

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

Навчально-науковий інститут інформаційних технологій

Кафедра системного аналізу

Пояснювальна записка

до кваліфікаційної роботи
на ступінь вищої освіти бакалавр

на тему: **«РОЗРОБКА СИСТЕМИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ДЛЯ
ОПТИМІЗАЦІЇ ПРОЦЕСІВ ОБІГУ КРИПТОВАЛЮТ»**

Виконав: студент 4 курсу, групи САД-41
спеціальності 124, Системний аналіз
Вавринчук Е.М.

Керівник Штіммерман А. М.
(прізвище, ініціали)

Рецензент Сторчак К.П.
(прізвище, ініціали)

Нормоконтроль Ставицька Ю.В.
(прізвище, ініціали)

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

Навчально-науковий інститут інформаційних технологій

Кафедра Системного аналізу

Ступінь вищої освіти бакалавр

Спеціальність 124, Системний аналіз

ЗАТВЕРДЖУЮ

Завідувач кафедри
системного аналізу

— О.А.Золотухіна
“ _____ ” _____ 2020 року

З А В Д А Н Н Я НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТУ

Вавринчуку Едуарду Миколайовичу

(прізвище, ім'я, по батькові)

1. Тема роботи: Розробка системи підтримки прийняття рішень для оптимізації процесів обігу криптовалюти

Керівник роботи Штіммерман Аксенія Миколаївна

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

затверджені наказом вищого навчального закладу від “ _____ ” 2020 року № _____

2. Строк подання студентом роботи _____

3. Вхідні дані до роботи: Науково-технічна література та електронні джерела для криптовалют, математичних методів прогнозування та прогнозування методом штучного інтелекту, а саме нейронними мережами.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити).

4.1. Аналіз існуючих рішень для криптовалют, розгляд криптовалют

4.2. Пошук математичних методів прогнозування

4.3. Пошук методів прогнозування засобами штучного інтелекту(нейронні мережі)

4.4. Розробка СППР для криптовалют із використанням досліджених методів

4.5. Порівняльний аналіз розроблених систем

5. Перелік графічного матеріалу

1. Об'єкт, предмет, мета бакалаврської роботи

2. Актуальність бакалаврської роботи

3. Завдання на бакалаврську роботу

4. Пошук існуючих рішень

5. Аналіз популярних криптовалют

6. Механізм передбачення Хольта-Вінтерса і методи прогнозування нейрон. мережами

7. Система підтримки прийняття рішення

6. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Підбір науково-технічної літератури	01.03.2020	Виконав
2	Пошук існуючих рішень систем для криптовалют	15.03.2020	Виконав
3	Пошук існуючих засобів прогнозування математичними моделями	12.04.2020	Виконав
4	Пошук існуючих засобів прогнозування штучним інтелектом	20.04.2020	Виконав
5	Розробка СППР із використанням математичної моделі	5.05.2020	Виконав
6	Розробка СППР із використанням штучного інтелекту	15.05.2020	Виконав
7	Порівняльний аналіз роботи двох методів	31.05.2020	Виконав
8	Розробка обов'язкових демонстраційних матеріалів	02.06.2020	Виконав
9	Попередній захист роботи	04.06.2020	Виконав
10	Здача роботи в деканат	15.06.2020	Виконав

Студент

(підпис)

Керівник роботи

(підпис)

Вавринчук Е.М.

(прізвище та ініціали)

Штіммерман А.М.

(прізвище та ініціали)

РЕФЕРАТ

Текстова частина бакалаврської роботи 40 с., 8 табл., 32 рис., 1 дод., 17 джерел.

БЛОКЧЕЙН, КРИПТОВАЛЮТА, НЕЙРОННА МЕРЕЖА, МАТЕМАТИЧНІ МЕТОДИ ПЕРЕДБАЧЕННЯ, ЧАСОВІ РЯДИ, ДОВГОТРИВАЛА МОДЕЛЬ ПАМ'ЯТІ, СИСТЕМА ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕННЯ.

Об'єктом дослідження даної роботи є методи передбачення курсів, що використовуватимуться в аналізі курсів для криптовалют.

Предметом дослідження виступають математичні методи передбачення зміни курсів а також передбачення використовуючи можливості штучного інтелекту.

Мета роботи передбачає розробку системи, що буде проводити аналіз інформації та надавати подальшу інформацію користувачеві, щодо майбутніх змін курсу криптовалют із мінімальною похибкою зміни курсу використовуючи методи передбачення.

Методами дослідження являлись аналіз існуючих рішень, пошук методів передбачення змін курсу математичними методами та методами штучного інтелекту.

Досліджено методи для передбачення, як математичні так і засобами штучного інтелекту і визначено для аналізу даних математичні методи на основі часових рядів та методи штучного інтелекту на основі довготривалої моделі пам'яті.

На основі результатів виконаних досліджень розроблено систему підтримки прийняття рішення а також проведено порівняльний аналіз розроблених систем, що надасть користувачеві зрозумілу інформацію щодо змін курсу криптовалют.

Впровадження розробленої методики дозволяє надати користувачу інформацію щодо змін курсу валют на основі попередніх значень курсу.

ЗМІСТ

ВСТУП	8
1 АНАЛІЗ РІШЕНЬ НЕОБХІДНИХ В РОЗРОБЦІ СППР ДЛЯ ОПТИМІЗАЦІЇ ОБІГУ КРИПТОВАЛЮТИ	9
1.1 Аналіз існуючих рішень для криптовалют	9
1.2 Аналіз популярних криптовалют	11
1.3 Аналіз методів передбачення математичними методами	19
1.4 Аналіз методів передбачення методами штучного інтелекту	22
2 РОЗРОБКА СИСТЕМИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕННЯ ДЛЯ ОПТИМІЗАЦІЇ ОБІГУ КРИПТОВАЛЮТ	25
2.1 Загальна структура систем підтримки прийняття рішення для оптимізації обігу криптовалют	25
2.2 Розробка СППР із використанням математичної моделі	26
2.3 Розробка СППР із використанням штучного інтелекту	32
3 ПОРІВНЯЛЬНИЙ АНАЛІЗ СППР ДЛЯ ОПТИМІЗАЦІЇ ОБІГУ КРИПТОВАЛЮТ	39
3.1 Визначення характеристик ефективності СППР для оптимізації обігу криптовалют	39
3.2 Розрахунок ефективності розроблених СППР для оптимізації обігу криптовалют	40
ВИСНОВОК	45
ПЕРЕЛІК ПОСИЛАНЬ	46
ДОДАТОК А	48
ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (ПРЕЗЕНТАЦІЯ)	58

ВСТУП

Бакалаврська робота присвячена аналізу та оптимізації процесів обігу криптовалют.

Робота над даною роботою є актуальною, оскільки недоліком існуючих рішень для криптовалют являються непередбачуванність курсу, що притаманний звичайним грошам, а значить не має демонстрації курсу на майбутні періоди, тобто як буде поводитись курсу протягом наступних декількох періодів.

Об'єктом дослідження даної роботи є методи передбачення курсів, що використовуватимуться в аналізі курсів для криптовалют.

Предметом дослідження виступають математичні методи передбачення зміни курсів а також передбачення використовуючи можливості штучного інтелекту.

Мета роботи передбачає розробку системи, що буде проводити аналіз інформації та надавати подальшу інформацію користувачеві, щодо майбутніх змін курсу криптовалют із мінімальною похибкою зміни курсу використовуючи методи передбачення.

Поставлена мета вимагає рішення низки завдань:

- Аналіз існуючих рішень для криптовалют;
- Аналіз популярних криптовалют;
- Пошук методів передбачення математичними методами;
- Пошук методів передбачення методами штучного інтелекту;
- Розробка системи підтримки прийняття рішення;
- Провести порівняльний аналіз двох методів.

Практична значущість результатів дослідження може полягати в можливості подальшого аналізу курсів для різних криптовалют на основі наведених рішень та використання даного методу для інших рішень.

1 АНАЛІЗ РІШЕНЬ НЕОБХІДНИХ В РОЗРОБЦІ СППР ДЛЯ ОПТИМІЗАЦІЇ ОБІГУ КРИПТОВАЛЮТИ

1.1 Аналіз існуючих рішень для криптовалют

Серед існуючих рішень будуть розглянуті системи що працюють із курсами криптовалют, тобто збирають курси валют, які потрібні будуть нам для подальшого аналізу цих даних в передбаченні курсів. Перше існуюче рішення - CoinMarketCap, що зображений на рисунку 1.1.



Рисунок 1.1- система CoinMarketCap.

Було визначено, що сервіс сам зберігає курси за кожен день, а значить можемо брати курси валют для аналізу даних із даного сервісу.

Наступним рішенням є Binance, що зображений на рисунку 1.2. Даний сервіс надає інформацію щодо зміни курсу щохвилини, як і наступний сервіс Coinbase, який зображений на рисунку 1.3. Найбільш корисним сервісом, що буде використовуватись як постачальник даних про курс валют для механізму передбачення.

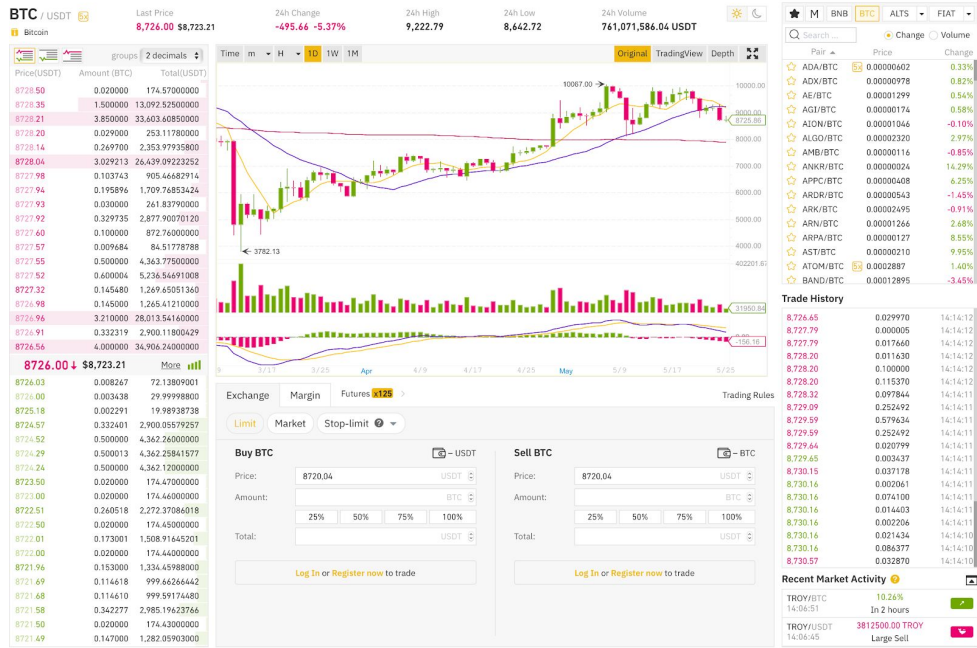


Рисунок 1.2 - система Binance



Рисунок 1.3 - система Coinbase

Далі розглянемо криптовалюту та виберемо одну для якої буде проводитись розробка механізму передбачення як приклад, який потім може бути використаних для будь-яких інших криптовалют.

1.2 Аналіз популярних криптовалют

Перша криптовалюта, що буде розглянута, є Bitcoin. Базовою складовою bitcoin є блок - одинична порція пов'язаних даних (інформації). Блокчейн Bitcoin - є реєстром або книгою (ledger), в якій записано абсолютно всі транзакції, що відбулись з монетами bitcoin.[1]

І якщо блокчейн - це бухгалтерська книга, то окремі сторінки цієї книги - це і є блоки блокчейна, в яких записуються фінансові транзакції і які пов'язані між собою за допомогою хешів так, що в заголовку кожного наступного блоку зберігається хеш-код попереднього.

У цій величезній бухгалтерській книзі неможливо ні видалити сторінку (блок), ні змінити її зміст, оскільки відразу ж зміниться зміст заголовків всіх наступних сторінок (блоків) і будь-який учасник розподіленої мережі bitcoin, у якого зберігається копія цієї книги-блокчейна, відразу ж помітить спробу підміни. Власне, сама мережа біткоіна це помітить автоматично і відкине спроби внести всі зміни в блокчейн.

Таким чином, всі блоки, які пишуться в блокчейн, не можуть змінюватись. Редагування чи зміна інформації, що вже міститься в блокчейн заборонена. Інформація являє собою транзакції. Блокчейн дозволяє лише запис нових блоків, що вже змінюють інформації на основі попередніх блоків.

Ця особливість притаманна лише блокчейну, що виступає в ролі децентралізованого реєстру для будь-яких фінансових транзакцій. І завдяки цьому блокчейн є набагато надійнішим ніж бухгалтерські книги чи, наприклад, реєстр операцій банку, адже блокчейн зберігає всю інформацію про всі блоки на кожному із комп'ютерів в мережі, які виступають в ролі серверів з інформацією, в децентралізованій мережі. [2]

Тепер розглянемо структуру блоку. Блок в блокчейні повинен містити заголовок(Head), що містить в собі службову інформацію. Також є корисна

інформація (Payload), що представляє собою тіло блоку, тобто інформацію, що блок зберігає. Структура блоків в блокчейн зображена на рисунку 1.4.

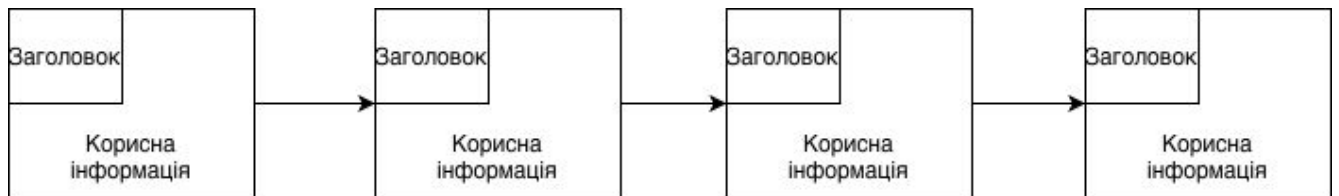


Рисунок 1.4 - Блоки в блокчейн і їх зв'язки між собою.

У заголовку блоку міститься наступна інформація:

- версія блоку;
- дата і час створення блоку;
- хеш-код заголовка блоку;
- хеш-код попереднього блоку;
- хеш-код всіх транзакцій в блоці;
- спеціальні параметри nonce і bits, ці змінні записуються коли відбувається майнінг;

Для розуміння нам важливі хеші в заголовку.

Хеш-код, що в заголовку блоку, виступає засобом, що з'єднує попередній блок та наступний блок в ланцюг блокчейна. Також хеш-код для нового блоку генерується із хеш-коду попереднього блоку.

Заголовок блоку зберігає хеш-код для транзакцій, що містяться в цьому блоці. Даний хеш-код транзакцій розраховується за допомогою алгоритму, що називається бінарне дерево хешей, або дерево Меркла (Merkle tree) і зображене на рисунку 1.5.

Генерування хеш-коду транзакцій будується наступним чином:

1. Обраховуються хеші для транзакцій, що містяться в даному блоці.
2. Обраховуються хеші від сум хеш-пар транзакцій блоку.
3. Обраховуються хеші від суми пар хешей із подальшим перерахунком, поки не отримаємо єдиний хеш код.

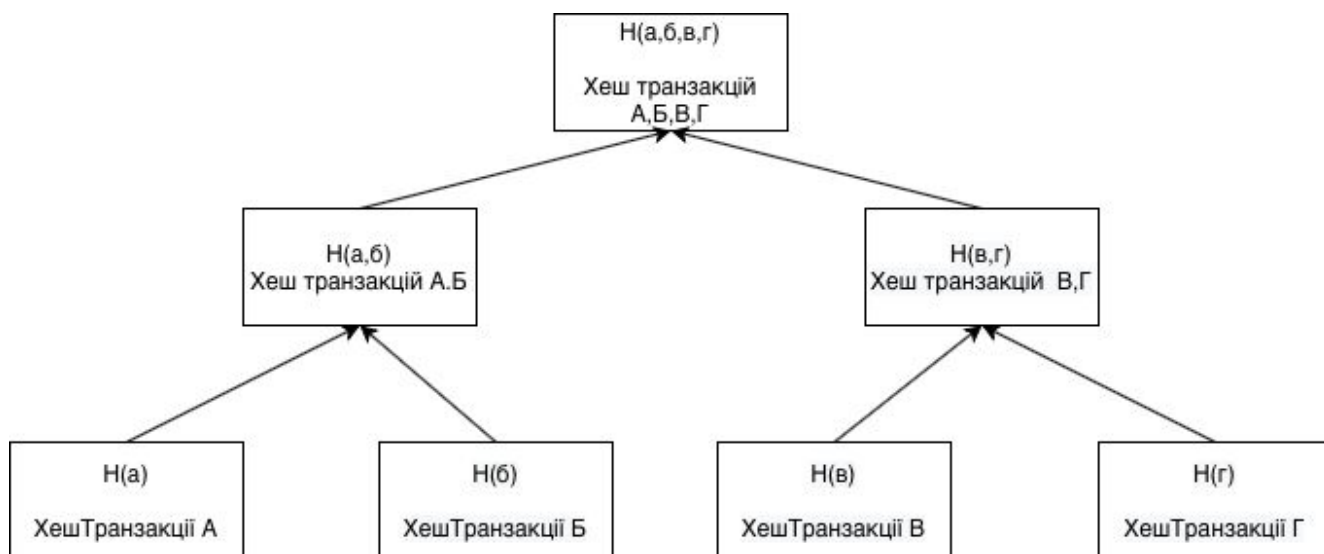


Рисунок 1.5 - Бінарне хеш-дерево.

Тут слід врахувати, що оскільки дерево бінарне (підрахунок йде парами), то на кожному кроці має бути парне число елементів. Тому якщо, на якомусь етапі виходить непарна кількість хешей, то останній просто дублюється для отримання пари.

Заголовки виконують функцію відслідковування цілісності кожного блоку.

Тепер перейдемо до власне вмісту (Payload) блоку. Як вже говорилося раніше, в блокчейн біткоїни записуються транзакції. Власне сама транзакція має вигляд:

З <адреса 1> відправити <N> біткоінів на <адреса 2>

Насправді транзакція, записана в блок блокчейна, складніша, оскільки сам протокол біткойна керує Входами (Inputs) і Виходами (Outputs).

Входи (In) - це транзакції, які поповнюють біткоін-адреса, а Виходи (Out) - це суми, які переводяться на інші біткоіни-адреси.

Тому в біткоін нова транзакція через Входи (один або кілька) посилається на Виходи (один або кілька) попередніх транзакцій і формує Виходи (може бути різна кількість виходів), що будуть використовуватись в подальших транзакціях, як зображено на рисунку 1.6.

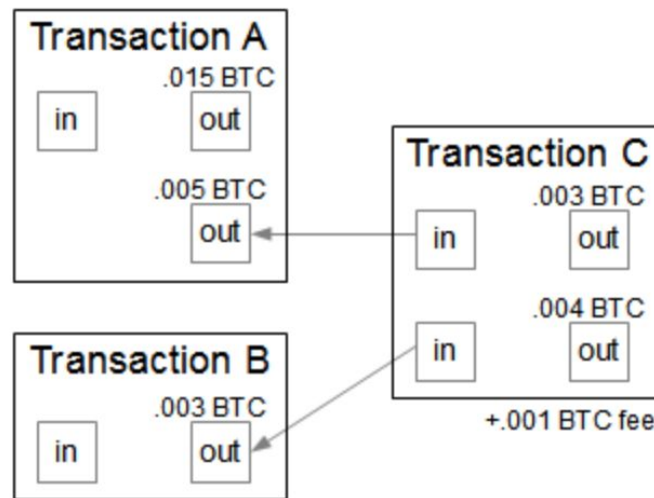


Рисунок 1.6 - Приклад транзакцій з входами і виходами та комісією.

Власне структура запису транзакцій в блокчейн біткоіна містить:

- кількість використовуваних входів;
- хеш-код і індекс кожного Входу, а також службова інформація;
- кількість використовуваних виходів;
- суми Виходів, а також службова інформація;
- мітка часу транзакції.

Ця, на перший погляд складна конструкція, насправді показує реальний рух грошей між біткоін-адресами і дає можливість досить швидко підрахувати баланс будь-якої адреси на підставі його Входів і Виходів.

Можливість вказати відразу кілька Виходів в транзакції - це дуже важлива річ, тому що транзакцію (точніше - її Вихід) можна використовувати як Вхід тільки один раз і тільки цілком.

Ще важливий момент - це так звана комісія за транзакцію (transaction fee), яку отримують майнери - люди, які записують нові блоки транзакцій в блокчейн.

Далі розглянемо наступний популярний блокчейн - Ethereum. По суті, це публічна база даних, яка веде постійний запис цифрових транзакцій. Важливо відзначити, що ця база даних не вимагає яких-небудь централізованих повноважень для її обслуговування та захисту. Замість цього вона працює як «не

підтверджена» транзакційна система - структура, в якій фізичні особи можуть здійснювати однорангові транзакції без необхідності довіряти третій стороні або один одному.

Ethereum реалізує парадигму блоку. Blockchain Ethereum по суті є transaction-based state machine. В інформатиці, стан машини посилається на щось, що буде читати ряд inputs і, на підставі цих inputs, буде зроблений перехід до нового стану. У Ethereum state machine ми починаємо з «стану генезису». Це буде аналогічно чистому аркушу або порожньому бланку, до того поки будь-які транзакції не відбудуться в мережі. Коли транзакції виконуються, "стан генезису" переходить в якийсь новий кінцевий стан. У будь-який момент часу цей кінцевий стан представляє поточний стан Ethereum. [3]

Основні компоненти які входять систему Ethereum:

- Аккаунт (accounts)
- Газ і оплата (gas and fees)

Ethereum складається з безлічі невеликих об'єктів акаунтів (облікових записів), які можуть взаємодіяти один з одним через інфраструктуру передачі повідомлень. У кожного аккаунта є пов'язаний з ним стан і 20-байтова адреса. Адреса в Ethereum - це 160-бітний ідентифікатор, який використовується для ідентифікації будь-якого облікового запису.[4]

Існує два типи акаунтів, які зображені на рисунку 1.7:

- Externally owned accounts (зовнішній аккаунт) - контролюються закритими ключами і не мають коду, пов'язаного з ними.
- Contract accounts (Аккаунт контракту) - контролюються кодом контракту і мають код асоційований з ним.

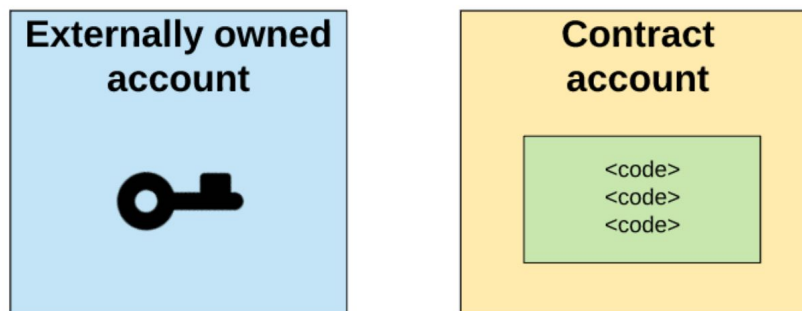


Рисунок 1.7 - Типи акаунтів.

Одна дуже важлива концепція в Ethereum - концепція комісій. Кожне обчислення, яке відбувається в результаті транзакції в мережі Ethereum, бере плату! Ця плата виплачується в номіналі «gas (газ)».

Gas - це одиниця, використовувана для вимірювання зборів, необхідних для конкретного розрахунку. Gas прайс - це кількість Ether, яке ви бажаєте витратити на кожну одиницю gas, вимірюється в «gwei». «Wei» - це найменша одиниця ефіру, де 10^{18} Wei = 1 Ether, а один gwei = 1 000 000 000 Wei.

Кінцевий блокчейн який буде розглянуто це EOS. EOS це останнє оновлення в блокчейн проектах, який здатний увійти в конкурентний криптовалютний ринок. Він спрямований на створення мережі, яка може обробляти мільйони транзакцій в секунду. Замість фіатних грошей ICO використовує для залучення коштів такі популярні криптовалюти як Ethereum або Litecoin. [5]

Основне завдання EOS створити децентралізований блокчейн, який здатний виконувати швидкі і безкоштовні транзакції. EOS не тільки дозволяє створювати смарт-контракти, а й дозволяє розробникам випускати DApps. EOS також намагається розробити платформу, яка працює як операційна система, що зробить її надзвичайно простою у використанні.

Ще одна мета EOS - виконувати мільйони безкоштовних транзакцій за лічені секунди. Це може вирішити велику проблему. Хоча і інші блокчейни здатні

виконувати смарт контракти, але жоден з них не може обробляти їх з достатньою швидкістю.

Як ми вже знаємо, як EOS, так і Ethereum здатні розгортати децентралізовані додатки, які підтримуються смарт-контрактами. Однак істотна відмінність полягає в обсязі транзакцій який кожна мережа може обробляти в певний момент часу.

Це може бути відомо також як "масштабованість" і є однією з найважливіших речей, яку слід враховувати при оцінці потенціалу блокчейна.

Як і будь-який інший блокчейн, EOS також має свою власну криптовалюту, під назвою EOS.

Якщо ви використовуєте блокчейн Ethereum, вам прийдеться кожного разу платити певну комісію за транзакцію під назвою GAS. І оскільки все більше і більше людей використовують мережу, вона з часом стає все дорожче.

Ця концепція є причиною того, що Ethereum стає непридатним для використання на глобальному рівні, оскільки на ньому не вигідно робити невеликі транзакції. Коли користувачеві необхідно використовувати мережу, йому доводиться орендувати EVM (Ethereum Virtual Machine), для якої потрібно комісія за кожену транзакцію.

EOS трохи відрізняється. Замість того, щоб «здавати в оренду» обчислювальну потужність, власники монет отримують право власності на частину мережі. Говорячи простими словами, якщо у вас є 1% токенів EOS, ви володієте 1% мережі. Це означає, що для виконання своїх транзакцій у вас буде 1% необхідної обчислювальної потужності. Таким чином, всі транзакції на EOS без комісій.

Токен EOS працює так само, як і будь-яка інша криптовалюта - ви можете відправляти, тримати і отримувати кошти між крипто гаманцями. Це робить його ефективною платіжною системою, так як ви можете абсолютно безкоштовно робити переклади кому завгодно в усьому світі протягом кількох секунд.

Проаналізувавши дані криптовалюти я обрав EOS криптовалюту як саму швидко-розвиваючу та найпопулярнішу на даний час.

Завершуючи аналіз криптовалют в таблиці 1.1 відображена коротка характеристика для трьох криптовалют.

Таблиця 1.1 - Порівняння криптовалют

Характеристика	Bitcoin	Ethereum	EOS
Модель токена	Інфляційна	Інфляційна	Інфляційна
Модель оплати транзакцій	Bitcoin	Gas	Bandwidth (потужності)
Швидкість транзакцій	78 хв	6 хв	1.5с
Віртуальна машина	-	EVM(Ethereum Virtual Machine)	стандарт WebAssembly
Мова розробки контрактів	-	Solidity	C++/Rust
Консенсус	Proof-of-Work	Proof-of-Work	Delegated Proof-of-Work
Ресурс консенсуса	Майнингові пули	Майнингові пули	21 блок-продюсер

Згідно таблиці 1.1 всі три криптовалюти мають інфляційну модель токена і означає, що токени довивускаються для видачі премій майнерам. Щодо моделі оплати транзакцій в Bitcoin використовується монета bitcoin. Для Ethereum використовується газ. Кожна транзакція коштує певну кількість газу, але кількість газу обмежується блоком. Тому неможливо створити смарт-контракт із великою кількістю газу, і це забезпечує систему від спаму. В EOS транзакції умовно безкоштовні, адже газу в EOS не існує. Замість нього використовується потужність мережі (Bandwidth), яка для кожного учасника мережі видається в залежності від того яку кількість EOS він має. Щодо швидкості транзакцій в

мережі Bitcoin, то генерування блоку займає 78 хвилин. Щодо Ethereum та EOS 6 хвилин та 1.5 секунд відповідно. Віртуальної машини для Bitcoin не існує оскільки не існує підтримки смарт-контрактів. Для Ethereum використовується EVM (Ethereum Virtual Machine) і мова написання контрактів Solidity а для EOS використовується стандарт WebAssembly, що є більш сучасним і універсальним, оскільки використовує 2 мови написання смарт-контрактів - C++ і Rust. Щодо консенсусу, для Bitcoin і Ethereum обмеженим ресурсом виступає обчислювальні можливості майнерів - Proof-of-Work і для цього використовуються різноманітні майнінгові пули. Для EOS характерним є Delegated Proof-of-Work, в якому обмеженим ресурсом є голоси учасників мережі адже використовується 21 блок-продюсер, який має голос.

1.3 Аналіз методів передбачення математичними методами

Розглянемо рішення задачі прогнозування за допомогою математичних моделей. Було обрано прогнозування на основі часових рядів. Прогнозування на основі часових рядів – один із самих популярних підходів до прогнозування розвитку економічних процесів, об'ємів торгових операцій, об'ємів виробництва та накопичення продукції на складах, оцінювання альтернативних економічних стратегій, формування бюджетів підприємств та держави, прогнозування та менеджмент економічних і фінансових ризиків та інше. В нашому випадку буде використовуватись для прогнозування економічних процесів а саме їх можлива зміна.[6]

Серед характеристики часових рядів можна виділити:[7]

1. Сезонність:

1.1. Повні ряди - коли дані є мінімум за 1 повний цикл, тобто мінімум за 12 місяців, або за 4 кварталу і можна виділити сезонність;

1.2. Неповний ряд - коли даних ще немає за повний цикл;

2. Зріст:

2.1. Часовий ряд з ростом, коли можна виділити явне зростання курсу від року до року.

2.1.1. Зростання може бути постійний;

2.1.2. Зростання може бути затухаючий;

2.1.3. Зростання може бути експонентний;

2.1.4. Зростання може змінюватися падінням;

2.2. Статичний часовий ряд - це часовий ряд, який не містить зростання;

3. Важливість тенденцій для прогнозу:

3.1. Для прогнозу можуть бути важливі тенденції за тривалий період;

3.2. Для прогнозу можуть бути важливі тенденції за останній період;

4. Нерегулярні, разові зміни.

При вивченні методів прогнозування числових рядів я дійшов висновку використовувати Метод Хольта. Метод Хольта-Вінтерса - це модель прогнозування, що може враховувати наступні параметри:[8]

– згладжений експонентний ряд;

– сезонність.

– тренд;

Розрахування експоненціально-згладженого ряду виконується згідно формали:

$$L_i = \frac{k \cdot y_i}{S_{i-s}} + (1+k) \cdot (L_{i-1} - T_{i-1}), \quad (1.1)$$

де S_{i-s} – коефіцієнт для сезонності попередніх періодів.

Для першого періоду сезонність завжди дорівнює 1.

Визначення значення тренда

Сезонність оцінюється за формулою:

$$S_i = \frac{q \cdot y_i}{L_i} + (1 - q) \cdot S_{i-s}, \quad (1.2)$$

де S_i - коефіцієнт для сезонності поточного періоду;

q - коефіцієнт згладжування сезонності (знаходиться в межах від 0 до 1);

S_{i-s} - коефіцієнт сезонності за цей же період в попередньому сезоні.

Здійснюється прогнозування за методом Хольта-Вінтерса Прогноз на p періодів вперед розраховується за формулою:

$$\hat{Y}_{i-p} = (L_i + p \cdot T_i) \cdot S_{i-s+p}, \quad (1.3)$$

де Y_{i-p} - прогноз на p періодів вперед розраховується за формулою;

S_{i-s+p} - коефіцієнт для сезонності за цей же період в останньому сезоні.

Метод передбачає певні переваги. Складається метод Хольта-Вінтерса із трьох основних складових елементів: сезонність, експоненціально-згладжений ряд та саму тенденцію розвитку. При використанні тренду з експоненційним згладжуванням можна згладити дрібні коливання в ряду динаміки для пошуку приватних спадів і стрибків, а не тільки виявити напрямок розвитку ряду динаміки. Сезонність в методі дозволяє спрогнозувати значення на наступні періоди з урахуванням цієї сезонності, що буде наочно відображати побудований прогноз.[9]

Із використанням даного методу є можливість будувати прогнози протягом великого проміжку часу, адже він враховує декілька факторів при прогнозуванні і означає, що він буде більш точним при прогнозуванні на більший проміжок часу. Але метод має і певні обмеження.

Аби побудувати прогноз більш точно для цього потрібно буде багато даних протягом певного часу. Аби робити прогнози більш точно потрібно використовувати дані, що забезпечують 4-5 періодів. Є певна залежність, чим

більше даних було надано для прогнозування, тим на довший період може бути розроблений прогноз.

Метод потребує проведення додаткову оцінку точності прогнозу. Алгоритм, що був наведений призначений для оцінки прогнозу, адже коефіцієнти k , b і q підбираються користувачем вручну а значить впливають на оцінку точності.

Метод Хольта-Вінтерса призначений для використання:[10]

– при стратегічному плануванні: при побудові тренду може враховувати динаміку(низхідна/висхідна) досліджуваного явища.

– при оперативному і тактичному плануванні: визначена сезонна зміна може надати інформацію про нерівномірний розподіл по місяцях відносно динаміки. Експоненціальне згладжування в цьому випадку враховує спади а також підйоми динаміки і використовується воно для виявлення спадів і підйомів (при тактичному плануванні) і бути готовим до них.

Застосування даного методу передбачає велику сферу можливості застосуванн і призначений для аналізу великого обсягу статистичних даних. Також метод Хольта-Вінтерса може бути використаний при прогнозуванні використовуючи комбінування з експертними методами прогнозування.[11]

1.4 Аналіз методів передбачення методами штучного інтелекту

Розглянемо передбачення за допомогою штучного інтелекту і знову ж за допомогою аналізу часових рядів. Після проведення пошукових робіт, я визначив, що непоганою ідеєю реалізації передбачення може використовуватись механізм довготривалої моделі пам'яті - Long Short-Term Neural Network (LSTMs). Ця модель організована в комірки, які включають кілька операцій. LSTM має змінну внутрішнього стану, яка передається від однієї комірки до іншої та модифікується операційними воротами, як зображено на рисунку 1.8.

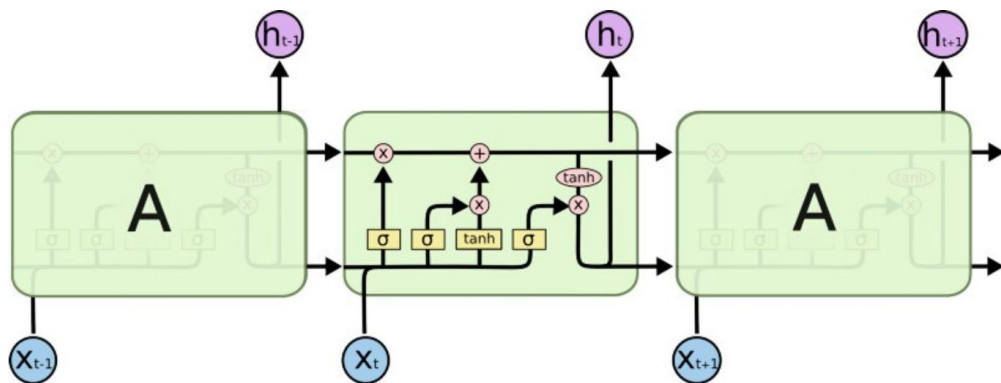


Рисунок 1.8 - Архітектура LSTM.

Архітектура LSTM передбачає 3 блоки:

- Правило входів - це сигмоподібний шар, який приймає вихід у $t-1$ та поточний вхід у момент t і з'єднує їх в єдиний тензор і застосовує лінійне перетворення з подальшим сигмоїдом. Через сигмоїду вихід цього затвору становить від 0 до 1. Це число множимо на внутрішній стан, і саме тому затвор називається воротом забуття. Якщо $f_t = 0$, то попередній внутрішній стан повністю забутий, тоді як, якщо $f_t = 1$, він буде переданий без змін.

- Вхідний затвор приймає попередній вихід і новий вхід і передає їх через інший сигмоподібний шар. Цей затвор повертає значення між 0 і 1. Значення вхідного шлюзу множиться на висновок шару-кандидата.

- Вихідний затвор керує тим, яка частина внутрішнього стану передається на вихід, і він працює аналогічно іншим воротам

Ці три правила, описані вище, мають незалежні ваги та ухили, отже, мережа дізнається, яку кількість минулого виходу потрібно зберегти, скільки поточного входу зберегти та скільки внутрішнього стану направити на вихід. [9]

У періодичній нейромережі ви не тільки надаєте мережі дані, але й стан мережі за одну мить до цього, Наприклад - обробка відео, де знадобиться періодична нейронна мережа. Те, що відбувається в поточному кадрі, сильно залежить від того, що було в останньому кадрі фільму більшу частину часу.

Протягом певного періоду періодична нейронна мережа намагається навчитися зберігати та скільки зберігати від минулого та скільки інформації зберігати від теперішнього стану, що робить її більш потужною порівняно з простою нейронною мережею.

2 РОЗРОБКА СИСТЕМИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕННЯ ДЛЯ ДЛЯ ОПТИМІЗАЦІЇ ОБІГУ КРИПТОВАЛЮТ

Системи підтримки прийняття рішень (СППР) в бакалаврській роботі полягає в розробці системи, що буде призначена для допомоги та підтримки діяльності людини при прийнятті рішення проблеми невизначеності курсів криптовалют. Застосування СППР забезпечує виконання аналізу області криптовалют, в яких даними виступають попередні курси криптовалют, та надання схеми зміни майбутніх курсів протягом наступного періоду.

Сама СППР було визначено задля ефективності обробки та аналізу інформації. В СППР бакалаврської роботи вкладено метод математичного моделювання та прогнозування та прогнозування засобами штучного інтелекту.

2.1 Загальна структура систем підтримки прийняття рішення для оптимізації обігу криптовалюти

Система підтримки прийняття рішення реалізовуватиметься у вигляді Python застосунку, який отримуватиме інформацію, а саме, актуальні курси криптовалют із веб-застосунку, CoinMarketCap, який було визначено раніше. Методом взаємодії та обробки результатів із веб-застосунку використовуватиметься python бібліотека pandas, що застосовується для моделювання даних та їх аналізу. Важливим етапом системи є збереження інформації під час її роботи, тому зберігатимуться саме дані, що отримуються із веб-додатку та подальше її оновлення під час роботи системи. Інформація зберігатиметься в табличному варіанті в форматі .csv. Далі реалізується модуль передбачення згідно алгоритму - математичної моделі чи штучного інтелекту.

Кінцевим результатом при роботі системи будуть покрокові відображення аналізу даних у вигляді графіків курсу які представлені в аналізі існуючих рішень.

Реалізуватиметься ця можливість завдяки бібліотеки matplotlib, що використовуються для візуалізації даних двовимірною 2D графікою.[12]

Загальна схема структури системи відображається на рисунку 3.1. Модуль передбачення буде розглянуто детальніше в кожному із моделей.

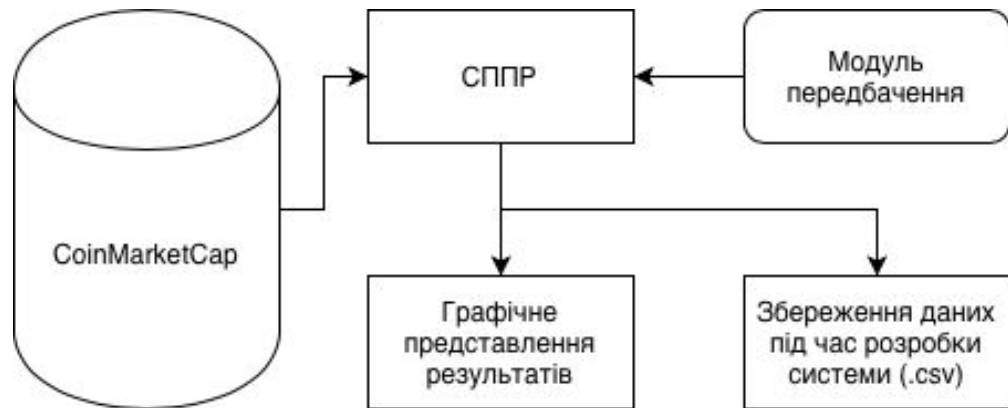


Рисунок 3.1 - Схема структури СППР

2.2 Розробка СППР із використанням математичної моделі

Основна задача в розробці системи це визначення схеми роботи модулю передбачення. Як було визначено, математичним методом передбачення буде використовуватись метод Хольта-Вінтерса.

Розглянемо детальніше схему модулю передбачення із використанням математичної моделі, що зображена на рисунку 3.2.



Рисунок 3.2 - Схема модулю передбачення Хольта-Вінтерса

Згідно рисунку 3.2 на модуль передбачення подаються самі дані із значенням дата і курс на цю дату, передаються періоди протягом яких буде проводитись аналіз, передається тренд та тип сезонності - адитивний чи мультиплікативний. При передачі даних на модуль відбувається ініціалізація та заповнення моделі необхідними даними із переданих й відбувається оптимізація даних згідно типу сезонності входу. Після проведення оптимізації відбувається саме обчислення згідно формул Хольта-Вінтерса, що відобразить лінію змін курсу криптовалют по формулі відносно оригінального курсу, це буде вважатись ефективністю моделі. Після обчислень відбувається генерування результатів в структуру даних `HoltWintersResult`, із якої буде викликатись метод прогнозування на основі підготовлених даних. Завершальним етапом моделі буде повернення структури `HoltWintersResult` із якої в подальшому буде будуватись графік курсів відносно оригінального курсу і сама крива передбачення.

Розробка з використанням математичної моделі, а саме, моделі Хольта-Вінтерса буде проводитись в декілька етапів. Кожен етап моделі розширює можливості попередніх, адже дана модель будується саме на основі попередніх моделей і їх покращень.

Перше, що потрібно, це імпортувати бібліотеки, які необхідні для роботи, як зображено на рисунку 3.3.

```
3 # імпорт бібліотек
4 from datetime import date
5 import requests
6 import pandas as pd
7 import matplotlib.pyplot as plt
8
```

Рисунок 3.3 - Імпорт бібліотек

Сама система починає роботу із отримання даних із веб-застосунку. Реалізація цієї частини відображена на рисунку 3.4.

```

9 # Отримання поточної дати
10 today = date.today()
11
12 # Формування URL із поточною датою
13 url = 'https://coinmarketcap.com/currencies/eos/historical-data/?start=20170101&end=20200526' + today.strftime("%d%m%Y")
14 # Отримання контенту сторінки
15 html = requests.get(url).content
16 # Отримання списку таблиць зі сторінки
17 df_list = pd.read_html(html)
18 # Отримання таблиці із курсом валют
19 df = df_list[2]
20

```

Рисунок 3.4 - Отримання початкових даних курсу криптовалют

Після отримання даних, результуючі дані виглядатимуть наступним чином, як зображено на рисунку 3.5. для подальшого використання цих даних необхідна певна трансформація відображення даних - індексом даних повинні бути дати а значеннями курсу повинні бути значення колонки “Open*”.

	Date	Open*	High	Low	Close**	Volume	Market Cap
0	May 27, 2020	2.500000	2.57	2.500000	2.56	1848037117	2391641596
1	May 26, 2020	2.540000	2.55	2.470000	2.50	1729219371	2330178499
2	May 25, 2020	2.510000	2.54	2.480000	2.54	1904348324	2368218553
3	May 24, 2020	2.570000	2.60	2.510000	2.51	2241154250	2340299894
4	May 23, 2020	2.600000	2.61	2.550000	2.57	1990987499	2398067877
...
1057	Jul 05, 2017	3.360000	3.52	2.730000	3.00	124339000	500626422
1058	Jul 04, 2017	4.100000	4.19	2.930000	3.37	218559008	550155157
1059	Jul 03, 2017	2.720000	5.40	2.630000	4.09	414950016	654930711
1060	Jul 02, 2017	0.996521	2.88	0.822648	2.71	320452000	0
1061	Jul 01, 2017	1.030000	1.07	0.989566	1.01	13613000	0

[1062 rows x 7 columns]

Рисунок 3.5 - Результуючі дані запиту

Результатом трансформації є дані, які будуть відображені у вигляді графу на рисунку 3.6. Цей графік відображає курс валют криптовалюти протягом усього їх часу згідно отриманих даних.

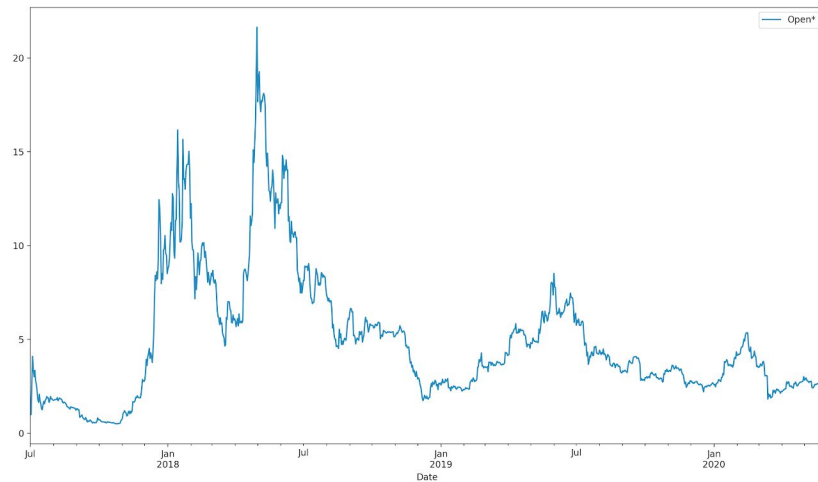


Рисунок 3.6 - Курс валют згідно отриманих даних

Перший етап є етап експоненціального згладжування. Прогноз в даному випадку розраховується із використанням середньозважених середніх значень, і передбачає, що ваги які мають найбільший коефіцієнт пов'язані в основному із останніми спостереженнями, а найменші ваги навпаки зв'язані із найновішими спостереженнями:[13]

$$\hat{y}_{T+1|T} = \alpha y_T + \alpha(1 - \alpha)y_{T-1} + \alpha(1 - \alpha)^2 y_{T-2} + \dots, \quad (3.1)$$

де $0 \leq \alpha \leq 1$ - параметр згладжування.

Залежність зміни зниження ваг в цьому випадку регулюється параметром згладжування α . Якщо α близький до 1, то більшу вагу надається для останніх спостережень. Передбачається 2 крайніх випадки:

- $\alpha = 0$: тобто при прогнозуванні значення дорівнюють середнім оригінальним значення і метод називається методом середніх значень.
- $\alpha = 1$: наївний метод - тобто при прогнозуванні значення становиться значенням останнього спостереження.

При цьому, якщо використовувати автоматичну оптимізацію, що надає статистичним моделям автоматичний пошук для знаходження оптимізованого значення передбачається кращий результат. Результат етапу відображено на

рисунку 3.7, а лістинг коду який проводить дані згладжування відображений на рисунку 3.8.

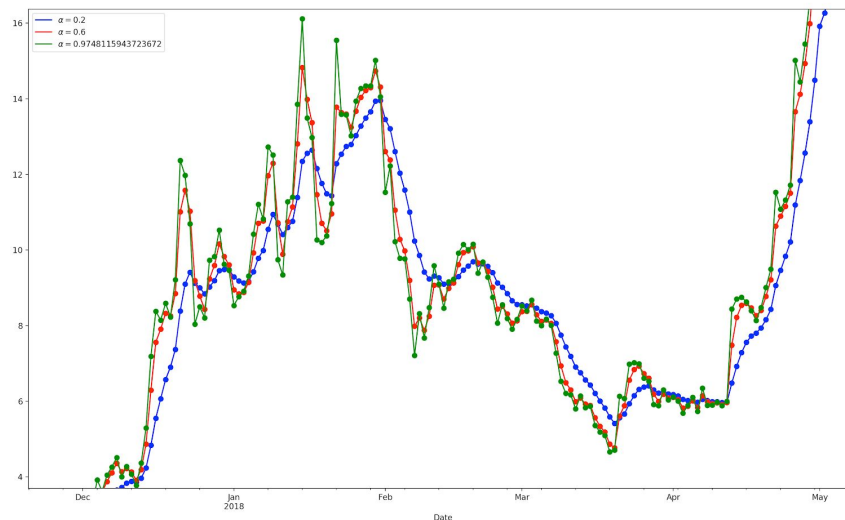


Рисунок 3.7 - Експоненціальне згладжування

```

37 # Експоненціальне згладжування
38 fit1 = SimpleExpSmoothing(data).fit(smoothing_level=0.2,optimized=False)
39 fcast1 = fit1.forecast(12).rename(r'\$\alpha=0.2$')
40 # Побудова графіку
41 fcast1.plot(marker='o', color='blue', legend=True)
42 fit1.fittedvalues.plot(marker='o', color='blue')
43

```

Рисунок 3.8 - Лістинг експоненціального згладжування

Другий етап є реалізація Хольта. При розширенні експоненціального згладжування є можливим прогнозування даних із тенденціями розвитку. Даних метод був названий методом Хольта. Метод Хольта передбачає використання рівняння прогнозу та два рівняння для згладжування (одне рівняння використовується для рівня та одне використовується для тренда):

$$\begin{aligned}
 \hat{y}_{t+h|t} &= l_t + hb_t \\
 l_t &= \alpha y_t + (1 - \alpha)(l_{t-1} + b_{t-1}) \\
 b_t &= \beta^*(l_t - l_{t-1}) + (1 - \beta^*)b_{t-1}, \quad (3.2)
 \end{aligned}$$

де $0 \leq \alpha \leq 1$ - параметр згладжування рівня, а $0 \leq \beta^* \leq 1$ - параметр згладжування тренду.

При довгостроковому прогнозуванні використовуючи прогнозування методом Хольта прогнозування буде нескінченно збільшуватись або нескінченно зменшуватись. Для уникнення цього потрібно використовувати метод демфірового тренду, що передбачає параметр демпфування $0 < \phi < 1$.

$$\begin{aligned}\hat{y}_{t+h|t} &= l_t + (\phi + \phi^2 + \dots + \phi^h)b_t \\ l_t &= \alpha y_t + (1 - \alpha)(l_{t-1} + \phi b_{t-1}) \\ b_t &= \beta^*(l_t - l_{t-1}) + (1 - \beta^*)\phi b_{t-1}.\end{aligned}\quad (3.3)$$

Результатом буде побудова ліній трендів, які зображено на рисунку 3.9, а лістинг коду, що виконує побудову тренду зображено на рисунку 3.10.

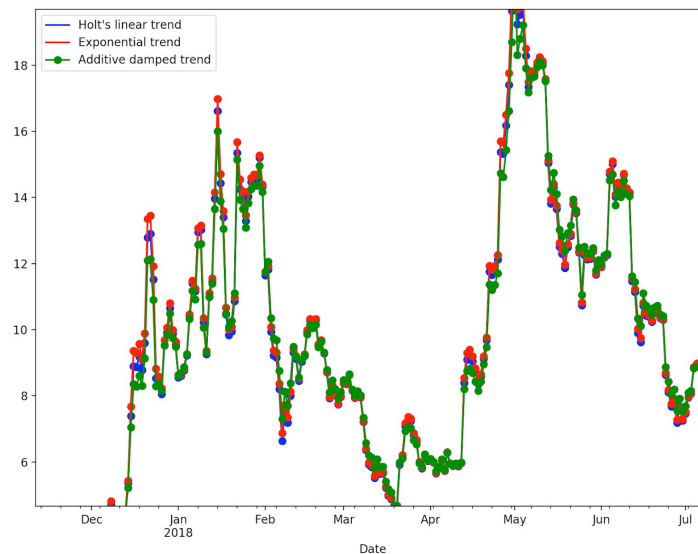


Рисунок 3.9 - Побудова ліній тренду

```
58 # Побудова ліній тренду методом Хольта
59 fit1 = Holt(data).fit(smoothing_level=0.8, smoothing_slope=0.2, optimized=False)
60 fcast1 = fit1.forecast(12).rename("Holt's linear trend")
61
```

Рисунок 3.10 - Лістинг для побудови ліній тренду

Останній етап є етап Хольта-Вінтерса. Метод Хольта-Вінтерса як було визначено раніше підходить для даних що мають тенденцію та сезонність, та використовується параметр згладжування сезонності γ . Передбачається 2 варіанти цього методу:[14]

- Адитивний метод: ряд має постійні середні коливання.

- Мультиплікативний метод: зміни в ряду відбуваються пропорційно зміні рівню серії.

Результатом кінцевого етапу буде побудова адитивного та мультиплікативного методів передбачення курсу криптовалют на 10 днів вперед, як зображено на рисунку 3.11, а лістинг, що виконує побудову передбачення зображено на рисунку 3.12.

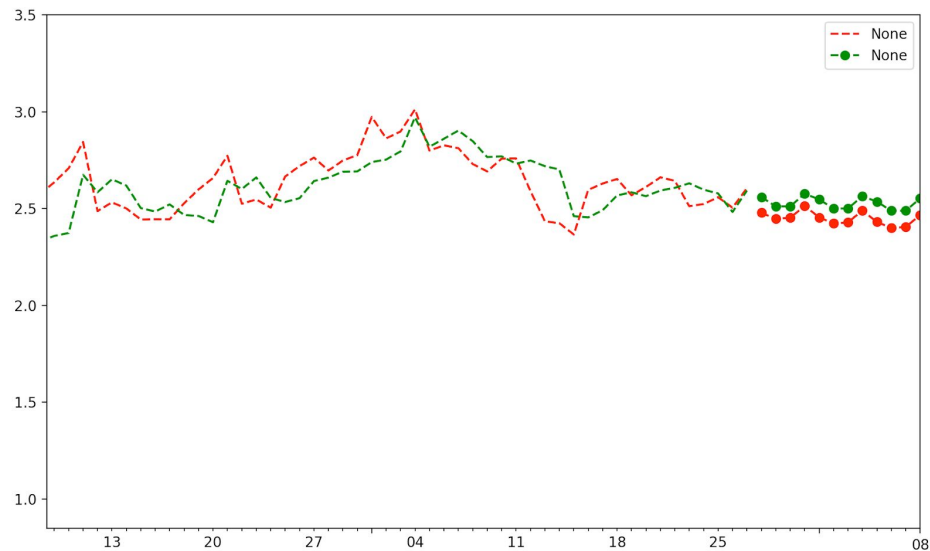


Рисунок 3.11 - Передбачення курсу методом Хольта-Вінтерса

```

78 # Побудова передбачення із подальшим методом Хольта-Вінтерса
79 fit1 = ExponentialSmoothing(data, seasonal_periods=4, trend='add', seasonal='add').fit(use_boxcox=True)
80 fit2 = ExponentialSmoothing(data, seasonal_periods=4, trend='add', seasonal='mul').fit(use_boxcox=True)
81 fit3 = ExponentialSmoothing(data, seasonal_periods=4, trend='add', seasonal='add', damped=True).fit(use_boxcox=True)
82 fit4 = ExponentialSmoothing(data, seasonal_periods=4, trend='add', seasonal='mul', damped=True).fit(use_boxcox=True)
83 fit1.fittedvalues.plot(style='--', color='red')
84 fit2.fittedvalues.plot(style='--', color='green')
85
86 fit1.forecast(12).plot(style='--', marker='o', color='red', legend=True)
87 fit2.forecast(12).plot(style='--', marker='o', color='green', legend=True)

```

Рисунок 3.12 - Лістинг побудови передбачення курсу методом Хольта-Вінтерса

2.3 Розробка СППР із використанням штучного інтелекту

Розробка системи підтримки прийняття рішення із використанням штучного інтелекту полягає в побудові моделі штучного інтелекту, її навчання, тестування і подальшого прогнозування. І спочатку потрібно визначити схему модулю передбачення із використанням штучного інтелекту, яка зображена на рисунку 3.13.[15]

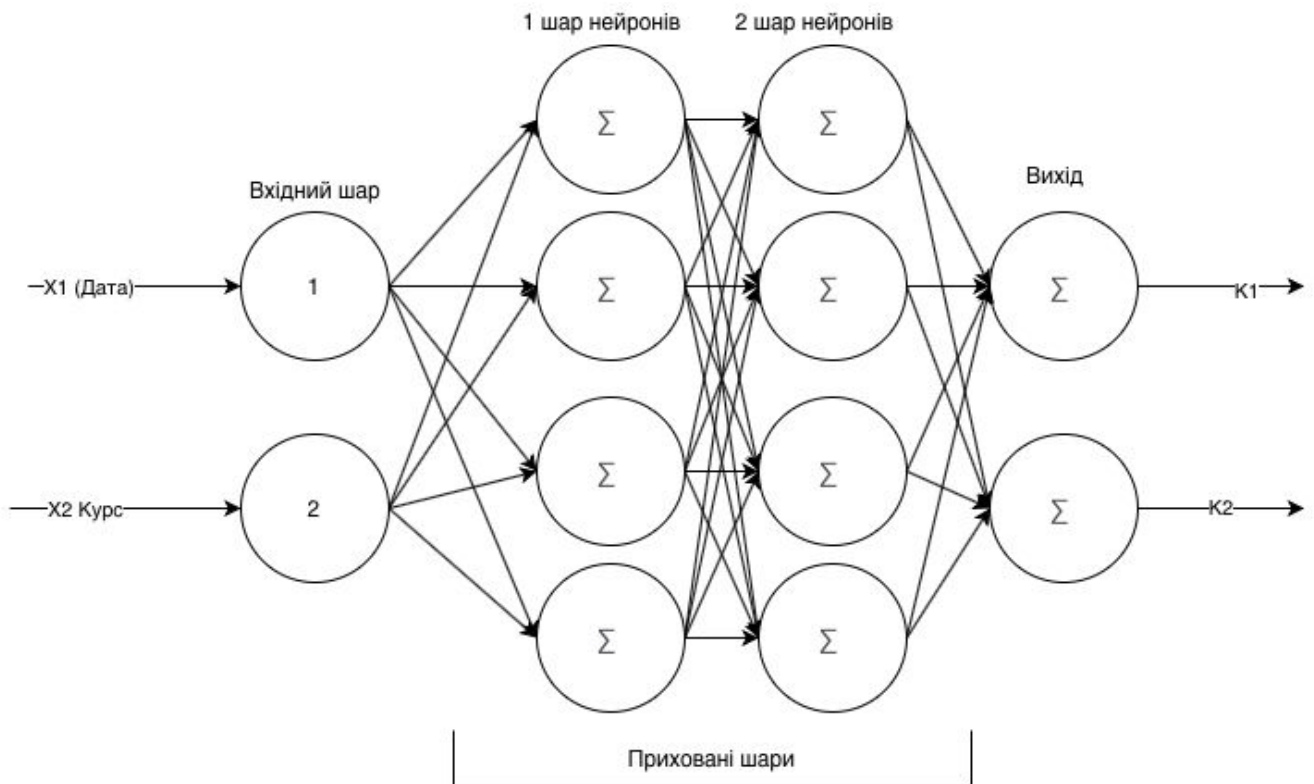


Рисунок 3.13 - Схема штучного інтелекту

Згідно схеми на нейронну мережу передаються дані, в нашому випадку вхідними даними для мережі є $X1$ і $X2$, які є датою та курсом відповідно. І вхідні дані утворюються наш вхідний шар для мережі. В моїй реалізації виділено 2 прихованих шари нейронів, вони будуть відображені в готовій структурі моделі далі. Кількість нейронів залежатиме від конкретної реалізації, в моєму випадку я передавав близько 50 нейронів в кожен шар. Модель також видає 2 виходи $K1$ і $K2$, що є датою та передбаченим курсом криптовалюти відповідно для відображення залежності курсу до дати для подальшої побудови курсу криптовалюти.[16]

Першим кроком в реалізації є імпорт бібліотек необхідних для роботи, які зображені на рисунку 3.14. Основними бібліотеками є Pandas, Keras - для побудови моделей штучного інтелекту, Numpy і Matplotlib.

```

1  from math import sqrt
2  from numpy import array
3  from numpy import mean
4  from numpy import std
5  from pandas import DataFrame
6  from pandas import concat
7  from pandas import read_csv
8  from sklearn.metrics import mean_squared_error
9  from keras.models import Sequential
10 from keras.layers import Dense
11 from matplotlib import pyplot
12 import requests
13 import pandas as pd
14 from datetime import date
15
16 import numpy as np
17 import pandas as pd
18 import matplotlib.pyplot as plt

```

Рисунок 3.14 - Імпорт бібліотек для роботи

Наступним кроком буде підготовка даних, як зображено на рисунку 3.4. Подальші дії системи це визначення продуктивності моделі, що відображені на рисунку 3.15 і код що виконує дану функцію відображений на рисунку 3.16.

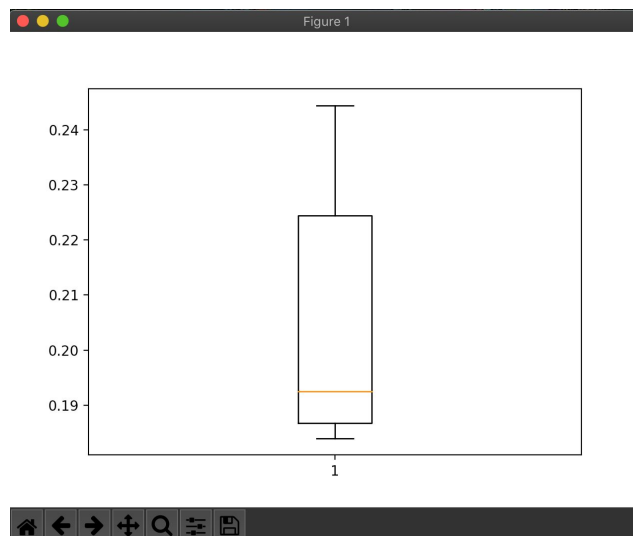


Рисунок 3.15 - Продуктивність моделі

```

104 ▼ def summarize_scores(name, scores):
105     # print a summary
106     scores_m, score_std = mean(scores), std(scores)
107     print('%s: %.3f RMSE (+/- %.3f)' % (name, scores_m, score_std))
108     # box and whisker plot
109     pyplot.boxplot(scores)
110     pyplot.show()

```

Рисунок 3.16 - Лістинг визначення продуктивності моделі

Подальшим кроком є побудова існуючого курсу криптовалют, що зображений на рисунку 3.6. Наступний крок системи це розподіл існуючих даних на 2 частини - набори даних для навчання системи і набори даних для тестування системи. Розподіл даних на два масиви виконувався розподілом по даті, як відображено на рисунку 3.17.

```
35 split_date = pd.Timestamp('01-01-2019')
36
37 train = data.loc[:split_date]
38 test = data.loc[split_date:]
```

Рисунок 3.17 - Розподіл даних

Розподілені дані тепер відображаються наступним чином, як показано на рисунку 3.18.

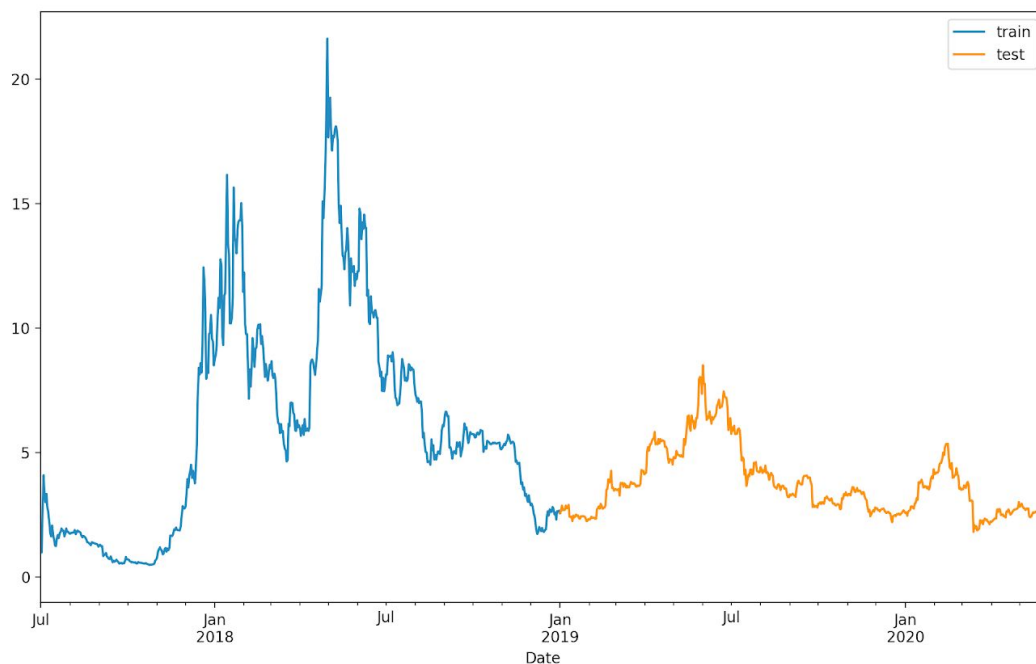


Рисунок 3.18 - Графік розподілу даних

Наступний кроком для розробки нашого модуля передбачення є навчання моделі на навчальних даних. Перше, що необхідно - це визначити саму модель, яка буде навчатись, лістинг створення моделі відображений на рисунку 3.19.[17]

```

114 model = Sequential()
115 model.add(Dense(1, input_shape=(X_test.shape[1],), activation='tanh',
kernel_initializer='lecun_uniform'))
116 model.compile(optimizer=Adam(lr=0.001), loss='mean_squared_error')
117 #print("X_train2")
118 #print(X_train)
119 model.fit(X_train, y_train, batch_size=16, epochs=20, verbose=1)
120
121 y_pred = model.predict(X_test)
122 plt.plot(y_test)
123 plt.plot(y_pred)
124 print('R-Squared: %f'%(r2_score(y_test, y_pred)))
125 print(model.summary())
126 plt.show()

```

Рисунок 3.19 - Лістинг створення моделі для навчання

Після створення моделі можемо переглянути структуру моделі, що визначена для навчання, яка відображена на рисунку 3.20.

```

Model: "sequential_1"
-----
Layer (type)                Output Shape         Param #
-----
dense_1 (Dense)             (None, 1)           2
-----
Total params: 2
Trainable params: 2
Non-trainable params: 0

```

Рисунок 3.20 - Структура моделі для навчання

Процес навчання моделі відбувається протягом певного часу на тренувальних даних. Результатом тренування буде визначено показник ефективності - тобто схожість результатів при навчанні із навчальними наборами даних, як зображено на рисунку 3.21. При навчанні моделі було визначено, що показник ефективності/схожості із оригінальними показниками сягнув 0.928600, що приблизно дорівнює 92% схожості даних із тренувальними наборами даних.

```

Epoch 12/20
549/549 [=====] - 0s 84us/step - loss: 0.0029
Epoch 13/20
549/549 [=====] - 0s 80us/step - loss: 0.0027
Epoch 14/20
549/549 [=====] - 0s 93us/step - loss: 0.0025
Epoch 15/20
549/549 [=====] - 0s 63us/step - loss: 0.0023
Epoch 16/20
549/549 [=====] - 0s 75us/step - loss: 0.0022
Epoch 17/20
549/549 [=====] - 0s 93us/step - loss: 0.0021
Epoch 18/20
549/549 [=====] - 0s 79us/step - loss: 0.0020
Epoch 19/20
549/549 [=====] - 0s 73us/step - loss: 0.0019
Epoch 20/20
549/549 [=====] - 0s 95us/step - loss: 0.0018
R-Squared: 0.928600

```

Рисунок 3.21 - Результат навчання моделі

Також після навчання є можливість переглянути як відображені результати навчання із оригінальними даними, відображено на рисунку 3.22.



Рисунок 3.22 - Відображення результатів навчання

Наступним етапом при розробці моделі є етап тестувальних наборів. На цьому етапі покращується попередня модель, а саме додаються шари, структура того як виглядає модель для тестування відображена на рисунку 3.23.

```
Model: "sequential_1"
```

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 50)	100
dense_2 (Dense)	(None, 50)	2550
dense_3 (Dense)	(None, 1)	51

```
Total params: 2,701
Trainable params: 2,701
Non-trainable params: 0
```

Рисунок 3.23 - Структура моделі для тестування

Результатом тестування моделі також є побудова графіку, що відображає схожість тестових результатів із оригінальними значеннями, як зображено на рисунку 3.24, а також є показник ефективності моделі, який під час тестування сягав 0.972039, тобто, приблизно, 97%.

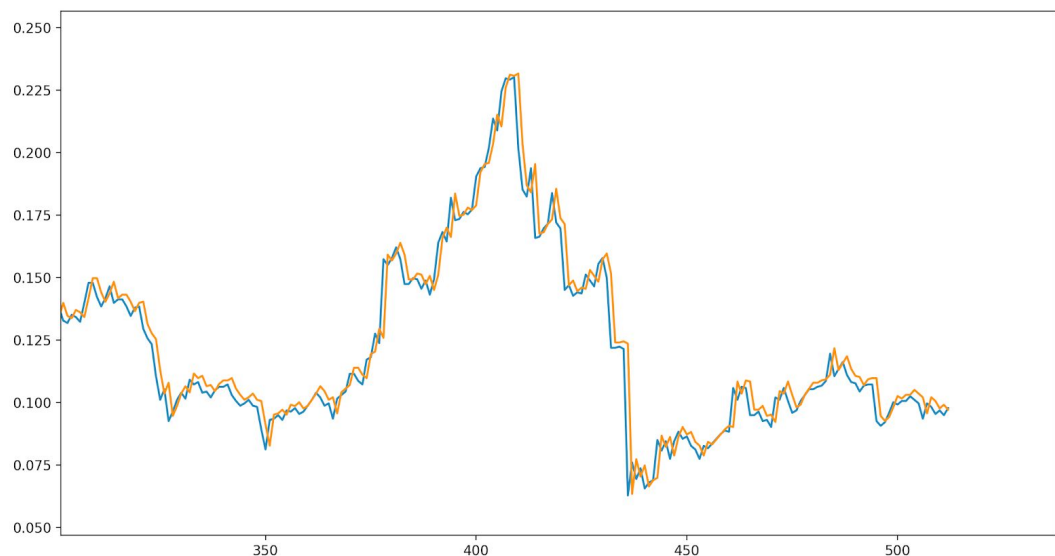


Рисунок 3.24 - Результат тестування моделі

Із отриманням кінцевого результату в вигляді ефективності моделі при тестуванні розробка модулю передбачення вважається завершеною.

3 ПОРІВНЯЛЬНИЙ АНАЛІЗ СППР ДЛЯ ОПТИМІЗАЦІЇ ОБІГУ КРИПТОВАЛЮТ

3.1 Визначення характеристик ефективності СППР для оптимізації обігу криптовалют

Кожна із методів в реалізованих системах надає свої результати із певними можливостями, тому важливим є визначити яка ж із двох моделей є кращою при прогнозуванні. Важливими характеристиками ефективності системи та методу і в таблиці 3.1 згідно стандарту якості ПЗ ISO 9126.

Таблиця 3.1 - Характеристики ефективності системи

№	Характеристика якості ПЗ згідно ISO 9126	Показник якості	Зміст показника	Вимога при проектуванні
1	Ефективність	Поведінка у часі	ступінь відповідності вимогам результату, часу обробки і пропускну здатності ПЗ під час виконання своїх функцій	<0.5
2	Функціональність	Точність	ступінь, в якому ПЗ забезпечує правильний результат з необхідним ступенем точності	>0.9
3	Надійність	Відмовостійкість	можливість роботи ПЗ як передбачалося, незважаючи на наявність апаратних або програмних збоїв відносно коду	>0.9

Таблиця 3.1 відображає інформацію згідно якої визначено основні характеристики ефективності для нашої СППР та визначено фактор важливості для кожної із характеристик, що відображає, як кожна із характеристик має прогнозоване значення на етапі проектування систем.

Згідно таблиці 3.1 ми будемо аналізувати ефективність двох розроблених методів та оберемо кращий метод для прогнозування згідно визначених під час проектування показників якості систем.

3.2 Розрахунок ефективності розроблених СППР для оптимізації обігу криптовалют

Розпочнемо аналіз зі швидкості систем при прогнозуванні. Кожна із систем передбачає швидку видачу прогнозу, але все ж кожна система працює з алгоритмом протягом певного часу, тому швидкість роботи кожної з систем буде наведено в таблиці 3.2. Розрахунок ефективності поведінки систем у часі розраховується із формули:

$$E_{\text{ч}} = \frac{Ч_{\text{м}}}{Ч_{\text{с}}} \quad (3.1)$$

, де $Ч_{\text{м}}$ - час роботи модулю передбачення, $Ч_{\text{с}}$ - загальний час системи. Ефективністю поведінки часу буде відношення роботи модулю до загального часу системи.

Таблиця 3.2 - Швидкість роботи систем

Назва СППР	Загальний час роботи системи, с	Час роботи модулю передбачення, с	Ефективність (поведінка у часі)
Хольта-Вінтерса	10.2712	3.3728	0.3283 (32%)
Нейронна мережа LSTM	13.1911	4.4858	0.3400 (34%)

Згідно таблиці 3.2, відображено загальний час роботи та час роботи саме модулю передбачення. Виходячи з цього краща характеристика по часу надається методу Хольта-Вінтерса, оскільки математична модель Хольта-Вінтерса працює швидше, оскільки не потрібно навчати модель як у випадку зі штучним інтелектом, та не потрібна додаткова трансформація даних.

Наступний етап порівняння характеристик буде точність прогнозування. В нашому випадку цій характеристиці надається найбільший коефіцієнт важливості, щоб надати користувачам якомога достовірнішу інформацію. Перше що розглянемо безпосередньо перед переходом до порівняння точності прогнозування будуть наведені набори та результуючі значення для кожної із систем. Значення похибки розраховуються за відносною похибкою по формулі 3.2.

$$\gamma = \frac{\Delta A_{\Pi}}{A_{H}} \cdot 100\% \quad (3.2)$$

А точність прогнозування буде розраховуватись за формулою 3.3

$$\gamma = 100 - \frac{\Delta A_{\Pi}}{A_{H}} \cdot 100 \quad (3.3)$$

Результуючі значення для методу Хольта-Вінтерса відображені в таблиці 3.3.

Таблиця 3.3 - Результат прогнозування методом Хольта-Вінтерса

Дата	Оригінальне значення курсу,\$	Результат передбачення курсу, \$	Точність прогнозування
May 30, 2020	2,56	2,39	93,359375
May 29, 2020	2,72	2,78	97,79411765
May 28, 2020	2,56	2,42	94,53125
May 27, 2020	2,5	2,37	94,8
May 26, 2020	2,54	2,81	89,37007874
May 25, 2020	2,51	2,14	85,25896414
May 24, 2020	2,57	2,21	85,9922179
May 23, 2020	2,6	2,93	87,30769231
May 22, 2020	2,47	2,29	92,71255061
May 21, 2020	2,6	2,12	81,53846154

May 20, 2020	2,63	2,26	85,93155894
....
Середня точність			89%

Згідно таблиці 3.3 наведено дані лише протягом 11 днів. Загальна кількість днів протягом яких розглядається прогнозування складає 1060 днів, тому наведена інформація є лише короткою частиною результату прогнозування системи. Результат точності для Хольта-Вінтерса при всіх даних складав 89%.

Таблиця 3.4 відображає результати прогнозування методом штучного інтелекту.

Таблиця 3.4 - Результат прогнозування методом нейронної мережі

Дата	Оригінальне значення курсу,\$	Результат передбачення курсу, \$	Точність прогнозування
May 30, 2020	2,56	2,48	96,875
May 29, 2020	2,72	2,66	97,79411765
May 28, 2020	2,56	2,5	97,65625
May 27, 2020	2,5	2,45	98
May 26, 2020	2,54	2,58	98,42519685
May 25, 2020	2,51	2,59	96,812749
May 24, 2020	2,57	2,49	96,88715953
May 23, 2020	2,6	2,66	97,69230769
May 22, 2020	2,47	2,4	97,1659919
May 21, 2020	2,6	2,68	96,92307692
May 20, 2020	2,63	2,6	98,85931559
....
Середня точність			97%

Згідно таблиці 3.4 результат точності для всіх даних склав 97%.

Навівши результуючі значення прогнозувань кожної із систем можемо провести порівняльну характеристику систем по точності прогнозування, порівняння відображено в таблиці 3.5.

Таблиця 3.5 - Точність проведення прогнозування

Назва СППР	Функціональна точність прогнозування
Хольта-Вінтерса	0.89
Нейронна мережа LSTM	0.97

В таблицях 3.3 і 3.4 відображено інформацію, щодо точності прогнозування систем. Точність прогнозування на тестових наборах відсутня для методу Хольта-Вінтерса, оскільки метод Хольта-Вінтерса не потребує навчання моделі. Виходячи із значень таблиці кращою характеристикою по точності прогнозування виступає метод штучного інтелекту.

Наступна характеристика - це безвідмовність систем, тобто ймовірність того, що при роботі система не зможе провести розрахунки та призведе до завершення роботи даної системи. В таблиці 3.6 відображені значення безвідмовності систем, розрахунок проводиться за формулою:

$$B_c = (100 - \frac{K_n}{K_c})/100 \quad (3.4)$$

, де K_n - кількість помилок, K_c - кількість строк коду.

Таблиця 3.6 - Значення безвідмовності систем

Назва СППР	Кількість помилок	Кількість строк коду	Ефективність безвідмовності системи
Хольта-Вінтерса	15	480	0.9996875
Нейронна мережа	3	120	0.99975

LSTM			
------	--	--	--

Згідно таблиці 3.6 отримано значення, що система Хольта-Вінтерса є менш безвідмовною, оскільки йде ускладнення коду формулами для прогнозування й збільшує кількість коду і кількість помилок, що можуть виникнути процесі роботи системи, а система на базі штучного інтелекту оперує значеннями курсу і суматорами штучного інтелекту та логіки штучного інтелекту, що є має меншу можливість помилки.

Таблиця 3.7 - Показники ефективності систем

№	Характеристика якості ПЗ згідно ISO 9126	Показник якості	СППР Хольта-Вінтерса	СППР нейронна мережа	Вимоги проектування
1	Ефективність	Поведінка у часі	0.3283	0.3400	<0.5
2	Функціональність	Точність	0.89	0.97	>0.9
3	Надійність	Відмовостійкість	0.9996875	0.99975	>0.9

Згідно таблиці 3.7, при розрахунку показників ефективності систем, було визначено, що система із штучним інтелектом є більш ефективною ніж система із використанням математичної моделі Хольта-Вінтерса.

ВИСНОВОК

Всі задачі, поставлені під час виконання бакалаврської роботи, було виконано в повному обсязі.

1. Проведено аналіз існуючих рішень, де було розглянуто такі системи як CoinMarketCap, Binance і Crypto.com. Розглянуто основні популярні криптовалюти, такі як: Bitcoin, Ethereum, EOS. Описано можливості використання криптовалют та розглянуто їх структуру. Проведено пошук математичних методів прогнозування та методів прогнозування засобами штучного інтелекту, а саме метод Хольта-Вінтерса та із використанням нейронної мережі з довготривалою моделлю пам'яті відповідно.

2. Розроблено систему підтримки прийняття рішення із використанням математичного прогнозування Хольта-Вінтерса та систему підтримки прийняття рішення із використанням нейронної мережі. Кожен етап розробки систем супроводжувався схемами систем, наведеними частинами коду та результатів системи.

3. Проведено порівняльний аналіз розроблених систем із визначенням ефективності кожної із систем. Було визначено, що система прогнозування на базі нейронної мережі є більш ефективною ніж система прогнозування на базі математичного методу.

4. Перспективою розвитку системи може бути покращення отримання даних із систем представлення курсу протягом дня, години. Може бути покращення відображення майбутнього курсу у вигляді таблиць а також використання даних систем як джерело отримання прогнозу для вже існуючих криптовалютних систем.

ПЕРЕЛІК ПОСИЛАНЬ

1. Алекс Тапскотт, Дон Тапскотт - Технологія блокчейн, в-во: Ексмо, 2017, ст. 99-120.
2. Сейфеддин Аммуc - Коротка історія грошей, або все, що потрібно знати про біткойн, в-во: Манн, Іванов і Фербер, 2019, ст. 187-259.
3. Олександр Фролов - Створення смарт-контрактів Solidity для блокчейна Ethereum, в-во: SelfPub, 2019, ст. 20-40.
4. Інтернет джерело - Ethereum смарт-контракти. [Електронний ресурс] <https://medium.com/@bakulinar/ethereum-смарт-контракт-на-solidity-от-простого-2f753fe4125e>
5. Інтернет джерело - Що таке криптовалюта EOS. [Електронний ресурс] <https://altcoinlog.com/what-is-eos-cryptocurrency/>
6. М. Кендел - Часові ряди, 1981, ст. 62-75.
7. Методи та моделі економічного прогнозування [Електронний ресурс] – 2013. – Режим доступу до ресурсу: <https://library.if.ua/book/72/5247.html>
8. Holt-Winters' seasonal method | Forecasting: Principles and Practice [Електронний ресурс] – 2017. – Режим доступу до ресурсу: <https://otexts.com/fpp2/holt-winters.html>
9. Revenue Forecasting using Holt–Winters Exponential Smoothing [Електронний ресурс] – 2017. – Режим доступу до ресурсу: https://www.researchgate.net/publication/311945797_Revenue_Forecasting_using_Holt-Winters_Exponential_Smoothing.
10. Інтернет джерело - Часові ряди і моделі прогнозування. [Електронний ресурс]: <https://4analytics.ru/prognozirovanie/vremennye-ryadi-i-modeli-prognozirovaniya.html>

- 11.Єріна А.М. Статистичне моделювання та прогнозування : [навч. посіб.] / А.М. Єріна. – К. : КНЕУ, 2001. – 170 с.
- 12.World Wide Web: habr: Хеш-алгоритми: [Електронний ресурс] – Режим доступу: <https://habr.com/post/93226/>
- 13.А.А. Грешилов, В.А. Стакун, А.А Стакун - Математичні методи побудови прогнозів, 1997, ст. 62-82.
- 14.Прогноз по методу експоненціального згладжування з трендом і сезонністю Хольта-Винтерса [Електронний ресурс]. – 2016 — Режим доступу до ресурсу: <http://4analytics.ru/prognozirovanie/prognoz-po-metodueksponencialnogo-sglajivaniya-s-trendom-i-sezonnostyu-xoltavintersa>.
- 15.Воронкова В.Г. Планування та прогнозування в умовах ринку. – Навчальний посібник / Під ред. д.ф.н., проф. В.Г. Воронкової. – К.: ВД «Професіонал», 2006. – 608 с.
- 16.Геєць В. М. Моделі і методи соціально-економічного прогнозування : підручник / В. М. Геєць, Т. С. Клебанова, О. І. Черняк та ін. - Х. : ВД "ІНЖЕК", 2005. - 396 с.
- 17.Інтернет джерело - Прогнозування часових рядів використовуючи нейронні мережі.[Електронний ресурс]: <https://blog.statsbot.co/time-series-prediction-using-recurrent-neural-networks-lstms-807fa6ca7f>

ДОДАТОК А

Лістинг програми із використанням математичної моделі

```
from HoltVintersLib import ExponentialSmoothing, SimpleExpSmoothing, Holt
# імпорт бібліотек
from datetime import date
import requests
import pandas as pd
import matplotlib.pyplot as plt

# Отримання поточної дати
today = date.today()

# Формування URL із поточною датою
url =
'https://coinmarketcap.com/currencies/eos/historical-data/?start=20170101&end=20200
526' + today.strftime("%d%m%Y")
# Отримання контенту сторінки
html = requests.get(url).content
# Отримання списку таблиць зі сторінки
df_list = pd.read_html(html)
# Отримання таблиці із курсом валют
df = df_list[2]

print(df)

# Отримання колонки з датою
df['Date'] = pd.to_datetime(df["Date"])
# Вставка колонки "Дата" як індекс
df_idx = df.set_index(["Date"], drop=True)

# Сортування по індексу
df_idx = df_idx.sort_index(axis=1, ascending=True)
df_idx = df_idx.iloc[::-1]

# Побудова графіку
```



```

data = df_idx[['Open*']]
data.plot(y='Open*')

plt.show()

# Експоненціальне згладжування
fit1 = SimpleExpSmoothing(data).fit(smoothing_level=0.2,optimized=False)
fcast1 = fit1.forecast(12).rename(r'\alpha=0.2$')
# Побудова графіку
fcast1.plot(marker='o', color='blue', legend=True)
fit1.fittedvalues.plot(marker='o', color='blue')

fit2 = SimpleExpSmoothing(data).fit(smoothing_level=0.6,optimized=False)
fcast2 = fit2.forecast(12).rename(r'\alpha=0.6$')
# Побудова графіку
fcast2.plot(marker='o', color='red', legend=True)
fit2.fittedvalues.plot(marker='o', color='red')

fit3 = SimpleExpSmoothing(data).fit()
fcast3 =
fit3.forecast(12).rename(r'\alpha=%s$'%fit3.model.params['smoothing_level'])
# Побудова графіку
fcast3.plot(marker='o', color='green', legend=True)
fit3.fittedvalues.plot(marker='o', color='green')

plt.show()

# Побудова ліній тренду методом Хольта
fit1 = Holt(data).fit(smoothing_level=0.8, smoothing_slope=0.2, optimized=False)
fcast1 = fit1.forecast(12).rename("Holt's linear trend")

fit2 = Holt(data, exponential=True).fit(smoothing_level=0.8, smoothing_slope=0.2,
optimized=False)
fcast2 = fit2.forecast(12).rename("Exponential trend")

fit3 = Holt(data, damped=True).fit(smoothing_level=0.8, smoothing_slope=0.2)
fcast3 = fit3.forecast(12).rename("Additive damped trend")

```

```
# Побудова графіку
fit1.fittedvalues.plot(marker="o", color='blue')
fcast1.plot(color='blue', marker="o", legend=True)
fit2.fittedvalues.plot(marker="o", color='red')
fcast2.plot(color='red', marker="o", legend=True)
fit3.fittedvalues.plot(marker="o", color='green')
fcast3.plot(color='green', marker="o", legend=True)

plt.show()
```

```
# Побудова передбачення із подальшим методом Хольта-Вінтерса
fit1 = ExponentialSmoothing(data, seasonal_periods=4, trend='add',
seasonal='add').fit(use_boxcox=True)
fit2 = ExponentialSmoothing(data, seasonal_periods=4, trend='add',
seasonal='mul').fit(use_boxcox=True)
fit3 = ExponentialSmoothing(data, seasonal_periods=4, trend='add', seasonal='add',
damped=True).fit(use_boxcox=True)
fit4 = ExponentialSmoothing(data, seasonal_periods=4, trend='add', seasonal='mul',
damped=True).fit(use_boxcox=True)
fit1.fittedvalues.plot(style='--', color='red')
fit2.fittedvalues.plot(style='--', color='green')

fit1.forecast(12).plot(style='--', marker='o', color='red', legend=True)
fit2.forecast(12).plot(style='--', marker='o', color='green', legend=True)

plt.show()
print("Forecasting rates of properties using Holt-Winters method with both additive and
multiplicative seasonality.")
```

Лістинг програми із використанням штучного інтелекту

```
from math import sqrt
from numpy import array
from numpy import mean
from numpy import std
from pandas import DataFrame
```

```
from pandas import concat
from pandas import read_csv
from sklearn.metrics import mean_squared_error
from keras.models import Sequential
from keras.layers import Dense
from matplotlib import pyplot
import requests
import pandas as pd
from datetime import date

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

# Розділення набору на тренувальні та тестові
def train_test_split(data, n_test):
    return data[:-n_test], data[-n_test:]

# трансформація списку в правильний формат
def series_to_supervised(data, n_in, n_out=1):
    df = DataFrame(data)
    cols = list()
    # input sequence (t-n, ... t-1)
    for i in range(n_in, 0, -1):
        cols.append(df.shift(i))
    # forecast sequence (t, t+1, ... t+n)
    for i in range(0, n_out):
        cols.append(df.shift(-i))
    # put it all together
    agg = concat(cols, axis=1)
    # drop rows with NaN values
    agg.dropna(inplace=True)
    return agg.values

# визначення похибки
def measure_rmse(actual, predicted):
    return sqrt(mean_squared_error(actual, predicted))
```

Заповнення моделі

```
def model_fit(train, config):
    # unpack config
    n_input, n_nodes, n_epochs, n_batch = config
    # prepare data
    data = series_to_supervised(train, n_input)
    train_x, train_y = data[:, :-1], data[:, -1]
    # define model
    model = Sequential()
    model.add(Dense(n_nodes, activation='relu', input_dim=n_input))
    model.add(Dense(1))
    model.compile(loss='mse', optimizer='adam')
    # fit
    model.fit(train_x, train_y, epochs=n_epochs, batch_size=n_batch, verbose=0)
    return model
```

передбачення із заповненою моделлю

```
def model_predict(model, history, config):
    # unpack config
    n_input, _, _, _ = config
    # prepare data
    x_input = array(history[-n_input:]).reshape(1, n_input)
    # forecast
    yhat = model.predict(x_input, verbose=0)
    return yhat[0]
```

перевірка даних

```
def walk_forward_validation(data, n_test, cfg):
    predictions = list()
    # split dataset
    train, test = train_test_split(data, n_test)
    # fit model
    model = model_fit(train, cfg)
    # seed history with training dataset
    history = [x for x in train]
    # step over each time-step in the test set
```

```

for i in range(len(test)):
    # fit model and make forecast for history
    yhat = model_predict(model, history, cfg)
    # store forecast in list of predictions
    predictions.append(yhat)
    # add actual observation to history for the next loop
    history.append(test[i])
# estimate prediction error
error = measure_rmse(test, predictions)
print(' > %.3f % error)
return [error,predictions]

# повторення налаштувань
def repeat_evaluate(data, config, n_test, n_repeats=5):
    # fit and evaluate the model n times
    scores = []
    predictions = []
    for _ in range(n_repeats):
        tmp_scores, tmp_predictions = walk_forward_validation(data, n_test, config)
        scores.append(tmp_scores)
        predictions.append(tmp_predictions)
    return [scores,predictions]

# обрахування ефективності моделі
def summarize_scores(name, scores):
    # print a summary
    scores_m, score_std = mean(scores), std(scores)
    print('%s: %.3f RMSE (+/- %.3f)' % (name, scores_m, score_std))
    # box and whisker plot
    pyplot.boxplot(scores)
    pyplot.show()

def find_best_solution(series,name,method="nn"):
    data = series.values
    n_test = int(len(data)*0.2) # percentage used for test
    config = [24, 500, 100, 100]
    scores,predictions = repeat_evaluate(data, config, n_test)

```

```

summarize_scores('mlp', scores)
plt.savefig("scores_%s_%s.png" % (name,method))
plt.show()
train,test = train_test_split(data,n_test)
prediction =
pd.DataFrame(list(train.flatten()+np.array(predictions[0]).flatten().tolist()))
ax = pd.DataFrame(data).plot(label="Original") # main data
# print(test)
prediction.plot(ax=ax, alpha=.7, figsize=(14,7))
plt.savefig("pred_%s_%s.png" %(name,method))
plt.show()

```

```

today = date.today()
url =
'https://coinmarketcap.com/currencies/eos/historical-data/?start=20170101&end=2020
526' + today.strftime("%d%m%Y")
html = requests.get(url).content
#df_list = pd.read_html(html)
#df = df_list[2]
#df.to_csv("MyData.csv")
df = pd.read_csv("myData.csv")
print(df)
df['Date'] = pd.to_datetime(df["Date"])
df_idx = df.set_index(["Date"], drop=True)
df_idx.head(5)

```

```

df_idx = df_idx.sort_index(axis=1, ascending=True)
df_idx = df_idx.iloc[::-1]

```

```

data = df_idx[['Open*']]
find_best_solution(data,"Rates")
diff = data.index.values[-1] - data.index.values[0]
days = diff.astype('timedelta64[D]')
days = days / np.timedelta64(1, 'D')
years = int(days/365)
print("Total data: %d years"%years)

```

```

print("80 percent data = 2017 to %d"%(2017 + int(0.8*years)))

split_date = pd.Timestamp('01-01-2019')

train = data.loc[:split_date]
test = data.loc[split_date:]
print("Split")
print(train)
print(test)

ax = train.plot(figsize=(10,12))
test.plot(ax=ax)
plt.legend(['train', 'test'])

from sklearn.preprocessing import MinMaxScaler
sc = MinMaxScaler()
train_sc = sc.fit_transform(train)
test_sc = sc.transform(test)

train_sc_df = pd.DataFrame(train_sc, columns=['Y'], index=train.index)
test_sc_df = pd.DataFrame(test_sc, columns=['Y'], index=test.index)
print("train_sc_df")
print(train_sc_df)

for s in range(1,2):
    train_sc_df['X_{}'.format(s)] = train_sc_df['Y'].shift(s)
    test_sc_df['X_{}'.format(s)] = test_sc_df['Y'].shift(s)

print("train_sc_df1")
print(train_sc_df)

print("test_sc_df1")
print(test_sc_df)

X_train = train_sc_df.dropna().drop('Y', axis=1)
y_train = train_sc_df.dropna().drop('X_1', axis=1)

```

```
X_test = test_sc_df.dropna().drop('Y', axis=1)
y_test = test_sc_df.dropna().drop('X_1', axis=1)
```

```
X_train = X_train.as_matrix()
y_train = y_train.as_matrix()
```

```
X_test = X_test.as_matrix()
y_test = y_test.as_matrix()
```

```
print('Train size: (%d x %d)%'(X_train.shape[0], X_train.shape[1]))
print('Test size: (%d x %d)%'(X_test.shape[0], X_test.shape[1]))
```

```
from sklearn.svm import SVR
regressor = SVR(kernel='rbf')
```

```
regressor.fit(X_train, y_train)
y_pred = regressor.predict(X_test)
```

```
plt.plot(y_test)
plt.plot(y_pred)
```

```
from sklearn.metrics import r2_score
```

```
def adj_r2_score(r2, n, k):
    return 1-(((1-r2)*((n-1)/(n-k-1))))
```

```
r2_test = r2_score(y_test, y_pred)
print("R-squared is: %f"%r2_test)
```

```
from keras.models import Sequential
from keras.layers import Dense
from keras.optimizers import Adam
import keras.backend as K
```

```
K.clear_session()
model = Sequential()
```



```
model.add(Dense(1, input_shape=(X_test.shape[1],), activation='tanh',
kernel_initializer='lecun_uniform'))
model.compile(optimizer=Adam(lr=0.001), loss='mean_squared_error')
#print("X_train2")
#print(X_train)
model.fit(X_train, y_train, batch_size=16, epochs=20, verbose=1)
```

```
y_pred = model.predict(X_test)
plt.plot(y_test)
plt.plot(y_pred)
print('R-Squared: %f%(r2_score(y_test, y_pred)))
print(model.summary())
plt.show()
```

```
K.clear_session()
model = Sequential()
model.add(Dense(50, input_shape=(X_test.shape[1],), activation='relu',
kernel_initializer='lecun_uniform'))
model.add(Dense(50, input_shape=(X_test.shape[1],), activation='relu'))
model.add(Dense(1))
model.compile(optimizer=Adam(lr=0.001), loss='mean_squared_error')
model.fit(X_train, y_train, batch_size=16, epochs=20, verbose=1)
```

```
y_pred = model.predict(X_test)
```

```
plt.plot(y_test)
plt.plot(y_pred)
print('R-Squared: %f%(r2_score(y_test, y_pred)))
print(model.summary())
plt.show()
```

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (ПРЕЗЕНТАЦІЯ)



Міністерство освіти і науки України
Державний університет телекомунікацій
Навчально-науковий інститут інформаційних
технологій



Кафедра системного аналізу

Бакаларська робота

за темою «Розробка системи підтримки прийняття
рішень для оптимізації процесів обігу криптовалюти»

Виконав: студент групи САД-41

Вавринчук Едуард Миколайович

Науковий керівник:

ст.вик. кафедри Системного аналізу, Штіммерман

Аксенія Миколаївна

Київ-2020

Об'єкт, предмет, мета бакалаврської роботи

- **Об'єкт дослідження**- методи передбачення курсів для криптовалют
- **Предмет дослідження**– математичні методи передбачення та передбачення шляхом штучного інтелекту
- **Мета роботи**– аналіз інформації та їх обробка з видачею майбутніх змін криптовалют для користувачів з максимальною ефективністю

Актуальність бакалаврської роботи

- Недоліком існуючих рішень є непередбачуваність зміни курсу, що не залежить від звичайних грошей.
- Не передбачається демонстрація курсу, тобто як він поведе себе в наступний тиждень/місяць.
- Задачею є представлення користувачу можливості зміни курсу криптовалюти на основі попередніх значень курсу з максимальною ефективністю.
- Показниками ефективності для рішення є швидкість, точність та безвідмовність прогнозування систем.

Завдання на бакалаврську роботу

- Провести аналітичний огляд існуючих рішень для процесів обігу криптовалюти із загальнодоступних джерел, в т.ч. із мережі Інтернет.
 - Провести пошук засобів передбачення математичними методами та штучним інтелектом.
 - Розробка систем підтримки прийняття рішення із використанням математичного методу та штучним інтелектом.
 - Порівняльний аналіз систем і визначення кращого методу прогнозування для криптовалют.
-

Пошук існуючих рішень

CoinMarketCap (Рис. 1)
 Crypto.com (Рис. 2)
 Binance (Рис. 3)

Рис. 1 - CoinMarketCap

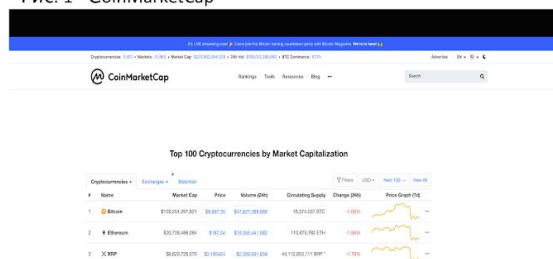


Рис. 2 - Crypto.com

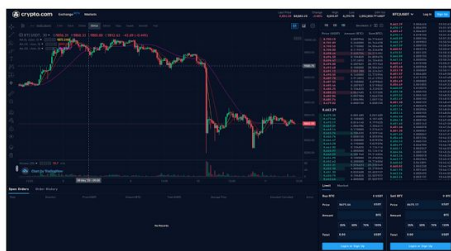
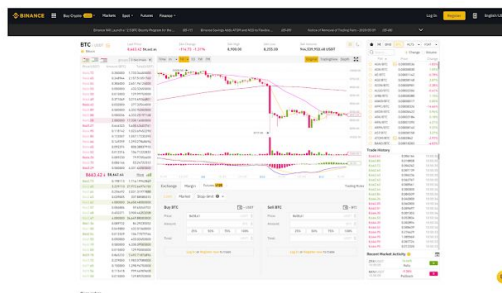


Рис. 3 - Binance



Аналіз популярних криптовалют

Характеристика	Bitcoin	Ethereum	EOS
Модель токена	Інфляційна	Інфляційна	Інфляційна
Модель оплати транзакцій	Bitcoin	Gas	Bandwidth(потужності)
Швидкість транзакцій	78 хв	6 хв	1.5с
Віртуальна машина	-	EVM(Ethereum Virtual Machine)	стандарт WebAssembly
Мова розробки контрактів	-	Solidity	C++/Rust
Консенсус	Proof-of-Work	Proof-of-Work	Delegated Proof-of-Work
Ресурс консенсуса	Майнінгові пули	Майнінгові пули	21 блок-продюсер

Механізм передбачення Хольта-Вінтерса і методи що базуються на штучних нейронних мережах

Формула експоненціального згладжування

$$L_t = \frac{k \cdot y_t + (1+k) \cdot (L_{t-1} - T_{t-1})}{S_{t-s}}$$

Формула тренду

$$S_t = \frac{q \cdot y_t + (1-q) \cdot S_{t-s}}{L_t}$$

Формула передбачення

$$\hat{Y}_{t-p} = (L_t + p \cdot T_t) \cdot S_{t-s+p}$$

Рис. 1 - Приклад прогнозування методом Хольта-Вінтерса

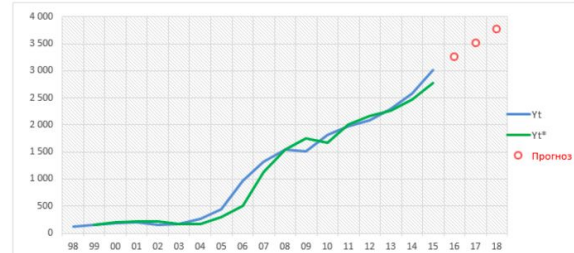
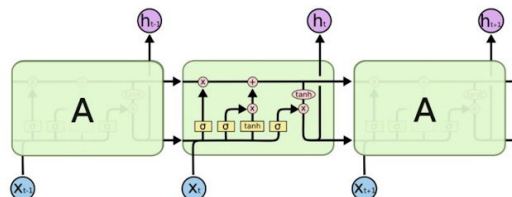


Рис. 2 - LSTM нейронна мережа

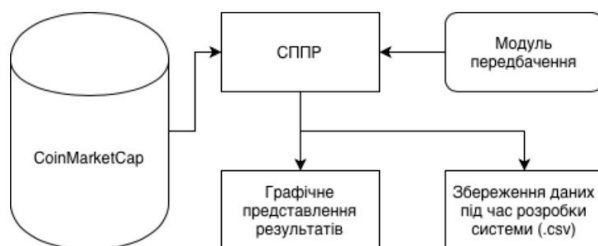


В бакалаврській роботі була розроблена система підтримки прийняття рішення

Рис. 1

Currency in USD						
Date	Open*	High	Low	Close**	Volume	Market Cap
Jun 02, 2020	2.82	2.89	2.64	2.68	2,931,746,699	2,499,160,414
Jun 01, 2020	2.67	2.82	2.67	2.81	2,380,045,715	2,626,478,790
May 31, 2020	2.76	2.76	2.66	2.67	2,161,363,681	2,491,563,531
May 30, 2020	2.61	2.77	2.59	2.78	2,210,704,952	2,576,884,318
May 29, 2020	2.62	2.67	2.60	2.61	2,001,356,933	2,434,453,956
May 28, 2020	2.56	2.62	2.53	2.62	1,863,387,715	2,441,711,521
May 27, 2020	2.50	2.57	2.50	2.56	1,848,037,117	2,391,641,598
May 26, 2020	2.54	2.55	2.47	2.50	1,729,219,371	2,330,178,499
May 25, 2020	2.51	2.54	2.48	2.54	1,904,348,324	2,368,218,553
May 24, 2020	2.57	2.60	2.51	2.51	2,241,154,250	2,340,299,894
May 23, 2020	2.60	2.61	2.55	2.57	1,960,987,499	2,388,067,877
May 22, 2020	2.47	2.62	2.45	2.60	2,362,296,582	2,422,717,866
May 21, 2020	2.60	2.61	2.43	2.47	2,505,696,721	2,305,367,706
May 20, 2020	2.63	2.66	2.56	2.60	2,834,851,770	2,421,830,449
May 19, 2020	2.66	2.66	2.59	2.63	2,735,454,853	2,457,601,291
May 18, 2020	2.62	2.73	2.61	2.66	3,408,005,115	2,452,069,102
May 17, 2020	2.62	2.66	2.60	2.62	2,822,225,030	2,413,634,126
May 16, 2020	2.59	2.67	2.57	2.63	3,304,379,145	2,422,299,125
May 15, 2020	2.61	2.65	2.56	2.59	3,790,363,647	2,389,693,107

Рис. 2



Структура та результати СППР із використанням математичного методу

Рис. 1. Модуль передбачення із використанням математичної моделі



Рис. 2 - побудова оригінального курсу

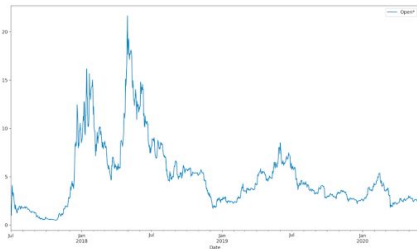
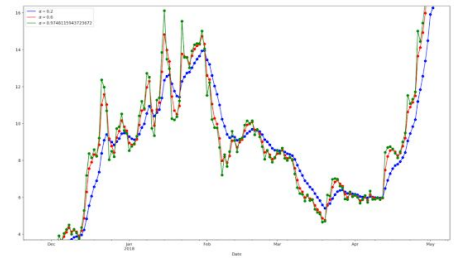


Рис. 3 - Результат розрахунку формули експоненціального згладжування згідно формули

$$L_i = \frac{k \cdot y_i}{S_{i-s}} + (1+k) \cdot (L_{i-1} - T_{i-1}),$$



Структура та результати СППР із використанням математичного методу

Рис. 1 - побудова ліній тренду згідно формули $S_i = \frac{q \cdot y_i}{L_i} + (1-q) \cdot S_{i-s},$

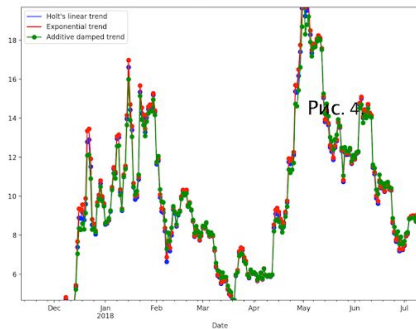
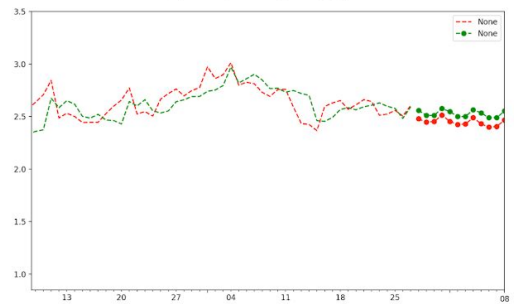


Рис. 2 - побудова передбачення згідно формули $\hat{Y}_{i-p} = (L_i + p \cdot T_i) \cdot S_{i-s+p},$



В розробленій бакалаврській роботі СППР використовується наступна архітектура НМ

Рис. 1 - Схема нейронної мережі

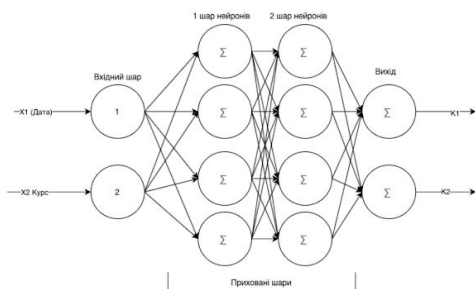


Рис. 2 - Визначення продуктивності мережі

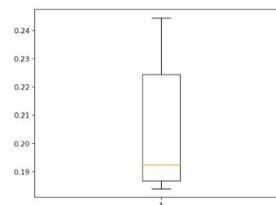
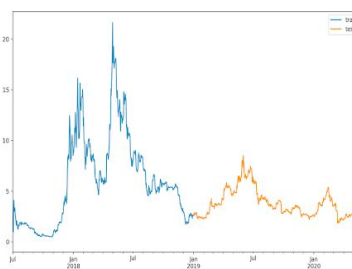


Рис. 3 - Розподіл даних та навчальні та тестувальні

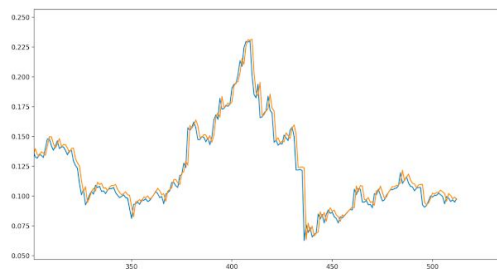


В розробленій бакалаврській роботі СППР використовується наступна архітектура НМ

Рис. 1 - Результат створення нейронної мережі із шарами

```
Model: "sequential_1"
Layer (type)                Output Shape         Param #
-----
dense_1 (Dense)             (None, 50)          100
dense_2 (Dense)             (None, 50)          2550
dense_3 (Dense)             (None, 1)           51
-----
Total params: 2,701
Trainable params: 2,701
Non-trainable params: 0
```

Рис. 2 - Результат прогнозування мережі і відображення оригінального курсу



Порівняльний аналіз систем

№	Характеристика якості ПЗ згідно ISO 9126	Показник якості	Зміст показника	Прогнозоване значення
1	Ефективність	Поведінка у часі	ступінь відповідності вимогам результату, часу обробки і пропускової здатності ПЗ під час виконання своїх функцій	<0.5
2	Функціональність	Точність	ступінь, в якому ПЗ забезпечує правильний результат з необхідним ступенем точності	>0.9
3	Надійність	Відмовостійкість	можливість роботи ПЗ як передбачалося, незважаючи на наявність апаратних або програмних збоїв відносно коду	>0.9

Порівняльний аналіз швидкості прогнозування

$$E_{\text{ч}} = \frac{Ч_{\text{м}}}{Ч_{\text{с}}}$$

Назва СППР	Загальний час роботи системи, с	Час роботи модулю передбачення, с	Ефективність (поведінка у часі)
Хольта-Вінтерса	10.2712	3.3728	0.3283
Нейронна мережа LSTM	13.1911	4.4858	0.3400

Порівняльний аналіз точності прогнозування

формула розрахунку відносної похибки

$$\gamma = \frac{\Delta A_{\text{п}}}{A_{\text{н}}} \cdot 100\%$$

формула точності $\gamma = 100 - \frac{\Delta A_{\text{п}}}{A_{\text{н}}} \cdot 100$

Результат прогнозування математичним методом

Дата	Оригінальне значення курсу, \$	Результат передбачення курсу, \$	Точність прогнозування
May 30, 2020	2,56	2,39	93,359375
May 29, 2020	2,72	2,78	97,79411765
May 28, 2020	2,56	2,42	94,53125
May 27, 2020	2,5	2,37	94,8
May 26, 2020	2,54	2,81	89,37007874
May 25, 2020	2,51	2,14	85,25896414
May 24, 2020	2,57	2,21	85,9922179
May 23, 2020	2,6	2,93	87,30769231
May 22, 2020	2,47	2,29	92,71255061
May 21, 2020	2,6	2,12	81,53846154
May 20, 2020	2,63	2,26	85,93155894
....
Середня точність			89%

Результат прогнозування нейронною мережею

Дата	Оригінальне значення курсу, \$	Результат передбачення курсу, \$	Точність прогнозування
May 30, 2020	2,56	2,48	96,875
May 29, 2020	2,72	2,66	97,79411765
May 28, 2020	2,56	2,5	97,65625
May 27, 2020	2,5	2,45	98
May 26, 2020	2,54	2,58	98,42519685
May 25, 2020	2,51	2,59	96,812749
May 24, 2020	2,57	2,49	96,88715953
May 23, 2020	2,6	2,66	97,69230769
May 22, 2020	2,47	2,4	97,1659919
May 21, 2020	2,6	2,68	96,92307692
May 20, 2020	2,63	2,6	98,85931559
....
Середня точність			97%

Порівняльний аналіз систем

Порівняльний аналіз безвідмовності систем

Назва СППР	Кількість помилок	Кількість строк коду	Ефективність безвідмовності системи
Хольта-Вінтерса	15	480	99.96875
Нейронна мережа LSTM	3	120	99.975

$$B_c = 100 - \frac{K_p}{K_c}$$

Порівняльна характеристика систем

№	Характеристика якості ПЗ згідно ISO 9126	Показник якості	СППР Хольта-Вінтерса	СППР нейронна мережа	Вимоги проектування
1	Ефективність	Поведінка у часі	0.3283	0.3400	<0.5
2	Функціональність	Точність	0.89	0.97	>0.9
3	Надійність	Відмовостійкість	99.96875	99.975	>0.9

Висновок

Протягом виконання бакалаврської роботи було проаналізовано можливі методи прогнозування як засобами математичних моделей і визначено математичну модель Хольта-Вінтерса так і засобами штучних нейронних мереж.

В етапі розробки систем підтримки прийняття рішень були розроблені 2 системи із використанням математичної моделі Хольта-Вінтерса та із використанням штучного інтелекту на базі довготривалої моделі пам'яті (LSTM).

Було проведено порівняльний аналіз розроблених систем із визначенням ефективності кожної із систем і визначено, що система прогнозування на базі нейронної мережі є більш ефективною ніж система прогнозування на базі математичного методу.

Перспективою розвитку системи може бути покращення отримання даних із систем представлення курсу протягом дня, години. Може бути покращення відображення майбутнього курсу у вигляді таблиць а також використання даних систем як джерело отримання прогнозу для вже існуючих криптовалютних систем.

Апробація роботи

1. Вавринчук Е.М. АНАЛІЗ РОЗВИТКУ КРИПТОВАЛЮТ У СВІТІ І НА УКРАЇНІ / Е. М. Вавринчук Науково-технічна конференція "Сучасні інфокомунікаційні технології">// Тезиси. К. ДУТ, 5 грудня 2019р.
2. Вавринчук Е.М. РОЗРОБКА СИСТЕМИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ДЛЯ ГАЛУЗІ КРИПТОВАЛЮТИ / Е. М. Вавринчук Всеукраїнська науково-практична конференція "СИСТЕМНИЙ АНАЛІЗ В БІЗНЕСІ ТА УПРАВЛІННІ". // Збірник тез. К. ДУТ, 17 квітня 2020р.

17

Дякую за увагу!

