

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет прикладної математики

Кафедра програмного забезпечення комп'ютерних систем

«На правах рукопису»
УДК 681.3.01

«До захисту допущено»

Завідувач кафедри

_____ Євгенія СУЛЕМА

« ____ » _____ 2024 р.

Магістерська дисертація

на здобуття ступеня магістра

за освітньо-професійною програмою

**«Інженерія програмного забезпечення мультимедійних та
інформаційно-пошукових систем»**

**зі спеціальності 121 Інженерія програмного забезпечення на тему:
«Спосіб та програмне забезпечення для автоматизованого аналізу та
прогнозування ризиків операцій з криптовалютами»**

Виконав:

студент II курсу, групи КП-11мп

Кондрацький Владислав Вадимович

Керівник:

Доцент кафедри ПЗКС, к.т.н., доцент

Люшенко Леся Анатоліївна

Консультант з нормоконтролю:

Доцент кафедри ПЗКС, к.т.н., доцент,

Онай Микола Володимирович

Рецензент:

Доцент кафедри СПСКС, к.т.н., доцент,

Тарасенко-Клятченко Оксана Володимирівна

Засвідчую, що у цій магістерській
дисертації немає запозичень з праць інших
авторів без відповідних посилань.

Студент _____

Київ – 2024 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет прикладної математики

Кафедра програмного забезпечення комп'ютерних систем

Рівень вищої освіти – другий (магістерський)

Спеціальність (спеціалізація) – 121 «Інженерія програмного забезпечення»

Освітньо-професійна програма «Інженерія програмного забезпечення мультимедійних та інформаційно-пошукових систем»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Євгенія СУЛЕМА

«___» _____ 2022 р.

ЗАВДАННЯ
на магістерську дисертацію студенту
Кондрацькому Владиславу Вадимовичу

1. Тема дисертації «Спосіб та програмне забезпечення для автоматизованого аналізу та прогнозування ризиків операцій з криптовалютами», науковий керівник дисертації Люшенко Леся Анатоліївна, ст. викладач кафедри ПЗКС, к.т.н., затверджені наказом по університету від «19» листопада 2023 р. № 5217-С
2. Термін подання студентом дисертації «15» січня 2024 р.
3. Об'єкт дослідження: способи автоматизованого аналізу та прогнозування ризиків операцій з криптовалютами.
4. Предмет дослідження: розробка комплексного способу автоматизованого аналізу та прогнозування ризиків операцій з криптовалютами.
5. Перелік завдань, які потрібно розробити:
 - проведення аналізу існуючих автоматизованих способів прогнозування ризиків операцій з криптовалютами;
 - розробка комбінованого способу аналізу та прогнозування ризиків операцій з криптовалютами із урахуванням знайдених недоліків у існуючих способах;
 - реалізація і тестування розробленого способу та програмного забезпечення;
 - проведення аналізу отриманих результатів;
 - побудова бізнес-моделі.
6. Орієнтовний перелік графічного (ілюстративного) матеріалу:
 - схема архітектури програмного забезпечення;
 - схема бази даних;
 - діаграма роботи модулю прогнозування;
 - графік порівняння точності прогнозу комбінованого способу та базової моделі;
 - дерево проблем.

7. Орієнтовний перелік публікацій:

- Тези доповіді “Спосіб та програмне забезпечення для автоматизованого аналізу та прогнозування ризиків операцій з криптовалютами”

8. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Онай М. В., доцент кафедри ПЗКС		

9. Дата видачі завдання «10» жовтня 2022 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1.	Грунтовне ознайомлення з предметною галуззю	17.10.2022	
2.	Визначення структури магістерської дисертації; вивчення літератури, пошук додаткової літератури, патентний пошук	12.12.2022	
3.	Робота над першим розділом магістерської дисертації; проведення наукового дослідження	13.02.2023	
4.	Проведення наукового дослідження; робота над другим розділом магістерської дисертації; розроблення програмного забезпечення	03.04.2023	
5.	Проведення наукового дослідження; робота над статтею за результатами наукового дослідження	08.05.2023	
6.	Проведення наукового дослідження; робота над третім розділом магістерської дисертації	22.06.2023	
7.	Завершення роботи над основною частиною магістерської дисертації; підготовка ілюстративного матеріалу; підготовка матеріалів доповіді на конференції ПМК-2023	11.09.2023	
8.	Оформлення текстової і графічної частини магістерської дисертації	04.12.2023	

Студент

Владислав КОНДРАЦЬКИЙ

Науковий керівник

Леся ЛЮШЕНКО

РЕФЕРАТ

Актуальність теми. За останні роки попит на альтернативні форми цифрових активів зростає, як наслідок, ринок потребує вдосконалення методів для програмного аналізу та прогнозування ризиків, які виникають під час операцій з ними. Криптовалютний ринок характеризується високою волатильністю та швидкими змінами, що вимагає глибокого розуміння ризику для прийняття обґрунтованих інвестиційних рішень. Прогнозування ризиків дозволяє інвесторам підготуватися до можливих негативних сценаріїв і розробити стратегії їх мінімізації або уникнення.

Для даної магістерської дисертації, актуальним є розробка програмного забезпечення, яке виконує аналізування та прогнозування ризиків операцій з криптовалютою на основі комбінованого способу, який включає, методи машинного навчання, алгоритми штучного інтелекту та статистичні методи.

Об'єктом дослідження програмне аналізування та прогнозування ризиків операції з криптовалютами.

Предметом дослідження є методи та способи програмного аналізування та прогнозування ризиків операцій з криптовалютами.

Мета роботи. Розробка програмного забезпечення з використанням комбінованого способу програмного прогнозування ризиків операцій з криптовалютами в реальному часі для підвищення точності прогнозування.

Наукова новизна. Вперше запропоновано комбінований метод програмного прогнозування ризиків в операціях з криптовалютами, який завдяки експертному аналізу комбінує статистичні методи та алгоритми машинного навчання для збільшення точності прогнозування на 9%.

Практична цінність отриманих результатів роботи полягає в тому що, розроблений спосіб та створене програмне ззабезпечення дає змогу підвищити точність прогнозування курсу криптовалюти та оцінку ризиків при високій волативності та динамічності ринку криптовалют.

Апробація роботи. Основні положення і результати даної роботи були представлені на науковій конференції магістрантів та аспірантів. “Прикладна математика та комп’ютинг ПМК-2023”, та опубліковано у збірнику тез доповідей: Кондрацький В.В., Струцинський О.А., Спосіб та програмне забезпечення для автоматизованого аналізу та прогнозування ризиків операцій з криптовалютами. ”Прикладна математика та комп’ютинг ПМК-2023”, (Київ, 28 – 30 листопада 2023 р.), с. 474 – 480.

Структура та обсяг роботи. Магістерська дисертація складається з вступу, п’яти розділів, висновків та додатків.

У вступі надано загальну характеристику роботи, описано проблеми, обґрунтовано актуальність напрямку досліджень, сформульовано мету і завдання дослідження.

У першому розділі розглянуто існуючі методи прогнозування та аналізу ризиків операцій з крипто валютами, їх обмеження та способи вирішення цих обмежень. Розглянуто існуючі програмні рішення та виокремлено їх недоліки.

У другому розділі розглянуті способи для автоматизованого збору інформації. Розглянуто інтегровані методи аналізу та прогнозування ризиків у криптовалютному ринку. Розібрано запропонований комбінований спосіб прогнозування курсу та ризиків криптовалюти за допомогою експертного аналізу та стекінгу моделей.

У третьому розділі були виокремлені засоби реалізації програмного забезпечення. Описана загальна архітектура програмного забезпечення. Описана система для керування базами даних та розібрана її структура. Описана взаємодія із застосунком та основні сторінки які доступні для користувача.

У четвертому розділі було проведено аналіз та оцінку розробленого програмного забезпечення, фокусуючись на оцінюванні прогнозування ризиків операцій з криптовалютами, тестуванні різних компонентів

системи та шляхах її вдосконалення. Проаналізовано точність комбінованого способу в порівнянні з базовою моделлю.

У п'ятому розділі проаналізовано ключові проблеми прогнозування ризиків, визначено зацікавлені сторони, їхній вплив та очікування. Проведено аналіз ринку, сформовано унікальну ціннісну пропозицію, враховані потенційні витрати та доходи. Розроблено та описано бізнес-модель типу lean canvas.

У висновках проаналізовано отримані результати роботи.

У додатках наведено фрагменти вихідного коду програмної реалізації способу, копії графічних матеріалів та слайдів презентації

Робота виконана на 156 аркушах, містить 3 додатки та посилання на список використаних літературних джерел з 31 найменувань. У роботі використано 12 рисунків.

Ключові слова: ризик, прогнозування, нейронні мережі, машинне навчання, експертний аналіз, стекінг, комбінований спосіб, прогнозування ризиків.

ABSTRACT

Actuality. In recent years, the demand for alternative forms of digital assets has been growing, as a result, the market needs to improve methods for software analysis and forecasting of risks that arise during transactions with them. The cryptocurrency market is characterized by high volatility and rapid changes, which requires a deep understanding of risk to make informed investment decisions. Risk forecasting allows investors to prepare for possible negative scenarios and develop strategies to minimize or avoid them.

For this master's thesis, it was decided to develop and test a method and software for automated risk prediction of transactions with cryptocurrencies.

The object of the research is the market of cryptocurrencies and operations with cryptocurrencies. This includes alternative forms of digital assets, electronic payments, and methods of their analysis and forecasting.

The subject of research is methods and software for automated risk analysis and forecasting in the cryptocurrency market. Development of an integrated risk forecasting approach that includes statistical and machine learning models such as ARIMA, LSTM, GB, RF, and neural networks.

The goal of the work. Development of software using a combined method of software risk forecasting of transactions with cryptocurrencies in real time to increase the accuracy of forecasting..

The scientific novelty. For the first time, a combined method of software risk forecasting in crypto-currency operations has been proposed, which, thanks to expert analysis, combines statistical methods and machine learning algorithms to increase forecasting accuracy by 9%.

The practical value of the obtained work results is that the developed method makes it possible to increase the accuracy of rate forecasting in unstable situations on the cryptocurrency market.

Approbation of work. The main provisions and results of this work were presented at the scientific conference of master's and postgraduate students.

"Applied mathematics and computing PMK-2023", and published in the collection of theses of reports: V.V. Kondratskyi, O.A. Strutsynskyi, Ctool and software for automated analysis and risk forecasting of transactions with cryptocurrencies. "Applied mathematics and computing PMK-2023", (Kyiv, November 28-30, 2023), p. 474 - 480.

Structure and scope of work. The master's thesis consists of an introduction, five chapters, conclusions and appendices.

The introduction provides a general description of the work, describes the problems, substantiates the relevance of the research direction, formulates the goal and task of the research.

In the first chapter, the existing methods of forecasting and risk analysis of operations with crypto currencies, their limitations and ways to solve these limitations are considered. The existing software solutions are considered and their shortcomings are highlighted.

In the second chapter, methods for automated information collection are considered. Integrated methods of risk analysis and forecasting in the cryptocurrency market are considered. The proposed combined method of forecasting the rate and risks of cryptocurrency using expert analysis and model stacking is analyzed.

In the third chapter, the software implementation tools were highlighted. The general architecture of the software is described. The database management system is described and its structure is analyzed. The interaction with the application and the main pages available to the user are described.

In the fourth chapter, the analysis and evaluation of the developed software was carried out, focusing on the assessment of forecasting the risks of operations with cryptocurrencies, testing of various components of the system and ways of its improvement. The accuracy of the combined method was analyzed in comparison with the basic model.

The fifth chapter analyzes the key issues of risk forecasting, identifies stakeholders, their influence and expectations. A market analysis was conducted,

a unique value proposition was formed, and potential costs and revenues were taken into account. A lean canvas business model was developed and described.

The conclusion of the work is analyzed in the conclusions.

The appendices contain fragments of the source code of the software implementation of the method, copies of graphic materials and presentation slides

The work is completed on 156 sheets, contains 3 appendices and references to the list of used literary sources from 31 names. 12 drawings are used in the work.

Key words: risk, forecasting, neural networks, machine learning, expert analysis, stacking, combined method, risk forecasting.

ЗМІСТ

СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ	4
ВСТУП	5
1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ АНАЛІЗУ ТА ПРОГНОЗУВАННЯ РИЗИКІВ ОПЕРАЦІЙ З КРИПТОВАЛЮТАМИ	7
1.1. Ризики операцій з криптовалютами	7
1.2. Аналіз ризиків операцій з криптовалютами	8
1.3. Прогнозування курсу криптовалюти для визначення ризиків операцій з криптовалютами	10
1.4. Аналіз існуючих програмних рішень	15
1.5. Актуальність створення програмного забезпечення	19
1.6. Висновки до розділу	21
2. РОЗРОБЛЕНИЙ СПОСІБ ПРОГРАМНОГО ПРОГНОЗУВАННЯ РИЗИКІВ ОПЕРАЦІЙ З КРИПТОВАЛЮТАМИ	23
2.1. Збір необхідних даних та аналіз криптовалютного ринку	23
2.2. Інтегровані методи аналізу та прогнозування ризиків у криптовалютному ринку	28
2.3. Комбінований спосіб прогнозування курсу та ризиків криптовалюти на основі експертного аналізу та стекінгу моделей	35
2.4. Висновки до розділу	40
3. ОСОБЛИВОСТІ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	42
3.1. Вимоги до програмного забезпечення	42
3.2. Засоби реалізації програмного забезпечення	43
3.3. Архітектура програмного забезпечення	52

3.4. Взаємодія користувача із застосунком	63
3.5. Висновки до розділу	66
4. АНАЛІЗ РОЗРОБЛЕНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	68
4.1. Оцінювання прогнозування ризиків операцій з криптовалютами	68
4.2. Порівняльний аналіз способів прогнозування ризиків операцій з криптовалютами.....	70
4.3. Тестування та аналіз розробленого програмного забезпечення ...	74
4.4. Вдосконалення розробленого програмного забезпечення.....	76
4.5. Висновки до розділу	77
5. ПОБУДОВА БІЗНЕС-МОДЕЛІ	79
5.1. Опис проблеми	79
5.2. Зацікавлені сторони	81
5.3. Комерційне рішення. Основні характеристики	83
5.4. Конкурентні переваги рішення.....	84
5.5. Клієнти. Сегменти ринку споживання.....	85
5.6. Унікальна ціннісна пропозиція.....	87
5.7. Доходи і витрати	88
5.8. Бізнес-модель.....	90
5.9. Висновки до розділу	92
ВИСНОВКИ.....	93
СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ.....	96
ДОДАТКИ.....	99

СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ

Застосунок (англ. application, application software, app) – спеціалізована комп'ютерна програма, яка дозволяє користувачеві вирішувати конкретні прикладні завдання.

Вебзастосунок (часом називають *вебдодаток*) – розподілена програма, де клієнтом є браузер, а сервером є веб-сервер.

Програмне забезпечення (програмні засоби) (ПЗ; англ. software) – набір програм для системи обробки інформації та програмні файли, необхідні для роботи цих програм.

Майнинг криптовалют – це процес використання обчислювальної потужності для вирішення складних математичних завдань, які забезпечують безпеку блокчейн мереж.

Фреймворк (англ. Framework, каркас, платформа, структура, інфраструктура) – інфраструктура для програмних рішень, які полегшують розробку складних систем.

Багатоплатформність (кросплатформність, мультиплатформність) – властивість програмного забезпечення, яке може працювати на кількох програмних (включаючи операційні системи) або апаратних платформах.

Сервер (англ. server – «служба», від англ. to serve – служити, множ. сервери) – у комп'ютерній термінології цей термін може стосуватися окремого комп'ютера або програми. Головною особливістю є здатність машини або програми більшу частину часу працювати автономно без втручання людини, реагуючи на зовнішні події відповідно до встановленого програмного забезпечення.

API (англ. Application Programming Interface) – це набір правил та протоколів для створення та інтеграції програмного забезпечення.

ВСТУП

За останні роки попит на альтернативні форми цифрових активів зростає, як наслідок, ринок потребує вдосконалення методів для програмного аналізу та прогнозування ризиків, які виникають під час операцій з ними. Криптовалютний ринок характеризується високою волатильністю та швидкими змінами, що вимагає глибокого розуміння ризику для прийняття обґрунтованих інвестиційних рішень. Прогнозування ризиків дозволяє інвесторам підготуватися до можливих негативних сценаріїв і розробити стратегії їх мінімізації або уникнення [1].

Управління ризиками в цій сфері є складним процесом, який включає ідентифікацію, аналіз, моніторинг та прийняття рішень щодо операцій з криптовалютою. Недбале управління або нехтування ризиками може призвести навіть до повної втрати інвестицій. Ризики операцій з криптовалютою пов'язані з раптовими та значними змінами цін, шахрайством, інформаційними вкидами та іншими чинниками впливу на ринок криптовалюти [1]. Тому важливо провести ретельне дослідження та аналіз ризиків, пов'язаних з використанням конкретної криптовалюти, перш ніж прийняти рішення про її купівлю чи продаж.

Після визначення потенційних ризиків необхідно розробити стратегію управління ними. Цей процес передбачає розробку підходу до зменшення негативного впливу ризику на портфель активів. Важливий компонент також враховує фактори, які можуть вплинути на ринкові умови, і прогнозує майбутні тенденції.

Розробка програмного забезпечення з використанням нових способів аналізування та прогнозування ризиків операцій з криптовалютами в реальному часі з високою точністю є актуальною для сучасних ринків пов'язаних з цифровими активами. Застосування штучного інтелекту (AI) і машинного навчання (ML) у прогнозуванні курсів криптовалют є перспективним напрямком, який відкриває нові горизонти для розуміння

та управління ризиками. Складність, висока волатильність і наявність численних факторів впливу на ринку криптовалют роблять ці технології особливо цінними.

Дана магістерська дисертація присвячена дослідженню існуючих методів та способів прогнозування ризиків операцій з криптовалютами та розробці комбінованого способу програмного прогнозування ризиків таких операцій в реальному часу з застосуванням нейронних мереж та алгоритмів машинного навчання, а також створенню програмного забезпечення для прогнозування ризиків з криптовалютами, яке забезпечує збір та аналіз історичних даних, оцінку волатильності ринку, прогнозуванню курсу валют та оцінку ризиків з використанням розробленого способу.

У рамках дослідження проаналізовані вже існуючі та рекомендовані підходи до прогнозування ризиків в операціях з криптовалютами, а саме: аналіз ризиків на основі чутливості, моделювання Монте-Карло, диверсифікація, стрес-тестування та аналіз сценаріїв. Результатом магістерської роботи є програмне рішення, в основу якого покладено розроблений спосіб прогнозування ризиків в операціях з криптовалютами в реальному часі для поліпшення точності прогнозування за мінімально можливий час.

1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ АНАЛІЗУ ТА ПРОГНОЗУВАННЯ РИЗИКІВ ОПЕРАЦІЙ З КРИПТОВАЛЮТАМИ

1.1. Ризики операцій з криптовалютами

Перш ніж говорити про аналіз та прогнозування ризиків, пов'язаних з операціями з криптовалютою, важливо зрозуміти поняття ризиків у сфері фінансів. Ризик – властивість ситуації, яка має невизначений результат, з обов'язковою наявністю певних наслідків. При інвестуванні ризики асоціюється з невизначеністю та можливістю втрати, або набуття прибутку. Розуміння ризику та управління ним є ключовими аспектами будь-якого інвестиційного процесу. Криптовалюта є відносно новим класом активів та характеризуються меншою передбачуваністю та більшою волатильністю цін, ніж традиційні інвестиційні активи, такі як акції чи облігації. Це означає, що хоча криптовалюти можуть запропонувати величезний потенціал для високої прибутковості, вони також несуть потенційно високі ризики отримати значні збитки. Ризики мають не тільки негативний, а й позитивний вплив, якщо їх правильно оцінювати та керувати ними [2], а саме:

- *потенціал для отримання високих прибутків:* висока волатильність криптовалют означає, що ціни можуть швидко зростати, надаючи інвесторам можливість отримати величезні прибутки. Інвестори, які розуміють ринкові тенденції та знають, як правильно реагувати на зміни ринку, можуть використовувати ці коливання у своїх інтересах;
- *розширення інвестиційних можливостей:* криптовалюти відкривають двері для нових інвестиційних можливостей, яких немає на традиційних фінансових ринках. Це включає інвестиції в нові, нестабільні і ризиковані технології, які можуть мати досить вагомий вплив на різні сектори економіки в майбутньому якщо інвестиція виявиться успішною;

- *використання ризику як індикатора*: іноді ризик може служити важливим індикатором загального стану ринку або активів. Інвестори, які можуть правильно інтерпретувати ці сигнали, використовують їх для покращення своїх інвестиційних стратегій.

Розглянемо чотири основних типи ризиків які зустрічаються найчастіше на криптовалютному ринку [3]:

- ризики волатильності ринку;
- регуляторні ризики;
- ризики безпеки;
- ризик неплатоспроможності криптоактиву.

Волатильність є однією з ключових характеристик ринку криптовалют. Курс криптовалюти може швидко змінюватися через низку факторів, включаючи новини, технологічні прориви або дії великих компаній на ринку. Ця непередбачуваність може призвести до високих прибутків, а також до ризику значних збитків.

Кіберзлочинність є значним ризиком у сфері криптовалют. Зломи бірж, фішингові атаки та інші типи кіберзлочинів можуть призвести до втрати криптовалютних активів. Зберігання та торгівля криптовалютами вимагає високого рівня кібербезпеки, тому трейдери повинні використовувати надійні методи захисту своїх активів. Може виникнути ризик банкрутства криптовалютної біржі чи іншої фінансової установи. Це може статися через погане управління, кібератаки або раптові коливання ринку. Таке банкрутство може призвести до втрати коштів користувачів, тому важливо вибрати надійну та стабільну торгову платформу.

1.2. Аналіз ризиків операцій з криптовалютами

Аналіз криптовалютних ризиків має вирішальне значення в управлінні ризиками, оскільки він допомагає зрозуміти й оцінити потенційні загрози або можливості, які пов'язані з інвестуванням.

Аналіз ризиків у сфері обмінних операцій криптовалют починається з якісного підходу, заснованого на досвіді, інтуїції та якісному аналізі. Цей підхід включає оцінку впливу різних факторів, таких як новини, нормативні зміни та репутація криптовалютного проєкту та його команди. Аналіз цих аспектів може допомогти визначити потенційні ризики, які можуть вплинути на курс та стабільність криптовалюти. У той же час кількісний аналіз використовує числові дані для більш об'єктивного вимірювання та оцінки ризику. Це може включати статистичний аналіз історичних даних, вивчення нестабільності ринку та оцінку кореляції між різними активами та впливами на їх вартість. Тому використання математичних моделей дозволяє інвесторам точніше прогнозувати ймовірність виникнення і потенційний вплив ризиків.

Аналіз сценаріїв є ще одним важливим інструментом для визначення та оцінки ризиків. Цей підхід передбачає розгляд різних сценаріїв «що-якщо», які можуть статися, та аналіз того, як вони можуть вплинути на портфель. Так проаналізувавши, як портфель відреагує на різке падіння ринку або масштабне регуляторне втручання, можна підготуватися до ризиків, які можуть виникнути в майбутньому.

Аналіз чутливості використовується для оцінки реакції інвестицій на зміни зовнішніх умов, наприклад, коливання процентних ставок або цін на товари. Цей підхід може допомогти інвесторам зрозуміти, як зовнішні фактори впливають на вартість криптовалютних активів.

Не менш важливим є аналіз портфеля, який включає оцінку ризику на рівні всього портфеля (а не лише окремих активів). Аналіз показує, як різноманітні активи в портфелі взаємодіють один з одним, а також загальний баланс ризику між ризикованими та безризиковими інвестиціями. Важливим аспектом є оцінка диверсифікації портфеля, яка допомагає знизити загальний рівень ризику.

1.3. Прогнозування курсу криптовалюти для визначення ризиків операцій з криптовалютами

Різноманітність спеціальних аспектів, чинників та неможливість однозначно визначити вплив конкретного з них на ризики під час операцій з криптовалютою, призводить до створення та вдосконалення великої кількості методів прогнозування ризиків.

До найбільш розповсюджених належать [5]:

- фундаментальний аналіз;
- технічний аналіз;
- аналіз ризиків на основі чутливості;
- моделювання Монте-Карло;
- портфельний аналіз і диверсифікація;
- стрес-тестування та аналіз сценаріїв.

Прогнозування ризиків у сфері криптовалют вимагає використання методів та способів, кожний з яких забезпечує унікальний внесок у розуміння потенційних загроз. Одним із основних видів аналізу є фундаментальний аналіз, який дозволяє визначити внутрішню цінність криптовалюти з урахуванням економічних, фінансових, політичних та інших відповідних факторів. Цей аналіз допомагає визначити фактори, які можуть вплинути на майбутній курс криптовалюти а отже і на ризики під час операцій з нею.

В свою чергу технічний аналіз використовує історичні ринкові дані для визначення можливих трендів. Хоча цей метод не може точно передбачити майбутні зміни цін, він має вирішальне значення для розуміння основних поведінкових трендів ринку.

Оцінка ризику також можлива завдяки аналізу чутливості, який допомагає оцінити реакцію курсу криптовалюти на зміни в економічних і фінансових умовах, наприклад, коливання процентних ставок або цін на нафту. Такий аналіз базується на зв'язку між різними факторами ринку та курсу криптовалюти.

Метод Монте-Карло дозволяє оцінити ризики шляхом створення потенційних сценаріїв та оцінки їх ймовірності. Як наслідок, це покращує розуміння ризиків, які виникають під час операцій з криптовалютою.

Аналіз і диверсифікація майбутнього портфеля криптовалютних активів передбачає оцінку різних активів у портфелі, що допомагає зменшити загальний ризик шляхом диверсифікації інвестицій між різними класами активів.

Останній, але не менш важливий метод – це стрес-тестування та аналіз майбутніх сценаріїв. Цей підхід допомагає моделювати екстремальні ринкові умови та оцінювати реакцію портфеля активів на різні надзвичайні ситуації, які можуть виникнути на ринку.

Аналізуючи та прогнозуючи ризики операцій з криптовалютою, важливо враховувати різні підходи, кожен зі своїми унікальними перевагами та обмеженнями. Фундаментальний аналіз оцінює внутрішню цінність криптовалюти на основі економічних і фінансових факторів, які можуть враховувати ширші умови ринку. Хоча цей підхід може надати цінну інформацію, він вимагає глибокого розуміння та не завжди може точно передбачити короткострокові коливання ринку.

З іншого боку, технічний аналіз зосереджується на історичних даних і ринкових моделях, допомагаючи визначити основні тренди, а також рівні опору або підтримки. Хоча цей підхід корисний для розуміння загальних тенденцій, він може не враховувати раптові зовнішні події, які впливають на ринок.

Аналіз чутливості показує, як курс криптовалюти реагує на зовнішні економічні зміни, такі як коливання процентних ставок. Це дає важливу інформацію про зв'язок між криптовалютами та глобальними економічними факторами. Однак точна оцінка всіх зовнішніх ефектів може бути важкою через складність ринкових умов.

Моделювання за методом Монте-Карло використовує випадково згенеровані вхідні дані для генерації розподілу можливих результатів, що

дозволяє інвесторам оцінювати різні сценарії та їхній потенційний вплив на їхні інвестиції. Хоча цей метод корисний для прогнозування, його здатність враховувати всі можливі майбутні ситуації може бути обмеженою.

З іншого боку, аналіз портфеля та диверсифікація допомагають диверсифікувати інвестиційні ризики та зменшити потенційні втрати через розподіл активів. Такий підхід важливий для забезпечення стабільності портфеля, але не вирішує всіх проблем ідентифікації та оцінки окремих ризиків.

Стрес-тестування та аналіз сценаріїв дозволяють змодельовати екстремальні ринкові умови та оцінити реакцію портфеля на різні кризові сценарії. Ці методи є важливими для розуміння реакції інвестицій на несподівані події, хоча вони можуть не враховувати всі основні ринкові умови.

Беручи до уваги ці аспекти, очевидно, що існує потреба у створенні комбінованого способу для програмного аналізу та прогнозування ризиків під час операцій з криптовалютами. Такий спосіб може поєднувати ці різні підходи в єдину систему, надаючи трейдерам та інвесторам комплексні інструменти для ефективного управління ризиками, що є ключовим для успішного інвестування в цю складну галузь.

У процесі створення програмного способу для прогнозування ризиків, пов'язаних з криптовалютою, вирішальну роль відіграє можливість точного прогнозування курсів криптовалют валют. Прогнози курсу є не тільки інструментом для оцінки потенційних прибутків або збитків, вони також є фактором у виявленні та оцінці ризиків. Ключовим аспектом цього процесу є розуміння майбутніх тенденцій волатильності, що дає змогу оцінити рівень ризику. Так висока волатильність, як правило, супроводжується вищим ризиком, але також може надавати більше можливостей для отримання прибутку. Тому точне прогнозування волатильності є ключовим для розуміння загального профілю ризику.

У той же час, аналіз тренду курсів криптовалюти може допомогти визначити загальний напрям ринкових трендів. Отже знання того, чи знаходиться ринок у висхідному чи спадному тренді, може допомогти інвесторам скоригувати свої інвестиційні стратегії відповідно до поточних умов. Прогнози курсу криптовалюти включають аналіз зовнішніх факторів, таких як економічні новини, політичні події або нормативні зміни, і їх вплив на курс криптовалют. Розуміння цих зовнішніх факторів є важливим для оцінки ризиків, пов'язаних із непередбаченими подіями, які можуть раптово вплинути на ринок.

Використання технологій, таких як штучний інтелект (AI) і машинне навчання (ML), може значно підвищити точність прогнозів і здатність адаптуватися до швидко мінливих умов ринку.

Використання AI та ML для прогнозування обмінних курсів криптовалют відкриває нові можливості для глибшого розуміння ринкових тенденцій. Алгоритм ML здатний аналізувати великі обсяги даних, включаючи історичні ціни та обсяги торгів, щоб ідентифікувати складні закономірності та взаємозв'язки, які можуть вплинути на курси криптовалюти [6].

Ці технології дозволяють моделювати різні сценарії та виявляти потенційні ризики, значно підвищуючи ефективність процесу управління ризиками. Наприклад, можна змоделювати вплив глобальних економічних подій або змін регуляторної політики на процентні ставки криптовалюти, що може краще підготувати інвесторів до можливих коливань ринку.

Розглянемо модель ARIMA (модель авторегресійної інтегрованої ковзної середньої). ARIMA – це статистична модель, яка використовується для аналізу та прогнозування часових рядів. Цю модель можна використовувати для прогнозування обмінних курсів криптовалюти, оскільки вона дозволяє визначати й аналізувати тенденції, сезонність та інші моделі в історичних даних.

Побудова моделі ARIMA складається з кількох етапів [7]:

1. Ідентифікація: на цьому етапі визначається структура моделі ARIMA, включаючи визначення порядків авторегресії (AR), інтегралу (I) і ковзного середнього (MA);
2. Оцінка: на основі вибраних параметрів моделі проводиться оцінка коефіцієнтів;
3. Перевірка: після оцінки моделі проводиться аналіз залишків для перевірки її адекватності;
4. Прогнозування: використання моделі для визначення майбутніх значень часового ряду.

Що ж до використання VaR та ES для оцінки ризиків, після розробки прогнозу за допомогою моделі ARIMA, можна використовувати такі інструменти ризик-менеджменту, як Value at Risk (VaR) та Expected Shortfall (ES), для оцінки ризиків. VaR допомагає визначити максимальну втрату, яку може зазнати портфель протягом певного періоду часу з заданим рівнем довіри, ES, у свою чергу, оцінює середню втрату портфеля в умовах, коли втрати перевищують VaR.

Однак ці моделі мають певні обмеження. Наприклад, ARIMA не завжди адекватно враховує раптові та непередбачувані зміни на ринку, а VaR не враховує ризики в ситуаціях поза нормальними умовами. Крім того, вони не завжди ефективні у виявленні складних залежностей у зашифрованих даних.

Це підкреслює необхідність розробки комплексних моделей, які поєднують різні методи та алгоритми для створення більш точних прогнозів курсу та ризику. Інтеграція різних аналітичних методів, включаючи штучний інтелект і машинне навчання, допомагає створити систему, здатну адаптуватися до швидко мінливих умов ринку криптовалют і надати інвесторам надійні інструменти управління ризиками.

1.4. Аналіз існуючих програмних рішень

Проаналізуємо сучасні програмні рішення для управління ризиками операцій з криптовалютами.

CoinMetrics є провідним програмним постачальником даних аналізу ринку криптовалют (рис. 1.1). Він містить різноманітний набір аналітичних інструментів, включаючи графіки динаміки цін, обсяги торгів, а також кореляційний аналіз [8]. Крім того, CoinMetrics надає доступ до базової інформації, такої як ринкова капіталізація, обсяг торгівлі та вартість майнінгу. Ця система використовує надійні алгоритми та статистичні моделі для прогнозування небезпек і надання користувачам інформації, яка є актуальною та фактичною для прийняття рішень. CoinMetrics пропонує інструменти для аналізу інформації про криптовалюту, включаючи ринкові дані, аналіз блокчейну та основну інформацію про активи, ця інформація потім використовується для оцінки стану криптовалюти. Додатково надається доступ до таких показників, як ціна, обсяг торгівлі, капіталізація ринку, активність спільноти тощо. Крім того, він використовує додаткові показники, такі як NVT (відношення вартості мережі до транзакцій) і MVRV (ринкова вартість до реалізованої вартості), щоб оцінити потенційну переоцінку або недооцінку активів.

CryptoCompare – це ще одна популярна система, яка надає інструменти та дані, які використовуються для оцінки ризику на ринку криптовалют (рис. 1.2) [9]. Він забезпечує комплексний аналіз даних ринку, включаючи ціни, обсяг торгів, капіталізацію та інші показники. CryptoCompare також має функцію побудови діаграм і графіків, які візуалізують дані та полегшують прийняття рішень. CryptoCompare надає детальний опис даних ринку криптовалют, включаючи ціни, обсяги торгів, котирування на різних біржах, ліквідність та інші показники. Включає здатність порівнювати та впорядковувати активи криптовалюти за різними критеріями.

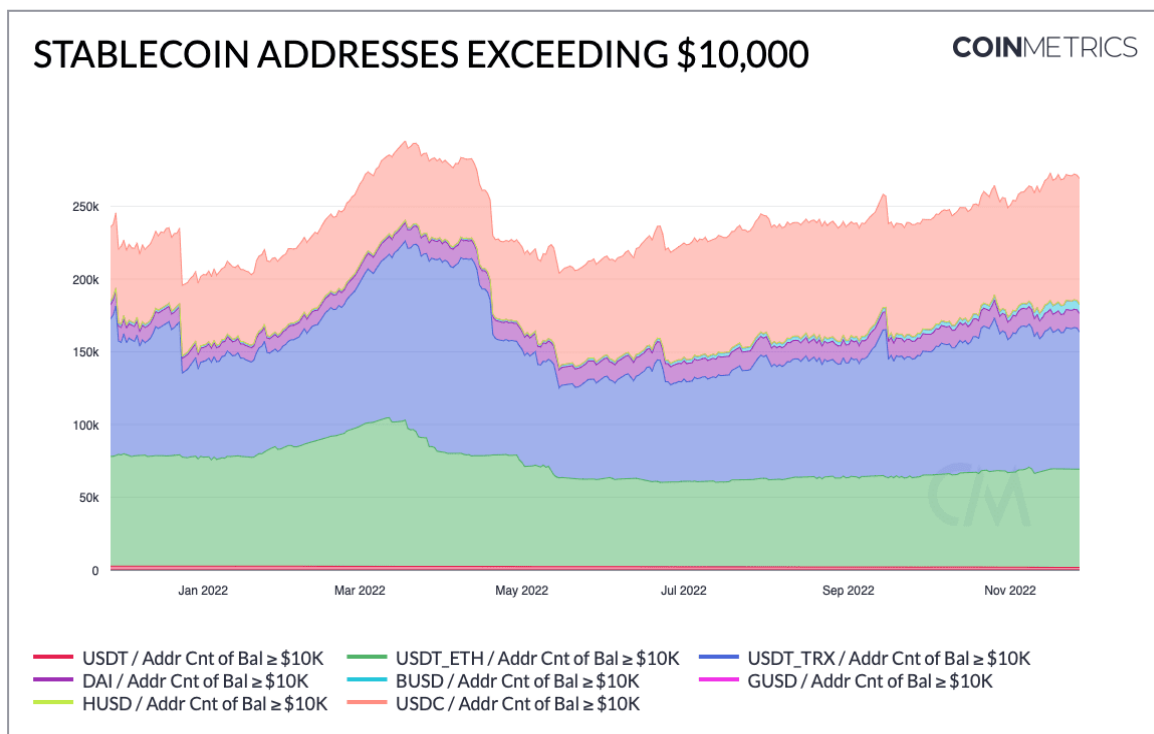


Рис. 1.1. Графік прогнозу системи «CoinMetrics»

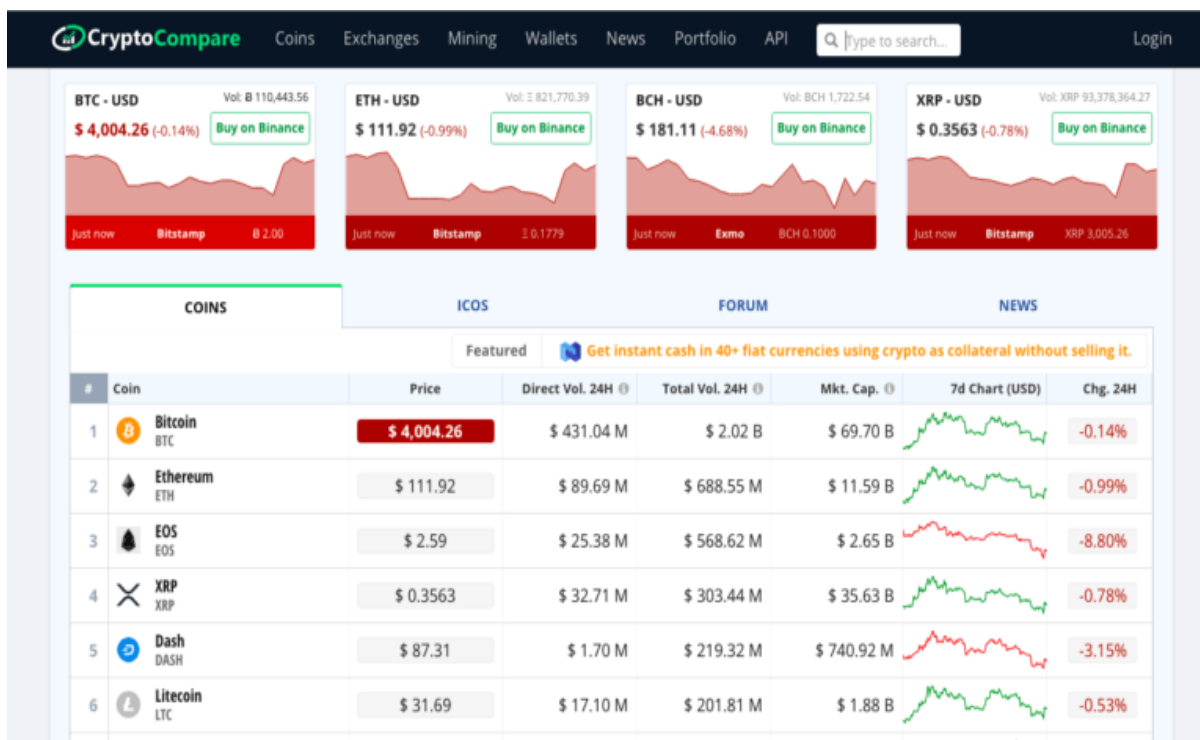


Рис. 1.2. Знімок інтерфейсу системи «CryptoCompare»

Santiment – це нова система аналізу криптовалют, яка використовує штучний інтелект і технологію блокчейн для отримання та ретельного вивчення великого обсягу даних (рис. 1.3)[10]. Santiment має велику кількість розширеного аналізу щодо цінкових тенденцій, обсягу торгів, інформації про гаманець і поведінкової поведінки. Система використовує складні алгоритми та моделі машинного навчання для оцінки ризику та визначення напрямку ринків криптовалют. Santiment також сприяє взаємодії з громадою та обміну аналітичними думками та прогнозами.

Santiment – це нова система, яка використовує штучний інтелект і технологію блокчейн для аналізу даних про криптовалюту та оцінки ризиків. Він також надає інструменти для розпізнавання ринкового обману, злочинної діяльності та незвичної поведінки. Також важливо визнати, що вона цікавиться поведінкою спільноти та аналізом соціальних мереж, щоб визначити настрої та тенденції ринку.

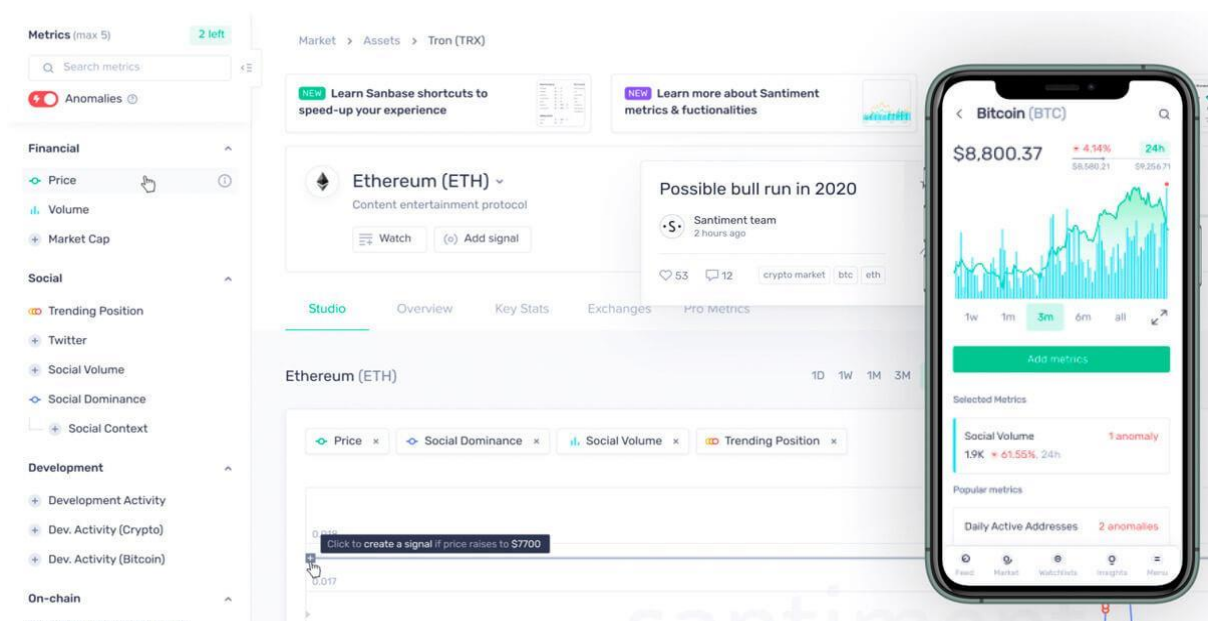


Рис. 1.3. Веб сайт та мобільний застосунок системи «Santiment»

Системи CoinMetrics, CryptoCompare та Santiment, мають різні можливості та підходи до аналізу даних та оцінки ризиків.

В таблиці 1.4 наведено порівняльний аналіз властивосте описаних вище систем.

Таблиця 1.4

Порівняння систем CoinMetrics, CryptoCompare та Santiment

Критерії	CoinMetrics	CryptoCompare	Santiment
Зручний інтерфейс	відсутній	присутній	присутній
Прогнозування курсу	присутнє	присутнє	присутнє
Відкритий системний код	присутній	відсутній	відсутній
Використання штучного інтелекту	присутнє	присутнє	присутнє
Прогнозування та оцінка ризиків	відсутня	відсутня	відсутня
Графічні інструменти	відсутні	присутні	присутні

Як видно з таблиці 1.4 системи CryptoCompare і Santiment мають зручний інтерфейс, який полегшує доступ і навігацію. Також вони пропонують різноманітні параметри налаштування, які дозволяють адаптувати систему до особистих уподобань і потреб користувача. Відкрита система Santiment дозволяє розширювати та змінювати функціональність системи в будь-який час. CoinMetrics і Santiment використовують штучний інтелект для оцінки та прогнозування небезпек, що забезпечує більш точні результати. CoinMetrics демонструє високий ступінь якості прогнозованого курсу криптовалюти. Усі три системи мають багатий набір візуальних засобів, які полегшують візуальну інтерпретацію даних і тенденцій на ринку криптовалют.

Незважаючи на розширені можливості таких систем, як CoinMetrics, CryptoCompare та Santiment, існують певні обмеження щодо прогнозування та оцінки ризику криптовалюти. Ці системи надають велику кількість

корисної інформації, але вони зосереджені на зборі й аналізі ринкових даних, тенденцій і поведінки спільноти, не приділяючи належної уваги аналізу ризиків та прогнозування. CoinMetrics зосереджений більше на ринкових показниках NVT і MVRV, які не дають повного уявлення про потенційні ризики. CryptoCompare надає велику кількість інформації про ринкові умови, але відсутність спеціальних інструментів оцінки ризиків обмежує його корисність для управління ризиками. Хоча Santiment використовує передові технології, такі як штучний інтелект для аналізу даних, він більше спеціалізується на аналізі поведінки та настроях ринку, а не на оцінці та прогнозуванню ризиків.

1.5. Актуальність створення програмного забезпечення

Однією з ключових проблем ринку криптовалют є відсутність програмних інструментів аналізу ризиків операцій з криптовалютою, що має вирішальне значення для інвесторів, трейдерів тощо.

Аналіз існуючих рішень виявляє наступні ключові проблеми:

- способи та методи: недосконалість існуючих способів аналізування та прогнозування ризиків операцій з криптовалютами в умовах високої волативності;
- терміни створення прогнозів: більшість існуючих рішень не можуть забезпечити дійсні прогнози в короткостроковій перспективі, що важливо для ринку криптовалют, що швидко змінюється;
- середня точність прогнозів невизначена: відсутність інформації про точність прогнозів робить їх ненадійними та збільшує кількість користувачів ризик матеріальних втрат;
- функціональна надлишковість: деякі рішення пропонують надмірну функціональність, яка може не знадобитися середньому користувачеві чи дрібному інвестору.

Існує актуальна потреба у створенні способу та програмного забезпечення, яке вирішує вищезазначені проблеми та надає користувачам функціональні можливості:

1. Налаштування терміну прогнозу: можливість вибрати короткий, середній і довгий періоди для прогнозування, що дає користувачам гнучкість і можливість своєчасно реагувати на зміни ринку.
2. Індикація ймовірної точності прогнозу: інформує про середню точність результатів прогнозу, що дає користувачам змогу краще зрозуміти потенційні ризики.
3. Визначення факторів, що впливають на прогнози: можливість вибору різних економічних, політичних і ринкових факторів, які можуть вплинути на прогнози.
4. Адаптованість до вимог користувача: здатність програми налаштовувати свою роботу на основі факторів, вибраних користувачем.

Виходячи з вище зазначеного, актуальним є програмне забезпечити, яке виконує аналізування та прогнозування ризиків операцій з криптовалютою з використанням методів машинного навчання та алгоритмів штучного інтелекту, що дозволить краще оцінити потенційні загрози та протистояти ним.

Отже метою магістерської дисертації є розробка комбінованого способу програмного прогнозування ризиків операцій з криптовалютами в реальному часі з застосуванням нейронних мереж та алгоритмів машинного навчання, а також створення програмного забезпечення, яке забезпечує збір та аналіз історичних даних криптовалютного ринку, оцінку волатильності цього ринку, прогнозування курсу валют та оцінку ризиків з використанням розробленого способу

1.6. Висновки до розділу

Основні ризики під час операцій з криптовалютами, включають: волатильність ринку; законодавчі та нормативні зміни; безпека криптовалютних активів. Ці фактори впливають на стабільність і прибутковість інвестицій.

Для ефективного управління ризиками необхідна оцінка, яка включає в себе якісні та кількісні методи, а саме:

- метод Монте-Карло дозволяє оцінити ризики шляхом створення потенційних сценаріїв та оцінки їх ймовірності;
- аналіз і диверсифікація майбутнього портфеля криптовалютних активів, що дозволяє зменшити загальний ризик шляхом диверсифікації інвестицій між різними класами активів;
- метод стрес-тестування та аналіз майбутніх сценаріїв, який дає можливість моделювати екстремальні ринкові умови та оцінювати реакцію портфеля активів на різні надзвичайні ситуації.
- фундаментальний аналіз оцінює внутрішню цінність криптовалюти на основі економічних і фінансових факторів, які можуть враховувати умови ринку.
- технічний аналіз зосереджується на історичних даних і ринкових моделях, допомагаючи визначити основні тренди, а також рівні опору або підтримки.

Після дослідження та аналізу було виявлено, що ці методи не враховують короткострокові тренди, спрощують складні фінансові взаємозв'язки і не враховують всі можливі ризики на ринку криптовалют.

Дослідження сучасних програмних систем таких, як CoinMetrics, CryptoCompare та Santiment, які процюють на ринку криптовалют, показує, що ці системи мають корисні інструменти, але вони зосереджені на зборі й аналізі ринкових даних, тенденцій і поведінки спільноти, не приділяючи

належної уваги аналізу та можливості прогнозуванню ризиків операцій з криптовалютами.

Враховуючи це, було прийнято рішення зосередитися на створенні програмного забезпечення на основі розробленого комбінованого способу аналізування та прогнозування ризиків операцій з криптовалютами з використанням методів машинного навчання та алгоритмів штучного інтелекту. Це програмне забезпечення має виконувати:

- Точний аналіз ризиків. Це означає, що не лише збираються та аналізуються біржові дані, але також використовує алгоритми та аналітичні інструменти для аналізування ризиків.
- Надання даних про середню точність прогнозованих результатів, що дозволить приймати більш обґрунтовані рішення та розуміти потенційні ризики та невизначеності. Це особливо важливо для зниження ризику непередбачених втрат.
- Зручний інтерфейс і формат відображення інформації, який буде інтуїтивно зрозумілим і зручним. Це не тільки забезпечує простоту використання, але й робить аналіз і прогнози більш доступними для широкого кола інвесторів.

Отже актуальним є програмне забезпечити, яке виконує аналізування та прогнозування ризиків операцій з криптовалютою з використанням методів машинного навчання та алгоритмів штучного інтелекту, що дозволить краще оцінити потенційні загрози на криптовалютному ринку та протистояти ним.

2. РОЗРОБЛЕНИЙ СПОСІБ ПРОГРАМНОГО ПРОГНОЗУВАННЯ РИЗИКІВ ОПЕРАЦІЙ З КРИПТОВАЛЮТАМИ

2.1. Збір необхідних даних та аналіз криптовалютного ринку

Дані, які необхідні для прогнозування валютного курсу та ризиків, збираються автоматизовано з загальнодоступних онлайн-ресурсів. API зазвичай використовується для отримання поточної інформації з біржових ресурсів про курси, показники ринку криптовалют, новини тощо. API надає розробникам доступ до функцій і даних, пов'язаних із певною послугою, у даному випадку – ринком крипто валют [11].

CoinAPI-арі було обрано для дослідження та аналізу ринку криптовалют. Це надійний і широко використовуваний програмний сервіс, який надає доступ до фінансової інформації щодо криптовалют. Цей вибір є виправданим з кількох причин, а саме:

- Широкий спектр даних. CoinAPI надає доступ до великої кількості даних, включаючи історичні та поточні курси криптовалют, дані про обсяги торгів, історія купівель та інформацію про операції на різних обмінних платформах, а також надає інформацію про регуляторні та технічні показники. Така широта даних є критично важливою для точного аналізу ринку з подальшим прогнозуванням ризиків.
- Актуальність інформації. Також не менш важливим аспектом цього API є актуальність інформації та висока якість даних. CoinAPI надає дані у реальному часі, що є важливим для трекінгу швидкозмінних тенденцій на криптовалютному ринку.
- Історичні дані. CoinAPI надає велику базу історичних даних, які використовуються в даній роботі для тренування моделей. Ці данні також відомі своєю надійністю і точністю. Це

важливо для розробки надійних моделей прогнозування, які мають базуватися на точних і актуальних даних.

- Агригація даних. CoinAPI агрегує інформацію з множини криптовалютних бірж, що дає змогу отримати більш всебічний огляд ринку, ніж при використанні даних лише з однієї біржі.

Враховуючи всі ці аспекти, використання даних з CoinAPI є обґрунтованим вибором, як джерело даних для дослідження.

Наведемо нижче код для отримання даних валютної пари, поточну ціну певної валюти або кількох різних валют, а також для запиту історичної інформації щодо ціни та важливої інформації про криптовалюту, доступ до якої здійснюється через CoinAPI API. Використовуючи запит HTTP_GET, відповідь від сервера API, який потім перетворюється на JSON [12].

Розглянемо перший приклад:

```
import requests
api_key = 'FF4AACC3-C6FF-47A1-8F4D-5EB8DC574699'
base_url = 'https://rest.coinapi.io/v1/'
```

Запит на отримання списку доступних криптовалют

```
url = f'{base_url}assets'
headers = {'X-CoinAPI-Key': api_key}
response = requests.get(url, headers=headers)
data = response.json()
print('Список доступних криптовалют:')
for asset in data:
    print(asset['name'])
```

Відповідь яку було отримано:

Bitcoin

Litecoin

Ripple

Yen

Наведений код є прикладом запиту до API CoinAPI для отримання списку доступних криптовалют. Для взаємодії з API використовується бібліотека requests. Розберемо його детальніше, щоб надалі розуміти логіку побудови запитів.

- Визначається `api_key` – ключ доступу до API, який потрібно отримати на офіційному веб-сайті CoinAPI.
- Встановлюється `base_url`, який вказує на базову адресу API.
- Формується URL для запиту до API шляхом додавання `/assets` до `base_url`.
- Створюється `headers` – заголовки HTTP-запиту, в якому вказується `X-CoinAPI-Key` зі значенням `api_key`.
- Виконується HTTP-запит GET за допомогою `requests.get()`, передаючи URL та заголовки.
- Відповідь, яку повертає сервер API, записується у змінну відповіді.
- Відповідь перетворюється на JSON за допомогою методу `response.json()`, а результат зберігається в змінній `data`.
- Кожна крипто валюта впорядкована, відображаються її назви (назва активу – `asset['name']`).

Отримана відповідь містить список доступних крипто валют, задокументованих в CoinAPI. У цьому прикладі показано перші кілька одиниць, але список набагато більший. Використання API полегшує автоматизацію отримання поточної інформації та новин про ринок крипто валют, потім ця інформація та новини аналізуються. Крім того, API надає засоби для інтеграції з іншими програмними та апаратними засобами для розширення функціональності та глибини аналізу.

Розглянемо другий приклад, отримання останніх цін для певної криптовалюти:

Запит на отримання останніх цін для певної криптовалюти

```
symbol_id = 'BTC'
url = f'{base_url}exchangerate/{symbol_id}/USD'
response = requests.get(url, headers=headers)
data = response.json()
```

```
print('Остання ціна Bitcoin (BTC):')
print(data['rate'])
```

Отримана відповідь:

```
Остання ціна Bitcoin (BTC):
25902.276419810856
```

Тут по аналогії з попереднім прикладом отримана остання ціна на таку крипто валюту як Bitcoin. Запити такого формату використовуються для актуалізації інформації на клієнтській частині програмного забезпечення. Кожного разу коли користувач хоче дізнатися інформацію про криптоактив що міститься в його інвестиційному портфелі, виконується запит такого типу. Одночасно із цим запитом, якщо актив який переглядає користувач знаходиться в його портфелі, потрібно надати всю доступну інформацію про майбутні прогнозовані ризики та курс. Для цього система використовує запити на отримання історичних цін для подальшого аналізу, навчання моделей та відображення графіків за допомогою розробленого алгоритму.

Код отримання історичних цін без додаткових налаштувань:

Запит на отримання історичних даних цін для певної криптовалюти

```
symbol_id = 'BTC'
headers = {'X-CoinAPI-Key': api_key}
url=f'{base_url}ohlcv/{id}/USD/history?period_id=1DAY&limit=10'
response = requests.get(url, headers=headers)
if response.status_code == 200:
    data = response.json()
```

```

if 'error' in data:
    print('Помилка:', data['error'])
else:
    print('Історичні дані цін Bitcoin (BTC):')
    for item in data:
        print(item[start], '-', item[end], ':', item[price_close])
else:
    print('Помилка при виконанні запиту:', response.status_code))

```

Відповідь яка була отримана:

Історичні дані цін Bitcoin (BTC):

```

2022-06-01T00:00:00.0000000Z - 2022-06-02T00:00:00.0000000Z :
37500.0
2022-06-02T00:00:00.0000000Z - 2022-06-03T00:00:00.0000000Z :
38000.0
2022-06-03T00:00:00.0000000Z - 2022-06-04T00:00:00.0000000Z :
38500.0

```

Проаналізувавши наведений запит для отримання історичних курсів для певної криптовалюти можна зробити наступний висновок: данні які було отримано досить не інформативні тому вони потребують додаткового опрацювання та нормалізації з подальшими модифікаціями.

За допомогою вибраного API створено програмне забезпечення, які регулярно запитуватимуть API для отримання поточних даних. Програмне забезпечення також передбачає заплановані запити на отримання нової інформації в певний період часу (наприклад, щохвилини або щогодини) залежно від обраних налаштувань, щоб зберегти інформацію у відповідному форматі у базі даних.

Мова програмування Python і пов'язані з нею бібліотеки (pandas, plot та інші) будуть використані для обробки, очищення та аналізу зібраних даних. Python також використовуватиметься для розрахунку статистичних

показників, створення графіків, щоб мати більш зручну взаємодію з інформацією, яку надає система.

2.2. Інтегровані методи аналізу та прогнозування ризиків у криптовалютному ринку

Основний виклик у прогнозуванні ризиків на криптовалютному ринку полягає в необхідності адаптації до його особливостей, таких як висока волатильність та непостійність ринкових трендів. Традиційні моделі прогнозування, такі як фундаментальний або технічний аналіз, часто виявляються недостатньо гнучкими аби ефективно працювати з цими особливостями.

Отже, потрібен новий підхід, який дасть можливість швидко обробляти та аналізувати великі обсяги даних, передбачати та прогнозувати ринкові зміни. Такий підхід включає в себе не лише збір даних, але й їх аналіз для виявлення потенційних ризиків, які враховують реальні умови криптовалютного ринку.

ARIMA

В аналізі та прогнозуванні ризиків операцій з криптовалютами використовується комбінований підхід, який інтегрує кілька моделей і методів. Ключовим елементом цього підходу є ARIMA, класичний статистичний метод, що використовує авторегресію, диференціювання та ковзне середнє для аналізу та прогнозування часових рядів.

Метод базується на ідентифікації та використанні зв'язків між різними точками часового ряду для прогнозування майбутніх значень. $ARIMA(p, d, q)$, де p – порядок компоненти авторегресії, d – порядок різниць для досягнення стаціонарності, і q – порядок ковзного середнього [13]. Метод може бути представлений також як комбінація трьох основних компонентів: авторегресія (AR), інтеграція (I), ковзне середнє (MA).

Авторегресія (AR) – це компонент p , що вказує на кількість затримок минулих значень, які використовуються в моделі. Формула для AR виглядає так:

$$Y_t = \varphi_1 Y_{t-1} + \dots + \varphi_p Y_{t-p} + \epsilon_t,$$

де ϵ_t є поточним значенням, φ – коефіцієнти авто регресії, а ϵ_t – термін помилки.

Термін помилки у статистичних моделях, особливо в регресійному аналізі, представляє невідому або невраховану змінну в моделі. Він також може відображати будь-яку випадковість або невизначеність у даних. Тобто, термін помилки відображає відхилення фактичних значень від тих, що прогнозуються моделлю.

Інтеграція (I) – компонент d відповідає порядку різниць, необхідних для досягнення стаціонарності часового ряду. Це може бути виражено як:

$$\nabla^d Y_t,$$

де ∇ є оператором різниці.

Ковзне середнє (MA) – компонент q представляє ковзне середнє і вказує на кількість затримок помилкових термінів у моделі. У контексті статистичного моделювання, помилкові терміни відносяться до різниці між спостережуваними значеннями залежної змінної та значеннями, прогнозованими моделлю. Також помилкові терміни використовуються для оцінки точності моделі та її здатності відтворювати спостережувані дані. Формула для MA може бути представлена як:

$$Y_t = \mu + \epsilon_t + \theta_1 \epsilon_{t-1} + \dots + \theta_q \epsilon_{t-q},$$

де μ – середнє значення ряду, а θ – коефіцієнти ковзного середнього.

У підсумку, ARIMA ефективно визначає та коригує тренди в історичних даних, але цей метод має свої обмеження. Обмеження методу ARIMA в контексті криптовалютного ринку впливають з його нездатності реагувати на високу волатильність та раптові зміни. Але цей метод найкраще підходить для даних із стабільними трендами та сезонністю які

теж зустрічаються на крипторинку. Ці обмеження компенсуються інтегруванням LSTM (Long Short-Term Memory) моделі.

LSTM

Архітектура LSTM яка є варіантом рекурентних нейронних мереж, розроблена для ефективного вирішення проблем з довготривалими залежностями в часових рядах [14]. Вона відрізняється унікальною структурою, що включає модулі пам'яті з інтегрованими спеціальними вентилями: вхідним, забуваючим та вихідним.

Вентилі керують потоком інформації всередині клітини та між клітинами. Вхідний вентиль визначає, яку нову інформацію слід додати до стану клітини. Забуваючий вентиль вирішує, які дані слід забути або відкинути зі стану клітини. Вихідний вентиль контролює, яка інформація зі стану клітини має бути передана до наступного шару мережі або як вихід з LSTM.

У контексті LSTM, "клітина" відноситься до фундаментального елемента в архітектурі цих мереж. Клітина в LSTM представлена як "запам'ятовувальний блок", який зберігає інформацію протягом певного періоду часу. Основна функція клітини полягає в здатності підтримувати стан (або інформацію) на протязі тривалого періоду, що є ключовим для розуміння довготривалих залежностей у часових рядах або послідовностях даних.

Клітина LSTM складається з декількох компонентів, які разом керують процесом зберігання та модифікації інформації. Ці компоненти включають стан клітини та вентилі.

Стан клітини представлений як основний компонент, який зберігає інформацію. Стан клітини здатний транспортувати важливі дані протягом усього процесу обробки послідовності.

Оновлення стану клітини відбувається за формулою:

$$c_t = f_t \times c_{t-1} + i_t \times \tilde{c}_t,$$

де $\tilde{c}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c)$ є кандидатом на новий стан клітини, з вагами W_c та U_c та зміщенням b_c . Вихід h_t з моделі LSTM розраховується як $h_t = o_t \times \tanh(c_t)$.

Однією з ключових переваг LSTM є її здатність уникати проблем градієнтного затухання чи вибуху, завдяки своїм унікальним вентилям, які регулюють потік інформації. Ця властивість робить LSTM ефективною для завдань, де потрібно врахувати довгострокові залежності під час прогнозування часових рядів.

У процесі аналізу та прогнозування курсу і визначення ризиків, модель LSTM відіграє важливу роль, однак вона також має деякі обмеження, які потребують компенсації за допомогою інших методів. Основними обмеженнями LSTM є висока складність обчислень та складнощі в роботі з даними, які містять високий рівень шуму або аномалій через непередбачувані скачки в ціні крипто валюти [15]. Ця модель, хоч і ефективна у виявленні довготривалих залежностей у часових рядах, може виявитися не настільки ефективною при швидкому обробленні великих обсягів таких проблемних даних. Для компенсації цих обмежень LSTM, було прийнято рішення інтегрувати такі методи як Random Forest та Gradient Boosting.

Random Forest

Random Forest є прикладом ансамблевого методу, який включає множину дерев рішень. Кожне дерево навчається на відмінному наборі даних, і їх прогнози об'єднуються для досягнення загального результату. Ця методика знижує ризик перетренування, яке може виникати при використанні одного дерева рішень, і підвищує точність прогнозування, взявши середнє з усіх дерев рішень.

Це доповнює потенціал LSTM, особливо в ситуаціях, де необхідно робити прогнози на основі проблемних даних [16].

$$Y = \frac{1}{n} \sum_{i=1}^n y_i,$$

де n = кількість дерев [16].

Gradient Boosting

Gradient Boosting, на відміну від попереднього, підходить до задачі послідовно, де кожне нове дерево виправляє помилки попередніх. Це дозволяє моделі покращуватись з кожним кроком, враховуючи помилки та слабкі сторони попередніх дерев, і формує кінцевий прогноз як вагову суму прогнозів кожного дерева. Тобто це послідовно побудовані дерева рішень, де кожне наступне дерево намагається виправити помилки попереднього [17]. Кінцевий прогноз є ваговою сумою прогнозів кожного дерева. Ця послідовна корекція робить метод особливо ефективним у випадках, де LSTM може не враховувати всі відхилення у даних

$$F(x) = \sum_{i=1}^N \gamma_i h_i(x),$$

де $h_i(x)$ – прогноз i -го дерева, а γ_i – вага цього прогнозу [17].

Таким чином, комбінація LSTM з Random Forest та Gradient Boosting створює комплексний підхід, де сильні сторони кожного методу використовуються для вдосконалення загальної точності та надійності алгоритму прогнозування. Використання цієї комбінації методів дозволяє ефективно аналізувати та прогнозувати складні часові ряди криптовалютного ринку, компенсуючи обмеження одиначної моделі.

CNN

Інтеграція конволюційних нейронних мереж (CNN) у загальну стратегію аналізу та прогнозування ризиків пов'язаних з криптовалютами, розширює та поглиблює аналітичний потенціал наявного набору інструментів, що включає ARIMA, LSTM, Random Forest та Gradient

Boosting, забезпечуючи додаткові можливості для адаптивного та ефективного прогнозування.

На відміну від LSTM та ARIMA, які оптимально підходять для аналізу часових рядів, CNN ефективно обробляє просторові залежності в даних. Це означає, що CNN може розпізнавати складні шаблони та особливості в структурованих даних, що є особливо корисним для аналізу складних наборів даних.

У CNN, процес оновлення ваг між нейронами відбувається через механізм градієнтного спуску, де ваги кожного зв'язку адаптуються шляхом ітеративного уточнення. Це здійснюється з метою мінімізації функції втрат, яка вимірює відхилення прогнозованих результатів від фактичних. Швидкість навчання регулює інтенсивність цих оновлень, дозволяючи системі ефективно навчатися та оптимізувати свою точність на основі навчальних даних.

$$S(x) = \sum_{i=1}^n \omega_i p_i(x),$$

де J – функція втрат, а α – швидкість навчання [18].

CNN також має свої обмеження. Одним із них є відносно висока складність обчислень, особливо при роботі з великими наборами даних. Крім того, CNN може бути менш ефективною при роботі з даними, що мають високий рівень шуму, або коли необхідно аналізувати довготривалі часові залежності.

Для компенсації цих обмежень, CNN інтегрується з LSTM та Random Forest. LSTM доповнює CNN, ефективно обробляючи довготривалі залежності в часових рядах, тоді як Random Forest може ефективно обробляти шумні дані, що допомагає зменшити вплив шуму на аналіз CNN. Таким чином, інтеграція CNN з названими моделями та методами дозволяє створити більш гнучку та потужну систему прогнозування, яка використовує сильні сторони кожного підходу для автоматизованої оптимізації точності та надійності аналізу даних на криптовалютному ринку.

Стекінг

Стекінг являє собою розширений метод ансамблевого машинного навчання, що інтегрує прогнози від моделей для формування узагальненого прогнозу з підвищеною точністю. Цей метод базується на концепції мета-моделювання, де вихідні дані з первинних моделей навчання використовуються як вхідні дані для другорядної моделі, яка здійснює кінцеве прогнозування. Це робиться шляхом призначення ваги кожному прогнозу від окремих моделей, створюючи композитний прогноз.

$$S(x) = \sum_{i=1}^n \omega_i p_i(x),$$

де $p_i(x)$ – прогноз i -ї моделі, а ω_i – вага цього прогнозу [19].

Value-at-Risk (VaR) та Expected Shortfall (ES)

Value-at-Risk (VaR) та Expected Shortfall (ES) є ключовими метриками у визначенні ризиків під час операцій з криптовалютами. Обидві ці метрики надають інформацію про потенційні втрати, але кожна з них використовується для оцінки ризику з різних перспектив [20].

Value-at-Risk, це метрика, що використовується для оцінки максимальної втрати, протягом певного періоду часу з заданою ймовірністю яка представляє собою рівень довіри. VaR розглядається як порогове значення, нижче якого лежить заданий відсоток втрат певної криптовалюти.

У контексті фінансового ризик-менеджменту, "втрати" визначаються як негативні фінансові зміни вартості активів, які можуть виникнути в результаті ринкових умов. Це дає чітке уявлення про потенційні ризики, з якими можна зіткнутися. Формула обчислення має такий вигляд:

$$V_{\alpha}R_{\alpha}(x) = F^{-1}(1 - \alpha),$$

де F^{-1} – обернена функція розподілу доходності портфеля. X , α – рівень довіри [20].

Expected Shortfall (ES) це метрика, також відома як Conditional VaR, виходить за рамки VaR, оцінюючи середній розмір втрат у ситуаціях, коли фактичні втрати перевищують розрахунковий VaR. ES дає глибше

уявлення про потенційні втрати, вказуючи на можливу величину втрат в найгірших сценаріях [21]. Його обчислення можна представити як:

$$ES_{\alpha}(x) = \frac{1}{\alpha} \int_0^{\alpha} V_{\alpha}R_{\alpha}(x) d\gamma,$$

де $V_{\alpha}R_{\alpha}(x)$ VaR на рівні γ , α – рівень довіри [21].

У контексті використання ARIMA, LSTM, Random Forest, Gradient Boosting та стекінгу, ці метрики відіграють роль інструментів оцінки ризиків, що впливають з прогнозів, сформованих за допомогою цих моделей та методів. Моделі, як ARIMA та LSTM, ефективні в аналізі часових рядів і можуть допомогти в прогнозуванні змін у криптовалютних ринках, від яких залежить VaR та ES. Random Forest та Gradient Boosting, будучи ансамблевими методами, дозволяють підвищити точність та надійність прогнозів, що впливає на точність розрахунку VaR та ES. Стекінг, в свою чергу, інтегрує висновки з цих різних моделей, створюючи більш точний та надійний прогноз, який використовується для оцінки ризиків за допомогою VaR та ES.

2.3. Комбінований спосіб прогнозування курсу та ризиків криптовалюти на основі експертного аналізу та стекінгу моделей

Для прогнозування ризиків операцій з крипто валютами прийнято рішення застосувати комбінований спосіб прогнозування за допомогою експертного аналізу та стекінгу моделей. Поєднання експертного аналізу разом із стекінгом є відносно новим напрямом. Він не лише надає глибше розуміння стабільності ринкових даних за допомогою визначених ризиків, але й виявляє складні залежності та закономірності цих даних демонструючи кращі результати ніж традиційні способи такі як технічний аналіз.

Експертний аналіз даних включає кілька ключових аспектів, таких як стаціонарність, сезонність, автокореляція та аналіз волатильності. Це не

тільки визначає внутрішні закономірності в даних, але й оцінює вплив різних факторів на коливання курсу криптовалюти. Наприклад, аналіз стаціонарності (з використанням тестів ADF і KPSS) дозволяє визначити, чи має часовий ряд постійне середнє значення та дисперсію, що важливо для прогнозування ризиків.

Стекінг моделей в свою чергу дозволяє поєднувати переваги різних моделей прогнозування, для досягнення вищої точності за допомогою різних методів для різних аспектів даних, таких як тренд, сезонність або шум. Застосування перехресної перевірки за допомогою методу TimeSeriesSplit допомагає оцінити ефективність кожної моделі окремо, а потім вибрати найкращу комбінацію для кінцевого прогнозу.

Для оцінки ефективності моделі використовуються різні показники. Розгляд цих показників потрібно почати з середньої квадратичної помилки (MSE). MSE визначає середнє відхилення прогнозованих значень від фактичних, що дозволяє оцінити точність та ефективність моделі. Тобто високі значення MSE вказують на нижчу точність прогнозування моделі, що є вирішальним для точності прогнозу та визначення ризиків під час операцій з криптовалютою.

Таким чином, розроблений алгоритм аналізує результати кожної моделі за допомогою MSE і визначає ті моделі, які працюють погано для окремого набору даних. Це дає змогу виключати неефективні моделі з подальшого аналізу та використовувати лише ті, які забезпечують найбільш точні та надійні прогнози.

Процес експертного аналізу, стекінгу та кінцевого прогнозу для визначення ризиків можна розділили на чотири основні етапи:

- 1) експертний аналіз історичних даних;
- 2) крос-валідація моделей;
- 3) визначення найкращих комбінацій моделей;
- 4) стекінг моделей та отримання кінцевого прогнозу;

Експертний аналіз історичних даних охоплює кілька ключових елементів: аналіз стаціонарності, аналіз сезонності, аналіз автокореляції та аналіз волатильності.

Під час аналізу стаціонарності використовуються розширені тести Дікі-Фуллера (ADF) і Квятковського-Філіпса-Шмідта-Шина (KPSS). Вони потрібні для визначення того, чи є часовий ряд стаціонарним. Стаціонарність важлива для статистичних моделей прогнозування тому є важливим критерієм оцінки ефективності моделі для такого типу набору даних.

Розширений тест Дікі-Фуллера (ADF):

$$Y_t = a_1 y_{t-1} + a_2 y_{t-2} + \varepsilon_t,$$

де Y_t – часовий ряд, а ε – помилка [22].

Розширений тест Квятковського-Філіпса-Шмідта-Шина (KPSS):

$$KPSS_{statistic} = kpss_test[0]\rho,$$

де ρ – значення $= kpss_test$ [22].

Аналіз сезонності використовує адитивну побудову графіків, щоб визначити сезонні зміни у даних. Під час оцінки сезонної сили, визначається наскільки суттєвим є сезонність і чи вплине вона на подальший прогноз. Аналіз автокореляції в свою чергу використовує функцію автокореляції (ACF) і функцію часткової автокореляції (PACF), щоб визначити залежність даних від минулих значень.

Для аналізу волатильності застосовується тест Левена, він перевіряє однорідність волатильності між максимумами та мінімумами. Це важливо для визначення стабільності та лінійності курсу криптовалюти.

$$H_0: \sigma_1 = \sigma_2 = \dots = \sigma_k \text{ vs } H_1: \sigma_i \neq \sigma_i,$$

принаймні для однієї пари (i, j) .

Наступним етапом є кросвалідація. Вона забезпечує оцінку різних моделей прогнозування за допомогою перехресної перевірки. Оцінка продуктивності моделей на фактичних даних виконується після їх

нормалізації та розділення на навчальні і тестові набори. На цьому кроці використовується Time Series cross-validator. Саме це надає індекси для даних часових рядів, які спостерігаються через певні проміжки часу. У кожному розділі індекси тесту повинні бути більшими, ніж у попередньому. Тобто у контексті розроблюваного алгоритму, у k-му розподілі він повертає перші k згорток як навчальний набір і (k+1)-ту згортку як тестовий набір що в кінцевому результаті допомагає провести швидко але ефективну оцінку ефективності моделей.

Після цього алгоритм розраховує середню квадратичну помилку на основі прогнозів кожної моделі за допомогою фактичних даних, що і є кінцевою оцінкою ефективності моделей – MSE.

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

де y_i – фактичні значення, $\hat{y}_i$ – прогнозовані значення.

Якщо кросвалідація пройшла успішно та середньо квадратична помилка була обчислена для кожної моделі, наступним етапом є вибір найкращої комбінації моделей які підійдуть для прогнозу курсу та ризиків. Алгоритм порівнює середні похибки різних комбінацій моделей та обирає ту, яка мінімізує загальну похибку.

Після завершення процесу прогнозування, наступним кроком є визначення потенційних ризиків. В цьому контексті ризики визначаються як потенційні фінансові втрати, які можуть виникнути через непередбачувані зміни на ринку криптовалют. Для оцінки цих ризиків використовуються методи Value-at-Risk (VaR) та Expected Shortfall (ES), які дозволяють розрахувати максимально можливі втрати з певною ймовірністю.

Важливо відзначити, що ефективність прогностичних моделей має безпосередній вплив на оцінку ризиків. Наприклад, якщо ефективність моделі знижується або стає менш стабільною, це може вказувати на наявність стрибків або нерегулярностей у даних, що збільшує потенційні

ризиків. Зміни в точності та надійності моделі можуть сигналізувати про зміни в ринкових умовах, що, в свою чергу, впливає на розрахунок VaR та ES і, відповідно, на загальну оцінку ризиків операцій з криптовалютою.

Опишімо алгоритм розробленого способу прогнозування ризиків операцій з криптовалютами.

Цей алгоритм може бути розділений на кілька ключових етапів, які виконуються послідовно:

1. Збір даних з відкритого API: описаний алгоритм починається зі збору даних. Цей процес передбачає збір історичних даних, таких як ціни відкриття та закриття, найвищі та найнижчі ціни, обсяг торгів тощо.
2. Попередня обробка та експертний аналіз: після збору даних з API вони нормалізуються та стандартизуються, що допомагає підвищити ефективність алгоритмів машинного навчання та методів що використовуються для прогнозування. Потім дані розбиваються на навчальні та тестові вибірки, що є ключовим кроком для подальшої оцінки ефективності та точності моделі.
3. Перевірка на успішність обробки даних: перевірка відповідності даних, яка перевіряє, чи відповідають зібрані дані формату який потрібен для належної роботи моделей та методів. Якщо дані не відповідають цим критеріям, дані повторно обробляються та нормалізуються, доки не будуть виконані необхідні умови.
4. Крос-валідація моделей. Визначення найкращої комбінації моделей та методів: цей етап оцінює моделі та методи прогнозування за допомогою перехресної перевірки. Оцінка точності моделей на фактичних даних виконується після їх нормалізації та розділення на навчальні і тестові набори.
5. Побудова та навчання комбінації моделей та методів для прогнозування: на цьому етапі використовуються данні які

вже були попередньо нормалізовані та розділені на два набори, тестовий і навчальний.

6. Прогнозування курсу та стекінг моделей: на цьому етапі відбувається прогнозування курсу крипто валюти. Після отримання прогнозу від комбінації моделей та методів відбувається стекінг.
7. Розрахунок ризиків відповідно до отриманого прогнозу: по завершенню прогнозування, наступна фаза полягає у визначенні потенційних ризиків, що розглядаються як можливі фінансові втрати від несподіваних ринкових коливань. Оцінка таких ризиків здійснюється через застосування методів Value-at-Risk та Expected Shortfall для розрахунку максимальних можливих втрат із заданою імовірністю.
8. Надсилання сповіщень про результат прогнозування та ризики: перед остаточним формуванням результатів прогнозування алгоритм надсилає сповіщення про можливі ризики та звіт про прогнозування в якому описані використані методи та точність кожної використаної моделі чи методу.
9. Повернення результатів для відображення: на цьому етапі алгоритм формує прогнозовані данні у JSON який використовується для побудови графіків та таблиць.

2.4. Висновки до розділу

У рамках розробки алгоритму для прогнозування ризиків на ринку криптовалют, було використано комбінований підхід, який об'єднує як традиційні методи прогнозування часових рядів, так і техніки машинного навчання. Для збору даних було використання зовнішніх API, що забезпечило надійність та стабільність отриманих даних.

Для програмного прогнозування ризиків операцій з криптовалютами було прийнято рішення використовувати комбінований спосіб за допомогою експертного аналізу та стекінгу моделей. Вибір цього способу зумовлений його більшою точністю та адаптивністю під час виявлення неочевидних, нелінійних взаємозв'язків та шаблонів у даних курсу криптовалюти.

Використання експертного аналізу дозволяє аналізувати дані на предмет стаціонарності, сезонності, автокореляції та волатильності, що є критично важливим для точного прогнозування. Стекінг моделей, у свою чергу, забезпечує об'єднання прогностичних моделей та методів, з метою покращення загальної точності прогнозів шляхом призначення ваги для моделі чи методу в залежності від прогностичної точності.

Ключові компоненти алгоритму включають використання моделей ARIMA, LSTM, RF, GB та машинне навчання, що дозволяє проводити аналіз та ідентифікацію потенційних ризиків. Ці моделі допомагають визначити та передбачити різні ринкові тенденції та моделі поведінки, включаючи ризики та сезонність, які є критичними для розуміння ринку криптовалют.

Використання таких методів, як VaR (Value at Risk) і ES (Expected Shortfall), оцінює можливість потенційних фінансових втрат та прийняти рішення для мінімізації ризику.

Розроблений спосіб дає можливості для більш точного прогнозування ризиків у динамічному середовищі ринку криптовалют. Цей спосіб дохволяє зменшити невизначеність, що надає можливість інвесторам і аналітикам в складних ринкових умовах приймати обґрунтовані рішення з приводу інвестування.

3. ОСОБЛИВОСТІ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1. Вимоги до програмного забезпечення

Програмне забезпечення має автоматизувати аналіз ринків криптовалют, ідентифікувати потенційні небезпеки та створювати прогнози курсів та ризиків операцій з криптовалютами на основі досліджених даних, тому застосунок має:

- швидко реагувати на запити користувачів і надавати відповіді в режимі реального часу, що забезпечить своєчасну реакцію на потенційну небезпеку крипторинку;
- гарантувати високий ступінь захисту даних для користувачів, використовуючи сучасні методи шифрування та аутентифікації..
- бути адаптивним і здатним корегувати інформацію до змін ринку, що передбачає можливість швидкого внесення змін до аналізу та прогнозів на основі нових даних або змін на ринку.
- співпрацювати з різними криптовалютами, дані з цих транзакцій збиратимуться та оцінюватимуться.
- легко масштабованим і забезпечувати стабільну відповідь на великий обсяг запитів.

Реалізація цих вимог гарантуватиме ефективність, надійність і корисність програмного рішення для користувачів, які використовують криптовалютні операції та шукають надійні засоби прогнозування ризиків. Оскільки розроблюване програмне забезпечення представлене у вигляді вебзастосунку, мова програмування має відповідати таким вимогам:

- широкі функціональні можливості, які надають змогу розроблювати складні вебзастосунки;
- велика кількість відкритих для використання бібліотек;
- постійна підтримка та оновлення;
- висока ефективність.

Застосунок також має давати можливість користувачам налаштування певних системних компонентів, щоб персоналізувати застосунок відповідно до їхніх конкретних потреб і вподобань.

У ситуаціях із високим ризиком або іншими критичними подіями користувачі повинні отримувати негайні сповіщення, щоб своєчасно реагувати на зміни на ринку.

Особиста інформація повинна зберігатися в безпечному середовищі. Система повинна гарантувати конфіденційність і доступність даних користувачів.

У випадках, коли виникають запитання чи проблеми, користувачі повинні мати можливість швидко зв'язатися зі службою підтримки та отримати якісну допомогу.

3.2. Засоби реалізації програмного забезпечення

3.2.1. Вибір мови програмування для модулю прогнозування ризиків та курсу криптовалют

Існує кілька мов програмування які можуть бути обрані для реалізації розроблюваного алгоритму. Серед таких мов Java, C++, і Ruby, кожна з них має свої унікальні переваги та особливості. Java відома своєю ефективністю у великих проєктах, C++ – своєю швидкістю та гнучкістю, а Ruby вирізняється простотою синтаксису. Однак, серед усіх цих мов, Python виокремлюється завдяки своїй гнучкості та широкому спектру застосувань.

Код на Python часто називають «псевдокодом» через його схожість з алгоритмічними інструкціями [23]. Python використовується в широкому діапазоні програм, від веб-розробки (з використанням фреймворків, таких як Django та Flask) до наукових і числових обчислень (з використанням бібліотек, таких як NumPy і SciPy), аналізу даних (Pandas) і машинного навчання (Scikit -learn). Однією з найбільших переваг Python є його широка

екосистема бібліотек і фреймворків, які пропонують готові до використання рішення для різноманітних завдань. Це значно прискорює процес розробки. Ця мова особливо популярна у наукових обчисленнях та аналізі даних завдяки своїм потужним бібліотекам для математичного моделювання, статистики, візуалізації даних і машинного навчання які використані в даній роботі. Що до наукових досліджень, Python також займає лідерство завдяки своїй здатності обробляти великі набори даних і складні обчислення. Він використовує динамічний тип програмування, який дозволяє розробникам писати менше коду, але вимагає більш ретельного тестування, особливо для великих і складних програм.

Через ці особливості Python було обрано для розробки модулів прогнозування ризиків та курсів валют, де важлива здатність ефективно обробляти великі обсяги даних і виконувати складні обчислення.

3.2.2. Вибір мови програмування для розробки серверної та клієнтської частини

В області фронтенд-розробки, існують різні мови та фреймворки, такі як TypeScript, CoffeeScript, та Dart. TypeScript розширює можливості JavaScript, додаючи строгу типізацію та об'єктно-орієнтовані функції, що забезпечують додаткову надійність коду. CoffeeScript пропонує більш зручний синтаксис, який у свою чергу компілюється в JavaScript. Dart, розроблений Google, спрямований на підвищення продуктивності та швидкодії веб-додатків. Проте, JavaScript залишається найпопулярнішим вибором серед розробників, завдяки своїй універсальності, широкій підтримці та здатності до інтеграції з різноманітними фреймворками.

JavaScript – це об'єктно-орієнтована мова скриптів, яка пропонує кросплатформну функціональність і в основному використовується для створення інтерактивних вебзастосунків [24].

На сьогодні JavaScript є ключовим компонентом більшості популярних веб-браузерів. Переважена більшість веб-розробники зазвичай

включають JavaScript у свій стек технологій. Опитування розробників Stack Overflow постійно визначає JavaScript як одну з найпоширеніших мов серед програмістів

Завдяки своїй популярності, JavaScript має широку підтримку, постійно оновлюється та вдосконалюється. Ця мова програмування спеціально орієнтована для веб-розробки. Вона є надзвичайно популярною, постійно розвивається, підтримується та ідеально підходить для розробки клієнтської і серверної частин вебзастосунку.

Її недолік у відсутності строгої типізації, та як наслідок дещо ускладненому процесі знаходження та виправлення помилок компенсується величезною кількістю відкритих бібліотек, які навпаки, пришвидшують процес розробки.

Тому, після порівняння всіх переваг та недоліків інших мов програмування, для реалізації клієнтської та серверної частини вебзастосунку була обрана мова програмування JavaScript.

3.2.3. Вибір фреймворків для розробки серверної та клієнтської частини

Вибір клієнтської структури веб-додатку є ключовим рішенням у процесі розробки, яке безпосередньо впливає на продуктивність, масштабованість і простоту підтримки проєкту. Розглянемо три популярні фреймворки: React/React-Native, Angular і Vue.js, порівняємо їх особливості та визначимо, чому React Native є найкращим вибором для розробки програмних комплексів, орієнтованих на виявлення ризиків.

React/React-Native, створений Facebook, є однією з найпопулярніших бібліотек JavaScript, які використовуються для створення інтерфейсів користувача [25]. Одним з його основних атрибутів є віртуальний DOM, цей атрибут підвищує ефективність додатків шляхом оптимізації процесу оновлення інтерфейсу. Крім того, він відрізняється простою структурою та інтеграцією з іншими бібліотеками та фреймворками, що робить його

ідеальним вибором для розробників, які хочуть мати гнучкість у своїх проєктах. Крім того, велика кількість членів спільноти сприяють тому, що React надає розробникам широкий вибір інструментів і компонентів, необхідних для створення сучасних комплексних веб-додатків.

Angular в свою чергу, розроблений Google, є повноцінною платформою та фреймворком, який використовується для створення веб-додатків. Angular надає більш компактний фреймворк для додатків, який можна використовувати для розробки у модульний та організований спосіб. Цей фреймворк містить кілька корисних функцій, включаючи маршрутизацію, форми та HTTP-клієнт, що зменшує потребу в додаткових бібліотеках. Однак дуже важливо зазначити, що Angular має вищий бар'єр для входу, ніж React і Vue.js, що може вимагати від розробників більшого рівня знань і досвіду.

Vue.js – це прогресивний фреймворк для JavaScript, який має інтуїтивно зрозумілий інтерфейс. Його легкість і гнучкість сприяють швидкому розгортанню додатків на його основі. Підтримка однофайлових компонентів (SFC) разом з легкістю інтеграції та можливістю включати в розробку бібліотеки спільноти розробників, роблять його універсальним для багатьох завдань. Незважаючи на молоді спільноту Vue.js порівняно з React і Angular, вона зростає швидше та привертає увагу розробників завдяки простій архітектурі та порівняно простого використання.

Виходячи з аналізу доступних варіантів, прийнято рішення використовувати React Native для розробки клієнтської частини програмного комплексу для прогнозування ризиків операцій з криптовалютами. Ось чотири основні причини такого вибору:

1. React Native дозволяє створювати нативні мобільні програми для iOS та Android. Враховуючи, що пакет програмного забезпечення зосереджений на виявленні ризиків, мобільна доступність має вирішальне значення. Користувачі зможуть

отримувати актуальну інформацію та сповіщення прямо на свої смартфони.

2. React Native спрощує розробку та підтримку, дозволяючи використовувати майже однаковий код на багатьох платформах замість створення окремих версій для різних платформ.
3. React Native є крос-платформним рішенням, він пропонує високу продуктивність нативної програми завдяки використанню вбудованих нативних компонентів.
4. React Native також має велику й активну спільноту, яка постійно розширює базу знань в якій містяться бібліотеки та інструменти для розробки.

Отже, React Native, потужний інструмент для створення мобільних додатків, який пропонує не тільки високу продуктивність і зручність використання, але також гнучкість і ефективність у розробці та підтримці такого типу застосунків. Це дозволить нам надавати користувачам постійний доступ до найновішої інформації, яка є життєво необхідною для складних процедур, спрямованих на виявлення та аналіз ризиків.

Розглядаючи серверні фреймворки JavaScript, є кілька ключових варіантів, зокрема Express, Кoa та Meteor. Кожен із цих фреймворків має свої унікальні характеристики, які роблять їх хорошим вибором для різних типів проєктів.

Express – один із найпопулярніших фреймворків Node.js. Його головними перевагами є мінімалізм і гнучкість, що дозволяє розробникам легко створювати ефективні веб-додатки та API. Express не нав'язує суворої архітектури, надаючи розробникам свободу вибору найкращих інструментів і практик для своїх проєктів. Він забезпечує базову маршрутизацію, обробку запитів і відповідей, а також підтримує численні проміжні обробники, що дозволяє легко розширити його функціональність [26]. Express також має велику спільноту та багатий

вибір бібліотек і модулів, доступних через NPM, що робить його чудовим вибором для швидкого розгортання проєкту.

Створений тими ж розробниками, що й Express, Коа є сучаснішим фреймворком, розробленим, щоб бути ще більш легким і гнучким. Він використовує сучасні функції JavaScript, такі як `async/await`, щоб покращити читабельність коду та спростити обробку асинхронних операцій. Коа не має вбудованих інструментів маршрутизації чи інших функцій із в базовому варіанті та зосереджується на забезпеченні мінімалістичного, але потужного середовища розробки.

Meteor, ще один фреймворк JavaScript, який надає повний набір інструментів для розробки веб і мобільних додатків. Він забезпечує інтеграцію як з інтерфейсом, так і з сервером, реалізуючи концепцію реактивних програм, де зміни в базі даних автоматично відображаються в інтерфейсі користувача. Meteor також пропонує широкий спектр функціональних можливостей, що є дуже корисним але в одночас складнішим для вивчення, ніж Express чи Коа.

Беручи до уваги аспекти перераховані вище, Express є найкращим вибором для програмного комплексу який спеціалізується на прогнозуванні ризиків операцій з крипто валютами завдяки своїй простоті та гнучкості. Відсутність вимог до архітектури дозволяє розробникам швидко адаптуватися до потреб проєкту та легко інтегрувати його з іншими інструментами. Це робить Express ідеальною платформою та чудовим вибором у даному випадку.

3.2.4. Вибір фреймворку для розробки серверної частини модулю прогнозування ризиків та курсу криптовалюти

Django було обрано як основний фреймворк для розробки веб-додатку і API через ключові переваги, які пропонує цей фреймворк. Як потужна високорівнева структура для Python, Django виділяється своєю

здатністю створювати безпечні, масштабовані та гнучкі веб-додатки. Дуже схожий по своїй суті на Express у випадку JS.

Безпека є однією з ключових переваг Django. Фреймворк надає розробникам широкі вбудовані функції безпеки, які захищають веб-додатки від багатьох відомих уразливостей. Це включає захист від міжсайтового сценарію (XSS), підробки міжсайтового запиту (CSRF), впровадження SQL та інших потенційних загроз. Цей рівень безпеки важливий для будь-якої веб-програми, особливо під час обробки конфіденційних даних.

Масштабованість і гнучкість є іншими важливими аспектами, які пропонує Django. Фреймворк підтримує розробку великих складних веб-додатків, надаючи розробникам потужні інструменти для керування та організації коду [27]. Ці інструменти дозволяють легко масштабувати проєкт у міру зміни потреб і зростання кількості запитів за певний проміжок часу, забезпечуючи високу продуктивність і стабільність програми незалежно від її розміру.

Динамічне створення API є ще однією значною перевагою використання Django, особливо завдяки інтеграції з Django Rest Framework. Інструмент дозволяє розробникам ефективно створювати, тестувати та керувати RESTful API. У сучасному світі веб-розробки API відіграють ключову роль у забезпеченні функціональності та інтеграції веб-додатків, і ця функція є надзвичайно цінною.

Ці функції разом роблять Django чудовим вибором для розробки веб-додатків. Його здатність забезпечувати високий рівень безпеки, масштабованості, гнучкості розробки та ефективного керування API робить структуру ідеальним рішенням для сучасних веб-проєктів, які потребують надійності, продуктивності та простоти обслуговування.

Роль Docker у запуску Python API

Запуск API за допомогою Docker є ключовим компонентом розробки сучасних веб-додатків, особливо в поєднанні з Django. Docker надає важливі переваги, які спрощують розгортання та керування веб-додатками.

Послідовність і ізоляція середовища є одними з найважливіших переваг Docker. Контейнеризація Docker забезпечує уніфіковане середовище для всіх етапів розробки, від локальних до робочих серверів. Це гарантує, що програми працюють однаково в усіх середовищах, значно знижуючи ризики, пов'язані з відмінностями конфігурації [28]. Узгодженість цього середовища має вирішальне значення для запобігання проблемам, викликаним відмінностями в налаштуваннях або версіях залежностей.

Простота розгортання та масштабування є ще однією значною перевагою використання Docker. Контейнеризація, розгортання та масштабування роблять API легким для розгортання. Docker дозволяє легко керувати робочими навантаженнями та ресурсами незалежно від структури API, забезпечуючи гнучкість і швидкість реагування на зміни.

Кожен контейнер містить усі необхідні бібліотеки, необхідні для роботи програми, що забезпечує стабільність та ізоляцію API що безпосередньо використовується в цьому контейнері. Це дозволяє уникнути конфліктів залежностей між різними проєктами та гарантує надійність API.

Таким чином, поєднання Django і Docker для розробки та запуску Python API є стратегічно правильним вибором, який забезпечує безпеку, ефективність і гнучкість для керування веб-додатками.

3.2.5. Вибір системи керування базами даних

Вибір системи керування базами даних (СУБД) є критично важливим рішенням під час розробки веб-додатків, оскільки воно безпосередньо впливає на те, як дані зберігаються, керуються та обробляються. У цьому

випадку дві популярні СУБД – PostgreSQL і MongoDB – часто вважаються потенційним вибором. Пояснення вибору PostgreSQL у контексті цього проєкту вимагає розуміння можливостей обох систем та їх придатності для конкретних завдань.

Для проєктів, які передбачають обробку та аналіз великих обсягів даних, особливо коли дані мають складну структуру або вимагають високого рівня цілісності та надійності, вибір правильної СУБД є вирішальним. База даних повинна підтримувати ефективне управління даними, забезпечувати їх безпеку та дозволяти легко розширювати систему.

PostgreSQL – це об’єктно-реляційна база даних, відома своєю надійністю, гнучкістю та відповідністю стандартам SQL [29]. Однією з його головних переваг є підтримка складних запитів, забезпечення високого рівня безпеки даних. PostgreSQL також підтримує розширені функції, такі як зберігання JSON, геопросторові дані та розширений пошук, що робить його чудовим вибором для систем, які потребують високого ступеня гнучкості обробки даних.

З іншого боку, MongoDB – це документно-орієнтована СУБД NoSQL, оптимізована для швидкого зберігання та отримання даних у JSON-подібному форматі. Ця система забезпечує високу продуктивність при обробці великих обсягів неструктурованих даних і часто використовується в системах, які потребують швидких змін у схемі даних.

Для цього проєкту я вирішив обрати PostgreSQL з кількох причин:

1. PostgreSQL надає потужні механізми для забезпечення цілісності даних, включаючи підтримку ізоляції транзакцій і потужне керування помилками. Це важливо для проєктів, у яких пріоритетом є точність і надійність даних;
2. завдяки розширеним функціям SQL PostgreSQL може реалізувати складні запити та операції аналізу, що є вкрай

важливим для проєктів, які вимагають глибокого аналізу даних;

3. PostgreSQL підтримує масштабованість, необхідну для обробки зростаючих обсягів даних, і надає гнучкість для обробки різних типів даних, включаючи структуровані, напівструктуровані (JSON) і геопросторові дані;
4. наявність активної спільноти та широке використання PostgreSQL комерційними та некомерційними проєктами забезпечує регулярне оновлення, покращення та надійну підтримку.

Вибір PostgreSQL для цього проєкту відображає потребу у безпеці даних, гнучкості обробки та здатності виконувати складні запити та аналізи. Тому цей вибір є правильним та обґрунтованим.

3.3. Архітектура програмного забезпечення

Застосунок був розроблений у вигляді клієнт-серверної архітектури з використанням інструментів які були описані вище. Архітектура клієнт-сервер – це фундаментальний принцип сучасних веб-додатків, який визначає взаємодію між пристроєм користувача (клієнтом) і сервером, який обробляє запити та надсилає відповіді [30]. На зображенні показано діаграму архітектури клієнт-сервер, яка ілюструє взаємодію між різними компонентами розробленої системи (рис 3.1).

Клієнтська частина – це інтерфейс для взаємодії з користувачем. У цьому випадку для розробки клієнта використовується React-Native, популярна бібліотека JavaScript, яка дозволяє створювати крос платформні, динамічні та адаптивні інтерфейси користувача.

Серверна частина обробляє запити від клієнтської частини, виконує логіку програми, доступ до бази даних та інтеграцію із зовнішніми API. У цьому випадку він побудований на Express, легкій і гнучкій структурі Node.js для створення веб-серверів і API.

База даних у цій архітектурі реалізована за допомогою потужної об'єктно-реляційної СУБД PostgreSQL. PostgreSQL було обрано за його надійність, продуктивність, підтримку складних запитів і високий рівень безпеки даних.

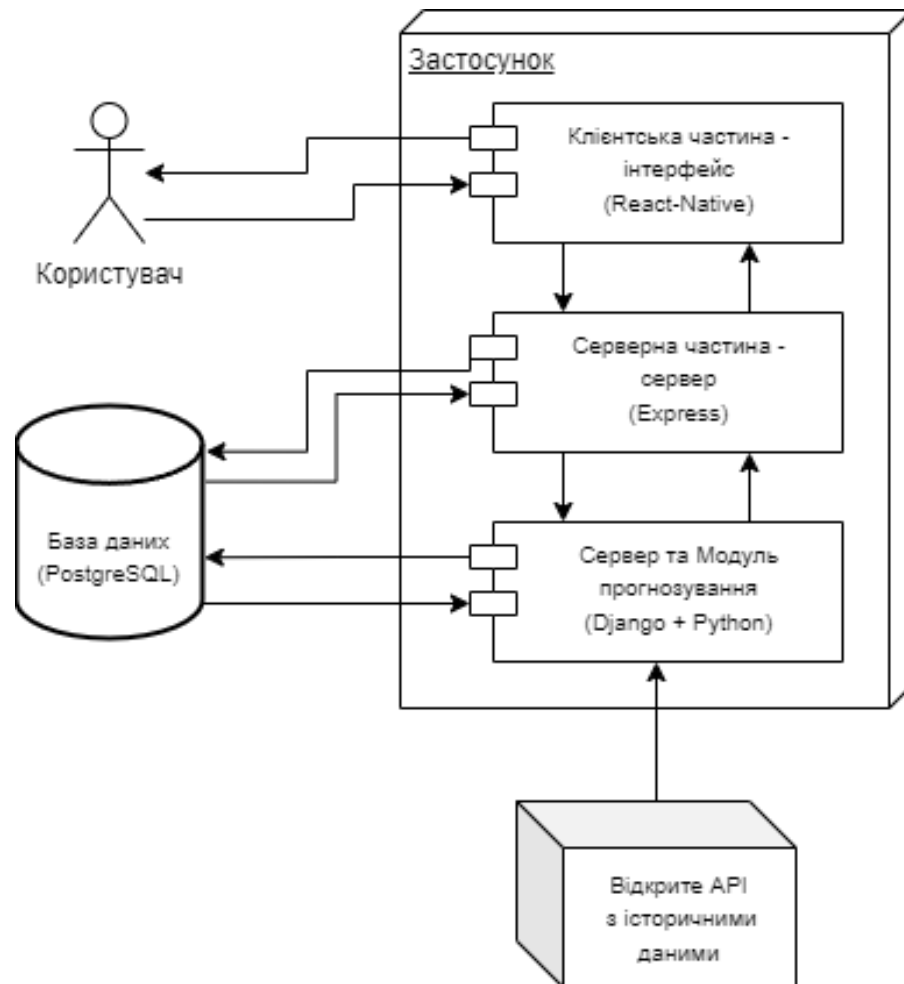


Рис. 3.1. Клієнт-серверна архітектура розробленого застосунку

Сервер і модулі прогнозування створені за допомогою Django і Python. Django використовується для створення потужної логіки на стороні сервера, що забезпечує високу якість веб-додатків і підтримує чистоту коду. Python відіграє важливу роль у складних обчислювальних процесах, таких як аналіз даних і машинне навчання, що робить його ідеальним для розробки модулів прогнозування.

Відкритий API історичних даних інтегровано в систему, щоб забезпечити модуль прогнозування поточною та історичною інформацією, необхідною для аналізу та прогнозування ризиків.

Загалом ця клієнт-серверна архітектура забезпечує міцну основу для ефективної взаємодії між користувачем і сервером, обробки даних, аналітичних запитів і прогнозування. Використання такої архітектури дозволяє створювати надійні, масштабовані та продуктивні веб-застосунки.

Патерн Model-View-Controller (MVC) – це архітектурний шаблон, який поділяє програму на три основні логічні компоненти: модель, представлення та контролер. Такий підхід не тільки спрощує процес розробки, але й полегшує підтримку та розширення системи (рис 3.2) [31].

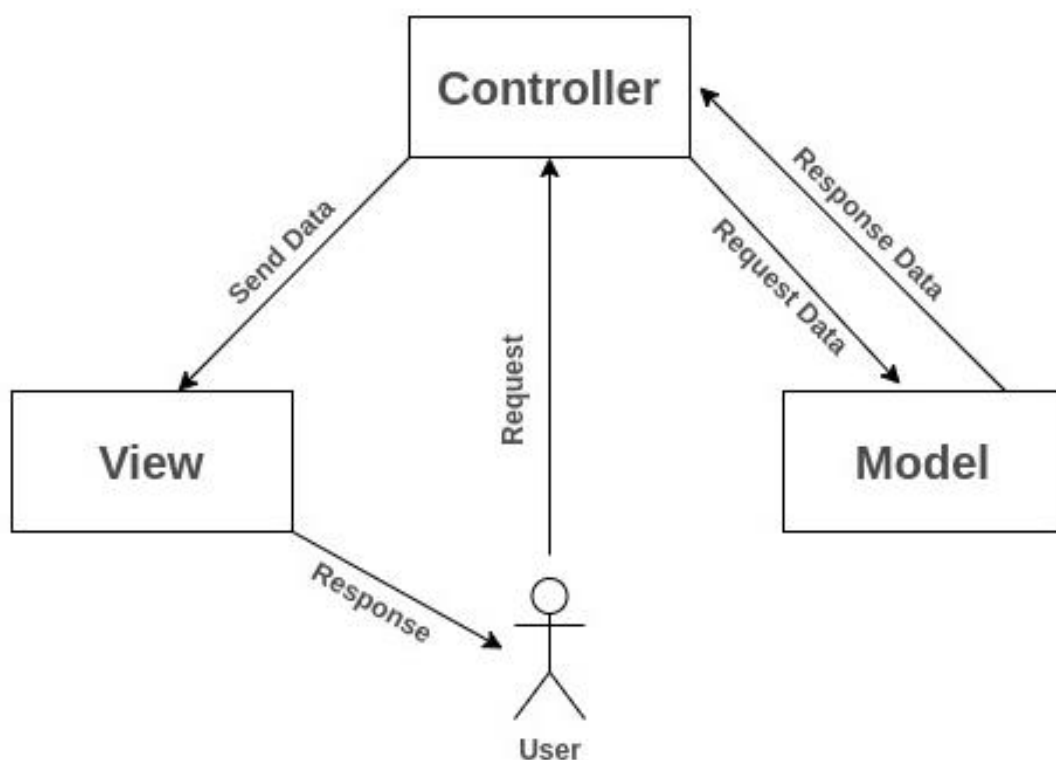


Рис. 3.2. Структура шаблону MVC

Моделі – це компоненти, які представляють структури даних, бізнес-логіку та системні правила. Усі важливі дані та відповідні методи обробки

зосереджені в моделі. Цей блок відповідає за доступ до бази даних, включаючи запити, вставлення, оновлення та видалення записів. У контексті використовуваної архітектури PostgreSQL діє як сховище моделі, забезпечуючи надійне та ефективне керування даними.

Представлення – це частина програми, яка відображає дані користувачеві, тобто інтерфейс, через який користувач сприймає інформацію та взаємодіє з програмою. У контексті даної роботи цей блок використовує React Native, що дозволяє створювати динамічні та візуально зрозумілі інтерфейси для мобільних додатків.

Контролер – це компонент, який приймає дані користувача, перетворює їх за допомогою моделі та відображає в поданні. У цій архітектурі Express відіграє роль контролера, керуючи взаємодією між користувачами та серверами та виконуючи логіку маршрутизації для обробки запитів HTTP та відповіді на них.

Використання шаблону MVC дозволяє чітко розділяти функціональність програмного комплексу, спрощуючи тестування та обслуговування. Моделі можна розробляти та тестувати окремо від інтерфейсу користувача, забезпечуючи вищу якість коду. У той же час ізоляція інтерфейсу дозволяє окремо фокусуватись на UX/UI, не турбуючись про інші частини шаблону.

Для задоволення потреб розроблюваного програмного комплексу було створено 6 основних таблиць та одну додаткову для додаткової інформації (рис 3.3). *User*

Таблиця, що містить у собі інформацію про користувачів які зареєстровані у системі.

Назви всіх полів таблиці та їх відповідні типи даних наведено у таблиці 3.4.

Таблиця 3.4

Таблиця User

<i>Назва поля</i>	<i>Тип даних</i>
id	INTEGER
email	STRING
password	STRING
role	STRING

UserNotifications

Таблиця, що містить у собі всі сповіщення які отримують користувачі. Доступ до сповіщень окремого користувача відбувається за допомогою.

Таблиця 3.5

Таблиця UserNotofication

<i>Назва поля</i>	<i>Тип даних</i>
id	INTEGER
email	STRING
password	STRING
role	STRING

Cryptocurrency

Таблиця, що містить у собі інформацію про крипто валюти з якими може працювати розроблювана система.

Назви всіх полів таблиці та їх відповідні типи даних наведено у таблиці 3.6.

Таблиця 3.6

Поля таблиці Cryptocurrency

<i>Назва поля</i>	<i>Тип даних</i>
id	INTEGER
name	STRING
code	STRING

CryptocurrencyDescription

Таблиця, що містить у собі додаткову інформацію про крипто валюти. До цієї інформації відноситься тип даних, джерело даних, та важлива інформація яка може впливати на результат прогнозу ризиків. Назви полів таблиці та їх відповідні типи даних наведено у таблиці 3.7.

Таблиця 3.7

Поля таблиці CryptocurrencyDescription

<i>Назва поля</i>	<i>Тип даних</i>
id	INTEGER
title	STRING
description	STRING

UserCryptocurrencies

Таблиця, що містить у собі інформацію про крипто валюти які користувач додав до свого інвестиційного портфелю.

Назви всіх полів таблиці та їх відповідні типи даних наведено у таблиці 3.8.

Таблиця 3.8

Поля таблиці UserCryptocurrencies

<i>Назва поля</i>	<i>Тип даних</i>
id	INTEGER
userId	INTEGER
cryptocurrencyId	INTEGER
count	INTEGER

ExchangeRates

Таблиця, що містить у собі інформацію про ціну крипто валюти відповідно до гривні, долара, та євро. Назви всіх полів таблиці та їх відповідні типи даних наведено у таблиці 3.9.

Таблиця 3.9

Поля таблиці ExchangeRates

<i>Назва поля</i>	<i>Тип даних</i>
id	INTEGER
cryptocurrencyId	INTEGER
priceUSD	INTEGER
priceEUR	INTEGER
priceUAH	INTEGER
timestamp	STRING

ForecastData

Таблиця, що містить у собі інформацію про прогнозований курс і відповідно до цього курсу ризику. Назви всіх полів таблиці та їх відповідні типи даних наведено у таблиці 3.10.

Таблиця 3.10

Поля таблиці ForecastData

<i>Назва поля</i>	<i>Тип даних</i>
id	INTEGER
cryptocurrencyId	INTEGER
data	JSON
forecastDate	STRING
cryptoCode	STRING
description	STRING

Модуль прогнозування курсу і ризиків відіграє життєво важливу роль у системі, оскільки він надає клієнту необхідну інформацію для задоволення потреб користувачів у аналізі та інформації. Розроблений модуль прогнозування виконує комплекс функцій, включаючи збір даних, аналіз, обробку та зберігання результатів прогнозування. Розберемо їх більш детально.

Збір даних: процес починається з автоматичного збору найновішої інформації про курси обміну криптовалюти з інтегрованого API. Дані, отримані з цього API, містять потрібну інформацію, включаючи поточні курси валют, обсяги торгів, історичні дані та інші важливі фінансові показники які обробляються та формуються в наступних етапах.

Обробка та аналіз: отримані дані проходять етап попередньої обробки, який нормалізує дані та очищає їх від можливих помилок. Підготовлені дані використовуються для аналізу та створення прогнозних моделей. Цей модуль використовує статистичні алгоритми та алгоритми машинного навчання які були детально описані вище, для оцінки потенційних ризиків, пов'язаних із операціями криптовалют.

Збереження та запис інформації: після обчислення прогнозу результати зберігаються в таблиці ForecastData, де кожен запис представляє окремий прогноз із пов'язаними деталями. Структура таблиці ForecastData дозволяє зберігати не тільки числові дані, але й метадані, яка може включати опис прогнозу, мітку часу прогнозу та ідентифікатор пов'язаної криптовалюти. Основна інформація зберігається у полі "data" у форматі JSON. У цьому JSON всі данні які потрібні для відображення та подальших обрахунків отримує клієнтська частина. Поле "data" у кінцевому результаті має багато інформації та ось такий вигляд:

```
{
  "success": true,
  "response": {
    "historical_data": [
      {
        "price_open": 9236.91,
        "price_high": 9280.0,
        "price_low": 9100.0,
        "price_close": 9256.94,
        "volume_traded": 7497.56309216,
        "trades_count": 61845,
        "RSI": 54.24668326791317,
```

```

    "MACD": -18.11730365989024,
    "Signal": -31.776197507723513,
    "Volatility": 180.0,
    "returns": 0.00216738912971004,
    "cluster": 0,
    "risk": 0.03228164795091376,
    "date": "2023-07-14T00:00:00.0000000Z"
  },
  ...
],
  "forecast": [
    9217.77491439582,
    9566.856883777164,
    9461.344570091776,
    9326.478507239186,
    ...
  ],
  "VaR": -0.041243393285825415,
  "ES": 0.0024771760765162766
}
}

```

Інтеграція з клієнтською частиною: модуль прогнозування тісно інтегрований з клієнтом через сервер на базі Express. Сервер обробляє запит від користувача та звертається до модуля прогнозування для отримання необхідної інформації. Коли користувач запитує дані про курси криптовалют або прогнози, сервер формує відповідні запити до модуля прогнозів. Цей процес передбачає збір прогнозів із бази даних і передачу їх

клієнту React Native через API. Кроки роботи можна побачити на схемі роботи модулю прогнозування (рис 3.11).

Оновлення даних: модуль прогнозування регулярно оновлює інформацію в базі даних, щоб впевнитись, що дані актуальні для користувачів. Це означає, що дані, які користувач бачить у мобільному застосунку, завжди відображають останні доступні прогнози та аналіз. Це постійне оновлення є важливим для забезпечення високоякісного обслуговування, особливо в сфері криптовалют, де ринкові умови швидко змінюються.

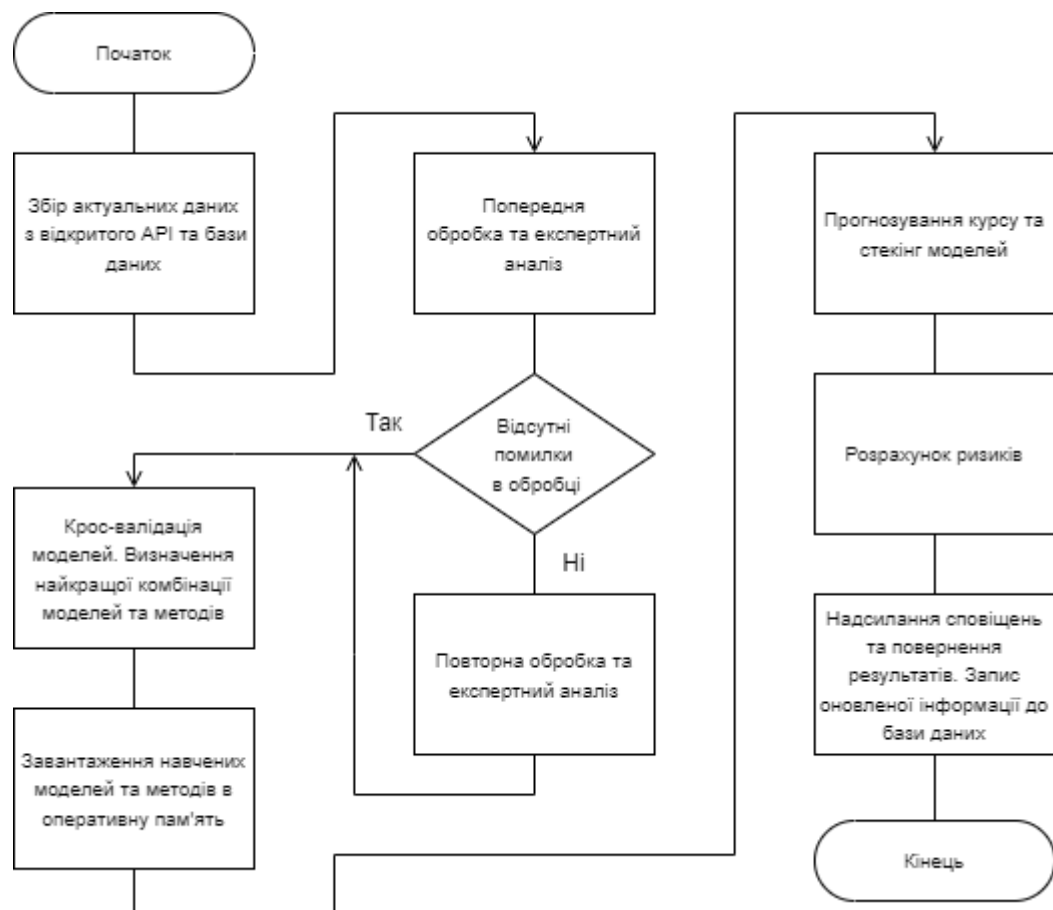


Рис. 3.11. Схема роботи модулю прогнозування

3.4. Взаємодія користувача із застосунком

Зареєстровані користувачі можуть взаємодіяти з веб-додатком у формі кількох веб-сторінок. Тобто після відвідування сторінки реєстрації

або логінку, авторизовані користувачі в системі можуть використовувати такі сторінки:

- Головна сторінка – відображає список підтримуваних системою крипто валют з можливістю сортування по категоріям.
- Сторінка інформації про криптовалюту – сторінка, яка з'являється по кліку на криптовалюту зі списку на головній сторінці. Відображає всю інформацію яка доступна на даний момент про даний криптоактив.
- Сторінка користувача – відображає інформацію баланс крипто портфелю, тобто суму всіх крипто валют та загальний приріст чи збиток який має користувач. Також на сторінці можна знайти список крипто валют що були додані до портфелю. Та список рекомендованих для додавання валют.
- Сторінка детальної інформації та статистики крипто валюти – відображає інформацію та деталі прогнозу у вигляді таблиць і графіків.
- Сторінка реєстрації та авторизації користувача в системі.

Розберемо основний сценарій взаємодії користувача із системою для отримання або оновлення даних прогнозованого курсу та інформації про можливі ризики відповідно до цього курсу (рис 3.12).

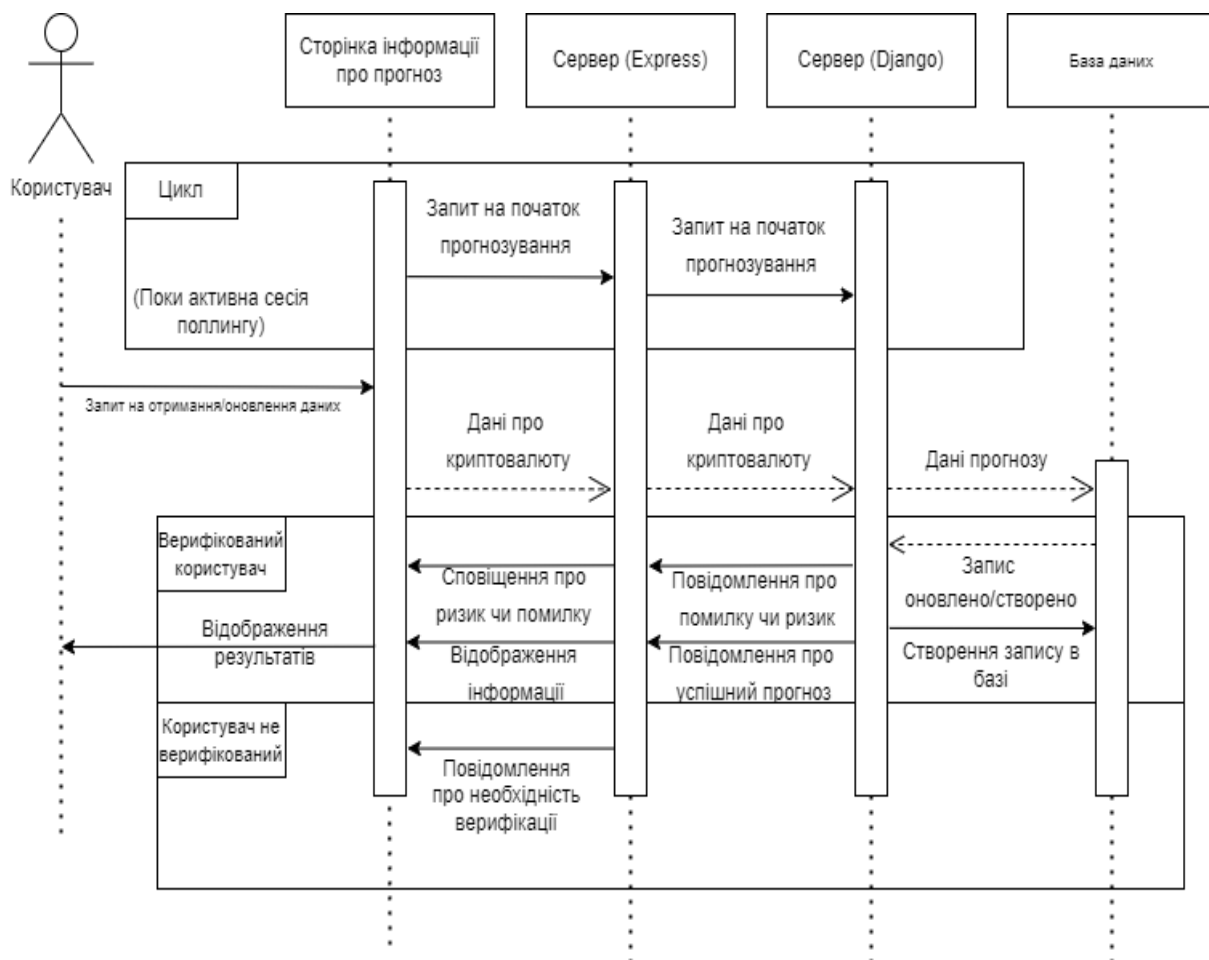


Рис. 3.12. Модель отримання або оновлення даних

На наданому зображенні показано модель отримання та оновлення даних у системі прогнозування ризиків операцій з криптовалютами. Модель представляє собою взаємодію між користувачами, клієнтськими сеансами, серверами та базою даних.

Користувачі починають взаємодіяти з системою, ініціюючи сеанс, який можна захищеним за допомогою автентифікації для підтвердження їх особи. Автентифіковані користувачі мають можливість запитувати інформацію про прогнози курсу та ризиків криптовалюти. Клієнтські запити надсилаються на сервер, створений на базі Express, який обробляє запити на початку прогнозування. Сервер діє як посередник між клієнтом і сервером прогнозів, реалізованим на Django. Сервер Django отримує запити від сервера Express і запускає процес аналізу та прогнозування. Він

збирає дані із відкритого API та ті, що зберігаються в базі даних, для створення прогнозів. Оновлені або нові прогнози зберігаються в таблиці ForecastData бази даних.

База даних діє як сховище для всієї інформації, зібраної та проаналізованої системою, і забезпечує швидкий доступ до цієї інформації, коли це необхідно. Після обробки та оновлення даних результати прогнозу надсилаються назад користувачеві через сервер.

Користувачі можуть переглядати цю інформацію через клієнтський інтерфейс, який відображає дані у легкому для розуміння форматі, тобто за допомогою графіків та таблиць. Користувачі також можуть отримувати додаткові повідомлення щодо ризиків або помилок що виникають під час прогнозу у вигляді сповіщень для яких відведена окрема сторінка. Отже, модель отримання або оновлення даних, відображає циклічний процес збору даних, їх аналізу, збереження результатів і надання цієї інформації користувачеві.

3.5. Висновки до розділу

У цьому розділі розглянуто основні засоби необхідні для створення програмного забезпечення, визначено мови програмування, які потрібні для зовнішніх та внутрішніх компонентів системи. Було проведено аналіз фреймворків, які використовуються для реалізації клієнтської та серверної частин веб-застосунку, разом із системами керування базами даних.

Рішення використовувати JavaScript як мову програмування для веб-додатку, було прийнято через орієнтованість на веб-розробку. Крім того, ця мова поєднує в собі гнучкість та динамічність, що дозволило легко інтегрувати її до проєкту. Доступність ресурсів та зручність стали вирішальною причиною для вибору React-Native як основного фреймворку для клієнтської частини.

Python було обрано як мову для впровадження модуля прогнозування через орієнтованість та доступність для обробки та аналізу великих

наборів даних. Express та Django фреймворки для серверної частини, було повністю інтегровано з PostgreSQL, обраною СУБД, завдяки її здатності відповідати вимогам проєкту щодо простоти, гнучкості, масштабованості та ефективного зберігання великої кількості даних. Цей вибір забезпечив як високу швидкість обробки, так і надійні заходи безпеки.

Використовуючи архітектуру клієнт-сервер, підкріплену шаблоном «Модель-Вид-Контролер», веб-застосунок отримує переваги від чітко визначеної структури та спрощеного керування. Отже, розуміння операційної логіки стає більш простим, а функціональні можливості системи можна легко розширити за допомогою гнучкості.

У структурі бази даних проєкту присутні сім ключових компонентів: шість основних сутностей, що складають основу системи, та одна додаткова сутність, яка потрібна для зберігання даних про криптовалюту. Модуль прогнозування, який був спеціально розроблений для цієї системи, реалізовано таким чином, щоб він міг взаємодіяти з цими сутностями, забезпечуючи швидку інтеграцію та взаємозв'язок у межах бази даних.

Для коректної взаємодії користувача із системою було створено та детально розібрано одну з основних моделей отримання та оновлення даних прогнозу ризиків та курсу криптовалюти.

4. АНАЛІЗ РОЗРОБЛЕНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1. Оцінювання прогнозування ризиків операцій з криптовалютами

Оскільки ризики визначаються за допомогою VaR та ES, ці два показники стануть ключовими критеріями оцінювання ризиків. Порівнявши їх з фактичними курсами криптовалюти, що були зафіксовані в минулому часі на криптовалютному ринку, оцінюється наскільки точним є прогноз цих ризиків. VaR визначає максимальну втрату, яку може зазнати портфель протягом певного періоду часу з заданим рівнем довіри, тоді як ES оцінює середню втрату в умовах, коли втрати перевищують VaR. Що ж до прогнозування курсу, він оцінюється за допомогою метрик контролю точності прогнозу RMSE (Root Mean Square Error) та MAE (Mean Absolute Error). Ці метрики дозволяють оцінити точність прогнозувань курсу моделі відносно фактичних значень. RMSE дає змогу оцінити середньоквадратичне відхилення прогнозів від реальних значень, тоді як MAE вимірює середню абсолютну помилку.

Етапи визначення точності прогнозу курсу і ризиків можна розділити на три основні етапи, а саме:

1. Підготовка даних.
2. Аналіз даних.
3. Формування прогнозу.
4. Оцінка точності прогнозу та ризиків.

На етапі 1. Підготовки даних відбувається форматування історичних даних, очищення від шумів та видалення викидів. Також відбувається нормалізація даних для забезпечення однорідності вхідних даних, що важливо для точності алгоритмів машинного навчання.

На етапі 2. Аналізування даних перед формуванням остаточного прогнозу, аналіз і кросвалідація відіграють ключову роль у визначенні більш точних моделей. Аналіз даних дозволяє ідентифікувати шаблони та

закономірності, що включають стаціонарність, сезонність, автокореляцію, та волатильність. Цей аналіз дає змогу краще зрозуміти ринкові тенденції та їх вплив на курс крипто валюти і визначення ефективності.

Під час кросвалідації, яка виконується за допомогою методу `TimeSeriesSplit`, прогнозовані дані моделей оцінюються порівнюючись з реальними даними. Це дозволяє визначити, які моделі найкраще працюють для конкретних наборів даних, і забезпечує можливість обрати комбінацію моделей та методів для прогнозування. Завершальний крок цього етапу полягає у комбінуванні обраних моделей через стекінг, що дає змогу використовувати переваги кожної моделі для підвищення точності та надійності кінцевого прогнозу. Цей підхід дозволяє не лише підвищити якість прогнозування курсу, але й забезпечити більш точне визначення ризиків, пов'язаних з криптовалютними операціями.

На етапі 3. Формування прогнозу дані передаються до попередньо навченої моделі або комбінації моделей для визначення прогнозу курсу та ризиків. Після успішного прогнозу данні зберігаються у форматі JSON для оцінки точності.

На етапі 4. У процесі оцінки точності прогнозу курсу та ризиків алгоритм послідовно використовує кілька ключових методів. На першому етапі, застосовуються метрики RMSE та MAE для вимірювання точності прогнозу курсу. RMSE, або Root Mean Square Error, визначає середнє квадратичне відхилення прогнозованих значень від реальних, тоді як MAE, або Mean Absolute Error, дозволяє оцінити середнє абсолютне відхилення.

Аналіз ризиків передбачає порівняння прогнозованих ризиків з фактичними втратами, що відбулися в минулому. Це надає інформацію про точність прогнозування цих ризиків що також враховується під час обчислення процентної точності прогнозу комбінованого способу.

Для оцінки швидкості роботи алгоритму використана бібліотека `time`, яка вимірює час, необхідний для вирахування прогнозу.

Для оцінки точності та адаптивності використовується базові моделі на основі ARIMA та LSTM. ARIMA та LSTM були обрані для порівняння з комбінованим методом через їхню адаптивність і ефективність у аналізі часових рядів. ARIMA більш точно враховує автокореляцію, тенденції і сезонність, тоді як LSTM здатна виявляти довгострокові залежності в даних.

4.2. Порівняльний аналіз способів прогнозування ризиків операцій з криптовалютами

Всі обчислені результати аналізу разом із оцінкою точності базової моделі та комбінованого способу окрім часу, зберігаються для обрахунку процентної точності (рис. 4.2).

Цілісний підхід до оцінки точності прогнозу надає нам можливість виміряти точність прогнозу, а й зрозуміти, як моделі працюють із різними типами даних. Результати порівняння точності комбінованого способу та базової моделі наведено у таблиці 4.1.

Таблиця 4.1

Порівняння точності комбінованого способу та базової моделі

Довжина прогнозу, днів	Комбінований спосіб прогнозування		Прогнозування на базовій моделі	
	Точність прогнозу, %	Час, секунди	Точність прогнозу, %	Час, секунди
7	87,44	3,12	73,47	1,68
10	86,72	3,48	71,19	1,82
14	84,30	3,64	70,73	1,92
18	80,91	3,72	67,65	2,24
24	76,21	4,06	62,03	2,41
30	74,37	4,24	59,94	2,64
40	69,51	4,52	56,11	2,86
55	67,92	4,64	54,26	3,18

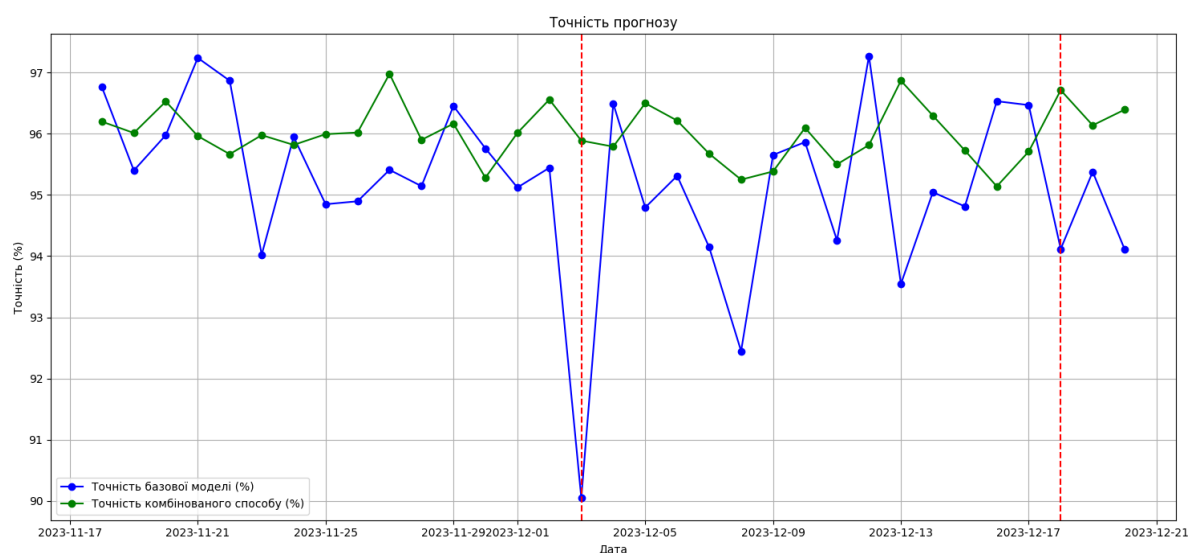


Рис. 4.2. Графік точності прогнозу для комбінованого способу та базової моделі

Як видно з графіків (рис. 4.2, рис. 4.3) та таблиці 4.1 комбінований спосіб демонструє вищу точність на всіх проміжках довжини прогнозу порівняно з базовою моделлю. Так при прогнозі на 7 днів комбінований метод досягає точності 87,44%, що значно перевищує 73,47% базової моделі. Ця тенденція точності у комбінованого методу зберігається навіть зі збільшенням терміну прогнозу, хоча загальна точність обох моделей знижується. Зокрема, при прогнозі на 55 днів комбінований метод все ще має вищу точність – 67,92% порівняно з 54,26% у базової моделі.

Зі збільшенням періоду прогнозування точність обох методів зменшується, але спад точності у комбінованого способу менш стрімкий. Наприклад, точність знижується з 87,44% до 67,92% для комбінованого способу, в той час як у базової моделі падіння більш різке – з 73,47% до 54,26%, якщо розглянути проміжок від 7-денного до 55-денного прогнозу. Це вказує на те, що комбінований метод краще адаптується до потенційних змін динаміки ринку в довгостроковій перспективі.

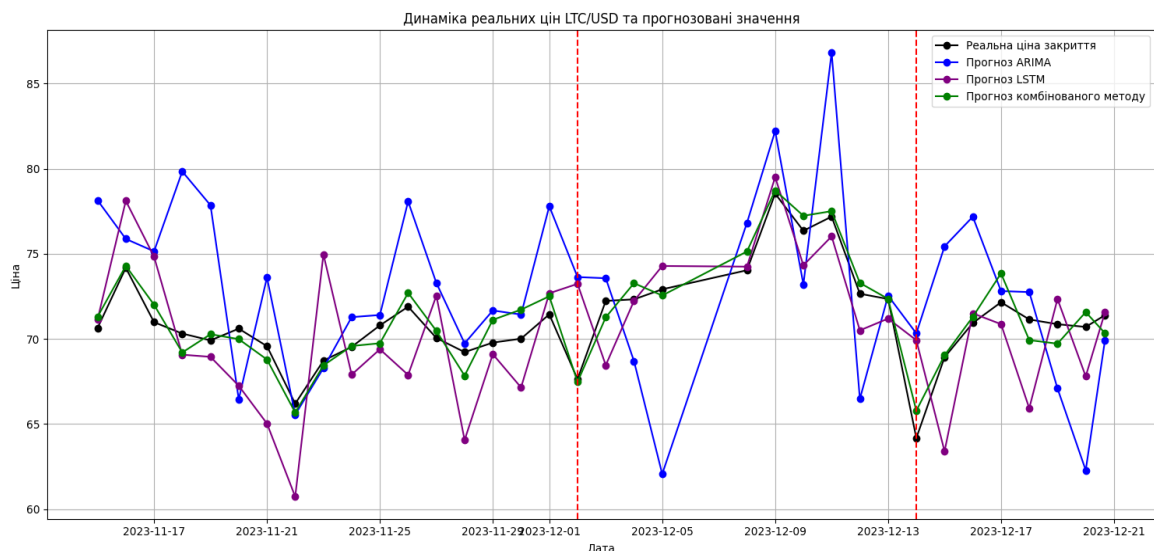


Рис. 4.3. Динаміка реальних цін LTC/USD та прогнозовані значення курсу

Час виконання для комбінованого методу вищий у порівнянні з базовою моделлю, що є очікуваним, адже комбінований метод включає

додаткові кроки глибокого аналізу та кросвалідації для знаходження найкращої комбінації прогнозу. Для 7-денного прогнозу комбінований метод займає 3,12 секунд, тоді як базова модель потребує лише 1,68 секунди. При 55-денному прогнозі час комбінованого методу зростає до 4,64 секунд, з 3,18 секунд для базової моделі.

У середньому, час прогнозування комбінованого способу складає близько 3-5 секунд, що для прогнозування ризиків для криптовалютних операцій, є цілком прийнятним. Ринкові коливання, які є визначальними для купівлі чи продажу активу, зазвичай розвиваються протягом значно більших часових відрізків – від кількох годин до декількох днів. У випадках, коли рішення про покупку або продаж криптовалюти базуються на даних прогнозах, точність є ключовим фактором для мінімізації ризиків.

Графік (рис. 4.3) додатково ілюструє вищу стабільність комбінованого методу, показуючи більшу точність прогнозування протягом часу без різких падінь. Це свідчить про те, що комбінований спосіб дозволяє алгоритму краще адаптуватися до ризикових ситуацій, зберігаючи вищий рівень точності. В свою чергу, ARIMA та LSTM показують значні спади в точності, що вказує на вразливість до раптових змін на ринку. Ці спади точності на графіку є критичними у ситуаціях, де необхідна швидка реакція на ризики, і де помилки у прогнозуванні можуть призвести до значних фінансових втрат.

Комбінований метод має точніші результати порівняно з ARIMA на 6% і LSTM на 4% , з середнім поліпшенням на 5%, що свідчить про його більшу надійність та адаптивність до різних умов ринку. Особливо важливим є той факт, що у випадках різкого падіння цін, метод показує ще більшу точність. Точність прогнозування курсу та ризиків в ситуаціях різкого падіння ціни точніше на 9% порівняно з ARIMA і LSTM . Це вказує на його здатність ефективно прогнозувати нестабільні та непередбачувані ринкові рухи.

Отже, комбінований спосіб незважаючи на довший час виконання має значні переваги у стабільності, точності та адаптивності, що робить його кращим вибором для прогнозування та аналізу ризиків під час операцій з криптовалютами.

4.3. Тестування та аналіз розробленого програмного забезпечення

Тестування розробленої системи можна розділити на частини:

- тестування клієнтської частини;
- тестування серверної частини (Node.js сервер);
- тестування роботи модулю прогнозування (Django сервер).

Клієнтська частина реалізована на основі React Native. Вона пройшла повне тестування для забезпечення стабільності та надійності. Автоматизоване тестування охоплює весь код застосунку, код взаємодії базою даних і сервером, для чого було використано:

- Unit Tests: перевірка окремих функцій та компонентів;
- UI Tests: валідація користувацького інтерфейсу та його відповідності до очікувань користувачів.

Використання Unit Tests дозволило детально оцінити кожен індивідуальну функцію та компонент, зосереджуючись на їх правильній роботі в ізоляції від інших частин системи. Це забезпечило точність і надійність функціональних блоків програми ще на етапі розробки. Результатом такого тестування є автоматична валідація вхідних даних що забезпечило стабільність роботи клієнтської частини.

Тестування інтерфейсу було ручним, воно включало в себе тестування візуальних елементів, навігації та інтерактивності програми, щоб забезпечити швидкий і безпомилковий досвід роботи з застосунком для користувачів.

Серверна частина проєкту, тобто два окремих модулі розроблені на основі Node.js та Django, також пройшли ретельне тестування, щоб забезпечити надійність та безпеку системи. Для Node.js-сервера, основний акцент був зроблений на тестуванні API-інтерфейсів, забезпечуючи, що всі запити та відповіді обробляються коректно. Це включало перевірку коректності вхідних даних, а також валідацію відповідей сервера на різноманітні запити, включаючи помилкові або нестандартні сценарії. Такий підхід забезпечив високий рівень стійкості API до потенційних помилок.

Сервер на Django, що відповідає за модуль прогнозування, тестувався вручну, з акцентом на перевірку стресостійкості та ефективності при обробці різних даних. Це включало аналіз реакції системи на зміну даних, які не підходять для прогнозування, що дозволило оцінити гнучкість та адаптивність алгоритмів. Такий підхід дозволив виявити потенційні слабкі місця в алгоритмах та оптимізувати їх для кращої продуктивності.

Обидва сервери також пройшли перевірку на здатність ефективно обробляти велику кількість запитів, забезпечуючи стабільну та швидку відповідь на запити користувачів. Це було важливо для забезпечення того, що система здатна витримувати високе навантаження, особливо в умовах реального часу, що є критичним для роботи з крипто валютними та прогнозування ризиків.

За допомогою тестування, знайдені помилки програмного забезпечення були виправлені. Отже в результаті веб-застосунок:

- автоматично збирає актуальні дані про курси криптовалют та готує їх для подальшої роботи;
- використовує підготовані данні для глибокого аналізу та визначення найефективнішої комбінації моделей;

- на основі визначеної комбінації моделей робить прогноз курсу криптовалюти комбінуючи результати моделей;
- на основі прогнозованого курсу автоматично прогнозує та обчислює можливі ризики які пов'язані з криптовалютою.
- надає результати прогнозованого курсу та ризиків за допомогою графіків, таблиць і сповіщень;

4.4. Вдосконалення розробленого програмного забезпечення

Вдосконалення розробленого програмного забезпечення є ключовим аспектом для підтримання його конкурентоспроможності на ринку та релевантності отриманих прогнозів. Незважаючи на високу якість і функціональність поточної версії, існує безліч способів для його удосконалення, які дозволять відповідати зростаючим вимогам користувачів і технологічним трендам.

Перше, що варто розглянути, це оновлення алгоритмів прогнозування. Криптовалюти постійно розвиваються, і алгоритми, які були ефективними вчора, можуть втратити свою точність сьогодні. Тому розроблене програмне забезпечення потребує постійного оновлення.

Другим напрямком може бути інтеграція з додатковими джерелами даних. Чим більше інформації може обробляти програмне забезпечення, тим точніші прогнози. Це може включати дані з соціальних медіа, новинних каналів, а також ринкові індикатори, що відображають загальний стан економіки.

Третім важливим аспектом є вдосконалення користувацького інтерфейсу. Користувачі прагнуть зазвичай простоти та зручності, тому важливо регулярно оновлювати інтерфейс, роблячи його більш інтуїтивним і доступним для широкого кола користувачів.

Четверте – це забезпечення вищої безпеки. У світі, де кібератаки стають все більш розповсюдженими, постійне оновлення заходів безпеки є

обов'язковим для забезпечення захисту користувацьких даних і фінансової інформації.

Останнім, але не менш важливим, є включення зворотного зв'язку від користувачів. Залучення спільноти користувачів до процесу вдосконалення дозволить зрозуміти їхні потреби та пріоритети, що сприятиме створенню продукту, який максимально відповідає очікуванням ринку.

Отже, для подальшого вдосконалення розробленого програмного забезпечення були обрані наступні перспективні напрямки:

- оновлення алгоритмів прогнозування.
- інтеграція з додатковими джерелами даних;
- удосконалення користувацького інтерфейсу;
- посилення заходів безпеки із збільшенням кількості користувачів у системі.
- включення зворотного зв'язку від користувачів для вдосконалення роботи системи;

4.5. Висновки до розділу

У даному розділі було розглянуто ключові аспекти оцінки ефективності розробленого способу для автоматизованого прогнозування ризиків під час операцій з криптовалютами.

Основні критерії оцінки включали точність прогнозу в різних умовах та за різний період часу разом із швидкодією. Окремо протестовано та оцінено точність прогнозування курсу та ризиків. Для визначення точності прогнозу були використані метрики RMSE та MAE, а швидкодію оцінювалася за допомогою бібліотеки time в Python.

Порівняльний аналіз курсу та ризиків показав, що розроблений спосіб забезпечує значно вищу точність прогнозів порівняно з

традиційними підходами, особливо у випадку динамічних і непередбачуваних умов криптовалютного ринку.

Тестування програмного забезпечення включало різні аспекти: від модульного тестування клієнтської частини, розробленої на основі React Native, до ручного тестування серверної частини на Node.js та Django. Особлива увага була приділена тестуванню роботи модулів прогнозування, де виявлені помилки були виправлені, а покриття коду тестами забезпечило високий рівень надійності.

У майбутньому планується подальше вдосконалення розробленого програмного забезпечення, включаючи оновлення алгоритмів прогнозування, інтеграцію з додатковими джерелами даних, покращення користувацького інтерфейсу, збільшення заходів безпеки, та включення зворотного зв'язку від користувачів для підвищення ефективності системи.

Отже, розроблене програмне забезпечення готове до випуску на ринок, оскільки воно успішно пройшло тестування, продемонструвало високу ефективність та надійність, також воно повністю відповідає потребам користувачів у точному та оперативному прогнозуванні ризиків під час операцій з криптовалютою.

5. ПОБУДОВА БІЗНЕС-МОДЕЛІ

5.1. Опис проблеми

Цей проєкт спрямований на створення автоматизованої системи для аналізу та прогнозування ризику, пов'язаного з криптовалютами. Процес розробки в першу чергу пов'язаний з вивченням проблем, дослідженням потенційних рішень і методів реалізації. Тестування передбачає перевірку виконаної роботи та скорочення часу, необхідного для її аналізу.

Станом на 2022 рік ринок криптовалют налічує приблизно 10 000 різних різновидів. Зростання популярності та швидкості розвитку пояснюється появою на біржах нових, перспективних проєктів. Біткойн продовжує займати більшість, але цей відсоток з кожним роком зменшується, тому інші валюти набувають популярності. Дані дослідження, проведеного компанією Chainalysis, свідчать про те, що Україна посідає третє місце за рівнем розвитку країни та створення нових криптовалют. Це важливо через актуальність теми. Однак швидкий розвиток і постійна передача інформації ускладнили компаніям проведення транзакцій і взаємодію зі змінами курсів у ручний спосіб. Вкрай важливо використовувати численні технології, методи та системи для автоматизації роботи криптовалют, щоб зменшити ймовірність проведення операцій із цими віртуальними активами. Збір даних, аналіз і прогнози про хід подій, поточний стан крипторинку та його найважливіші проблеми. Як правило такі операції виконуються професіональними трейдерами які обробляють інформацію зібрану на біржі вручну.

У цьому контексті першочерговою проблемою є витрати часу та витрати, пов'язані з підприємством. Очевидно, що потрібно наймати команду професіоналів і оплачувати додаткову роботу серверів, додатків та інших необхідних для цього програм. Як наслідок цих зусиль, криптовалюти знижуються в ціні, інвестори та біржі стурбовані стабільністю фінансів і фінансовою ситуацією, це ризики, згадані раніше.

Найбільш поширеним терміном, пов'язаним із криптовалютами, є волатильність. Це означає, що валюта може зростати або знижуватися протягом короткого періоду часу, що може призвести до проблем. З цього випливають ризики, спричинені людиною. Оскільки трейдери беруть участь в аналізі, розрахунках і прогнозах, не можна відкидати можливість того, що всі вони є предметними.

Організації, які спеціалізуються на операціях з криптовалютами, також використовують різноманітні додатки, які за окрему плату полегшують часткову або повну автоматизацію. Частота та якість дій розробників, як зазвичай очікується, залежить від процедури, яку вони дотримуються. Ці два аспекти є вирішальними, оскільки від реалізації цієї програми залежить фінансова спроможність фондової біржі та самого інвестора.

Незважаючи на переваги цих програм, потрібен додатковий захист від хакерів і зловмисного програмного забезпечення, що є найбільшою проблемою. Ті, хто використовує цифрову валюту та біржі, повинні використовувати ефективний захист, ретельно перевіряти всі адреси та посилання.

Таким чином, метою розробки є розробка способу автоматизації процесу аналізу та прогнозування ризиків операцій з криптовалютами за мінімальний час з високою точністю.

Дерево проблем

Нижче зображена схема на якій узагальнено описано наявні проблеми (рис. 5.1).

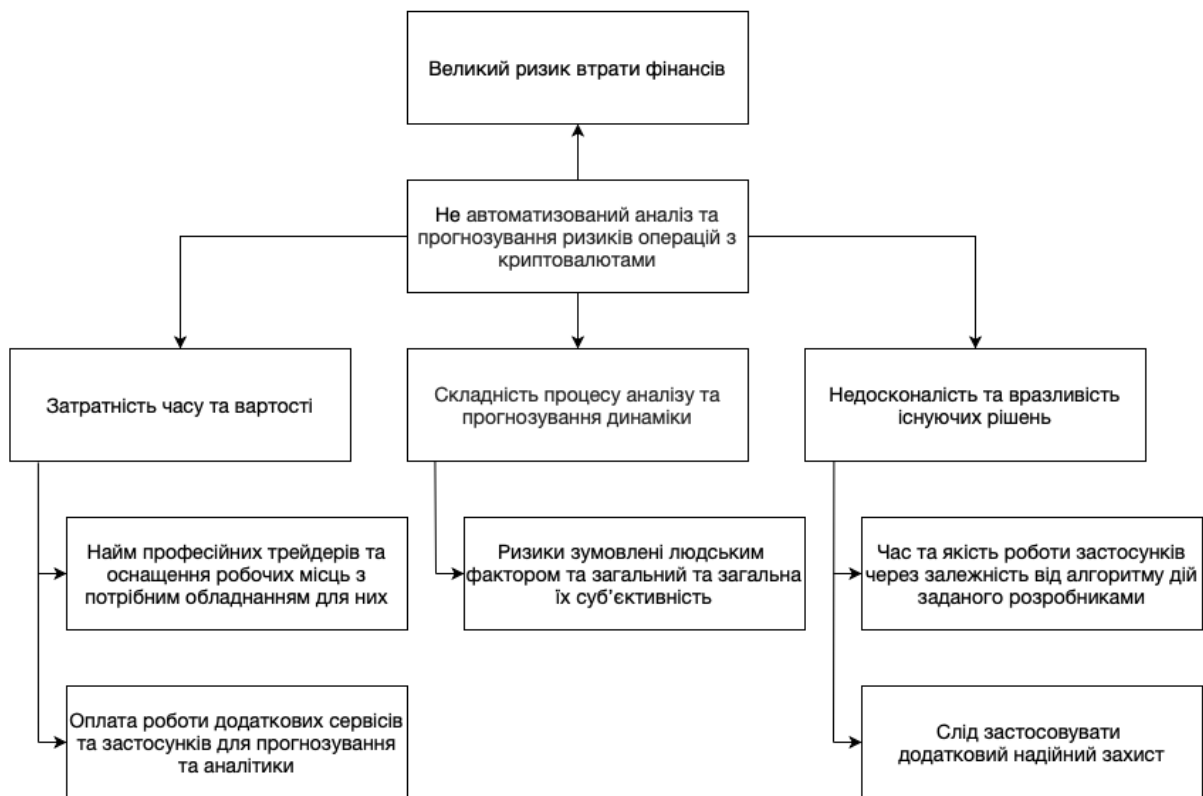


Рис. 5.1. Дерево проблем

5.2. Зацікавлені сторони

Узагальнимо перелік сторін, зацікавлених у розробці, їхні відповідні інтереси у використанні розробленого програмного забезпечення, їхній вплив на процес розробки та стратегії їх залучення в таблиці 5.2.

Таблиця 5.2

Зацікавлені сторони

№	Зацікавлена сторона	Сфера зацікавленості	B	3	$P=B*3$
1	Розробник застосунку	Розробник застосунку безперечно зацікавлений в його успішності. Також він має надію, що його продукт буде затребуваний більше ніж існуючі на ринку аналоги як в Україні, так і за її межами.	10	10	100
2	Компанії які спеціалізуються на прогнозуванні ризиків	Наявність такого типу застосунку допоможе компаніям автоматизувати та пришвидшити процес аналізу та прогнозування ризиків операцій з крипто валютами.	10	9	90
3	Крипто валютні біржі	Біржі зацікавлені в обмеженні ризиків під час обміну крипто валют чи інших активів.	8	8	64
4	Власники цифрових грошей	Власники цифрових грошей зможуть мінімізувати ризики при роботі із цифровими валютами на біржі чи при купівлі/продажі активів.	7	6	42
5	Розробники віртуальних монет	Розробка цілої крипто валюти зменшить потребу в знаннях теорії ймовірності та математики для обрахування ризиків під час тестування.	5	9	45
6	Трейдери	Частково зменшить потребу в сторонніх програмах та сервісах для обрахунку певної ймовірності ризику падіння/росту валюти чи загалом операцій із ними в даних проміжок часу.	6	8	48

7	Люди які зацікавлені в інвестиціях	Люди можуть швидше увійти у спілку інвесторів мінімізуювши ризики спричинені некомпетентністю	4	6	24
---	------------------------------------	---	---	---	----

У результаті розробник цього застосунку має найбільший вплив. Після завершення розрахунків необхідно розробити стратегію. Компанії, які спеціалізуються на прогнозуванні ризиків і розробці застосунків, мають найбільший потенціал впливу, тому саме з ними буде проведено найбільшу кількість ефективних комунікацій для досягнення мети та додаткової роботи.

Розробники криптовалюти та трейдери мають обмежений вплив, але вони нададуть корисні поради щодо технічних аспектів програми.

Для тих, хто зацікавлений в інвестуванні, вплив становить 24-45, тому, якщо вони можуть задовольнити свої основні вимоги щодо зовнішнього вигляду, мінімальної взаємодії з користувачем та набором інструментів, цільова програма буде мати постійну підтримку, що призведе до вищої продуктивності, розуміння та більшого інтересу до продукту в цілому.

5.3. Комерційне рішення. Основні характеристики

Відповідно до вищезазначеної інформації та загальних цілей залучених сторін також важливо визначити основні характеристики кінцевого продукту.

Програмне забезпечення, яке буде отримано в результаті дослідження способу автоматизованого аналізу та прогнозу ризиків операцій з криптовалютою, розроблене як вебдодаток, який розгорнуто з API, який надає дані з криптовалютних бірж. Розуміння та прогнозування

ризиків під час операцій з криптовалютами займе короткий час і матиме мінімум помилок.

Крім того, завдяки простоті використання він складатиметься з трьох окремих частин, які динамічно розділятимуть інтерфейс для швидкої взаємодії між собою.

Для трейдерів та інвесторів програмне забезпечення буде представлене у вигляді мобільного застосунку зі зручним інтерфейсом. Переваги цього підходу, дозволяють трейдерам повністю автоматизувати процес аналізу та прогнозування ризику операцій з криптовалютами та стежити за процесом у режимі реального часу. Цей спосіб задоволення потреб зрештою, сприятиме швидкому виявленню необхідної інформації та її моніторинг у реальному часі.

5.4. Конкурентні переваги рішення

У сфері торгівлі криптовалютами часто зустрічається ситуація, коли трейдери особисто беруть участь у оцінці та прогнозуванні ризиків, що іноді може призвести до суб'єктивних рішень. Такий підхід вимагає значних ресурсів, включаючи обладнання для кількох робочих місць та витрати на заробітну плату фахівців. Розроблений у рамках цього дослідження додаток пропонує інноваційний метод автоматизованого аналізу та прогнозування ризиків, пов'язаних з операціями криптовалют, що дозволяє знизити загальні витрати і зменшити потребу у фізичних робочих місцях.

Автоматизація не тільки допомагає керувати ризиками, пов'язаними з операціями криптовалют, але й підвищує швидкість здійснення транзакцій, враховуючи ризик і проведення аналізу. Ця швидкість відіграє ключову роль у фінансовому успіху, дозволяючи швидко реагувати на зміни у курсах валют та забезпечувати надійність угоди.

Програмне забезпечення, надає трейдерам зручні інструменти для прогнозування цін та ризику під час операцій з криптовалютою,

управління ризиками та їх оцінку. Індивідуалізація інтерфейсу дозволяє кожному користувачеві адаптувати додаток під свої унікальні потреби, підвищуючи тим самим ефективність ведення торгівлі.

В результаті конкурентними перевагами розробленого програмного забезпечення у вигляді веб-додатку є:

1. Скорочення витрат за рахунок скорочення кількості робочих місць і оплати додаткових послуг, що забезпечують прогнозування та аналіз курсу та ризиків.
2. Скорочення часу проведення операцій з криптовалютами за рахунок автоматичного аналізу і мінімізації ризиків.
3. Динамічний аналіз даних для отримання більш точного та фактичного прогнозу в рамках нестабільної ситуації на ринку криптовалют.
4. Підвищення ефективності програми як для індивідуального використання, так і для великих бірж або подібних послуг завдяки доступній інформації та аналітичним прогнозам у реальному часі.

5.5. Клієнти. Сегменти ринку споживання

При детальному аналізі ролі різних зацікавлених сторін у сфері криптовалют, особливу увагу слід звернути на криптобіржі, які є ключовими учасниками цього ринку. Виділяються чотири основні типи: централізовані, децентралізовані, криптосвопбери, та дилери криптовалютного свопу з попередньо встановленими гаманцями. Розроблене програмне забезпечення фокусується переважно на перших двох типах, пропонуючи автоматизовані рішення, які охоплюють широкий спектр потреб у цій області.

Централізовані біржі відрізняються здатністю управління активами, що дає користувачам можливість надійного зберігання та торгівлі криптовалютами. Відомі приклади таких бірж включають Kraken, Binance

та Bittrex. Ці платформи забезпечують високий рівень безпеки, простоту використання та надійність, що робить їх ідеальним вибором для широкого кола трейдерів.

У свою чергу, децентралізовані біржі, такі як Hodl Hodl, Bisq та LocalBitcoins, пропонують безпосередній обмін без участі посередників. Ці платформи використовують розумні контракти для забезпечення безпечної та прозорої торгівлі, відмовляючись від традиційних комісійних зборів. Цей підхід найкраще підходить для більш досвідчених користувачів, оскільки вимагає певних знань та навичок для ефективної торгівлі.

Тож розроблене програмне забезпечення спрямоване на інтеграцію з централізованими біржами, надаючи користувачам високий рівень безпеки та простоту використання. Це рішення дозволяє зменшити загальні витрати на обладнання та логістичні потреби бірж, водночас забезпечуючи ефективне управління активами з аналізом та прогнозуванням ризиків. Крім того, додаток надає користувачам можливість персоналізації інтерфейсу для задоволення їхніх індивідуальних потреб.

Розглядаючи ринок споживачів, розроблений застосунок призначений як для індивідуальних користувачів, так і для корпоративних клієнтів. Він включає веб-версію для зручного доступу для всіх типів користувачів, а також детально розроблений API для бізнес-клієнтів, які бажають інтегрувати рішення у свої системи.

Завдяки цьому підходу застосунок вирізняється своєю універсальністю, безпекою та ефективністю. Це робить його ідеальним інструментом для прогнозування та аналізу ризиків в операціях з криптовалютами. Також це дозволяє користувачам отримувати точні прогнози та аналітику, яка важлива для прийняття обґрунтованих рішень у цій динамічній і швидкозмінній сфері.

5.6. Унікальна ціннісна пропозиція

Ціннісна пропозиція програмного забезпечення для прогнозування ризиків операцій з криптовалютами полягає у наданні унікальної цінності, яка виходить за рамки традиційних підходів. Зважаючи на визначені вище завдання та потреби, які мають компанії, що спеціалізуються на прогнозуванні курсів криптовалют і ризиків пов'язаних із ними, розроблене програмне рішення пропонує значне поліпшення у цих процесах. Воно сприяє автоматизації аналізу та прогнозування, зменшуючи залежність від суб'єктивних оцінок професійних трейдерів і знижуючи ризик помилкових прогнозів.

На фондових біржах, де зараз задіяні професійні трейдери, розроблене програмне забезпечення дозволить автоматизувати аналітичні процеси, знижуючи вартість та підвищуючи точність прогнозування. Крім того, індивідуальні власники криптовалют та учасники операцій з крипто валютами отримають доступ до простого і ефективного інструменту, який допомагає у прийнятті обґрунтованих фінансових рішень.

На підставі аналізу потреб зацікавлених сторін, ідентифікована ключова проблема, яку вирішує представлене програмне рішення. Тобто, надання легкого доступу до комплексного аналізу ризиків та курсу криптовалют. Також завдяки веб-додатку, користувачі можуть отримувати всю необхідну інформацію про ринкові зміни та потенційні ризики у реальному часі.

Основною перевагою продукту є його клієнтоорієнтованість. Він розроблений з урахуванням потреб як великих компаній, які використовують відкрите API для інтеграції з власними системами, так і індивідуальних користувачів або трейдерів, які шукають простий у використанні інструмент для власних потреб. Веб-додаток та відкрите API постійно та автоматично розвиваються і адаптуються до змін ринкових умов, що забезпечує високий рівень задоволення користувачів і підтримку високої конкурентоспроможності продукту на ринку.

5.7. Доходи і витрати

Перед початком роботи над стартапом потрібно також точно впевнитись в його прибутковості. Для цього була створена таблиця плану доходу від продажу.

Таблиця 5.3

План доходу від продажу товарів та послуг на 2024 р.

№	Назва статті доходу	1 квартал 2024р.	2 квартал 2024р.	3 квартал 2024р.	4квартал 2024р.	Всього за 2024р.
1	Кількість ліцензій (шт.)	5	10	15	20	50
2	Ціна ліцензії (млн.грн)	7	7	7	7	28
3	Виторг від ліцензій (млн.грн)	35	70	105	140	350

Наступним кроком є визначення структури витрат.

Таблиця 5.4

Структура та розрахунок витрат (млн. грн.) на 2024 р.

№	Назва статті доходу	1 квартал 2024р.	2 квартал 2024р.	3 квартал 2024р.	4 квартал 2024р.	Всього за 2024р.
1	Заробітна плата (ЗП)	5	5	5	5	20
2	Податки на збори на ЗП	1.1	1.1	1.1	1.1	4.4
3	Оренда приміщень	3	3	3	3	12
4	Дослідна робота та розробка	4	4	4	4	32
5	Маркетинг та реклама	1	1	1	1	4
6	Комплектуючі та обладнання	2	2	2	2	8
7	Податкові платежі	1.25	2.5	3.75	5	12.5
8	Інші витрати	1	1	1	1	4
9	Всього витрат	18.35	19.6	20.85	22.1	96.9

Таблиця 5.5

План доходу від продажу товарів та послуг на 2024 р.

№	Назва статті доходу	1 квартал 2024р.	2 квартал 2024р.	3 квартал 2024р.	4квартал 2024р.	Всього за 2024р.
1	Прибуток	16,65	50,4	84,15	117,9	253,1

5.8. Бізнес-модель

Дотримуючись перелічених вище ознак, була створена бізнес-модель, яка відрізняється своєю простотою і водночас ефективністю. Основні клієнти – це компанії, що спеціалізуються на прогнозуванні ризиків, криптовалютні біржі, індивідуальні криптоінвестори та власники цифрових активів.

Проблема, яка вирішується, полягає у повільному та неефективному аналізі ризиків і прогнозуванні на ринку криптовалют, а також у високій вартості персоналу та офісних ресурсів, потрібних для ручного аналізу.

Розроблене рішення – це веб-додаток, який інтегрується з API бірж криптовалют для збору даних, аналізу та прогнозування або пропонується у вигляді мобільного застосунку.

Унікальна ціннісна пропозиція полягає в тому, що наш продукт виступає як оптимізатор автоматизованого аналізу та прогнозування ризиків для операцій з криптовалютами.

Структура витрат включає заробітну плату, податки на заробітну плату, оренду офісних приміщень, витрати на дослідження та розробку, маркетинг та рекламу, закупівлю обладнання, податкові платежі та інші непередбачувані витрати.

Канали розповсюдження включають спеціалізовані B2B партнерства, а також B2C залучення через соціальні мережі та інші рекламні заходи.

Ключові показники успіху включають загальний дохід від користувачів та кількість платних версій застосунку.

Додаткова прихована вигода включає скорочення часу та зниження ризику для автоматизованого аналізу та прогнозування ризиків операцій з криптовалютами. Ця бізнес-модель описана в таблиці 5.6 «Канва бізнес-моделі».

Канва бізнес моделі

Проблема	Рішення	Унікальна ціннісна пропозиція	Прихована перевага	Споживачі
Низька швидкість та якість аналізу та прогнозування ризиків операцій на криптовалютному ринку; суб'єктивність через людський фактор;	Програмне забезпечення у вигляді веб-додатку, що надає можливість автоматизовано отримувати данні із криптовалютних бірж для аналізу та прогнозування	Розроблюваний програмний продукт у вигляді веб-застосунку який оптимізує процес автоматизованого аналізу та прогнозування ризиків операцій з віртуальними збереженнями користувачів; дві версії додатку для зручної роботи із клієнтами та великими організаціями	Зменшення часу та можливого ризику в момент автоматизованого аналізу та прогнозування операцій з криптовалютами	компанії які спеціалізуються на прогнозуванні і ризиків, криптовалютні біржі, індивідуальні криптовалютні трейдери та власники цифрових грошей
	<i>Ключові метрики</i> Прибуток користувачів та сума придбаних платних версій застосунку з обмеженим терміном дії		<i>Канали</i> Спеціально відведені відділи для співпраці із ринком B2B; взаємодія із B2C через соціальні мережі та заходи спрямовані на зацікавлення	
Структура витрат			Потоки доходів	

Заробітна плата (ЗП); податки на збори на ЗП; оренда приміщень; Дослідна робота та розробка (виплата заробітних плат, соціальних виплат); маркетинг та реклама; комплектуючі та обладнання; податкові платежі; інші витрати (непередбачувані поломки обладнання, інші позапланові витрати)	Доходи від платної версії застосунку, комісія з отриманих прибутків
--	---

5.9. Висновки до розділу

У п'ятому розділі зосереджено увагу на детальному вивченні різних компонентів проєкту. Виявлені проблеми, які були чітко визначені в таблицях, надали розуміння ключових викликів. Також виявлено зацікавлені сторони і встановлено, що найбільший вплив та інтерес проявляють розробники та компанії, що займаються прогнозуванням ризиків. Ефективне спілкування та взаємодія з цими сторонами є ключовими для вдосконалення та розробки кінцевого продукту.

На основі цього створено комерційно привабливий продукт, що поєднує в собі усі необхідні функції високоякісного програмного забезпечення, що надає йому конкурентну перевагу. Розроблений застосунок, оптимізує та прискорює процес автоматизованого аналізу та прогнозування ризиків операцій з криптовалютою, визначається як унікальна пропозиція.

Також було визначено потенційні доходи та витрати, пов'язані з випуском та впровадженням програми на ринок. Результатом стала бізнес-модель, яка представлена у вигляді канви бізнес моделі. Вона відображає не тільки ефективність, але й необхідність розробки такого програмного забезпечення.

ВИСНОВКИ

Зростання попиту на альтернативні цифрові активи вимагає вдосконалення методів аналізу та прогнозування ризиків в операціях з криптовалютами. Магістерська робота присвячується розробці комбінованого способу програмного прогнозування ризиків операцій з криптовалютами в реальному часі з застосуванням нейронних мереж та алгоритмів машинного навчання, а також створення програмного забезпечення, яке забезпечує збір та аналіз історичних даних криптовалютного ринку, оцінку волатильності цього ринку, прогнозування курсу валют та оцінку ризиків з використанням розробленого способу для покращення точності.

У першому розділі були розглянуті існуючі методи прогнозування ризиків операцій з крипто валютами, серед яких фундаментальний, технічний, портфельний аналіз і диверсифікація, моделювання Монте-Карло та стрес-тестування з аналізом сценаріїв. Було виявлено, що ці методи не враховують короткострокові тренди, спрощують складні фінансові взаємозв'язки і не враховують всі можливі ризики на ринку криптовалют. Аналіз програмних рішень показав, що ці системи, зосереджені на прогнозуванні цін, а не на ризиках, також ці системи не мають визначеної точності методів прогнозування.

У другому розділі були розглянуті способи для автоматизованого збору інформації. Розглянуто інтегровані методи, ARIMA, LSTM, CNN, Random Forest, Gradient Boosting, Value-at-Risk, Expected Shortfall та стекінг моделей які потрібні для аналізу та прогнозування ризиків у криптовалютному ринку. Визначено їх ключові обмеження та способи їх вирішення. Описано комбінований спосіб який було обрано для автоматизованого прогнозування курсу та ризиків операцій з криптовалютами. Визначено ключові етапи прогнозування комбінованого способу, які включають в себе експертний аналіз історичних даних, крос-

валідацію моделей, визначення найкращої комбінації моделей, стекинг моделей та отримання кінцевого прогнозу. Переваги комбінованого способу, включають в себе точність та адаптивність під час виявлення неочевидних нелінійних взаємозв'язків та шаблонів у даних криптовалюти. Визначені метрики для оцінки ризиків Value-at-Risk та Expected Shortfall, які дозволяють розрахувати максимально можливі втрати з певною ймовірністю.

У третьому розділі було проаналізовано ключові технології для розробки програмного забезпечення, у тому числі вибір мов програмування та фреймворків для компонентів системи. JavaScript та React-Native обрано для реалізації клієнтської частини, оскільки вони забезпечують гнучкість та динамічність у веб-розробці. Для серверної частини та модуля прогнозування використано Python, Express і Django, що інтегровані з PostgreSQL, яка відповідає вимогам до обробки та зберігання великих обсягів даних. Використання клієнт-серверної архітектури та шаблону «Модель-Вид-Контролер» сприяє чіткій структурі та спрощенню керування. Запропоновано структуру бази даних із семи ключових компонентів, включно з модулем прогнозування для аналізу ризиків та курсу криптовалюти.

У четвертому розділі було проведено аналіз та оцінку розробленого програмного забезпечення, фокусуючись на оцінюванні прогнозування ризиків операцій з криптовалютами, тестуванні компонентів системи та шляхах її вдосконалення. Проаналізовано точність комбінованого способу в порівнянні з базовою моделлю, зосереджуючись на адаптивності до непередбачуваних змін на ринку криптовалюти. Результати показали, що комбінований метод прогнозування значно точніший за базові моделі на основі ARIMA та LSTM, особливо у довгостроковій перспективі незважаючи на більший час виконання що в середньому складає 3-5 секунд для комбінованого способу і 2-3 для ARIMA та LSTM. Оцінка точності в

ризиковій ситуації показала, що комбінований спосіб при 10-денному прогнозі має точність на 9% вище порівняно з одномодельними підходами.

У п'ятому розділі проаналізовано ключові проблеми прогнозування ризиків, визначено зацікавлені сторони, їхній вплив та очікування. Проведено аналіз ринку, сформовано унікальну ціннісну пропозицію, враховані потенційні витрати та доходи. Розроблена бізнес-модель типу lean canvas, яка підкреслює актуальність та перспективність розробленого програмного забезпечення.

СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. Cryptocurrency Risk Management: A Comprehensive Guide To Effective Risk Management: [Електронний ресурс]. – Режим доступу до ресурсу: financialcrimeacademy.org.
2. Assessing Crypto and Digital Asset Risks – KPMG United States: [Electronic resource]. – Resource access mode: kpmg.com.
3. Mastering Risk Management in Crypto Trading: All You Need to Know | KuCoin Learn: [Електронний ресурс]. – Режим доступу до ресурсу: kucoin.com.
4. How To Conduct A Great Crypto Risk Assessment: [Електронний ресурс]. – Режим доступу до ресурсу: riskpublishing.com.
5. 7 Unique Challenges in Cryptocurrency Risk Management: [Електронний ресурс]. – Режим доступу до ресурсу: garp.org.
6. Cryptocurrency price forecasting using the ARIMA model : [Електронний ресурс]. – Режим доступу до ресурсу: analyticsvidhya.com.
7. ARIMA on the Kaggle platform: [Електронний ресурс]. – Режим доступу до ресурсу: kaggle.com.
8. [Електронний ресурс] – Режим доступу <https://coinmetrics.io/>
9. [Електронний ресурс] – Режим доступу <https://www.cryptocompare.com/>
10. [Електронний ресурс] – Режим доступу <https://fastmining.org/>
11. Using the API to analyze the cryptocurrency market CoinAPI: [Електронний ресурс]. – Режим доступу до ресурсу: blog.coinapi.io.
12. Understanding the role of market data in cryptocurrency trading : [Електронний ресурс]. – Режим доступу до ресурсу: blog.coinapi.io.

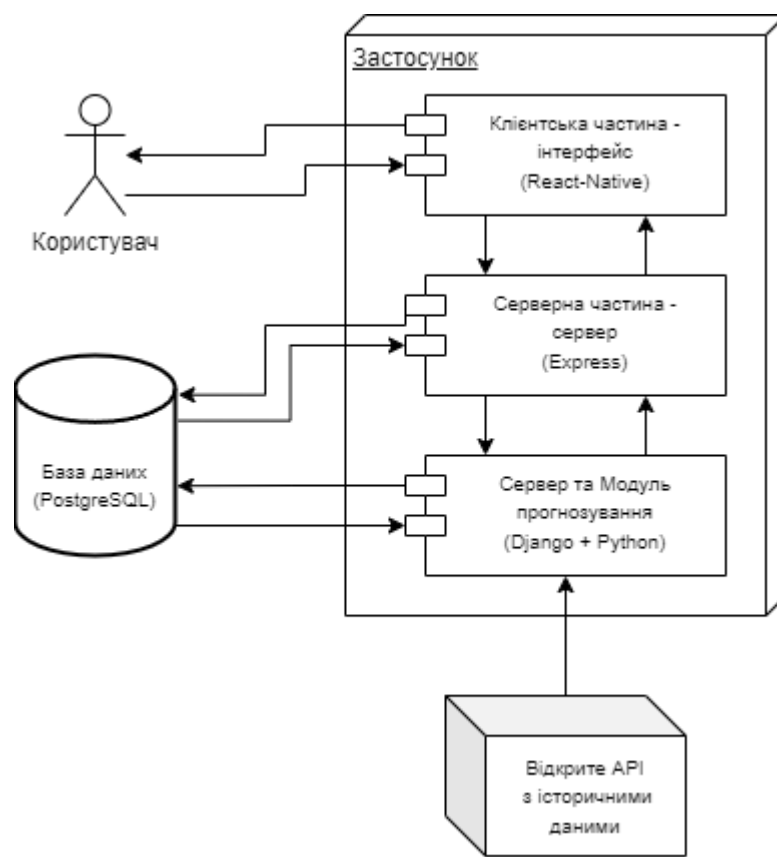
13. Technical analysis of cryptocurrencies: using moving averages, RSI and MACD: [Электронный ресурс]. – Режим доступа до ресурсу: coinmarketcap.com i cryptopolitan.com.
14. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press. – 800 с.
15. Hyndman, R. J., & Athanasopoulos, G. (2018). Forecasting: principles and practice. OTexts. – 420 с.
16. Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32. – 15 с.
17. Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. Annals of statistics, 1189-1232. – 1210 с.
18. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press. – 98 с.
19. Wolpert, D. H. (1992). Stacked generalization. Neural networks, 5(2), 241-259. – 249 с.
20. Jorion, P. (2007). Value at Risk: The New Benchmark for Managing Financial Risk. 3rd ed. McGraw-Hill. – 115 с.
21. Acerbi, C., & Tasche, D. (2002). On the coherence of expected shortfall. Journal of Banking & Finance, 26(7), 1487-1503. – 420 с.
22. Enders, W. (2014). Applied Econometric Time Series. 4th Edition. Wiley. – 496 с.
23. Lutz, M. (2013). Learning Python. 5th Edition. O'Reilly Media. – 1648 с.
24. Flanagan, D. (2020). JavaScript: The Definitive Guide. 7th Edition. O'Reilly Media. – 706 с.
25. Banks, A. & Porcello, E. (2020). Learning React: Modern Patterns for Developing React Apps. 2nd Edition. O'Reilly Media. – 350 с.

26. Brown, E. (2014). Web Development with Node and Express: Leveraging the JavaScript Stack. O'Reilly Media. – 332 с.
27. Greenfeld, D. A., & Roy, A. R. (2017). Two Scoops of Django 1.11: Best Practices for the Django Web Framework. Two Scoops Press. – 556 с.
28. Turnbull, J. (2014). The Docker Book: Containerization is the New Virtualization. James Turnbull. – 292 с.
29. Riggs, S., Krosing, H., & Balogh, K. (2020). PostgreSQL Up and Running: A Practical Guide to the Advanced Open Source Database. 4th Edition. O'Reilly Media. – 366 с.
30. Электронный ресурс: Client-Server Architecture: Concepts, Applications, and Trends – Techopedia.com.
31. Freeman, E., Robson, E., Bates, B., & Sierra, K. (2020). Head First Design Patterns: Building Extensible and Maintainable Object-Oriented Software. 2nd Edition. O'Reilly Media. – 672 с.

ДОДАТКИ

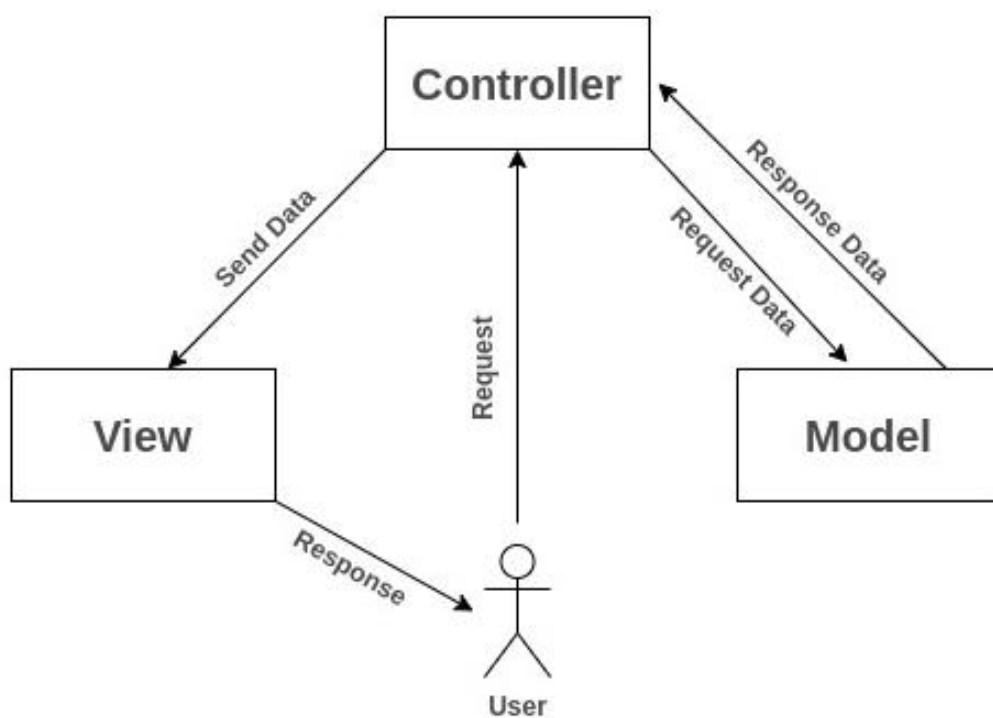
Додаток 1
Копії графічних матеріалів

Клієнт-серверна архітектура розробленого вебзастосунку



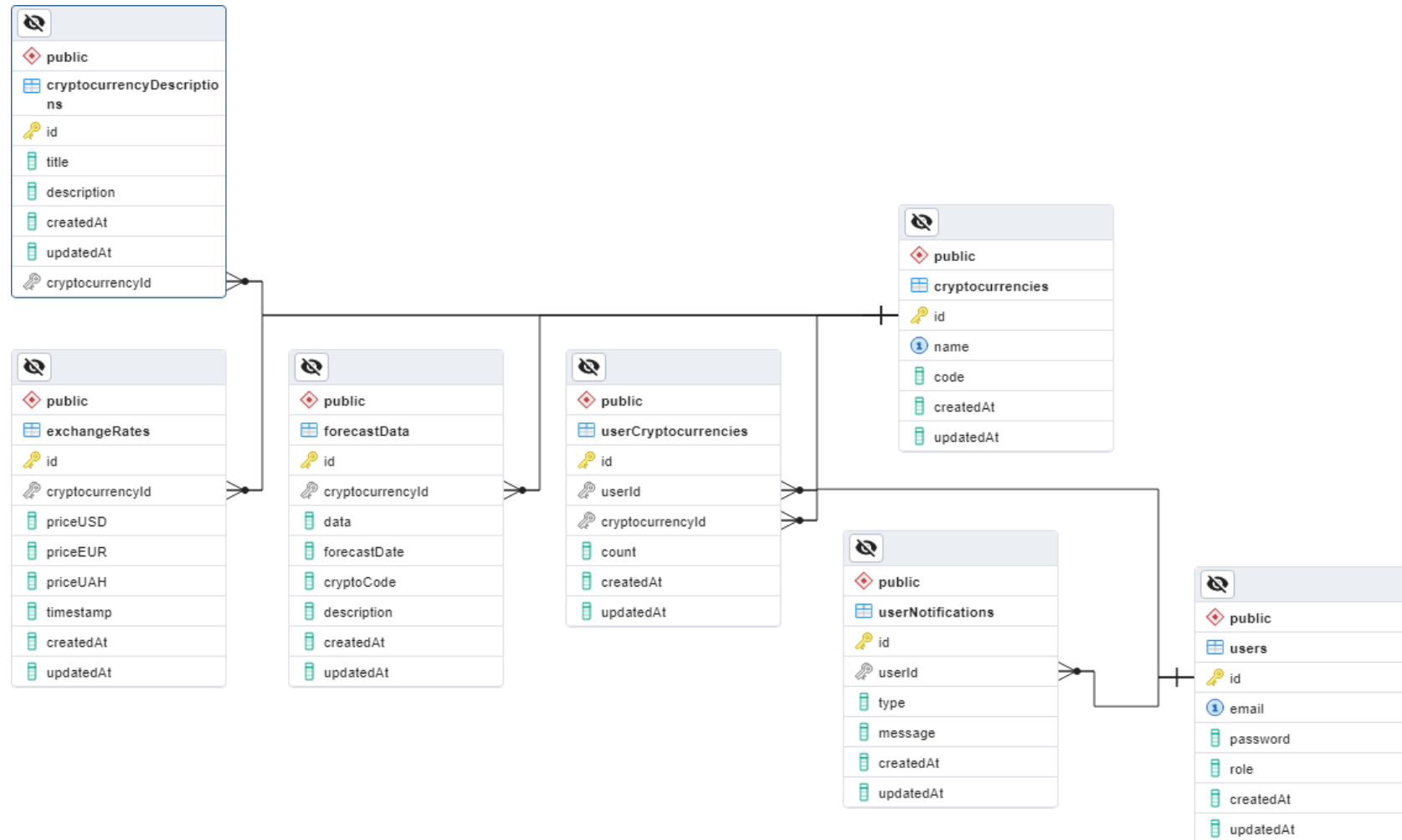
Кондрацький Владислав, КП-21мп

Клієнт-серверна архітектура розробленого вебзастосунку



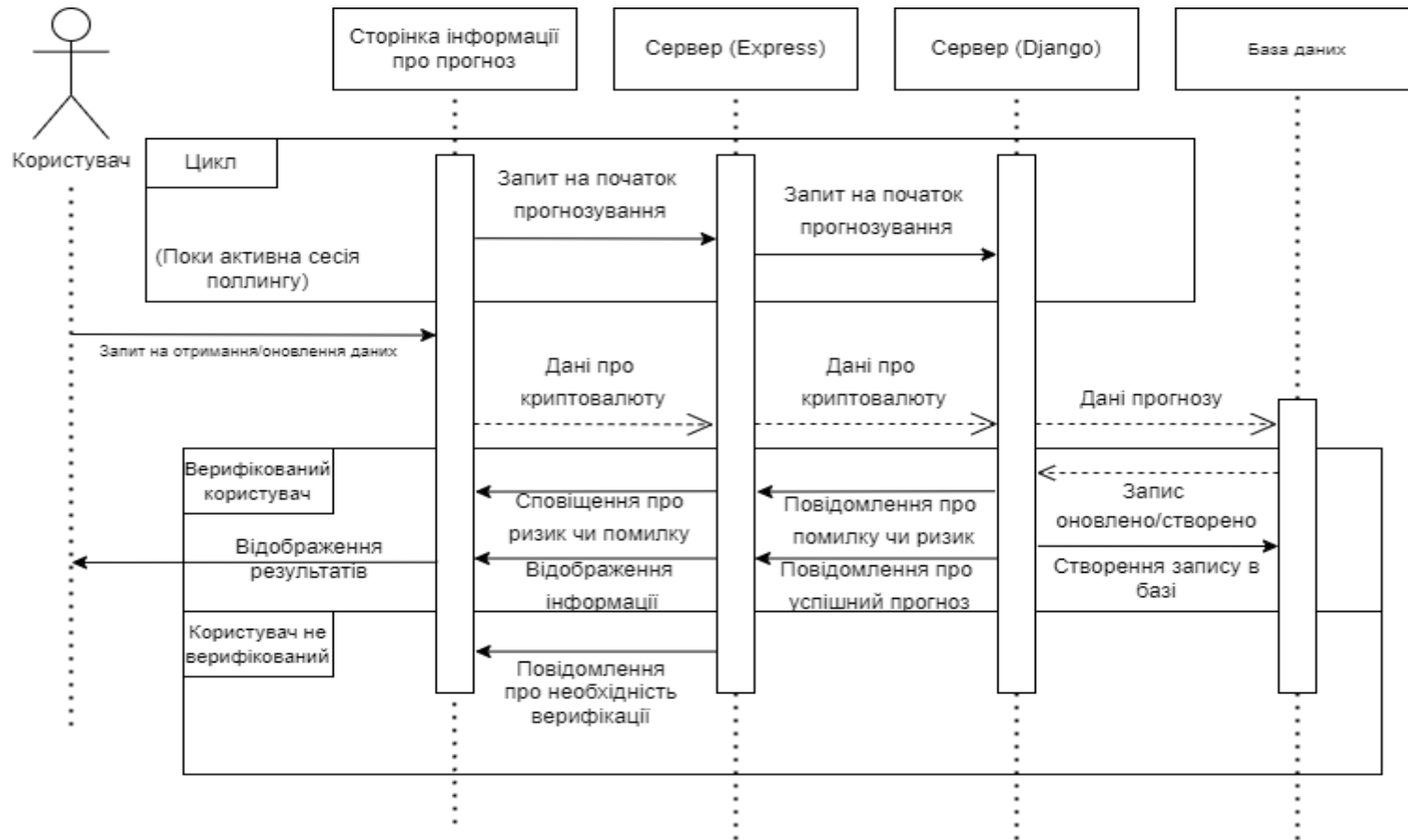
Кондрацький Владислав, КП-21мп

Модель бази даних



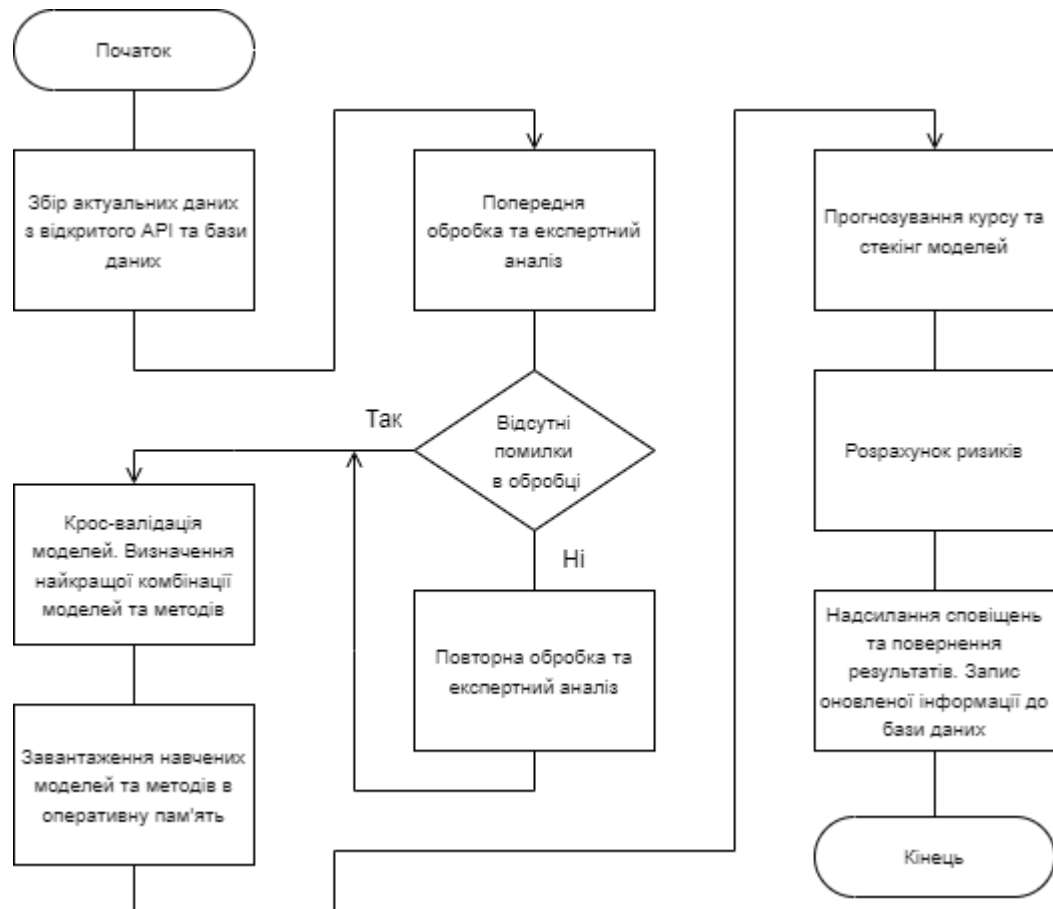
Кондрацький Владислав, КП-21мп

Модель отримання та оновлення даних



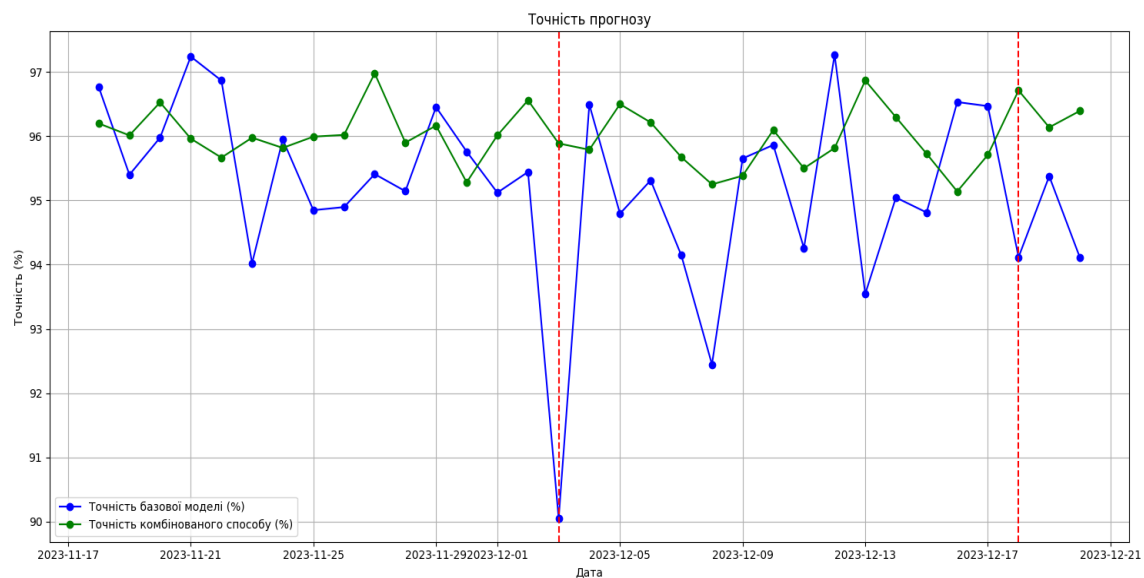
Кондрацький Владислав, КП-21мп

Схема роботи модулю прогнозування



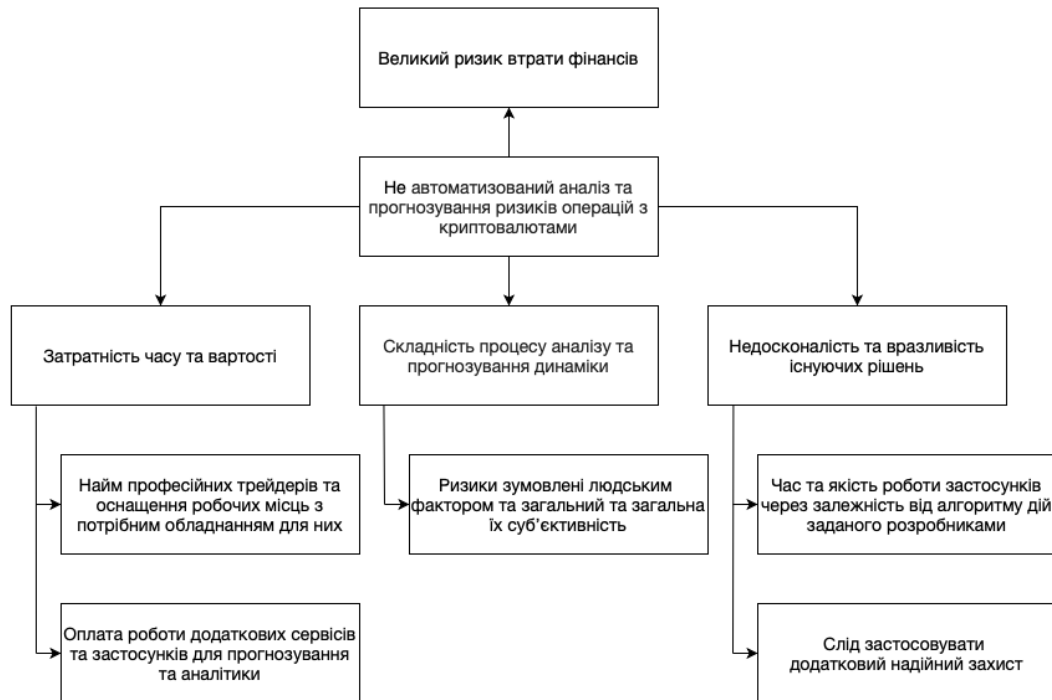
Кондрацький Владислав, КП-21мп

Графік точності прогнозу для комбінованого способу та базової моделі



Кондрацький Владислав, КП-21мп

Дерево проблем



Кондрацький Владислав, КП-21мп

Додаток 2
Лістинг програми

Лістинг 1. Urls.py

```
from django.urls import path
from .views import create_chart
from .views import get_historical_data
from .views import get_symbols
from .views import get_forecast_data

urlpatterns = [
    path('create/<str:name>/<str:start_time>/<str:end_time>', create_chart,
         name='chartCreator'),

    path('get_historical_data/<str:name>/<int:count>/<str:start_time>/<str:end_
time>',
         get_historical_data, name='chartCreator'),

    path('get_forecast_data/<str:code>/<str:user_id>/',
         get_forecast_data, name='chartCreator'),

    path('get_historical_data/<str:name>/<int:count>/<str:start_time>/',
         get_historical_data, name='chartCreator'),

    path('get_symbols', get_symbols, name='chartCreator'),
]
```

Лістинг 2. Forecast.py

```
import numpy as np
from django.http import JsonResponse
from sklearn.cluster import KMeans
from sklearn.preprocessing import MinMaxScaler
import pandas as pd
import requests
import json
from django.conf import settings
from pmdarima import auto_arima
from keras.models import Sequential
from keras.layers import LSTM, Dense
from sklearn.ensemble import RandomForestRegressor,
GradientBoostingRegressor
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from keras.layers import Conv1D, MaxPooling1D, Flatten
from keras.layers import Dropout
from keras.callbacks import EarlyStopping
from scipy.stats import t
from datetime import datetime, timedelta
from sklearn.model_selection import TimeSeriesSplit
from sklearn.metrics import mean_squared_error
from statsmodels.tsa.stattools import acf, pacf
from statsmodels.tsa.stattools import adfuller, kpss
from statsmodels.tsa.seasonal import seasonal_decompose
from scipy.stats import levene
from joblib import dump, load
import os
def save_model(model, filename):
    dump(model, filename)
```

```

def load_model(filename):
    return load(filename)

def train_or_load_model(model_name, train_function, *args, **kwargs):
    model_file = f'{model_name}.joblib'

    if os.path.exists(model_file):
        print(f"Завантаження моделі {model_name}")
        model = load_model(model_file)
    else:
        print(f"Навчання моделі {model_name}")
        model = train_function(*args, **kwargs)
        save_model(model, model_file)

    return model

def split_data(data, test_size=0.2):
    split_idx = int(len(data) * (1 - test_size))
    train, test = data[:split_idx], data[split_idx:]
    return train, test

def compute_RSI(series, period=14):
    delta = series.diff().dropna()
    loss = delta.where(delta < 0, 0)
    gain = -delta.where(delta > 0, 0)
    avg_loss = loss.rolling(window=period).mean()
    avg_gain = gain.rolling(window=period).mean()
    rs = avg_gain / avg_loss
    return 100 - (100 / (1 + rs))

def compute_MACD(series, fast=12, slow=26, signal=9):
    ema_fast = series.ewm(span=fast, adjust=False).mean()
    ema_slow = series.ewm(span=slow, adjust=False).mean()
    macd = ema_fast - ema_slow
    signal_line = macd.ewm(span=signal, adjust=False).mean()
    return macd, signal_line

def compute_RMSE(y_true, y_pred):
    differences = [true - pred for true, pred in zip(y_true, y_pred)]
    return np.sqrt(sum([diff ** 2 for diff in differences]) /
len(differences))

def compute_MAE(y_true, y_pred):
    differences = [true - pred for true, pred in zip(y_true, y_pred)]
    return sum([abs(diff) for diff in differences]) / len(differences)

def transform_data_for_cnn(data, lag=1):
    df = pd.DataFrame(data)
    columns = [df.shift(i) for i in range(1, lag + 1)]
    columns.append(df)
    df = pd.concat(columns, axis=1)
    df.fillna(0, inplace=True)

    X = df.iloc[:, :-1].values
    y = df.iloc[:, -1].values

```

```

X = X.reshape(X.shape[0], X.shape[1], 1)

return X, y

def train_cnn_model(X, y, epochs=50, batch_size=32):
    model = Sequential()
    model.add(Conv1D(filters=64, kernel_size=2, activation='relu',
input_shape=(X.shape[1], X.shape[2])))
    model.add(Flatten())
    model.add(Dense(50, activation='relu'))
    model.add(Dense(1))
    model.compile(optimizer='adam', loss='mse')
    model.fit(X, y, epochs=epochs, batch_size=batch_size, verbose=0)

    return model

def prepare_data(data):
    df = pd.DataFrame(data)
    original_dates = df['time_period_start'].copy()

    df['RSI'] = compute_RSI(df['price_close'])
    macd, signal = compute_MACD(df['price_close'])
    df['MACD'] = macd
    df['Signal'] = signal

    df['Volatility'] = df['price_high'] - df['price_low']
    columns_to_remove = ['time_period_end', 'time_open', 'time_close',
'time_period_start']
    df.drop(columns_to_remove, axis=1, inplace=True)

    return df, original_dates

def transform_data_for_lstm(data, lag=1):
    df = pd.DataFrame(data)

    scaler = MinMaxScaler(feature_range=(0, 1))
    df_scaled = scaler.fit_transform(df.values.reshape(-1, 1))

    df_scaled = pd.DataFrame(df_scaled)
    columns = [df_scaled.shift(i) for i in range(1, lag + 1)]
    columns.append(df_scaled)
    df = pd.concat(columns, axis=1)
    df.fillna(0, inplace=True)

    X = df.iloc[:, :-1].values
    y = df.iloc[:, -1].values

    X = X.reshape(X.shape[0], X.shape[1], 1)

    return X, y, scaler

def train_lstm_model(X, y, epochs=50, batch_size=32, neurons=50):
    model = Sequential()
    model.add(LSTM(neurons, input_shape=(X.shape[1], X.shape[2]),
return_sequences=True))
    model.add(Dropout(0.2))
    model.add(LSTM(neurons))
    model.add(Dropout(0.2))

```

```

        model.add(Dense(1))
        model.compile(loss='mean_squared_error', optimizer='adam')

        es = EarlyStopping(monitor='val_loss', patience=5, verbose=1,
            restore_best_weights=True)
        model.fit(X, y, epochs=epochs, batch_size=batch_size, verbose=0,
            validation_split=0.2,
            callbacks=[es])

    return model

def lstm_forecast(model, data, scaler):
    data = data.reshape(data.shape[0], 1, data.shape[1])
    forecast = model.predict(data)

    forecast = scaler.inverse_transform(forecast)
    return forecast

def forecast_prices(df, forecast_periods=30):
    print('testas_flag_for_print --', df.dtypes)

    price_close = df['price_close']

    if price_close.isnull().any():
        raise ValueError("price_close contains NaN values!")

    try:
        model_arima = auto_arima(price_close, seasonal=True, trace=False,
m=12)
        forecast_arima = model_arima.predict(n_periods=forecast_periods)
    except Exception as e:
        print('Error during ARIMA forecasting:', str(e))
        raise ValueError(f"Error forecasting with ARIMA model: {str(e)}")

    data_lstm_X, data_lstm_y, scaler =
transform_data_for_lstm(price_close.values)
    model_lstm = train_lstm_model(data_lstm_X, data_lstm_y)
    forecast_lstm = lstm_forecast(model_lstm, data_lstm_X[-
forecast_periods:], scaler)

    X = df.drop(['price_close'], axis=1)
    y = df['price_close']
    df.fillna(df.mean(), inplace=True)
    X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

    rf_model = RandomForestRegressor(n_estimators=100)
    rf_model.fit(X_train, y_train)
    rf_predictions = rf_model.predict(X_test)

    gb_model = GradientBoostingRegressor(n_estimators=100)
    gb_model.fit(X_train, y_train)
    gb_predictions = gb_model.predict(X_test)

    X_cnn, y_cnn = transform_data_for_cnn(price_close.values, lag=5)
    cnn_model = train_cnn_model(X_cnn, y_cnn)
    cnn_predictions = cnn_model.predict(X_cnn[-
forecast_periods:]).flatten()

    combined_forecast = (np.array(forecast_arima) + np.array(
        forecast_lstm.flatten()) + rf_predictions + gb_predictions +

```

```

cnn_predictions) / 5

    return combined_forecast

def stack_models(df, forecast_periods=30):
    try:
        model_efficiency = {}
        models_used = []
        rmse_values = {}
        mae_values = {}
        all_models = ['ARIMA', 'LSTM', 'RandomForest', 'GradientBoosting',
'CNN']

        model_scores = cross_validate_models(df)
        best_models = determine_best_model_combination(model_scores)
        models_used.extend(best_models)

        total_error = sum(model_scores.values())
        for model, error in model_scores.items():
            model_efficiency[model] = ((total_error - error) / total_error)
* 100

        train_df, test_df = split_data(df)
        X_train, y_train = train_df.drop(['price_close'], axis=1),
train_df['price_close']
        X_test, y_test = test_df.drop(['price_close'], axis=1),
test_df['price_close']

        predictions = []
        for model in best_models:
            pred, rmse, mae = perform_prediction_and_metrics(model,
X_train, X_test, y_train, y_test, df)
            if pred is not None:
                predictions.append(pred)
                rmse_values[model] = rmse
                mae_values[model] = mae

        min_length = min(len(pred) for pred in predictions)
        trimmed_predictions = [pred[:min_length] for pred in predictions]

        if trimmed_predictions:
            stacked_predictions = np.column_stack(trimmed_predictions)
            meta_model = LinearRegression()
            meta_model.fit(stacked_predictions, y_test[:min_length])
            forecast = meta_model.predict(stacked_predictions)
            stacking_rmse = compute_RMSE(y_test[:min_length], forecast)

            average_rmse = np.mean(list(rmse_values.values()))
            stackingEfficiency = 100 * (average_rmse - stacking_rmse) /
average_rmse if average_rmse else 0

        return {
            "forecast": forecast.tolist() if forecast is not None else "No
forecast",
            "rmse": rmse_values,
            "mae": mae_values,
            "modelsUsed": models_used,
            "models": all_models,
            "modelsEfficiency": model_efficiency,
            "stackingEfficiency": stackingEfficiency,
        }

```

```

except Exception as e:
    print("An error occurred during model stacking: ", str(e))
    return None

def perform_prediction_and_metrics(model_name, X_train, X_test, y_train,
y_test, df):
    try:
        predictions, rmse, mae = None, None, None

        if model_name == 'ARIMA':
            model_arima = train_or_load_model('arima', auto_arima, y_train,
seasonal=True, trace=False, m=12)
            predictions = model_arima.predict(n_periods=len(y_test))

        elif model_name == 'LSTM':
            lstm_train_data_X, lstm_train_data_y, lstm_train_scaler =
transform_data_for_lstm(y_train)
            if lstm_train_data_X.size == 0 or lstm_train_data_y.size == 0:
                raise ValueError("Empty training data for LSTM model.")
            model_lstm = train_or_load_model('lstm', train_lstm_model,
lstm_train_data_X, lstm_train_data_y)
            predictions = lstm_forecast(model_lstm, lstm_train_data_X,
lstm_train_scaler).flatten()

        elif model_name == 'RandomForest':
            model_rf = RandomForestRegressor(n_estimators=100)
            model_rf = train_or_load_model('random_forest', lambda X, y:
model_rf.fit(X, y), X_train, y_train)
            predictions = model_rf.predict(X_test)

        elif model_name == 'GradientBoosting':
            model_gb = GradientBoostingRegressor(n_estimators=100)
            model_gb = train_or_load_model('gradient_boosting', lambda X,
y: model_gb.fit(X, y), X_train, y_train)
            predictions = model_gb.predict(X_test)

        elif model_name == 'CNN':
            X_cnn, y_cnn = transform_data_for_cnn(y_train.values, lag=5)
            if X_cnn.size == 0 or y_cnn.size == 0:
                raise ValueError("Empty training data for CNN model.")
            model_cnn = train_or_load_model('cnn', train_cnn_model, X_cnn,
y_cnn)
            predictions =
model_cnn.predict(transform_data_for_cnn(y_test.values,
lag=5)[0]).flatten()

        else:
            raise ValueError(f"Model {model_name} is not recognized.")

        if predictions is not None:
            rmse = compute_RMSE(y_test, predictions)
            mae = compute_MAE(y_test, predictions)

        return predictions, rmse, mae

    except Exception as e:
        print(f"Error in model {model_name}: {str(e)}")
        return None, None, None

def calculate_returns(df):
    df['returns'] = df['price_close'].pct_change()

```

```

    return df

def calculate_future_returns(forecasted_prices):
    returns = np.diff(forecasted_prices) / forecasted_prices[:-1]
    return returns

def calculate_var(returns, confidence_level=0.95):
    if len(returns) == 0:
        return None
    sorted_returns = np.sort(returns)
    index = int((1 - confidence_level) * len(sorted_returns))
    var = sorted_returns[index]
    return var

def calculate_historical_var(returns, confidence_level=0.95):
    sorted_returns = np.sort(returns)
    index = int(confidence_level * len(sorted_returns))
    var = -sorted_returns[index]
    return var

def calculate_t_dist_es(returns, confidence_level=0.95):
    dof, mean, scale = t.fit(returns)
    es = t.expect(lambda x: x, args=(dof,), loc=mean, scale=scale, lb=mean
- scale * t.ppf(confidence_level, dof))
    return es

def calculate_es(returns, confidence_level=0.95):
    if len(returns) == 0:
        return None

    sorted_returns = np.sort(returns)
    index = int((1 - confidence_level) * len(sorted_returns))
    var = sorted_returns[index]
    es = sorted_returns[:index].mean()

    return es

def advanced_data_analysis(data):
    results = {}

    adf_test = adfuller(data['price_close'])
    kpss_test = kpss(data['price_close'], nlags='auto')
    results['stationary'] = {
        'adf_statistic': adf_test[0], 'adf_pvalue': adf_test[1],
        'kpss_statistic': kpss_test[0], 'kpss_pvalue': kpss_test[1]
    }

    decomposed = seasonal_decompose(data['price_close'], model='additive',
period=30)
    results['seasonality'] = {
        'seasonal': decomposed.seasonal,
        'trend': decomposed.trend,
        'resid': decomposed.resid,
        'seasonal_strength': np.std(decomposed.seasonal) /
np.std(data['price_close'])
    }

```

```

results['acf_values'] = acf(data['price_close'])
results['pacf_values'] = pacf(data['price_close'])

levene_test = levene(data['price_high'], data['price_low'])
results['volatility'] = {
    'levene_statistic': levene_test.statistic,
    'levene_pvalue': levene_test.pvalue
}

return results

def cross_validate_models(data, n_splits=5):
    tscv = TimeSeriesSplit(n_splits=n_splits)
    errors = {'ARIMA': [], 'LSTM': [], 'RandomForest': [],
              'GradientBoosting': [], 'CNN': []}

    for train_idx, test_idx in tscv.split(data):
        train, test = data.iloc[train_idx], data.iloc[test_idx]

        model_arima = train_or_load_model('arima_cv', auto_arima,
        train['price_close'], seasonal=True, trace=False, m=12)
        predictions_arima = model_arima.predict(n_periods=len(test))
        errors['ARIMA'].append(mean_squared_error(test['price_close'],
        predictions_arima))

        X_lstm, y_lstm, scaler_lstm =
        transform_data_for_lstm(train['price_close'])
        lstm_model = train_or_load_model('lstm_cv', train_lstm_model,
        X_lstm, y_lstm)
        X_test_lstm, y_test_lstm, _ =
        transform_data_for_lstm(test['price_close'])
        lstm_predictions = lstm_forecast(lstm_model, X_test_lstm,
        scaler_lstm)
        errors['LSTM'].append(mean_squared_error(test['price_close'],
        lstm_predictions.flatten()))

        rf_model = RandomForestRegressor(n_estimators=100)
        rf_model = train_or_load_model('random_forest_cv', lambda X, y:
        rf_model.fit(X, y), train.drop('price_close', axis=1),
        train['price_close'])
        rf_predictions = rf_model.predict(test.drop('price_close', axis=1))

        errors['RandomForest'].append(mean_squared_error(test['price_close'],
        rf_predictions))

        gb_model = GradientBoostingRegressor(n_estimators=100)
        gb_model = train_or_load_model('gradient_boosting_cv', lambda X, y:
        gb_model.fit(X, y), train.drop('price_close', axis=1),
        train['price_close'])
        gb_predictions = gb_model.predict(test.drop('price_close', axis=1))

        errors['GradientBoosting'].append(mean_squared_error(test['price_close'],
        gb_predictions))

        X_cnn, y_cnn = transform_data_for_cnn(train['price_close'], lag=2)
        cnn_model = train_or_load_model('cnn_cv', train_cnn_model, X_cnn,
        y_cnn)
        X_test_cnn, y_test_cnn =
        transform_data_for_cnn(test['price_close'], lag=2)[0:2]
        cnn_predictions = cnn_model.predict(X_test_cnn).flatten()
        errors['CNN'].append(mean_squared_error(y_test_cnn,
        cnn_predictions))

```

```

        return {model: np.mean(err) for model, err in errors.items()}

def determine_best_model_combination(model_scores):
    best_combination = []
    min_error = float('inf')

    models = list(model_scores.keys())
    for i in range(len(models)):
        for j in range(i + 1, len(models)):
            avg_error = (model_scores[models[i]] + model_scores[models[j]])
/ 2
            if avg_error < min_error:
                min_error = avg_error
                best_combination = [models[i], models[j]]

    return best_combination

def determine_notification_type(VaR, ES, rmse, mae):
    var_threshold_risk = -0.05
    es_threshold_risk = 0.1
    rmse_threshold_warning = 1000
    mae_threshold_warning = 500

    if VaR <= var_threshold_risk or ES >= es_threshold_risk:
        return "risk"

    average_rmse = np.mean(list(rmse.values()))
    average_mae = np.mean(list(mae.values()))
    if average_rmse > rmse_threshold_warning or average_mae >
mae_threshold_warning:
        return "warning"

    return "info"

def notification_delivery(forecast, rmse, mae, modelsUsed, models,
                        modelEfficiency, stackingEfficiency, VaR, ES,
user_id, code):
    try:
        forecast_message = f"Прогноз: {forecast}" if forecast != "No
forecast" else "Прогноз відсутній."

        forecast_for_code = f"Прогноз для {code}"

        modelsUsed_message = f"Використані моделі: {'',
'.join(modelsUsed)}."

        efficiency_message = "Ефективність моделей: " + ', '.join(
            [f"{model}: {efficiency:.2f}%" for model, efficiency in
modelEfficiency.items()])

        rmse_message = "RMSE: " + ', '.join([f"{model}: {value:.2f}" for
model, value in rmse.items()])
        mae_message = "MAE: " + ', '.join([f"{model}: {value:.2f}" for
model, value in mae.items()])

        stackingEfficiency_message = f"Ефективність стекинга:
{stackingEfficiency:.2f}%"

        var_message = f"VaR: {VaR:.2f}"

```

```

es_message = f"ES: {ES:.2f}"

notification = "\n".join(
    [forecast_for_code, modelsUsed_message, efficiency_message,
    rmse_message, mae_message,
    stackingEfficiency_message, var_message, es_message])

notification_type = determine_notification_type(VaR, ES, rmse, mae)

payload = {
    'userId': user_id,
    'type': notification_type,
    'message': notification,
}

post_url = 'http://192.168.1.5:5000/api/userNotifications'

try:
    post_response = requests.post(post_url, json=payload)
    if post_response.status_code == 200:
        return JsonResponse({
            "success": True,
            "message": 'Data successfully saved to database'
        }, status=200)
    else:
        return JsonResponse({
            "success": False,
            "message": f"Failed to save data. Status code:
{post_response.status_code}"
        }, status=200)
except requests.exceptions.RequestException as e:
    return JsonResponse({
        "success": False,
        "message": "An error occurred",
        "error": str(e)
    }, status=200)
except Exception as e:
    return f"Помилка при формуванні сповіщення: {str(e)}"

def get_forecast_data(request, code, user_id):
    url = f'http://192.168.1.5:5000/api/historicalData/{code}'
    response = requests.get(url)

    if response.status_code == 200:
        response_data = response.json()
        data = response_data.get('data')

        df, original_dates = prepare_data(data)
        df = calculate_returns(df)
        df.fillna(df.mean(), inplace=True)

        if 'price_close' not in df.columns:
            return JsonResponse({"success": False, "response": "price_close
column not found in DataFrame"}, status=400)

        try:
            scaler = MinMaxScaler()
            df_normalized =
pd.DataFrame(scaler.fit_transform(df[['price_close']]))
            kmeans = KMeans(n_clusters=3)
            kmeans.fit(df_normalized)
            df['cluster'] = kmeans.predict(df_normalized)

```

```

        distances = kmeans.transform(df_normalized)
        df['risk'] = distances.min(axis=1)
    except Exception as e:
        return JsonResponse({"success": False, "response": f"Error
during data processing: {str(e)}"}, status=500)

    try:
        model_results = stack_models(df)
        future_prices = model_results['forecast']
        future_returns = calculate_future_returns(future_prices)

        var = calculate_historical_var(future_returns)
        es = calculate_t_dist_es(future_returns)

        historical_data_with_dates =
df.assign(date=original_dates).to_dict(orient="records")

    except Exception as e:
        return JsonResponse({"success": False, "response": f"Error
during forecasting: {str(e)}"}, status=500)

    response_data = {
        "usedCode": code,
        "historical_data": historical_data_with_dates,
        "forecast": future_prices,
        "VaR": var,
        "ES": es,
        "rmse": model_results['rmse'],
        "mae": model_results['mae'],
        "modelsUsed": model_results['modelsUsed'],
        "models": model_results['models'],
        "modelsEfficiency": model_results['modelsEfficiency'],
        "stackingEfficiency": model_results['stackingEfficiency']
    }

    notification_delivery(
        future_prices,
        model_results['rmse'],
        model_results['mae'],
        model_results['modelsUsed'],
        model_results['models'],
        model_results['modelsEfficiency'],
        model_results['stackingEfficiency'],
        var,
        es,
        user_id,
        code
    )

    date_now_init = datetime.now()
    date_now_plus_2_hours = date_now_init + timedelta(hours=2)
    formatted_date = date_now_plus_2_hours.strftime('%Y-%m-%d
%H:%M:%S')
    date_now = formatted_date

    payload = {
        'cryptocurrencyId': 1,
        'cryptoCode': code,
        'forecastDate': date_now,
        'data': response_data
    }

    post_url = 'http://192.168.1.5:5000/api/forecastData'

```

```

try:
    post_response = requests.post(post_url, json=payload)
    if post_response.status_code == 200:
        return JsonResponse({
            "success": True,
            "message": 'Data successfully saved to database'
        }, status=200)
    else:
        return JsonResponse({
            "success": False,
            "message": f"Failed to save data. Status code:
{post_response.status_code}"
        }, status=200)
except requests.exceptions.RequestException as e:
    return JsonResponse({
        "success": False,
        "message": "An error occurred",
        "error": str(e)
    }, status=200)
else:
    error_message = response.json().get('error', 'Unknown error')
    return JsonResponse({"success": False, "response": f"Error:
{response.status_code}. Message: {error_message}"},
        status=response.status_code)

def send_to_database(name, data):
    crypto_code = name.split('_')[2]
    date_now_init = datetime.now()
    date_now_plus_2_hours = date_now_init + timedelta(hours=2)
    formatted_date = date_now_plus_2_hours.strftime('%Y-%m-%d %H:%M:%S')
    date_now = formatted_date
    payload = {
        'cryptocurrencyId': 1,
        'cryptoCode': crypto_code,
        'date': date_now,
        'data': data
    }
    post_url = 'http://192.168.1.5:5000/api/historicalData'

    try:
        post_response = requests.post(post_url, json=payload)
        if post_response.status_code == 200:
            return {"success": True, "message": "Data successfully saved to
database", "post_response": post_response}
        else:
            return {"success": False, "message": f"Failed to save data.
Status code: {post_response.status_code}",
                "response": post_response.text}
    except requests.exceptions.RequestException as e:
        return {"success": False, "message": "An error occurred", "error":
str(e)}

```

Лістинг 2. Server.js

```

require('dotenv').config()
const express = require('express')
const sequelize = require('./db')
const cors = require('cors')
const fileUpload = require('express-fileupload')
const router = require('./routes/index')
const errorHandler = require('./middleware/ErrorHandlerMiddleware')

```

```

const bodyParser = require('body-parser');

const PORT = process.env.PORT || 5000

const app = express()
app.use(cors())
app.use(bodyParser.json({ limit: '50mb' }));
app.use(bodyParser.urlencoded({ limit: '50mb', extended: true }));
app.use(express.json())
app.use(fileUpload({}))
app.use('/api', router)
app.use(errorHandler)

const start = async () => {
  try {
    await sequelize.authenticate()
    await sequelize.sync()
    app.listen(PORT, () => console.log(`Server started on port ${PORT}`))
  } catch (e) {
    console.log(e)
  }
}

start()

```

Лістинг 3. UserController.js

```

const ApiError = require('../errors/ApiError');
const bcrypt = require('bcrypt')
const jwt = require('jsonwebtoken')
const { User } = require('../models/models')

const generateJwt = (id, email, role) => {
  return jwt.sign(
    {id, email, role},
    process.env.SECRET_KEY,
    {expiresIn: '24h'}
  )
}

class UserController {
  async registration(req, res, next) {
    const {email, password, role} = req.body
    if (!email || !password) {
      return next(ApiError.badRequest('Password or email not found'))
    }
    const candidate = await User.findOne({where: {email}})
    if (candidate) {
      return next(ApiError.badRequest('Email already exist'))
    }
    const hashPassword = await bcrypt.hash(password, 5)
    const user = await User.create({email, role, password:
hashPassword})
    const token = generateJwt(user.id, user.email, user.role)
    return res.json({token})
  }

  async login(req, res, next) {
    const {email, password} = req.body
    const user = await User.findOne({where: {email}})
    if (!user) {

```

```

        return next(ApiError.internalError('User not found'))
    }
    let comparePassword = bcrypt.compareSync(password, user.password)
    if (!comparePassword) {
        return next(ApiError.internalError('Password uncorrected'))
    }
    const token = generateJwt(user.id, user.email, user.role)
    return res.json({token})
}

async getAll(req, res, next) {
    const users = await User.findAll()
    return res.json(users)
}

async checkAuth(req, res, next) {
    const token = generateJwt(req.user.id, req.user.email,
req.user.role)
    return res.json({token})
}
}

```

Додаток 3
Копія презентації

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ «КИЇВСЬКИЙ
ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ГОРЯСІКОРСЬКОГО»



ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

КАФЕДРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ
КОМП'ЮТЕРНИХ СИСТЕМ

СПОСІБ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ
АВТОМАТИЗОВАНОГО АНАЛІЗУ ТА ПРОГНОЗУВАННЯ
РИЗИКІВ ОПЕРАЦІЙ З КРИПТОВАЛЮТАМИ

Студент: Владислав КОНДРАЦЬКИЙ

Науковий Керівник: Ст. викладач кафедри ПЗКС, к.т.н., Люшенко Леся
Анатоліївна

Київ – 2024

ПОСТАНОВКА ЗАДАЧІ

Об'єкт дослідження: програмне аналізування та прогнозування ризиків операцій з криптовалютами

Предмет дослідження: методи та способи програмного аналізування та прогнозування ризиків операцій з криптовалютами

Мета роботи: розробка програмного забезпечення з використанням комбінованого способу програмного прогнозування ризиків операцій з криптовалютами в реальному часі для підвищення точності прогнозування

СПИСОК ЗАВДАНЬ

1. Проведення аналізу існуючих способів прогнозування ризиків операцій з криптовалютами.
2. Розробка комбінованого способу автоматизованого аналізу та прогнозування ризиків операцій з криптовалютами із урахуванням знайдених недоліків у існуючих способах.
3. Розробка програмного забезпечення.
4. Оцінка точності запропонованого способу та тестування програмного забезпечення.
5. Побудова бізнес-моделі.

АКТУАЛЬНІСТЬ ДОСЛІДЖЕННЯ ТА ІСНУЮЧІ РІШЕННЯ

- За останні роки відбулося значне зростання інтересу до цифрових активів як альтернативної форми платежів
- Криптовалюти характеризуються високою волатильністю, що може призводити до значних коливань у вартості
- Необхідність розвитку нових методів та програмного забезпечення для ефективного аналізу та прогнозування ризиків у операціях з криптовалютами

ІСНУЮЧІ РІШЕННЯ

- Фундаментальний аналіз;
- Технічний аналіз;
- Аналіз ризиків на основі чутливості;
- Моделювання монте-карло;
- Портфельний аналіз і диверсифікація;
- Стрес-тестування та аналіз сценаріїв.

НАЙВІДОМІШІ ПРОГРАМНІ РІШЕННЯ

Платформа	Основні недоліки
CryptoCompare	Недостатньо функціональних можливостей, прогнози доступні лише після підписки, невизначена середня точність, невідповідний період прогнозування
CoinMetrics	Обмежені функціональні можливості, потреба в підписці для доступу до повного спектру аналітики, невизначена точність прогнозів
TradingView	Прогнози доступні лише після підписки, відсутність спеціалізації на прогнозуванні ризиків, невизначена точність прогнозів
Messari	Обмежена функціональність без підписки, невизначена середня точність прогнозів, незазначений метод прогнозування, не регулярно виконується прогнозування



РОЗРОБЛЕНИЙ СПОСІБ ПРОГНОЗУВАННЯ РИЗИКІВ ОПЕРАЦІЙ З КРИПТОВАЛЮТАМИ

КОМБІНОВАНИЙ СПОСІБ ПРОГНОЗУВАННЯ КУРСУ ТА РИЗИКІВ КРИПТОВАЛЮТИ ЗА ДОПОМОГОЮ ЕКСПЕРТНОГО АНАЛІЗУ ТА СТЕКІНГУ МОДЕЛЕЙ

1. Експертний Аналіз:

- Стаціонарність (тести ADF і KPSS)
- Сезонність
- Автокореляція (ACF і PACF)
- Аналіз волатильності (тест Левена)

2. Експертизнайкращої комбінації:

- Крос-валідація (TimeSeriesSplit)
- Оцінка точності (MSE)
- Вибір комбінації, яка мінімізує загальну похибку

Розширений тест Дікі-Фуллера (ADF):

$$Y_t = a_1 Y_{t-1} + a_2 Y_{t-2} + \varepsilon_t$$

де Y_t – часовий ряд, а ε – помилка.

Розширений тест Квятковського-Філіпса-Шмідта-Шина (KPSS):

$$KPSS_{statistic} = kpss_test[0]\rho$$

де ρ – значення = `kpss_test[1]`.

3. Стекінг Моделей:

- Комбінування моделей (ARIMA, LSTM, CNN, Random Forest, Gradient Boosting)
- Кінцевий прогноз

4. Прогнозування ризиків:

- Value-at-Risk (VaR)
- Expected Shortfall (ES)

MSE: $\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$, де y_i – фактичні значення, $\hat{y}_i$ – прогнозовані значення.

тест Левена:

$$H_0: \sigma_1 = \sigma_2 = \dots = \sigma_k \text{ vs } H_1: \sigma_i \neq \sigma_j$$

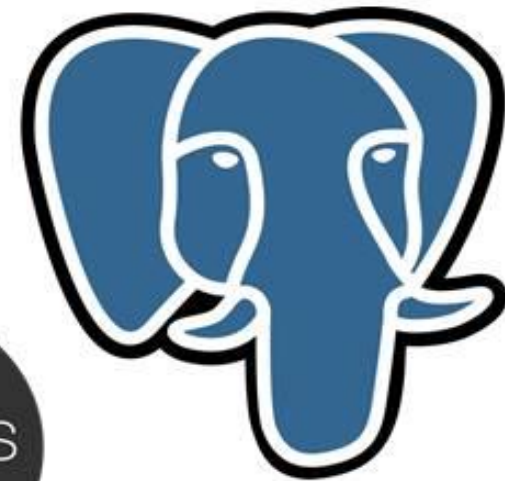
принаймні для однієї пари (i, j).

ОСОБЛИВОСТІ РЕАЛІЗАЦІЇ

ЗАСОБИ РЕАЛІЗАЦІЇ



django

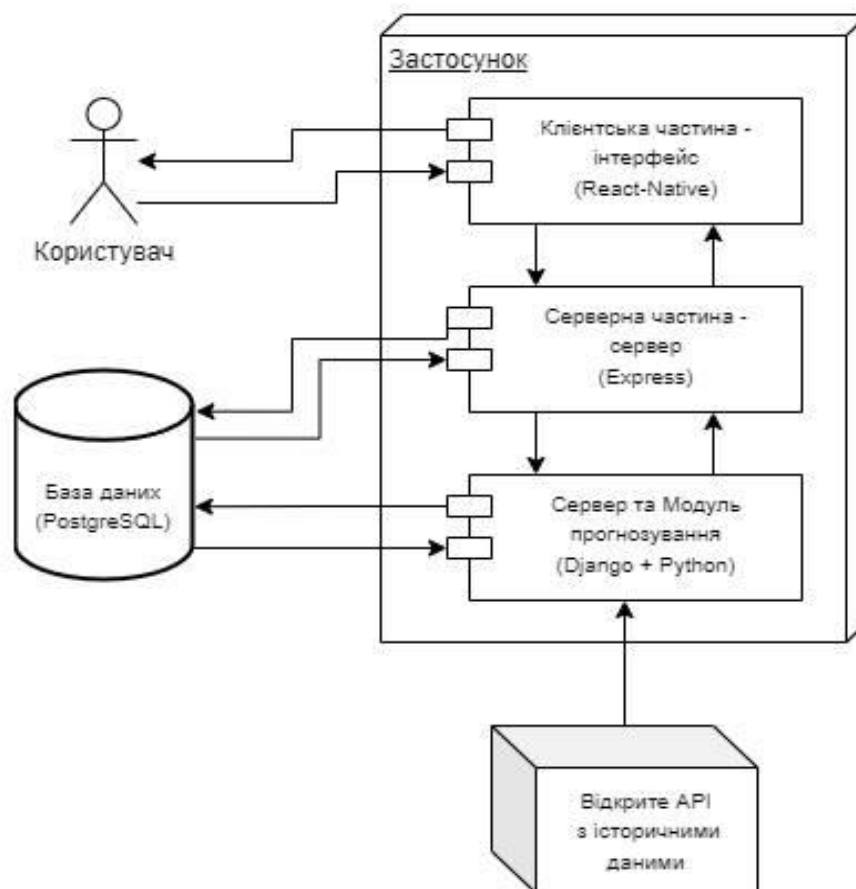


ЗБІР ДАНИХ

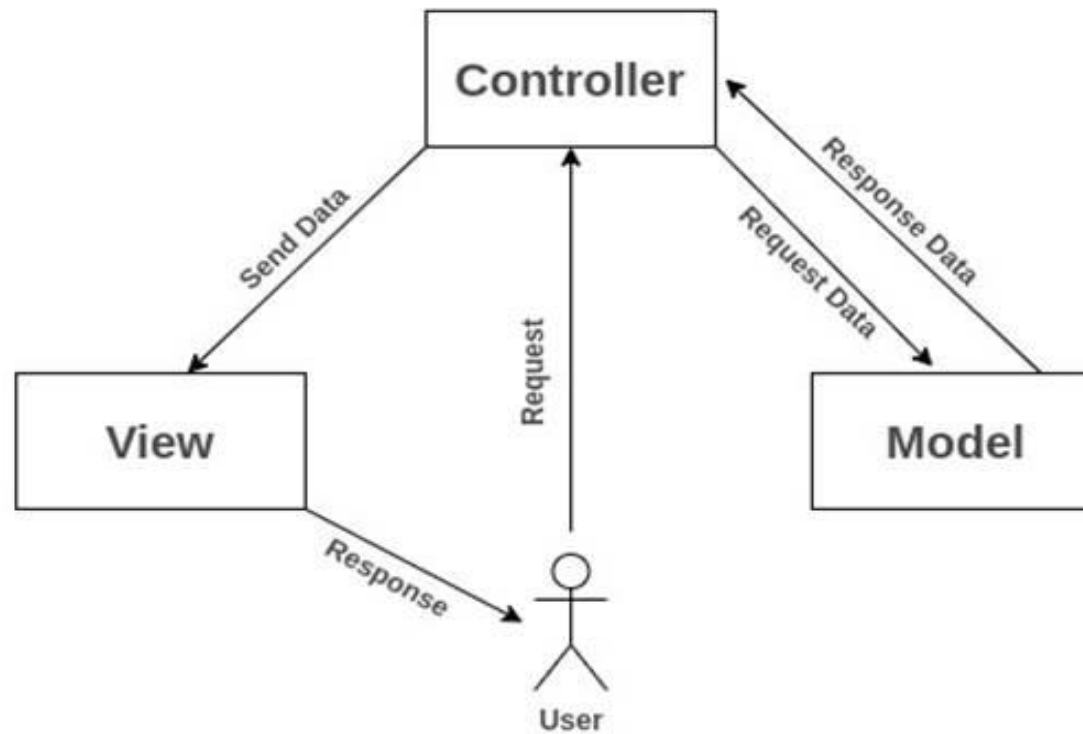
```
{
  "time_period_start": "2022-07-17T00:00:00.0000000Z",
  "time_period_end": "2022-07-18T00:00:00.0000000Z",
  "time_open": "2022-07-17T00:00:00.0349750Z",
  "time_close": "2022-07-17T23:59:59.9962050Z",
  "price_open": 21189.23,
  "price_high": 21667.1,
  "price_low": 20750.1,
  "price_close": 20791.74,
  "volume_traded": 11436.2821364,
  "trades_count": 376754
},
{
  "time_period_start": "2022-07-18T00:00:00.0000000Z",
  "time_period_end": "2022-07-19T00:00:00.0000000Z",
  "time_open": "2022-07-18T00:00:00.0133520Z",
  "time_close": "2022-07-18T23:59:59.8657470Z",
  "price_open": 20791.74,
  "price_high": 22777,
  "price_low": 20755.95,
  "price_close": 22427.56,
  "volume_traded": 31445.1413633,
  "trades_count": 587788
},
{
  "time_period_start": "2022-07-19T00:00:00.0000000Z",
  "time_period_end": "2022-07-20T00:00:00.0000000Z",
  "time_open": "2022-07-19T00:00:00.1043620Z",
  "time_close": "2022-07-19T23:59:59.7159420Z",
  "price_open": 22434.97,
  "price_high": 23800,
  "price_low": 21578.57,
  "price_close": 23395.06,
  "volume_traded": 34773.2963436,
  "trades_count": 617317
},
}
```

```
{
  "name": "Bitcoin",
  "code": "BTC",
  "price": "$39,279.19",
  "color": "#FF0700",
  "img": "https://cryptologos.cc/logos/bitcoin-btc-logo.png"
},
{
  "name": "Litecoin",
  "code": "LTC",
  "price": "$140.13",
  "color": "#000000",
  "img": "https://upload.wikimedia.org/wikipedia/commons/thumb/f/f8/LTC-400.png/2048px-LTC-400.png"
},
{
  "name": "Ether",
  "code": "ETH",
  "price": "$1,641.17",
  "color": "#9A9A9A",
  "img": "https://w7.pngwing.com/pngs/268/1013/png-transparent-ethereum-eth-hd-logo.png"
},
{
  "name": "Ripple",
  "code": "XRP",
  "price": "$0.76",
  "change": "-$0.02 (2.6%)",
  "color": "#0002FF",
  "img": "https://w7.pngwing.com/pngs/996/864/png-transparent-coin-ripple-xrp-crypto-currency-and-coin-icon.png"
},
}
```

КЛІЄНТ-СЕРВЕРНА АРХІТЕКТУРА



MODEL–VIEW–CONTROLLER



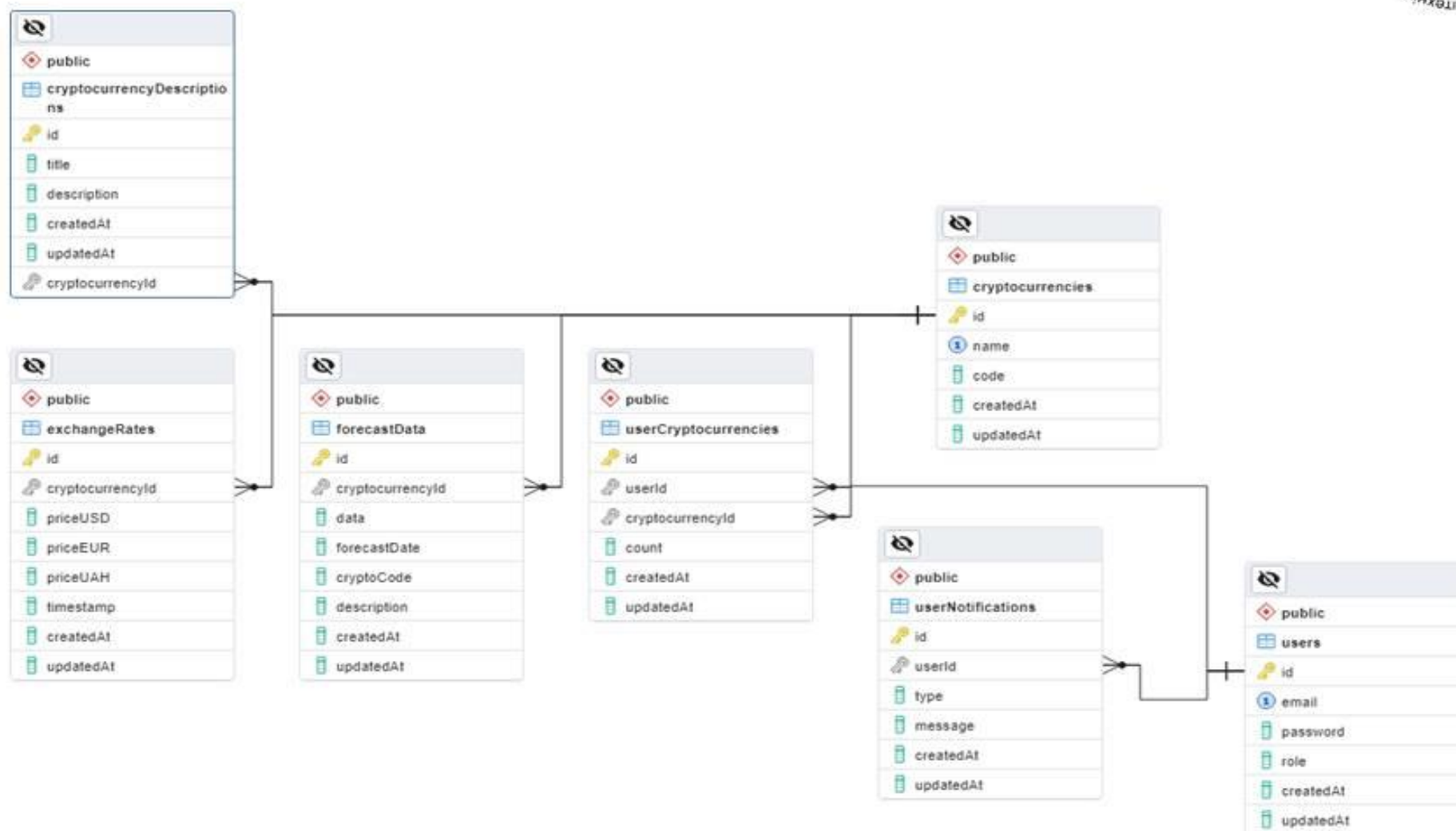
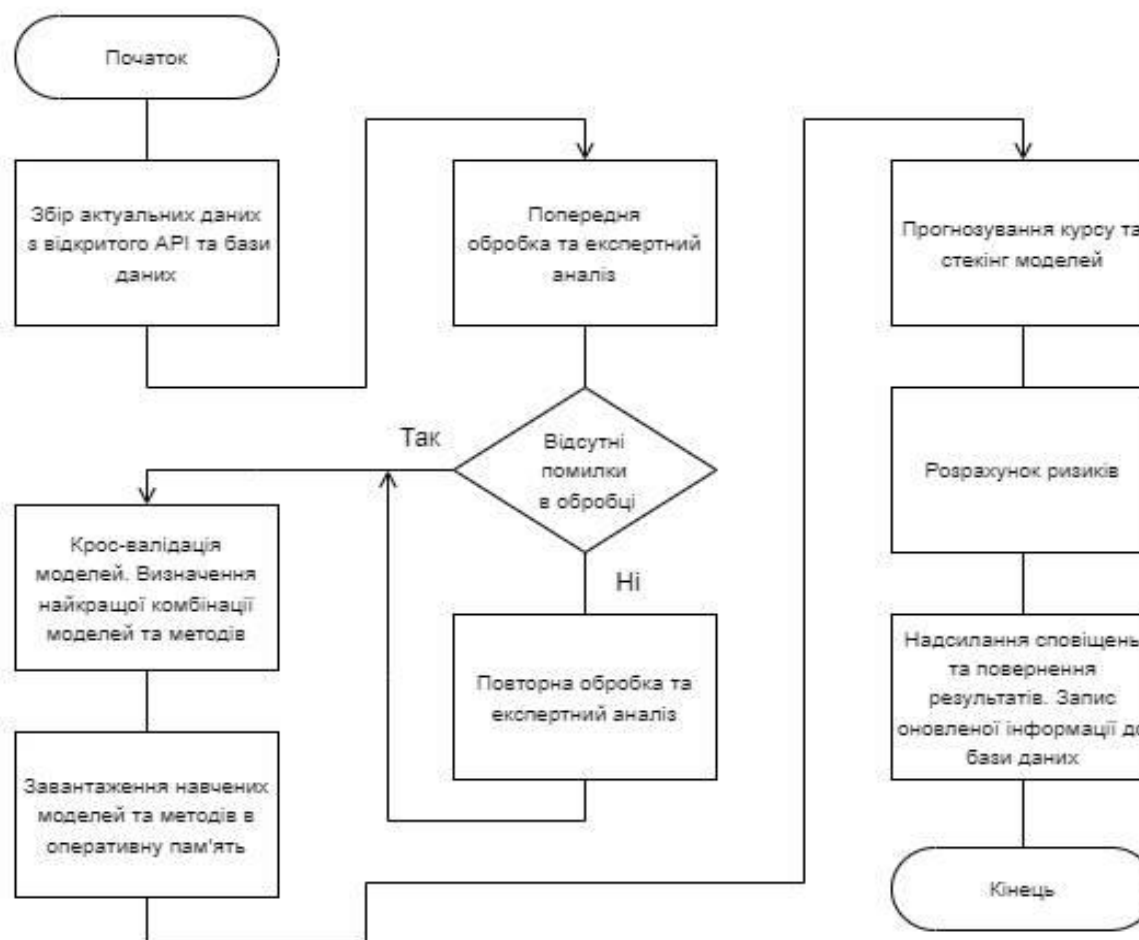


СХЕМА РОБОТИ МОДУЛЮ ПРОГНОЗУВАННЯ КУРСУ ТА РИЗИКІВ

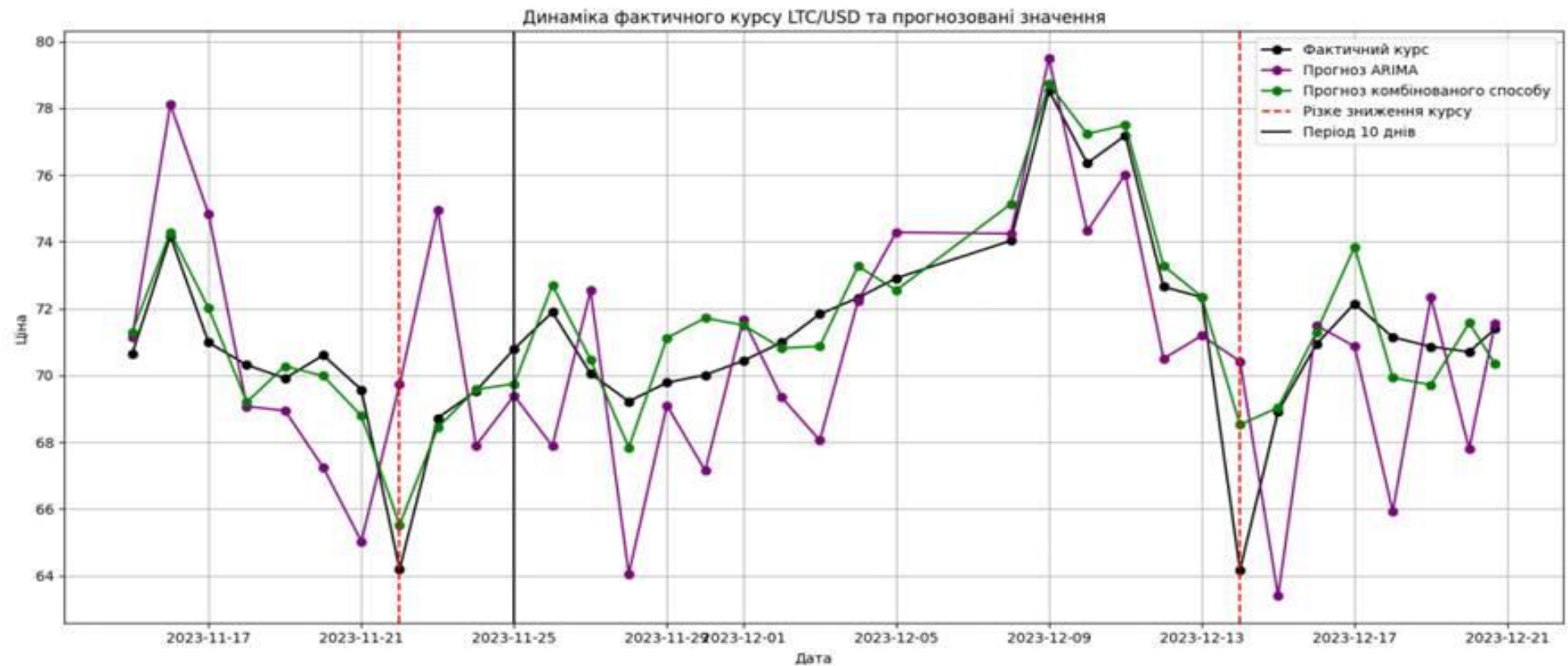


АНАЛІЗ РОЗРОБЛЕНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

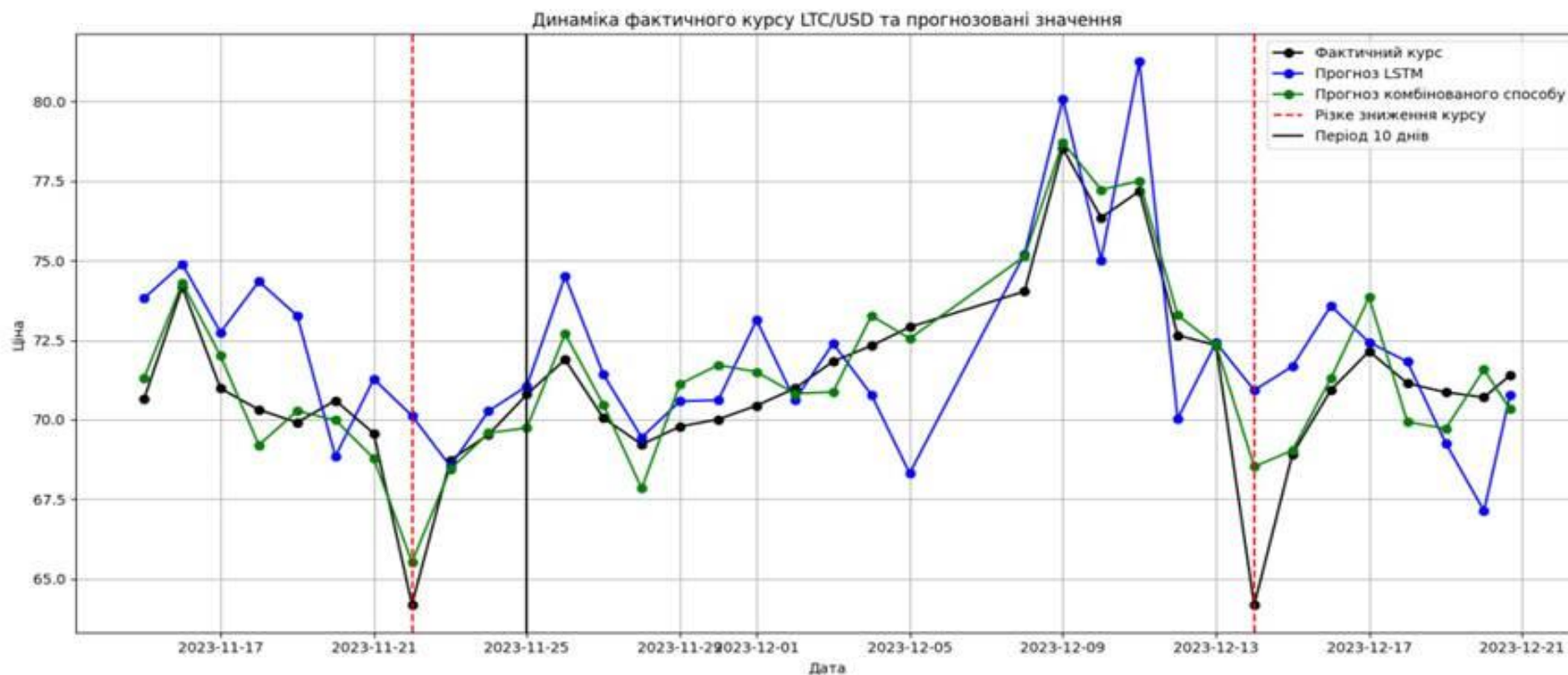
ПОРІВНЯННЯ КОМБІНОВАНОГО СПОСОБУ З БАЗОВОЮ МОДЕЛЛЮ

Довжина прогнозу, днів	Комбінований спосіб прогнозування		Прогнозування базової моделі	
	Точність прогнозу, %	Час, секунди	Точність прогнозу, %	Час, секунди
7	87,44	3,12	73,47	1,68
10	86,72	3,48	71,19	1,82
14	84,30	3,64	70,73	1,92
18	80,91	3,72	67,65	2,24
24	76,21	4,06	62,03	2,41
30	74,37	4,24	59,94	2,64
40	69,51	4,52	56,11	2,86
55	67,92	4,64	54,26	3,18

Порівняння точності комбінованого способу з ARIMA



Порівняння точності комбінованого способу з LSTM





ПОБУДОВА БІЗНЕС-МОДЕЛІ

КАНВА БІЗНЕС-МОДЕЛІ

Проблема Низька швидкість та якість або повна відсутність аналізу та прогнозування ризиків операцій на криптовалютному ринку; суб'єктивність через людський фактор;	Рішення Програмне забезпечення у вигляді веб-додатку, що надає можливість швидко та автоматизовано отримувати данні із криптовалютних бірж для аналізу та прогнозування ризиків	Унікальна ціннісна пропозиція Розроблюваний програмний продукт у вигляді веб-застосунку який автоматизує процес аналізу та прогнозування ризиків операцій з криптовалютами;	Прихована перевага Зменшення можливого ризику операцій з криптовалютами за допомогою автоматизованого аналізу та прогнозування.	Споживачі компанії які спеціалізуються на прогнозуванні ризиків, криптовалютні біржі, індивідуальні криптовалютні трейдери та власники цифрових активів
Ключові метрики Прибуток користувачів та сума придбаних платних версій застосунку з обмеженим терміном дії		Канали Спеціально відведені відділи для співпраці із ринком B2B та доходи від співпраці; взаємодія із B2C через соціальні мережі та заходи спрямовані на зацікавлення		
Структура витрат Заробітна плата (ЗП); податки на збори на ЗП; оренда приміщень; Дослідна робота та розробка (виплата заробітних плат, соціальних виплат); маркетинг та реклама; комплектуючі та обладнання; податкові платежі; інші витрати (непередбачувані поломки обладнання, інші позапланові витрати)			Потоки доходів Доходи від платної версії застосунку, комісія з отриманих прибутків	

НАУКОВА НОВИЗНА

Вперше запропоновано комбінований метод програмного прогнозування ризиків в операціях з криптовалютами, який завдяки експертному аналізу комбінує статистичні методи та алгоритми машинного навчання для збільшення точності прогнозування на 9%

ВИСНОВКИ

1. Розглянуті існуючі методи та наявні програмні рішення для прогнозування ризиків операцій з криптовалютами
2. Розроблено комбінований спосіб для автоматизованого прогнозування курсу та ризиків операцій з криптовалютами. Досліджено методи ARIMA, LSTM, CNN, Random Forest, Gradient Boosting, Value-at-Risk, Expected Shortfall та стекінг технологію.
3. Розроблена архітектура, модель бази даних та основні алгоритми реалізації. Зазначені особливості реалізації програмного забезпечення.
4. Середня точність комбінованого способу в ризикових ситуаціях вища на 9% в порівнянні з ARIMA та LSTM. Запропоновано покращення.
5. Розроблена бізнес-модель . Проведено аналіз ринку, сформовано унікальну ціннісну пропозицію розробленого програмного забезпечення

АПРОБАЦІЯ РОБОТИ

Основні положення і результати даної роботи були представлені на науковій конференції магістрантів та аспірантів

“Прикладна математика та комп’ютинг ПМК-2023”, та опубліковано у збірнику тез доповідей: Кондрацький В.В., Струцинський О.А., Спосіб та програмне забезпечення для автоматизованого аналізу та прогнозування ризиків операцій з криптовалютами.

”Прикладна математика та комп’ютинг ПМК-2023”
(Київ, 28 – 30 листопада 2023 р.), с. 474 – 480.



Доповідь завершено!
Дякую за увагу!