \quad (4.12)$$

де C_i – ціна придбання одиниці спецустаткування даного виду, марки, грн;

$C_{пр.і}$ – кількість одиниць устаткування відповідного найменування, які придбані для проведення досліджень, шт.;

K_i – коефіцієнт, що враховує доставку, монтаж, налагодження устаткування тощо, ($K_i = 1,10 \dots 1,12$);

k – кількість найменувань устаткування.

$$V_{спец} = 40000,00 \cdot 1 \cdot 1,11 = 44000 \text{ (грн.)}.$$

Отримані результати зведемо до таблиці 4.8:

Таблиця 4.8 – Витрати на придбання спецустаткування по кожному виду

Найменування устаткування	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
1. Комп'ютер	1	40000	44000
2. Клавіатура A4Tech KV-300	1	1270	1397
3. Миша HyperX Pulsfire core	1	1399	1538,9
4. Крісло Примтекс Плюс	1	3076	3383,6
5. Стіл RZTK eDesk	1	9000	9900
Всього			60219,5

4.3.5 Програмне забезпечення для наукових (експериментальних) робіт

До статті «Програмне забезпечення для наукових (експериментальних) робіт» належать витрати на розробку та придбання спеціальних програмних засобів і програмного забезпечення, (програм, алгоритмів, баз даних) необхідних для проведення досліджень, також витрати на їх проектування, формування та встановлення.

Балансову вартість програмного забезпечення розраховуємо за формулою:

$$B_{npz} = \sum_{i=1}^k C_{inprz} \cdot C_{npz.i} \cdot K_i, \quad (4.13)$$

де C_{inprz} – ціна придбання одиниці програмного засобу даного виду, грн;

$C_{npz.i}$ – кількість одиниць програмного забезпечення відповідного найменування, які придбані для проведення досліджень, шт.;

K_i – коефіцієнт, що враховує інсталяцію, налагодження програмного засобу тощо, ($K_i = 1, 10 \dots 1, 12$);

k – кількість найменувань програмних засобів.

$$B_{npz} = 600 \cdot 1 \cdot 1,11 = 660 \text{ (грн.)}$$

Отримані результати зведемо до таблиці 4.9:

Таблиця 4.9 – Витрати на придбання програмних засобів по кожному виду

Найменування програмного засобу	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
Підписка на Firebase	1	600	660
Оренда серверу	1	720	792
Всього			1452

4.3.6 Амортизація обладнання, програмних засобів та приміщень

В спрощеному вигляді амортизаційні відрахування по кожному виду обладнання, приміщень та програмному забезпеченню тощо, розраховуємо з використанням прямолінійного методу амортизації за формулою:

$$A_{обл} = \frac{Ц_{б}}{T_{г}} \cdot \frac{t_{вик}}{12}, \quad (4.14)$$

де $Ц_{б}$ – балансова вартість обладнання, програмних засобів, приміщень тощо, які використовувались для проведення досліджень, грн;

$t_{\text{вик}}$ – термін використання обладнання, програмних засобів, приміщень під час досліджень, місяців;

T_e – строк корисного використання обладнання, програмних засобів, приміщень тощо, років.

$$A_{\text{обл}} = (45000 \cdot 1) / (2 \cdot 12) = 1875 \text{ (грн.)}.$$

Проведені розрахунки зведемо до таблиці 4.10.

Таблиця 4.10 – Амортизаційні відрахування по кожному виду обладнання

Найменування обладнання	Балансова вартість, грн	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн
Комп'ютер	45000	2	1	1875,00
Приміщення лабораторії	190000	20	1	791,67
Всього				2666,67

4.3.7 Паливо та енергія для науково-виробничих цілей

Витрати на силову електроенергію (B_e) розраховуємо за формулою:

$$B_e = \sum_{i=1}^n \frac{W_{yi} \cdot t_i \cdot C_e \cdot K_{\text{ени}}}{\eta_i}, \quad (4.15)$$

де W_{yi} – встановлена потужність обладнання на визначеному етапі розробки, кВт;

t_i – тривалість роботи обладнання на етапі дослідження, год;

C_e – вартість 1 кВт-години електроенергії, грн; (вартість електроенергії визначається за даними енергопостачальної компанії), прийmemo $C_e = 7,5$ грн;

K_{eni} – коефіцієнт, що враховує використання потужності, $K_{eni} < 1$;

η_i – коефіцієнт корисної дії обладнання, $\eta_i < 1$.

$$B_e = 0,25 \cdot 250,0 \cdot 7,5 \cdot 0,5 / 0,8 = 292,97 \text{ (грн.)}$$

4.3.8 Службові відрядження

До статті «Службові відрядження» дослідної роботи на тему «Інформаційна технологія прогнозування курсу криптовалюти «Біткоїн»» належать витрати на відрядження штатних працівників, працівників організацій, які працюють за договорами цивільно-правового характеру, аспірантів, зайнятих розробленням досліджень, відрядження, пов'язані з проведенням випробувань машин та приладів, а також витрати на відрядження на наукові з'їзди, конференції, наради, пов'язані з виконанням конкретних досліджень.

Витрати за статтею «Службові відрядження» розраховуємо як 20...25% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{cv} = (Z_o + Z_p) \cdot \frac{H_{cv}}{100\%}, \quad (4.16)$$

де H_{cv} – норма нарахування за статтею «Службові відрядження», прийmemo $H_{cv} = 20\%$.

$$B_{cv} = (116591 + 1052,9) \cdot 20 / 100\% = 23528,75 \text{ (грн.)}$$

4.3.9 Інші витрати

До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками.

Витрати за статтею «Інші витрати» розраховуємо як 50...100% від суми основної заробітної плати дослідників та робітників за формулою:

$$I_e = (Z_o + Z_p) \cdot \frac{H_{ie}}{100\%}, \quad (4.17)$$

де H_{ie} – норма нарахування за статтею «Інші витрати», прийmemo $H_{ie} = 50\%$.

$$I_b = (116591 + 1052,9) \cdot 50 / 100\% = 58821,88 \text{ (грн.)}$$

4.3.10 Накладні (загальновиробничі) витрати

До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін.

Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуємо як 100...150% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{H3B} = (Z_o + Z_p) \cdot \frac{H_{H3B}}{100\%}, \quad (4.18)$$

де H_{H3B} – норма нарахування за статтею «Накладні (загальновиробничі) витрати», прийmemo $H_{H3B} = 100\%$.

$$B_{H3B} = (116591 + 1052,9) \cdot 100 / 100\% = 117643,77 \text{ (грн.)}$$

Витрати на проведення науково-дослідної роботи розраховуємо як суму всіх попередніх статей витрат за формулою:

$$B_{\text{заг}} = Z_o + Z_p + Z_{\text{дод}} + Z_n + M + K_e + B_{\text{спец}} + B_{\text{прз}} + A_{\text{обл}} + B_e + B_{\text{св}} + B_{\text{сп}} + I_e + B_{\text{нзв}}. \quad (4.19)$$

$$B_{\text{заг}} = 116591 + 1052,9 + 12940,81 + 28728,61 + 643,5 + 60219,5 + 1452 + 2666,67 + 292,97 + 23528,75 + 58821,88 + 117643,77 = 424675,98 \text{ (грн.)}$$

Загальні витрати ZB на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховується за формулою:

$$ZB = \frac{B_{\text{заг}}}{\eta}, \quad (4.20)$$

де η - коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи, прийmemo $\eta = 0,9$.

$$ZB = 232106,92 / 0,9 = 424675,98 \text{ (грн.)}$$

4.4 Розрахунок економічної ефективності науково-технічної розробки при її можливій комерціалізації потенційним інвестором

В ринкових умовах узагальнюючим позитивним результатом, що може отримати потенційний інвестор від можливого впровадження результатів тієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку.

Результати дослідження проведені за темою «Інформаційна технологія прогнозування курсу криптовалюти «Біткоїн»» передбачають комерціалізацію протягом 3-х років реалізації на ринку.

В цьому випадку основу майбутнього економічного ефекту будуть формувати:

ΔN – збільшення кількості споживачів яким надається відповідна інформаційна послуга у періоди часу, що аналізуються;

N – кількість споживачів яким надавалась відповідна інформаційна послуга у році до впровадження результатів нової науково-технічної розробки, прийmemo 1 особа

C_o – вартість послуги у році до впровадження інформаційної системи, прийmemo 2000,00 грн;

$\pm \Delta C_o$ – зміна вартості послуги від впровадження результатів, прийmemo зростання на 500,00 грн.

Можливе збільшення чистого прибутку у потенційного інвестора $\Delta \Pi_i$ для кожного із 3-х років, протягом яких очікується отримання позитивних результатів від можливого впровадження та комерціалізації науково-технічної розробки, розраховуємо за формулою [40]:

$$\Delta \Pi_i = (\pm \Delta C_o \cdot N + C_o \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\vartheta}{100}\right), \quad (4.21)$$

де λ – коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість. У 2023 році ставка податку на додану вартість складає 20%, а коефіцієнт $\lambda = 0,8333$;

ρ – коефіцієнт, який враховує рентабельність інноваційного продукту).
Прийmemo $\rho = 40\%$;

ϑ – ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2023 році $\vartheta = 18\%$;

Збільшення чистого прибутку 1-го року:

$$\Delta \Pi_1 = (1 \cdot 500 + 2000 \cdot 1500) \cdot 0,83 \cdot 0,4 \cdot (1 - 0,18/100\%) = 640684,79 \text{ грн.}$$

Збільшення чистого прибутку 2-го року:

$$\Delta\Pi_2 = (1 \cdot 500 + 2000 \cdot (1500 + 1200)) \cdot 0,83 \cdot 0,4 \cdot (1 - 0,18/100\%) = 1153578,9 \text{ грн.}$$

Збільшення чистого прибутку 3-го року:

$$\Delta\Pi_3 = (1 \cdot 500 + 2000 \cdot (1500 + 1200 + 850)) \cdot 0,83 \cdot 0,4 \cdot (1 - 0,18/100\%) = 1516585,2 \text{ грн.}$$

Приведена вартість збільшення всіх чистих прибутків $ПП$, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$ПП = \sum_{i=1}^T \frac{\Delta\Pi_i}{(1 + \tau)^t}, \quad (4.22)$$

де $\Delta\Pi_i$ – збільшення чистого прибутку у кожному з років, протягом яких виявляються результати впровадження науково-технічної розробки, грн;

T – період часу, протягом якого очікується отримання позитивних результатів від впровадження та комерціалізації науково-технічної розробки, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, $\tau = 18\%$;

t – період часу (в роках) від моменту початку впровадження науково-технічної розробки до моменту отримання потенційним інвестором додаткових чистих прибутків у цьому році.

$$\begin{aligned} ПП &= 640684,79 / (1 + 0,18)^1 + 1153578,9 / (1 + 0,18)^2 + 1516585,2 / (1 + 0,18)^3 = \\ &= 2216736,01 \text{ (грн.)}. \end{aligned}$$

Величина початкових інвестицій PV , які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки:

$$PV = k_{інв} \cdot 3B, \quad (4.23)$$

де $k_{инв}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію, приймаємо $k_{инв} = 2$;

$3B$ – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, приймаємо 424675,98 (грн.).

$$PV = k_{инв} \cdot 3B = 2 \cdot 424675,98 = 943724,39 \text{ (грн.)}$$

Абсолютний економічний ефект $E_{абс}$ для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{абс} = ПП - PV \quad (4.24)$$

де $ПП$ – приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки, 2216736,01 (грн.);

PV – теперішня вартість початкових інвестицій, 943724,39 (грн.).

$$E_{абс} = ПП - PV = 2216736,01 - 943724,39 = 1273011,61 \text{ (грн.)}$$

Внутрішня економічна дохідність інвестицій E_e , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$E_e = T_{ж} \sqrt{1 + \frac{E_{абс}}{PV}} - 1, \quad (4.25)$$

де $E_{абс}$ – абсолютний економічний ефект вкладених інвестицій, грн;

PV – теперішня вартість початкових інвестицій, грн;

$T_{ж}$ – життєвий цикл науково-технічної розробки, тобто час від початку її розробки до закінчення отримання позитивних результатів від її впровадження, 3 роки.

$$E_g = \sqrt[3]{1 + \frac{E_{abc}}{PV}} - 1 = (1 + 1273011,61 / 943724,39)^{1/3} - 1 = 0,55.$$

Мінімальна внутрішня економічна дохідність вкладених інвестицій τ_{min} :

$$\tau_{min} = d + f, \quad (4.26)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2023 році в Україні $d = 0,1$;

f – показник, що характеризує ризикованість вкладення інвестицій, приймемо 0,25.

$$\tau_{min} = 0,1 + 0,25 = 0,35 < 0,55,$$

свідчить про те, що внутрішня економічна дохідність інвестицій E_g , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки вища мінімальної внутрішньої дохідності. Тобто інвестувати в науково-дослідну роботу за темою «Інформаційна технологія онтологічного моделювання бази знань з організації бібліотеки» доцільно.

Період окупності інвестицій $T_{ок}$ які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$T_{ок} = \frac{1}{E_g}, \quad (4.27)$$

де E_g – внутрішня економічна дохідність вкладених інвестицій.

$$T_{ок} = 1 / 0,55 = 1,8 \text{ р.}$$

$T_{ок} < 3$ -х років, що може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

4.5 Висновок до розділу 4

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Інформаційна технологія прогнозування курсу криптовалюти «Біткоїн»» становить 43 бали, що, свідчить про комерційну важливість проведення цих досліджень, оскільки рівень комерційного потенціалу розробки високий.

При оцінюванні рівня конкурентоспроможності, згідно узагальненого коефіцієнту конкурентоспроможності розробки, науково-технічна розробка переважає існуючі аналоги приблизно в 2,45 рази.

Також термін окупності становить 1,8 роки, що менше 3-х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок. Отже можна зробити висновок про доцільність проведення науково-дослідної роботи за темою «Інформаційна технологія прогнозування курсу криптовалюти «Біткоїн»».

ВИСНОВКИ

Всі завдання, поставлені для реалізації інформаційної технології прогнозування курсу криптовалюти «Біткоїн» виконані в повному обсязі, а саме: проаналізовано технічний рівень існуючих систем прогнозування та обґрунтовано доцільність розробки інформаційної технології прогнозування курсу криптовалюти «Біткоїн»; розроблено математичну модель інформаційної технології прогнозування курсу криптовалюти «Біткоїн»; розглянуто методи та технології прогнозування курсу криптовалюти «Біткоїн»; спроектовано архітектуру та структуру програмного модуля прогнозування курсу криптовалюти «Біткоїн»; проаналізовано та обґрунтовано вибір інструментів для розробки; реалізовано програмний модуль інформаційної технології прогнозування курсу криптовалюти «Біткоїн»; виконано тестування програми та проведено аналіз отриманих результатів; обґрунтовано економічну доцільність розробки.

При виконанні магістерської кваліфікаційної роботи реалізовано інформаційну технологію прогнозування курсу криптовалюти «Біткоїн».

Проведено аналіз технічних рішень програм для прогнозування курсу криптовалют та доведено актуальність розробки інформаційної технології прогнозування курсу криптовалюти «Біткоїн». Розглянуті системи-аналоги є корисними та доволі актуальними, проте усі вони представлені у вигляді комерційного сервісу і не розкривають повних деталей реалізації або не мають достатнього функціоналу, обмежуючи користувача певними рамками прогнозування.

На основі аналізу існуючих моделей та методів, які застосовуються для вирішення поставленої задачі було обґрунтовано використання в інформаційній технології математичної моделі інтегрованого ковзного середнього (ARIMA). Модель відповідає за генерування прогнозів на основі переданої користувачем інформації.

Було розроблено структуру інформаційної технології прогнозування курсу криптовалюти «Біткоїн» та спроектовано схему алгоритму роботи системи. Обґрунтовано вибір архітектури інформаційної технології прогнозування курсу криптовалюти «Біткоїн». Здійснено проектування та розробку інформаційної технології прогнозування курсу криптовалюти «Біткоїн».

Обґрунтовано вибір мови програмування та інструментів для розробки. Реалізацію інформаційної технології проведено з використанням мови програмування Python для серверної частини та JavaScript для клієнтської частини.

Здійснено програмну реалізацію інформаційної технології прогнозування курсу криптовалюти «Біткоїн». Проведено тестування розробленої програми та підтверджено її працездатність. Порівняно з системами-аналогами розроблена інформаційна технологія забезпечує більшу точність прогнозування, а саме 73%, у той час коли BellInvestor має результат 68%, LongForecast – 63% та 55% – WalletInvestor, що означає доведення поставленої мети.

Економічно обґрунтовано доцільність розробки інформаційної технології прогнозування курсу криптовалюти «Біткоїн». Було спрогнозовано орієнтовану величину витрат по кожній з статей витрат, результатом стала сума 424 675 гривень. Шляхом аналізу розрахунків було доведено що розроблена інформаційна технологія прогнозування курсу криптовалюти «Біткоїн» переважає існуючі аналоги приблизно в 2,45 рази. Період окупності інформаційної технології прогнозування курсу криптовалюти «Біткоїн» складе близько 1,8 років.

Поставлена мета, а саме розширення функціональних можливостей інформаційної технології прогнозування курсу криптовалюти «Біткоїн» була досягнута за рахунок використання нових часових проміжків горизонту прогнозування. Це дозволило розробити зручний клієнтський додаток для інформаційної технології прогнозування курсу криптовалюти «Біткоїн», який відповідає всім запитам користувача, інтуїтивно зрозумілий та конкурує з іншими системами-аналогами на ринку.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Герус Я. К., Крилик Л. В. Аналіз передумов створення програмного модуля прогнозування курсу криптовалюти «Біткоїн». *Матеріали молодіжної науково-практичної інтернет-конференції студентів аспірантів та молодих науковців «Молодь в науці: дослідження, проблеми, перспективи (МН-2021)» : збірник матеріалів*. [Електронний ресурс]. Вінниця: ВНТУ, 2021. С. 485 – 486. URL: https://conferences.vntu.edu.ua/public/files/mn/mn-2021_netpub.pdf. (дата звернення 04.10.23).
2. Герус Я. К., Крилик Л. В. Перспективи розробки інформаційної технології для прогнозування курсу криптовалюти «БІТКОЇН». *III Науково-технічна конференція факультету інтелектуальних інформаційних технологій та автоматизації*. URL: <https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2023/paper/view/17175/14407> (дата звернення 04.10.23).
3. Галушка Є.О., Пакон О.Д «Сутність крипто валют та перспективи їх розвитку», «Young Scientist». № 4 (44). April, 2017, С.634–638.
4. Jay Clayton Statement on Cryptocurrencies and Initial Coin Offerings [Електронний ресурс]. URL: https://www.sec.gov/news/public-statement/statement-clayton-2017-12-11#_ftn2 (дата звернення: 06.10.2023).
5. Casey M. J. The Age of Cryptocurrency: How Bitcoin and the Blockchain Are Challenging the Global Economic Order, London: St. Martin's Press, 2015. 368 p.
6. Nathaniel Popper. Digital Gold: The Untold Story of Bitcoin, Allen Lane, 2015, 398p.
7. Walletinvestor. Cryptocurrency Forecast (Bitcoin & Altcoin, ICO Prediction, Prognosis 2023) [Електронний ресурс]. URL: <https://walletinvestor.com/forecast> (дата звернення 27.09.2023).
8. Belinvestor. Рубрика: Криптовалюти [Електронний ресурс]. URL: <https://belinvestor.com/cryptocurrencies/> (дата звернення 27.09.2023).
9. LongForecast. The Economy Forecast Agency [Електронний ресурс]. URL: <https://longforecast.com/> (дата звернення 27.09.2023).

10. Huckle S., Bhattacharya R., White M., Beloff N. Internet of things, blockchain and shared economy applications. *Procedia Computer Science*, 2016. V. 98. P. 461–466.

11. Молчанова Е., Солодковський Ю. Глобальна сервісна природа сучасних криптовалют. *Міжнародна економічна політика*. 2014. № 1. С. 60 – 79.

12. Haykin S. *Neural networks : a comprehensive foundation*. Upper Saddle River, New Jersey 07458 : Prentice Hall, Inc, 1999. 1104 p.

13. Андрієнко В. М., Семенов А. С. Методика статистичного аналізу економічних часових рядів. *Науковий вісник Ужгородського національного університету. Серія : Міжнародні економічні відносини та світове господарство*. 2018. Вип. 21(1). С. 5 – 13.

14. Smith S. How Cryptocurrencies Are Changing What CPAs Need to Know about Fraud Prevention. *Theoretical Economics Letters*/ October 2018 Vol.8 No.14.

15. Гур'янова Л.С., Клебанова Т.С., Сергієнко О.А., Прокопович С.В. Економетрика: навчальний посібник для студентів напряму підготовки «Економічна кібернетика» всіх форм навчання. Харків: ХНЕУ ім. С. Кузнеця, 2015. 384 с.

16. Березька, К. М. Методологічні аспекти застосування моделі нечітких часових рядів для прогнозування податкових надходжень. *Актуальні проблеми економіки*. 2011. № 1. С. 227–235.

17. Осипчук М.М. Аналіз часових рядів. Навчальний посібник Івано-Франківськ. Прикарпатський національний університет імені Василя Стефаника. 2009. 84 с.

18. Балан В. Г. Стохастичне моделювання процесу розробки й впровадження системи збалансованих показників у банківських установах. *Вісник КНУ ім. Т.Г. Шевченка. Серія: «Економіка»*. 2009. С. 37–42.

19. Інструкція як будувати UML діаграми. [Електронний ресурс]. URL: <https://dou.ua/forums/topic/40575/> (дата звернення 08.11.2023).

20. Грамотна клієнт-серверна архітектура: як правильно проектувати і розробляти web API [Електронний ресурс]. URL: <https://echo.lviv.ua/dev/6455> (дата звернення 27.09.2023).
21. PHP Загальна документація [Електронний ресурс]. URL: <https://www.php.net/docs.php/> (дата звернення 30.09.2023).
22. Python Загальна документація [Електронний ресурс]. URL: <https://wiki.python.org/moin/BeginnersGuide> (дата звернення 27.09.2023).
23. Яковенко А. В. *Основи програмування. Python. Частина 1*: навч. посіб. Київ КПІ ім. І. Сікорського, 2018. 195 с.
24. Joshua Bloch. *Effective Java: Programming Language Guide*. Publisher: Addison Wesley. 2001. 272 p.
25. Jon Skeet. *C# in depth. Fourth Edition*. Manning Publications Co. 2019. 528 p.
26. Javascript Загальна документація [Електронний ресурс]. URL: <https://www.ecmascriptinternational.org/publicationsandstandards/standards/ecma262> (дата звернення 30.09.2023)
27. Що таке JavaScript, та як функціонує JavaScript [Електронний ресурс]. URL: <http://ruszura.in.ua/uncategorized/scho-take-javascript-yakfunktsionuyu-javascript.html> (дата звернення 30.09.2023).
28. Angular. Загальна документація [Електронний ресурс]. URL: <https://angular.io/guide/what-is-angular> (дата звернення 06.11.2023).
29. Босько В. В., Константинова Л. В. Аналіз та дослідження фреймворку Angular як засобу розробки вебсайтів. *Центральноукраїнський науковий вісник. Технічні науки*. 2022. Вип. 5(36) с. 3.
30. Порівнюємо React, Angular і Vue – найпопулярніші бібліотеки й фреймворки у 2022 році [Електронний ресурс]. URL: <https://dou.ua/forums/topic/39933>. (дата звернення 06.11.2023).
31. Vue.js вступ. [Електронний ресурс]. URL: <https://vuejs.org/guide/introduction.html> (дата звернення 06.11.2023).

32. Подоба В., Літвінов Д. Веб-розробка з Python та Django для початківців. 2020. 65 с.
33. Введення в Django [Електронний ресурс]. URL: <https://developer.mozilla.org/ru/docs/Learn/Server-side/Django/Introduction> (дата звернення 08.11.2023).
34. Плюси і мінуси Django [Електронний ресурс]. URL: <https://blog.ukrnames.com/veb-master/plyusi-i-minusi-django> (дата звернення 08.11.2023).
35. Фреймворки в умовах програмування [Електронний ресурс]. URL: https://cloud.itstep.org/blog_3/frameworks-in-programming-languages-what-are-they-for-and-how-to-choose-them#13 (дата звернення 08.11.2023).
36. Firebase RealTime Database [Електронний ресурс]. URL: <https://firebase.google.com/docs/database> (дата звернення 08.11.2023).
37. Firebase Authentication. [Електронний ресурс]. URL: <https://firebase.google.com/docs/auth> (дата звернення 08.11.2023).
38. StatsModels Загальна документація [Електронний ресурс]. URL: <https://www.statsmodels.org/stable/gettingstarted.html#> (дата звернення 08.11.2023).
39. Технологічний аудит [Електронний ресурс]. URL: <https://studfile.net/preview/7336428/> (дата звернення 27.11.2023).
40. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт / Уклад.: В. О. Козловський, О. Й. Лесько, В. В. Кавецький. Вінниця : ВНТУ, 2021. 42 с.

Додаток А (обов'язковий)
Протокол перевірки кваліфікаційної роботи на наявність текстових запозичень

**ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ
 НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ**

Назва роботи: Інформаційна технологія прогнозування курсу криптовалюти «Біткоїн»

Тип роботи: магістерська кваліфікаційна робота
 (БДР, МКР)

Підрозділ кафедра комп'ютерних наук, ФІТА

(кафедра, факультет)

Показники звіту подібності Unicheck

Оригінальність 86,3% Схожість 13,7%

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 виконання автором. 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.

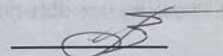
Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck щодо роботи.

Автор роботи



Герус Я. К.

Керівник роботи

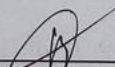


Крилик Л. В.

Опис прийнятого рішення

Магістерську кваліфікаційну роботу допущено до захисту

Особа, відповідальна за перевірку



Озеранський В. С.

Додаток Б (обов'язковий)

Лістинг програми

```

from flask import Flask, request, jsonify
from flask_cors import CORS
import numpy as np
import pandas as pd
import statsmodels.api as sm
from statsmodels.tsa.arima.model import ARIMA
from utils.stationarity_check import stationarity_check
from utils.get_prices import get_bitcoin_prices
from utils.seasonal_check import seasonal_check
from utils.pvalue_pacf import pvalue_pacf
from utils.qvalue_acf import qvalue_acf
import matplotlib.pyplot as plt
from arima.pacf import acf_pacf_draw
import pmdarima as pm
from statsmodels.tsa.stattools import acf
from sklearn.metrics import mean_squared_error, mean_absolute_percentage_error
import firebase_admin
import pyrebase
import datetime
from firebase_admin import credentials
from firebase_admin import db

app = Flask(__name__)
CORS(app)
# Отримання даних

cred = credentials.Certificate("service_account_key.json")
firebase_admin.initialize_app(
    cred,
    {
        "databaseURL": "https://forecast-b141f-default-rtdb.europe-west1.firebaseio.com/"
    },
)

firebaseConfig = {
    "apiKey": "AIzaSyBK2fO2qdfNp2L23s0910vTcuzG2VU1XjA",
    "authDomain": "forecast-b141f.firebaseio.com",
    "projectId": "forecast-b141f",
    "storageBucket": "forecast-b141f.appspot.com",
    "messagingSenderId": "940009115294",
    "appId": "1:940009115294:web:8c61408b5cd2384742358f",
}

firebase = pyrebase.initialize_app(firebaseConfig)

```

```

auth = firebase.auth()
ref = db.reference("forecasts")

@app.route("/forecast", methods=["POST"])
def forecast():
    try:
        data = request.get_json()
        start_time = int(data["start_time"])
        end_time = int(data["end_time"])
        interval = str(data["interval"])
        user_id = str(data["user_id"])
        bitcoin_data = get_bitcoin_prices(start_time, end_time, interval)
        rows = int(len(bitcoin_data) * 0.9)
        training_df = list(bitcoin_data[0:rows][["close"]])
        testing_df = list(bitcoin_data[rows:][["close"]])
        d_value = stationarity_check(training_df)
        diff = training_df
        for i in range(d_value):
            diff = np.diff(diff)

        p_value = pvalue_pacf(diff)
        q_value = qvalue_acf(diff)
        seasonal = seasonal_check(bitcoin_data, p_value, d_value, q_value)
        model_predictions = []
        target = len(testing_df)
        for i in range(target):
            model = ARIMA(
                training_df,
                order=(p_value, d_value, q_value),
                trend=None,
                seasonal_order=seasonal,
            )
            model_fit = model.fit()
            output = model_fit.forecast()
            result = list(output)[0]
            model_predictions.append(result)
            actual_test_value = testing_df[i]
            training_df.append(actual_test_value)

    def forecast_direction(forecast, testing_df):
        last_testing_price = testing_df[-1]
        forecast_price = forecast[0]

        return bool(forecast_price > last_testing_price)

    forecast = list(model_fit.forecast())
    automape = mean_absolute_percentage_error(testing_df, model_predictions)
    forecast_data = {

```

```

    "date": datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S"),
    "forecast": forecast,
    "mape": automape,
    "direction": forecast_direction(forecast, testing_df),
    "interval": interval,
  }
  ref.child(user_id).push(forecast_data)
  # ref.update(forecast_data)

```

```

response = {
  "date": datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S"),
  "forecast": forecast,
  "mape": automape,
  "direction": forecast_direction(forecast, testing_df),
  "interval": interval,
}
return jsonify(response)

```

```

except Exception as e:
    return jsonify({"error": str(e)})

```

```
@app.route("/login", methods=["POST"])
```

```
def login():
```

```

    try:
        data = request.get_json()
        email = str(data["email"])
        password = str(data["password"])
        # Авторизація користувача за допомогою Firebase Authentication
        user = auth.sign_in_with_email_and_password(email, password)
        print(user)
        # Отримайте токен доступу користувача
        user_id = user["localId"]
        id_token = user["idToken"]

```

```

    response = {
        "message": "User authenticated successfully.",
        "user_id": user_id,
        "id_token": id_token,
    }
    return jsonify(response)

```

```

except Exception as e:
    return jsonify({"error": str(e)})

```

```
@app.route("/user_forecasts", methods=["POST"])
```

```
def get_forecasts():
```

```

    try:
        data = request.get_json()

```

```

user_id = str(data["user_id"])
forecasts_ref = ref.child(user_id)
forecasts = forecasts_ref.get()

if forecasts is None:
    forecasts = []

response = {
    "message": "User forecasts retrieved successfully.",
    "forecasts": forecasts,
}
return jsonify(response)

except Exception as e:
    return jsonify({"error": str(e)})

if __name__ == "__main__":
    app.run(debug=True)

import requests
import pandas as pd

def get_bitcoin_prices(start_time, end_time, interval):
    url = "https://data-api.binance.vision/api/v3/klines"
    params = {
        "symbol": "BTCUSDT",
        "interval": interval,
        "startTime": start_time,
        "endTime": end_time,
    }

    response = requests.get(url, params=params)
    data = response.json()
    bitcoin_data_float = {"date": [], "close": []}
    for row in data:
        bitcoin_data_float["date"].append(float(row[0]))
        bitcoin_data_float["close"].append(float(row[4]))
    # print(bitcoin_data_float["close"])
    df = pd.DataFrame(bitcoin_data_float)
    df["date"] = pd.to_datetime(df["date"], unit="ms", origin="unix")
    df.set_index(["date"], inplace=True)
    return df

__all__ = ["get_prices"]

from statsmodels.tsa.stattools import pacf

def pvalue_pacf(data, max_pvalue=0.07, min_pvalue=-0.07, max_lag=5):

```

```

pacf_values = pacf(data, nlags=max_lag)
for lag in range(1, max_lag + 1):
    if pacf_values[lag] > max_pvalue or pacf_values[lag] < min_pvalue:
        return lag

return 0

__all__ = ["pvalue_pacf"]

from statsmodels.tsa.stattools import acf

def qvalue_acf(data, max_qvalue=0.07, min_qvalue=-0.07, max_lag=5):
    acf_values = acf(data, nlags=max_lag)
    for lag in range(1, max_lag + 1):
        if acf_values[lag] > max_qvalue or acf_values[lag] < min_qvalue:
            return lag

    return 0

__all__ = ["qvalue_acf"]

from statsmodels.tsa.seasonal import seasonal_decompose
import matplotlib.pyplot as plt
from statsmodels.graphics.tsaplots import acf

def seasonal_check(data, P, D, Q):
    decomp = seasonal_decompose(data)
    acf_result = acf(decomp.seasonal)
    max_corr_lag = acf_result.argmax()
    s = max_corr_lag
    if s == 0:
        return (0, 0, 0, 0)
    else:
        return (P, D, Q, s)

__all__ = ["seasonal_check"]

from statsmodels.tsa.stattools import adfuller
import numpy as np

def stationarity_check(data, max_diff=5):
    for i in range(max_diff + 1):
        stationarity_result = adfuller(data)

        print("ADF value:", stationarity_result[0])
        print("p-value:", stationarity_result[1])
        print("Critical values:")
        # for key, value in stationarity_result[4].items():

```

```

    # print("\t%s: %.3f" % (key, value))

    if (stationarity_result[1] <= 0.05) and (
        stationarity_result[4]["5%"] > stationarity_result[0]
    ):
        print("\u001b[32mStationary\u001b[0m")
        return i

    data = np.diff(data)

    print("\x1b[31mNon-stationary\x1b[0m")
    return max_diff

__all__ = ["stationarity_check"]

ReactDOM.createRoot(document.getElementById("root")).render(
    <BrowserRouter>
      <App />
    </BrowserRouter>
);

function App() {
  return (
    <>
      <Routes>
        <Route>
          <Route path="/login" element={<LoginPage />} />
        </Route>
        <Route element={<Layout />}>
          <Route path="/" element={<DashboardPage />} />
          <Route path="/day" element={<ForecastLastDayPage />} />
        </Route>
      </Routes>
    </>
  );
}

export const LoginPage = () => {
  const navigate = useNavigate();
  let location = useLocation();
  let from = location.state?.from?.pathname || "/";
  const handleSubmit = (event) => {
    event.preventDefault();
    const data = new FormData(event.currentTarget);
    axios
      .post("http://localhost:5000/login", {
        email: data.get("email"),

```

```

    password: data.get("password"),
  })
  .then((response) => {
    console.log(response);
    navigate(from, { replace: true });
  })
  .catch((error) => {
    console.error(error);
  });
};

return (
  <ThemeProvider theme={defaultTheme}>
    <Container component="main" maxWidth="xs">
      <CssBaseline />
      <Box
        sx={{
          marginTop: 8,
          display: "flex",
          flexDirection: "column",
          alignItems: "center",
        }}
      >
        <Avatar sx={{ m: 1, bgcolor: "secondary.main" }}>
          <LockOutlinedIcon />
        </Avatar>
        <Typography component="h1" variant="h5">
          Авторизація
        </Typography>
        <Box
          component="form"
          onSubmit={handleSubmit}
          noValidate
          sx={{ mt: 1 }}
        >
          <TextField
            margin="normal"
            required
            fullWidth
            id="email"
            label="Пошта"
            name="email"
            autoComplete="email"
            autoFocus
          />
          <TextField
            margin="normal"
            required

```



```

    fullWidth
    name="password"
    label="Пароль"
    type="password"
    id="password"
    autoComplete="current-password"
  />
  <FormControlLabel
    control={ <Checkbox value="remember" color="primary" /> }
    label="Запомняти мене"
  />
  <Button
    type="submit"
    fullWidth
    variant="contained"
    sx={{ mt: 3, mb: 2 }}
  >
    Авторизуватись
  </Button>
</Box>
</Box>
</Container>
</ThemeProvider>
);
};

```

```
const drawerWidth = 220;
```

```

const AppBar = styled(MuiAppBar, {
  shouldForwardProp: (prop) => prop !== "open",
})(({ theme, open }) => ({
  zIndex: theme.zIndex.drawer + 1,
  transition: theme.transitions.create(["width", "margin"], {
    easing: theme.transitions.easing.sharp,
    duration: theme.transitions.duration.leavingScreen,
  }),
  ...(open && {
    marginLeft: drawerWidth,
    width: `calc(100% - ${drawerWidth}px)`,
    transition: theme.transitions.create(["width", "margin"], {
      easing: theme.transitions.easing.sharp,
      duration: theme.transitions.duration.enteringScreen,
    }),
  }),
}));

```

```

const Drawer = styled(MuiDrawer, {
  shouldForwardProp: (prop) => prop !== "open",

```

```

))(({ theme, open }) => ({
  "& .MuiDrawer-paper": {
    position: "relative",
    whiteSpace: "nowrap",
    width: drawerWidth,
    transition: theme.transitions.create("width", {
      easing: theme.transitions.easing.sharp,
      duration: theme.transitions.duration.enteringScreen,
    }),
    boxSizing: "border-box",
    ...(!open && {
      overflowX: "hidden",
      transition: theme.transitions.create("width", {
        easing: theme.transitions.easing.sharp,
        duration: theme.transitions.duration.leavingScreen,
      }),
      width: theme.spacing(7),
      [theme.breakpoints.up("sm")]: {
        width: theme.spacing(9),
      },
    }),
  },
}));

```

```

const Layout = () => {
  const [open, setOpen] = React.useState(true);
  const toggleDrawer = () => {
    setOpen(!open);
  };
  return (
    <Box
      component="main"
      sx={{
        backgroundColor: (theme) =>
          theme.palette.mode === "light"
            ? theme.palette.grey[100]
            : theme.palette.grey[900],
        flexGrow: 1,
        width: "100%",
        height: "100vh",
        overflow: "auto",
        display: "flex",
        flexDirection: "row",
      }}
    >
    <CssBaseline />
    <AppBar position="absolute" open={open}>
      <Toolbar

```

```

sx={{
  pr: "24px", // keep right padding when drawer closed
  display: "flex",
}}
>
<IconButton
  edge="start"
  color="inherit"
  aria-label="open drawer"
  onClick={toggleDrawer}
  sx={{
    marginRight: "36px",
    ...(open && { display: "none" }),
  }}
>
  <MenuIcon />
</IconButton>
<Typography
  component="h1"
  variant="h6"
  color="inherit"
  noWrap
  sx={{ flexGrow: 1 }}
>
  Прогнозування курсу криптовалюти "Біткоїн"
</Typography>
<IconButton color="inherit">
  <Avatar />
</IconButton>
</Toolbar>
</AppBar>
<Drawer variant="permanent" open={open}>
<Toolbar
  sx={{
    display: "flex",
    alignItems: "center",
    justifyContent: "flex-end",
    px: [1],
  }}
>
  <IconButton onClick={toggleDrawer}>
    <ChevronLeftIcon />
  </IconButton>
</Toolbar>
<Divider />
<List component="nav">
  {mainListItems}
<Divider sx={{ my: 1 }} />

```

```

    {secondaryListItems}
  </List>
</Drawer>
<Box sx={{ width: "100%" }}>
  <Outlet />
</Box>
</Box>
);
};

export default Layout;

export function DashboardPage() {
  const [chartData, setChartData] = React.useState(null);
  React.useEffect(() => {
    const timeNow = moment().unix() * 1000;
    const timeMonthAgo = moment().subtract(1, "months").unix() * 1000;
    const params = {
      symbol: "BTCUSDT",
      interval: "1d",
      startTime: timeMonthAgo,
      endTime: timeNow,
    };

    axios
      .get("https://data-api.binance.vision/api/v3/klines", {
        params: params,
      })
      .then((response) => {
        const responseData = response.data.map((dayData) => {
          return [dayData[0], +dayData[4]];
        });
        setChartData(responseData);
      });
  }, []);

  return (
    <ThemeProvider theme={defaultTheme}>
      <Box>
        <Toolbar />
        <Container maxWidth="lg" sx={{ mt: 4, mb: 4 }}>
          <Grid container spacing={3}>
            { /* Chart */ }
            <Grid item xs={12} md={8} lg={9}>
              <Paper
                sx={{
                  p: 2,
                  display: "flex",

```

```

        flexDirection: "column",
        height: 370,
      }}
    >
    <Chart data={chartData} />
  </Paper>
</Grid>
<Grid item xs={12} md={4} lg={3}>
  <Paper
    sx={{
      p: 2,
      display: "flex",
      flexDirection: "column",
      height: 240,
    }}
  >
    <PriceCard />
  </Paper>
  {/* <Button
    size="large"
    variant="contained"
    sx={{ width: "250px", height: "50px", mt: 5 }}
  >
    Отримати прогноз
  </Button> */}
  <ForecastButton />
</Grid>

  {/* Recent ForecastHistory */}
  <Grid item xs={12}>
    <Paper sx={{ p: 2, display: "flex", flexDirection: "column" }}>
      <ForecastHistory />
    </Paper>
  </Grid>
</Grid>
</Container>
</Box>
</ThemeProvider>
);
}

```

```

const MiniGraph = ({ obj, onClick }) => {
  const getForecastType = (str) => {
    if (str === "1h") return "На годину";
    if (str === "1d") return "На день";
    if (str === "1w") return "На тиждень";
    if (str === "1M") return "На місяць";
  };

```

```

};

return (
  <Box
    sx={{
      display: "flex",
      flexDirection: "column",
      border: "1px solid lightgrey",
      p: 2,
    }}
    onClick={onClick}
  >
    <Stack direction="row" gap={2}>
      <Typography>
        BTC/USD. Тип прогнозу: {getForecastType(obj.interval)}
      </Typography>
    </Stack>
    <Stack direction="row" gap={2}>
      <Typography> Прогноз: {Math.trunc(obj.forecast[0])}</Typography>
    </Stack>
    <Stack direction="row" gap={2}>
      <Typography>Дата: {obj.date}</Typography>
    </Stack>
    <Stack direction="row" gap={2}>
      <Typography>
        Точність: {Math.trunc(100 - obj.mape * 1000 - 20)}%
      </Typography>
      {obj.direction ? (
        </img>
      ) : (
        </img>
      )}
    </Stack>
  </Box>
);
};

```

```

export const ForecastLastDayPage = () => {
  const [lastForecasts, setLastForecasts] = React.useState({
    forecasts: {},
  });
  const [openModal, setOpenModal] = React.useState(false);
  const [modalData, setModalData] = React.useState({});
  const [forecastData, setForecastData] = React.useState({});

```

```

const handleClick = (obj) => {
  const getInterval = (str) => {
    if (str === "1h") return "hours";
    if (str === "1d") return "days";
    if (str === "1w") return "weeks";
    if (str === "1M") return "months";
  };

  const setChartStartTime = (intervalOption) => {
    if (intervalOption === "1h") {
      return moment().subtract(2, "weeks").unix() * 1000;
    }
    if (intervalOption === "1d") {
      return moment().subtract(2, "months").unix() * 1000;
    }
    if (intervalOption === "1w") {
      return moment().subtract(1, "years").unix() * 1000;
    }
    if (intervalOption === "1M") {
      return moment().subtract(3, "years").unix() * 1000;
    }
  };

  const timeEnd = moment(obj.date).unix() * 1000;
  const timeNext =
    moment(timeEnd).add(1, getInterval(obj.interval)).unix() * 1000;
  console.log(
    moment(timeEnd).add(1, getInterval(obj.interval)).startOf("day").unix()
  );
  setOpenModal(true);
  setModalData({
    interval: obj.interval,
    startTime: setChartStartTime(obj.interval),
    endTime: timeEnd,
  });
  setForecastData({
    array: [timeNext, obj.forecast[0]],
    color: obj.direction ? "green" : "red",
  });
};

const fetchLastForecasts = async () => {
  const response = await axios.post("http://localhost:5000/user_forecasts", {
    user_id: "AvY1RNwsqUPBDM0KW9w9lvrCAnp1",
  });

  setLastForecasts(response.data);
};

```

```

React.useEffect(() => {
  fetchLastForecasts();
}, []);

const forecastInfo = Object.values(lastForecasts?.forecasts ?? {})
  .map((forecast) => {
    const today = moment().startOf("day").unix();
    const now = moment(forecast.date).unix();

    if (moment(now).isBetween(today, 1699142400000, undefined, "[)") {
      return forecast;
    }

    return undefined;
  })
  .filter((forecast) => forecast !== undefined);
return (
  <Box sx={{ mt: 10, ml: 5 }}>
    {forecastInfo?.map((obj) => (
      <MiniGraph key={obj.date} obj={obj} onClick={() => handleClick(obj)} />
    ))}
    <ModalHistory
      isOpen={openModal}
      title={`Прогноз BTC/USD`}
      handleCloseModal={() => setOpenModal(false)}
      modalData={modalData}
      forecastData={forecastData}
    />
  </Box>
);
};

```


Додаток В (обов'язковий)

ІЛЮСТРАТИВНА ЧАСТИНА

«ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ПРОГНОЗУВАННЯ КУРСУ
КРИПТОВАЛЮТИ «БІТКОІН»

Виконав: студент 2-го курсу, групи
2КН-22м


спеціальності 122 – «Комп'ютерні
науки»



Герус Я. К.

(прізвище та ініціали)

Керівник: д.т.н., доц. каф. КН



Крилик Л. В.

(прізвище та ініціали)

« 02 » 12 2023 р.

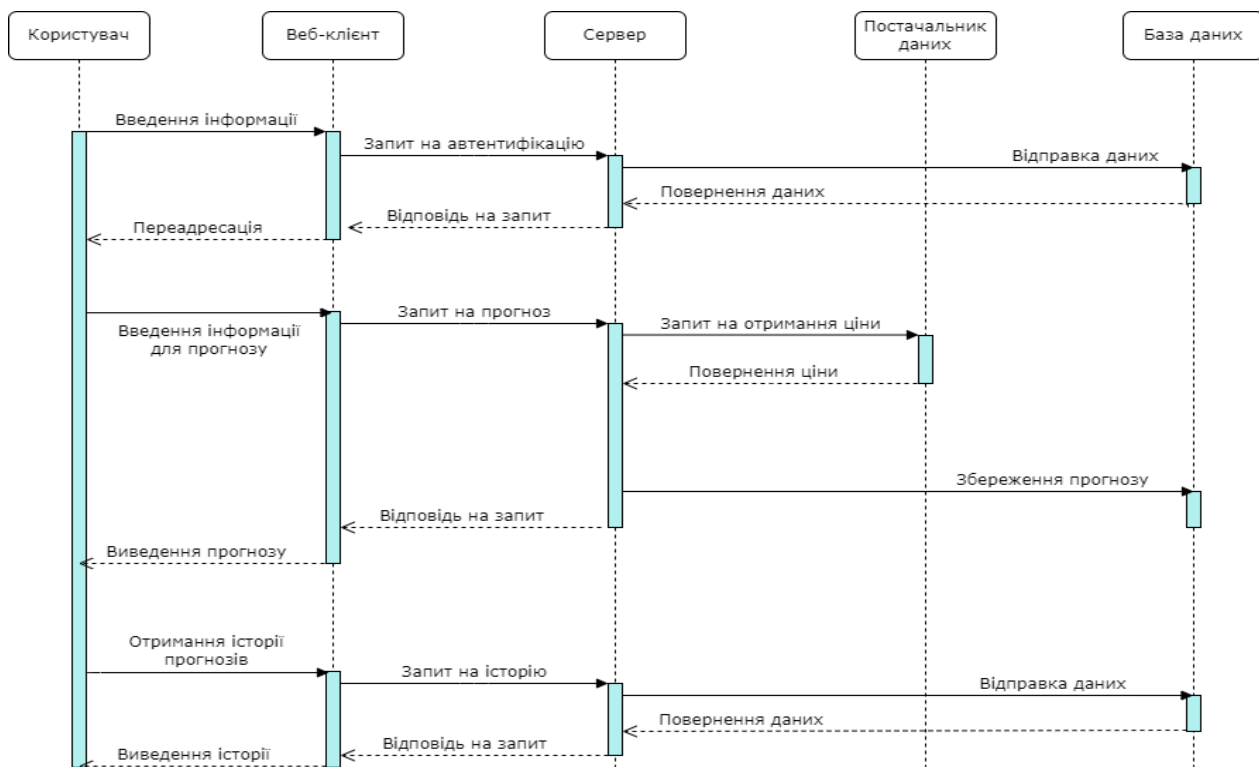


Рисунок В.1 – Діаграма послідовності взаємодії модулів інформаційної технології прогнозування курсу криптовалюти «Біткоїн»

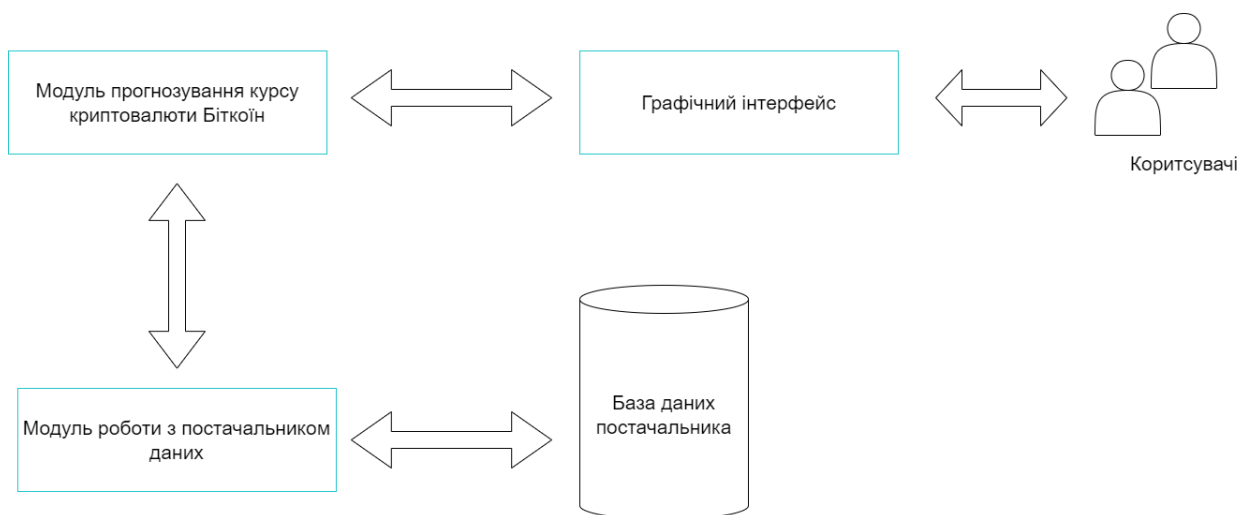


Рисунок В.2 – Загальна структурна модель програмного модуля

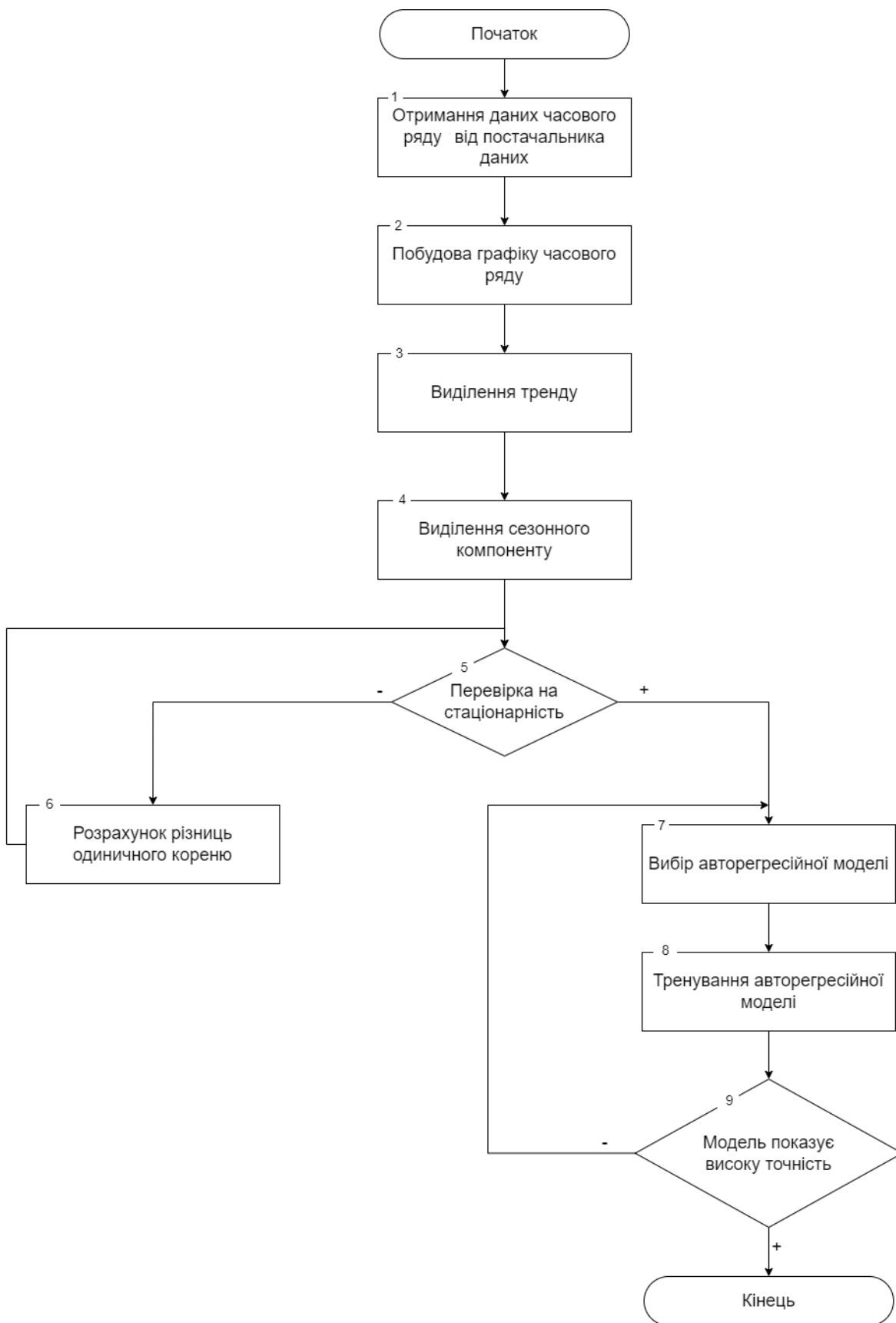


Рисунок В.3 – Схема алгоритму побудови математичної моделі прогнозування курсу криптовалюти «Біткоїн»



Рисунок В.4 – Схема загального алгоритму функціонування програмного модуля прогнозування курсу «Біткоїн»



Рисунок В.5 – Діаграма діяльності серверу під час процесу автентифікації користувача

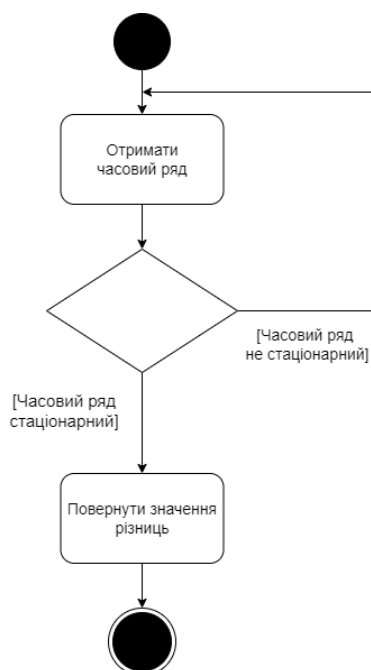


Рисунок В.6 – Діаграма діяльності серверу під час процесу перевірки на стаціонарність

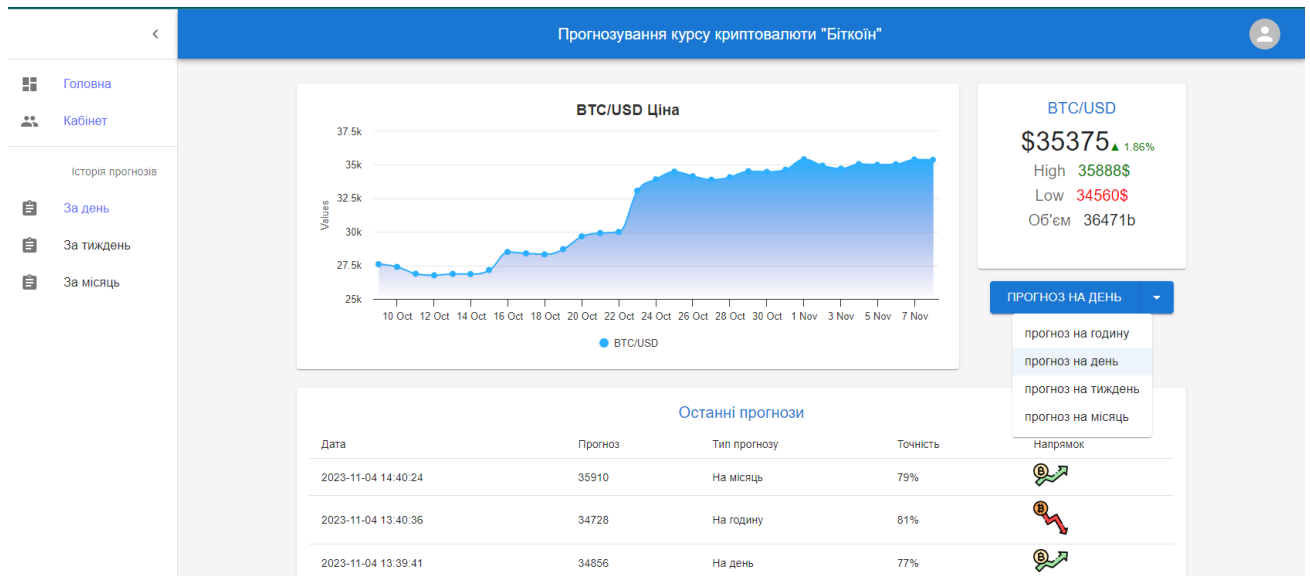
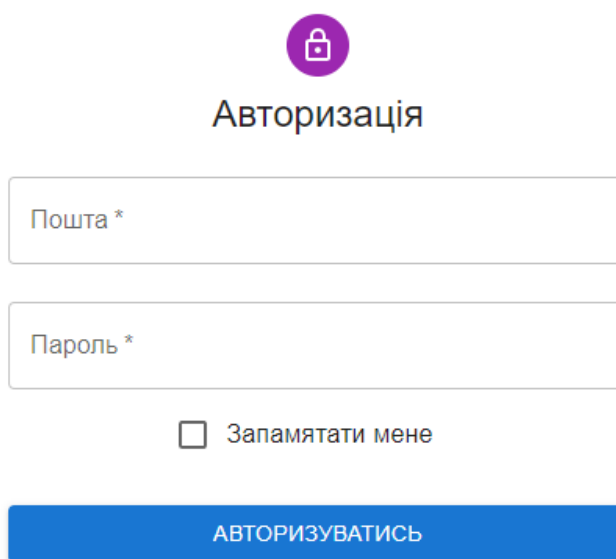


Рисунок В.7 – Загальний вигляд інтерфейсного вікна роботи програми

Додаток Г (довідниковий) Інструкція користувача

Для того щоб почати роботу з інформаційною технологією, потрібно мати персональний комп'ютер. Головною умовою є стабільне підключення до мобільної мережі інтернету або Wifi (рис. Г.1).



Авторизація

Пошта *

Пароль *

Запам'ятати мене

АВТОРИЗУВАТИСЬ

Рисунок Г.1 – Загальний вигляд інтерфейсного вікна авторизації користувача

Після успішної авторизації користувач переадресовується на вікно головного меню програми. Це вікно містить всю потрібну користувачу інформацію. Наприклад у цьому вікні відображається графік криптовалюти «Біткоїн» за останні 30 днів, деяка інформація про ціну за останній день, а саме: остання ціну, найбільшу та найменшу ціну, а також обсяг торгівлі. Також у цьому вікні знаходиться коротка таблиця історії останніх прогнозів користувача. Ця таблиця містить колонки: дата прогнозу, тип прогнозу, числове значення прогноз, приблизна точність прогнозу та напрямок руху ціни. Біля графіку знаходиться кнопка вибору горизонту прогнозування, яка запускає початок прогнозу. У лівій частині вікна знаходиться випадаюче меню, яке надає

навігацію по програмному модулю. Звідти можна перейти в головне вікно програми, в особистий кабінет, або в вікно де відобразатимуться прогнози за останній день, тиждень або місяць (рис. Г.2 –Г.4).

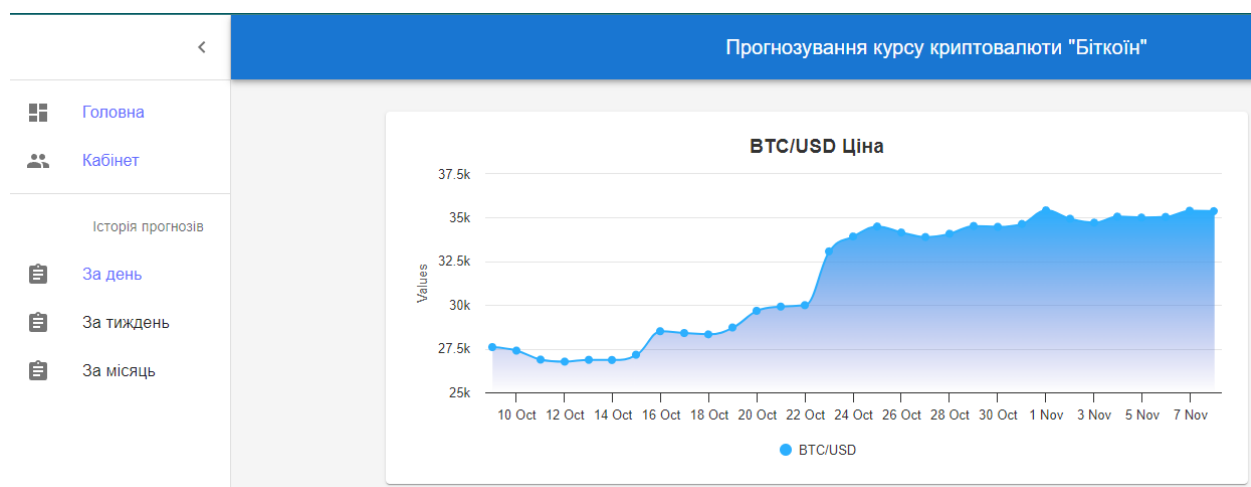


Рисунок Г.2 – Загальний вигляд інтерфейсного вікна головного меню з графіком

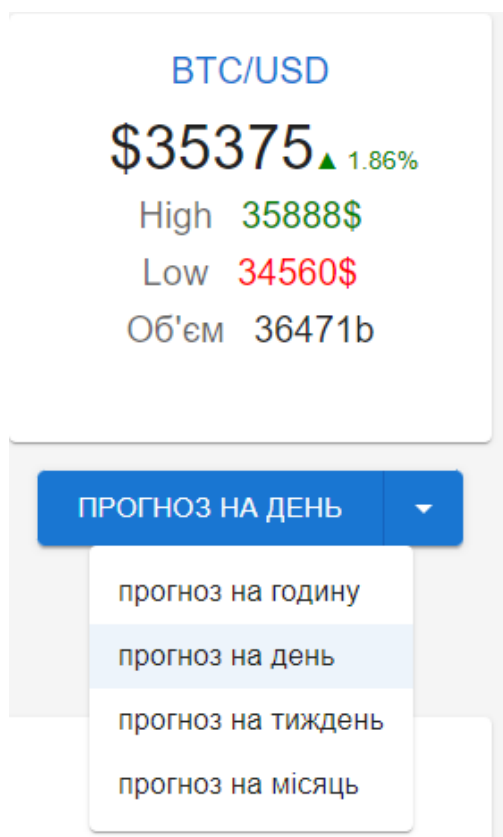


Рисунок Г.3 – Загальний вигляд інтерфейсного вікна головного меню з ціною та КНОПКОЮ





Останні прогнози				
Дата	Прогноз	Тип прогнозу	Точність	Напрямок
2023-11-08 17:03:21	35318	На годину	77%	
2023-11-08 16:58:23	35549	На день	71%	
2023-11-04 14:40:24	35910	На місяць	69%	
2023-11-04 13:40:36	34728	На годину	77%	

Рисунок Г.4 – Загальний вигляд інтерфейсного вікна головного меню з прогнозами

Для отримання прогнозу на бажаний період потрібно зі списку заданих періодів прогнозування вибрати потрібний і натиснути на кнопку «Прогноз на:». Оскільки процес тренування і прогнозування моделі займає проміжок часу, більший ніж звичайний http запит, який користувачі виконують щодня, спочатку відкриється модальне вікно з повідомленням про те, що відбувається прогноз. Після того як запит виконається виведеться графік прогнозу, де міститиметься інформація про дані ціни біткоїна, які були використані для прогнозу і сам прогноз. Такий графік можна приближувати і віддаляти для детальнішого огляду отриманих результатів (рис. Г.5–Г.7).



Рисунок Г.5 – Загальний вигляд інтерфейсного вікна очікування прогнозу



Рисунок Г.6 – Загальний вигляд інтерфейсного вікна отриманого прогнозу



Рисунок Г.7 – Загальний вигляд інтерфейсного масштабованого вікна прогнозу

Для того щоб переглянути історію прогнозів потрібно натиснути на одну з трьох кнопок «За день», «За тиждень», «За місяць» у випадаючому вікні в секції історія прогнозів. Після натиснення відбудеться переадресація на сторінку де виведуться усі прогнози за вибраний період часу. При натисканні на будь-який із прогнозів відкриється модальне вікно, де як і в випадку отриманого прогнозу можна буде переглянути графік (рис. Г.8).

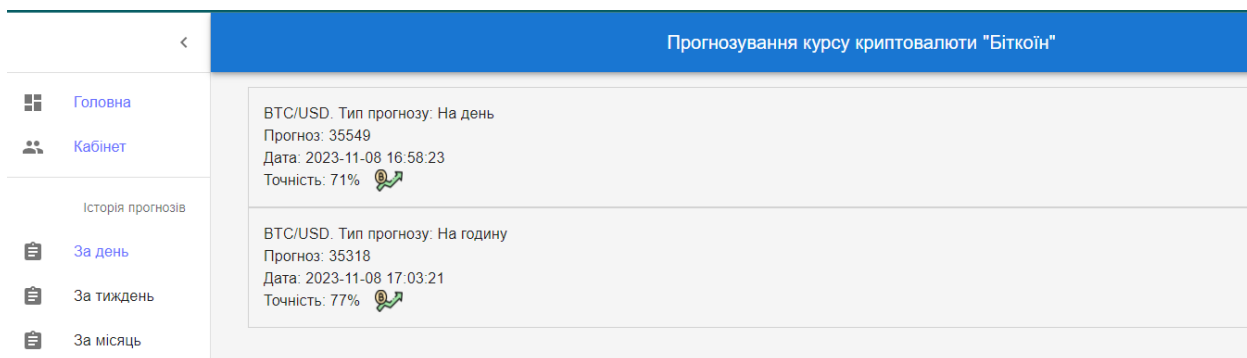


Рисунок Г.8 – Загальний вигляд інтерфейсного вікна історії прогнозів