

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет комп'ютерних наук та кібернетики
Кафедра прикладної статистики

Кваліфікаційна робота
на здобуття ступеня бакалавра
за спеціальністю 124 Системний аналіз
на тему:

**Розробка інформаційної системи прогнозу курсу Bitcoin
на основі алгоритмів аналізу часових рядів**

Виконав студент 4-го курсу
Белоусов Нікіта Олексійович



Науковий керівник:
асистент, кандидат фізико-математичних наук
Шевчук Юлія Михайлівна



Засвідчую, що в цій роботі немає запозичень з
праць інших авторів без відповідних посилань.

Студент



Роботу розглянуто й допущено до захисту на
засіданні кафедри прикладної статистики

«6» червня 2022 р., протокол № 11

В.о. завідувача кафедри
Розора Ірина Василівна



Зміст

| | |
|--|----|
| ВСТУП | 3 |
| 1 ІНФОРМАЦІЙНІ СИСТЕМИ..... | 5 |
| 1.1 Структурованість..... | 9 |
| 1.2 Рівень автоматизації..... | 11 |
| 1.3 Функціонально-прикладне призначення..... | 12 |
| 2 ІНФОРМАЦІЙНІ СИСТЕМИ ДЛЯ ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ | 17 |
| 2.1 Постановка проблеми | 17 |
| 2.2 Часові ряди | 18 |
| 2.3 Прогнозування часових рядів..... | 19 |
| 2.4 Огляд моделей прогнозування | 22 |
| 3 КРИПТОВАЛЮТИ ТА ЇХ ПРОГНОЗУВАННЯ | 25 |
| 3.1 Постановка проблеми | 25 |
| 3.2 Способи із використанням машинного навчання..... | 26 |
| 3.3 Способи із використанням нейронних мереж..... | 27 |
| 3.4 Класичні способи | 27 |
| 4 РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ | 29 |
| 4.1 Завантаження та попередня обробка даних | 29 |
| 4.2 Аналіз часового ряду | 30 |
| 4.3 Побудова моделі часового ряду | 33 |
| 4.4 Аналіз та оцінка моделі | 35 |
| ВИСНОВКИ..... | 38 |
| Список використаної літератури..... | 39 |
| Додаток А | 41 |
| Інформаційна система (код програми)..... | 41 |

ВСТУП

В останній час крипторинки отримав дуже великий ріст та популярність серед інвесторів та різних компаній. У 2017 році курс найвідомішої криптовалюти Bitcoin різко почав рости, через два роки він коштував майже в 10000 разів більше, чим привернув увагу многих людей. На цю валюту почав рости попит, що спровокувало ще більш стрімкий ріст ціни. Саме через таку неймовірну популярність виникла потреба в розробці спеціальних бірж для торгів криптографічною валютою. Ї з кожним днем їх кількість збільшується відповідно до об'ємів інформації, яку потрібно обробляти у режимі онлайн. Вирізняють два основні напрямки розвитку:

1. професійні криптотрейдери, які постійно власноруч аналізують та прогнозують напрямки руху цін. Такий спосіб створює вагомні затримки, що має негативний вплив й на результат.
2. веб-платформи, які надають у користування свою програму та інформаційну систему, які повністю автоматизовані, що дуже прискорює процес.

Недоліком веб-платформ є відсутність безкоштовного доступу. Окрім того, через великий потік користувачів, можуть виникати значні часові затримки, які можуть критично впливати на зменшення ефективності використання подібних веб-платформ.

Але не кожна людина, яка має бажання займатися криптотрейдингом, має доступ до професійних криптобірж або трейдерів, які допоможуть спрогнозувати рух ліній тренду. Але й так само не кожна людина має великі обчислювальні можливості та місця зберігання неосяжних об'ємів даних, щоб користуватися складними професійними системами аналізу. Тому виникає необхідність в розробці інформаційної системи для аналізу курсу криптовалют, яка б підходила за системними вимогами будь-якому користувачу та була якнайбільш точною у своїх прогнозах.

Дана робота присвячена дослідженню математичних моделей, які могли б змістовно моделювати курс Bitcoin, та при цьому не були вельми складними для використання на практиці. Задачею даної роботи є дослідити методи часових рядів, що використовуються для оцінки динаміки курсу Bitcoin, проаналізувати особливості використання даних методів та розробити прототип інформаційної системи для моделювання курсу Bitcoin.

Структурно робота складається з титульного аркушу, змісту, вступу, основної частини, що складається з чотирьох розділів з підрозділами, висновків, списку використаної літератури та додатків. Кваліфікаційна робота складається з 43 стор. та містить 20 рис., 20 джерел та 1 додаток.

1 ІНФОРМАЦІЙНІ СИСТЕМИ

Загалом інформаційну систему (ІС) можна визначити як сукупність організаційних і технічних засобів для збереження та обробки інформації з метою забезпечення інформаційних потреб користувачів (абонентів). ІС здавна (в тому чи іншому вигляді) широко застосовують у життєдіяльності людства. Це пов'язано з тим, що для існування цивілізації необхідним є обмін інформацією як між окремими членами і колективами суспільства, так і між різними поколіннями. Інформаційні системи, як і інформація та інформаційні технології, існували з моменту появи суспільства, оскільки на будь-якій стадії його розвитку є потреба в управлінні. А для управління необхідна систематизована, заздалегідь підготовлена інформація.

Інформаційна система (ІС) — це набір взаємопов'язаних компонентів, які збирають, обробляють, зберігають та поширюють дані та інформацію; інформаційна система забезпечує механізм зворотного зв'язку для моніторингу та контролю за її роботою, щоб переконатися, що вона продовжує виконувати свої цілі та завдання. Механізм зворотного зв'язку має вирішальне значення для того, щоб допомогти організаціям досягти своїх цілей, таких як збільшення прибутку або покращення обслуговування клієнтів. Комп'ютерна інформаційна система (СБІС) — це єдиний набір апаратних засобів, програмного забезпечення, баз даних, мереж, людей і процедур, які налаштовані для збору, маніпулювання, зберігання та обробки даних в інформацію. Компанії все частіше впроваджують комп'ютерні інформаційні системи у свої продукти та послуги. Інвестиційні компанії пропонують своїм клієнтам широкий спектр потужних інвестиційних інструментів, включаючи доступ до обширних онлайн-досліджень. Автомобілі доступні з розширеними навігаційними системами, які не потребують водія щоб довести вас до місця призначення, а по дорозі покаже інформацію про останні погодні умови та умови дорожнього руху, щоб допомогти вам уникнути заторів і затримок на дорогах. Годинники, цифрові камери, мобільні телефони, музичні плеєри та

інші пристрої покладаються на CBIS, щоб надати своїм користувачам новітні та найкращі функції. Компоненти CBIS показані на рисунку 1.1.

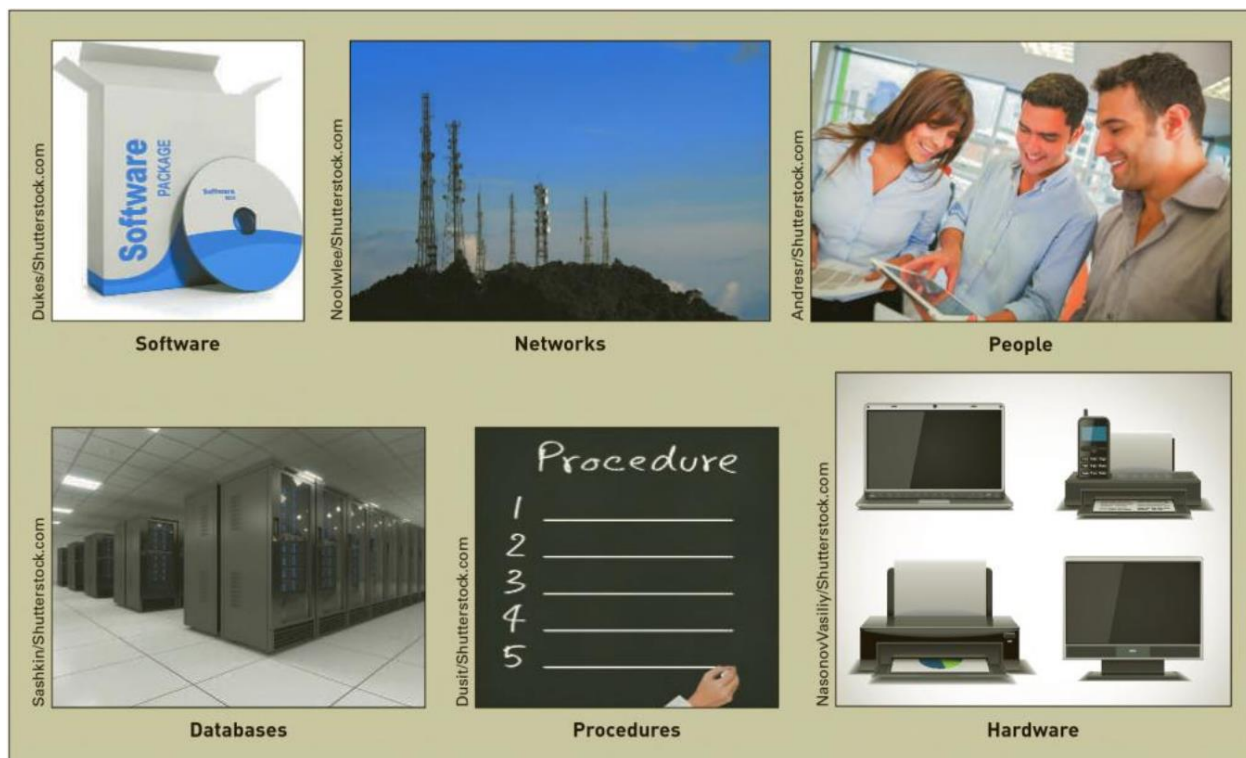


Рисунок 1.1 Компоненти комп'ютерної інформаційної системи

Технологічна інфраструктура організації включає в себе все апаратне забезпечення, програмне забезпечення, бази даних, мережі, людей і процедури, які налаштовані для збору, маніпулювання, зберігання та перероблення даних в інформацію. Технологічна інфраструктура — це набір спільних ресурсів ІС, які утворюють основу кожної комп'ютерної інформаційної системи. Люди визначають різницю між успіхом і невдачею в усіх організаціях. Відомий американський письменник та бізнес-консультант Джим Коллінз у своїй книзі «Від хорошого до великого» сказав: “Ті, хто створює великі компанії, розуміють, що кінцевим гальмівним механізмом зростання будь-якої великої компанії є не ринки, не технології, конкуренція чи продукти. Це одна річ: здатність отримувати та утримувати достатню кількість потрібних людей.” Таким чином не дивно, що люди є найважливішим елементом комп'ютерних інформаційних систем. Хороші системи можуть дозволити людям досягати надзвичайних результатів. Вони також можуть

підвищити роботу, задоволення та продуктивність працівників. Персонал інформаційних систем включає всіх людей, які керують, запускають, програмують та обслуговують систему, включаючи головного інформаційного директора, який очолює усю інформаційну систему. Кінцеві користувачі — це люди, які безпосередньо працюють з інформаційними системами щоб отримати результати. До них входять фінансові керівники, представники з маркетингу та оператори виробничих ліній. Процедура визначає кроки, які необхідно виконати для досягнення конкретного кінцевого результату. Наприклад ввести замовлення клієнта, оплатити рахунок постачальнику або запитати поточний звіт про запаси. Хороші процедури описують, як досягти бажаного кінцевого результату, хто, що і коли робить, і що робити, якщо щось піде не так. Коли люди добре навчені та дотримуються ефективних процедур, вони можуть виконувати роботу швидше, скорочувати витрати, краще використовувати ресурси та легше адаптуватися до змін. Якщо процедури добре задокументовані, вони можуть значно знизити витрати на навчання та скоротити час навчання. Використання CBIS передбачає налаштування та виконання багатьох процедур, у тому числі для експлуатації, обслуговування та безпеки системи. Наприклад, деякі процедури описують, як отримати доступ до системи за допомогою певної процедури входу та пароля. Інші описують, хто може отримати доступ до фактів у базі даних або що робити, якщо катастрофа, наприклад пожежа, землетрус або ураган, робить CBIS непридатним для використання. Якісні процедури можуть допомогти компаніям скористатися новими можливостями та уникнути тривалих перебоїв у бізнесі у разі стихійних лих. Погано розроблені та неадекватно реалізовані процедури, однак, можуть призвести до того, що люди витрачають свій час на марні правила або до неадекватного реагування на катастрофи. Отже, інформаційна система — це певна організаційно-технічна система, яка поєднує в собі комплекс програм, наприклад для збору, збереження та аналізу даних, правил, наприклад як вести себе в тій чи іншій ситуації, та людини, яка, маючи усі отримані дані приймає остаточне рішення. Термін «інформаційна

система» може бути віднесений до широкого класу систем — від найпростіших типу «телефон — секретар — комп'ютер — база даних» до міжнаціональних виробничих та невиробничих систем. У загальному вигляді структурнофункціональна схема ІС зображена на рисунку 1.2.

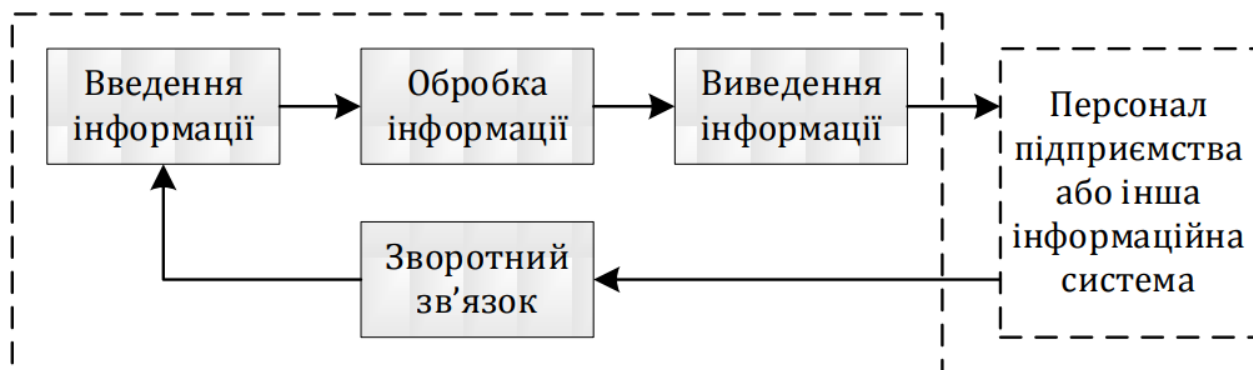


Рисунок 1.2 Структурна схема інформаційних систем

Як і у кожній великій та складній структурі інформаційні системи мають багато рівнів управління, але можна виділити три основні:

1. стратегічний;
2. тактичний;
3. оперативний.

На кожному з рівнів користувач потребує інформації. Тобто він має зробити запит в інформаційну систему. Вона в свою чергу знайде необхідно інформацію, обробить її, якщо це буде потрібно, сформулює відповідь на запит та передасть інформацію користувачеві. Таким чином на кожному з рівнів інформаційна система, а саме її відповідь на відповідний запит, може бути основою для прийняття рішення. Також для опису ІС використовуються такі класифікаційні ознаки:

1. структурованість завдань, розв'язуваних ІС;
2. рівень автоматизації інформаційного процесу;
3. функціонально-прикладне призначення ІС.

Тож розглянемо особливості використання наведених типів ІС.

1.1 Структурованість

Вирізняють три типи завдань у розробці та моделюванні ІС

(див. рисунок 1.3):

1. структуровані, або формалізовані;
2. неструктуровані, або неформалізовані;
3. частково структуровані.

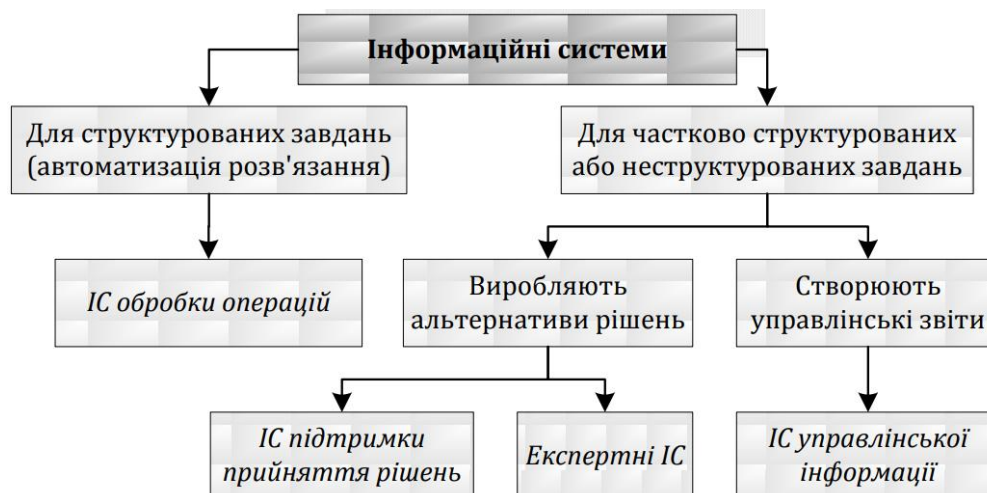


Рисунок 1.3 Поділ інформаційних систем на типи

Структурованими називають завдання, в яких відомі усі параметри, елементи та взаємозв'язки між ними. В свою чергу неструктуровані завдання це такі, у яких неможливо виділити елементи та їх взаємозв'язки.

Структуровані завдання – це такі, у яких ми маємо певну структуру. Наявність структури визначають відомість параметрів, елементів та зв'язків між ними.

Маючи усі дані ми можемо створити математичну модель нашої задачі, а далі й розробити алгоритм для розв'язання. Аналогічні завдання доводиться вирішувати доволі часто. Вони можуть бути не складні, але виявивши певний алгоритм можна скоротити час рахування усіх необхідних даних. Тож метою використання інформаційних систем є повна автоматизація процесу їх аналізу та розв'язання. Тобто необхідність присутності людини зводиться майже до нуля. Як приклад можна навести задачу реалізування розрахунку вартості перевезення вантажу. Нам відомі витрати за перевезення певного вантажу,

певної ваги та на певну дальність. Тож ця задача є структурованою й ми можемо розробити алгоритм для її вирішення. Бо цю задачу потрібно буде розв'язувати постійно с новими вхідними даними, а наявність інформаційної системи, розробленої з нашим алгоритмом, допоможе скоротити час та облегшити вирішення, користувачеві необхідно буде лише ввести нові дані.

В свою чергу неструктуровані задачі не мають яскраво виражених елементів та зв'язків між ними. Тому ми не маємо змоги розробити математичний опис й алгоритм, що дещо ускладнює розв'язання. Але й тут ми можемо використати інформаційні системи. Складно формалізувати та математично описати взаємовідносини між колегами, тому що психологічні й соціальні фактори майже не піддаються математичному опису. Але майже не існують повністю неструктурованих задач. Майже всі вони частково структуровані. Наприклад ми маємо дані ефеутивності праці того чи іншого колективу, або окремих людей у підрозділі. Цих даних достатньо щоб створити інформаційну систему, яка зможе зробити мінімальний аналіз усіх даних та видати звіт, на основі якого керуючий зможе прийняти більш вірне та проаналізоване рішення.

В цілому можна виділити два підтипи інформаційних систем, направлених на розв'язання частково структурованих задач:

1. інформаційні системи, що, на основі оброблених даних (в обробку може входити все: пошук, фільтрація, сортування) створює управлінські звіти. За допомогою цих звітів та інформації що в них керівник приймає певне рішення;

2. інформаційні системи, які, аналізуючи усі наявні дані, розробляють можливі альтернативні рішення. Тоді керівнику дається декілька варіантів, з яких він обирає найоптимальніший.

Останній підтип поділяють ще на два підтипи:

1. інформаційні системи підтримки прийняття рішень;
2. експертні інформаційні системи.

Перший підтип надає різні моделі, які можуть описати дані, будь то математичні, статистичні чи фінансові. Застосування цих моделей полегшує визначення й оцінювання альтернативних рішень. Користувач, зробивши запит до інформаційної системи й надавши ті дані, які в нього є, та після обробки отримає у відповідь інформацію, якої йому бракувало для прийняття рішення.

Експертні інформаційні системи мають більш автоматизовану структуру. Вони розробляють, порівнюють та оцінюють альтернативні рішення на основі попередніх розв'язань аналогічних або схожих задач та проведеного над ними аналізу.

1.2 Рівень автоматизації

Вище ми декілька разів казали про автоматизовані інформаційні системи. Але ІС мають ще декілька підтипів у цій сфері:

1. частково автоматизовані;
2. повністю автоматизовані.

Частково автоматизовані інформаційні системи, як можна зрозуміти з їх назви, потребують втручання людини та, можливо, певних технічних засобів для повної обробки даних.

В свою чергу автоматизовані інформаційні системи повністю беруть на себе увесь процес отримання та обробки даних. Також, урахувуючи їх широке застосування в організації процесів управління, вони мають різні модифікації й можуть бути класифіковані за деякими напрямками, представленими на рисунку 1.4.

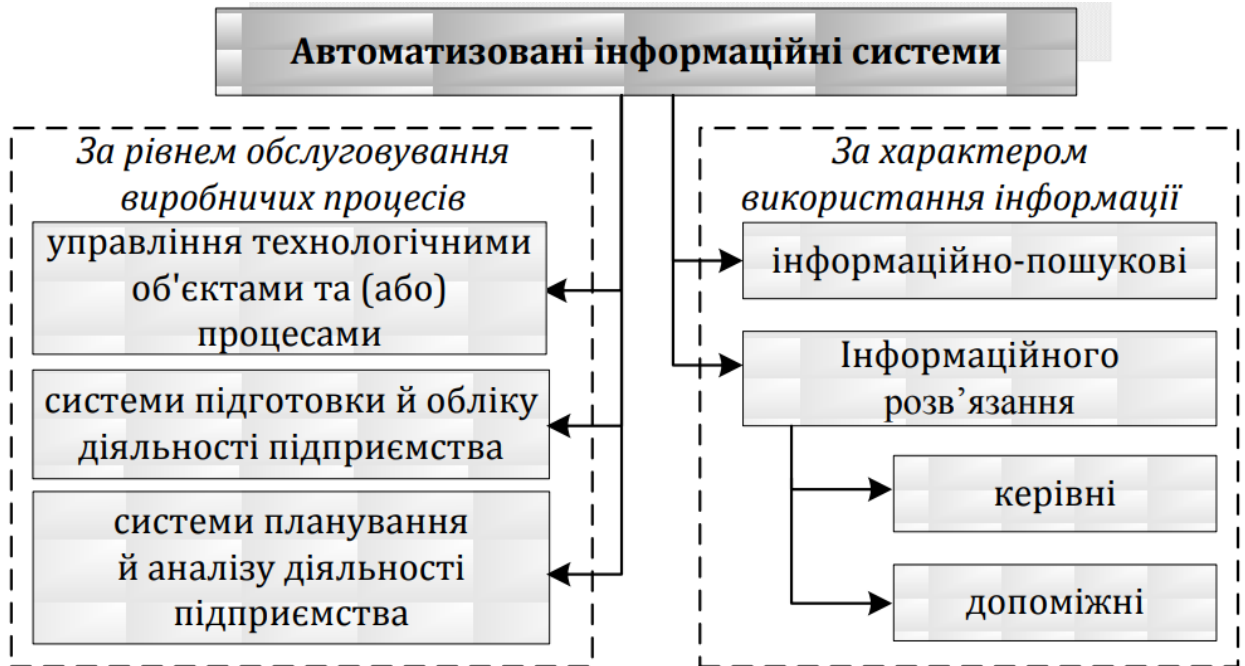


Рисунок 1.4 Поділ автоматизованих інформаційних систем

1.3 Функціонально-прикладне призначення

Також інформаційні системи поділяються за функціонально-прикладними сферами та управлінськими рівнями користувача. Це одні з найважливіших факторів визначення типу системи, на основі якого вона будується. Можна визначити зв'язок – чим вищий рівень управління, тим менший обсяг робіт, які виконує працівник за допомогою інформаційної системи. Однак чим вищий рівень управління, тим й складніша та більш інтелектуальна потрібна система. Відповідно зростає й її роль у прийнятті остаточного рішення.

Усі рівні потребують певної інформації. Але, в залежності від зони відповідальності, в різних обсягах та з різними методами обробки. На рисунку 1.5 представлена управлінська піраміда, в основі якої є інформаційні системи, які виконують первинну обробку даних, достатню для прийняття відповідних рішень на рівні оперативного управління. Вершиною ж піраміди буде стратегічний, найвищий рівень управління. На цьому етапі обробки даних інформаційні системи теж змінюють свою роль, становлячись стратегічними. На цьому рівні ми отримуємо найповніший звіт з обробки усіх наших даних, допомагаючи менеджерам вищої ланки прийняти рішення в умовах поганої

структурованості. Взагалі, згідно піраміди, інформаційні системи поділяються на три рівні:

1. оперативний;
2. функціональний;
3. стратегічний.

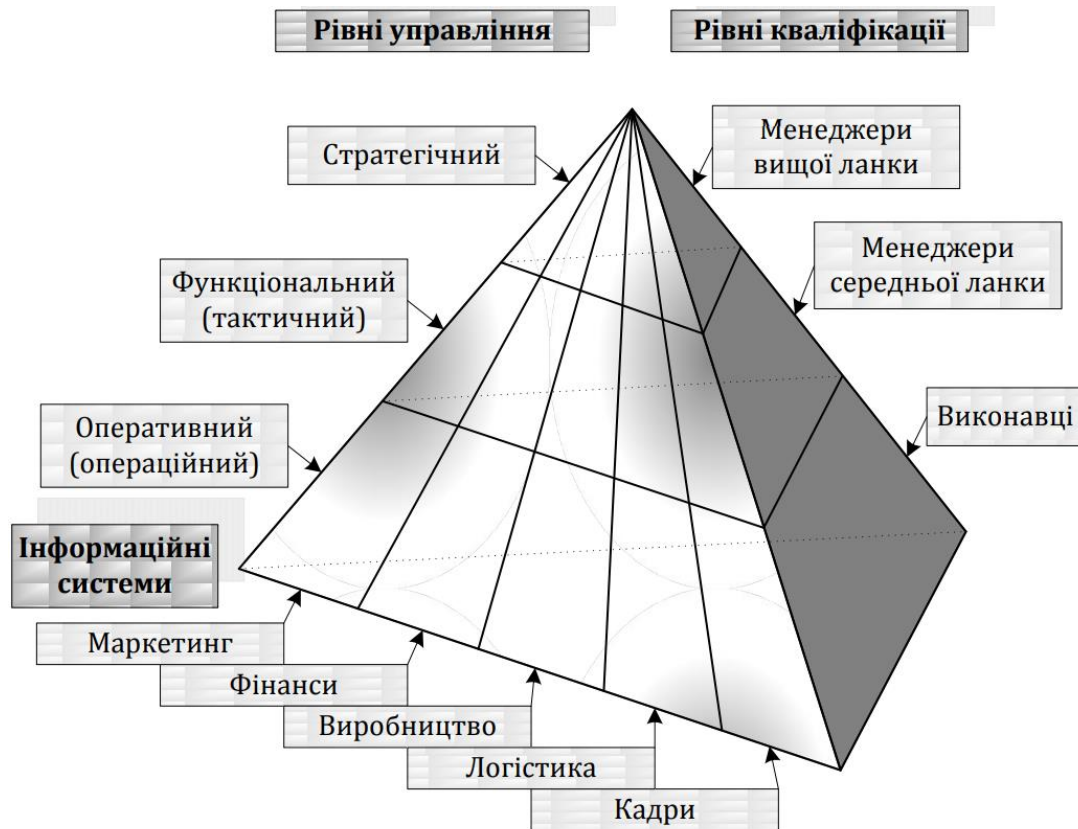


Рисунок 1.5 Пірамідальний вигляд структури інформаційних систем

По-перше розглянемо найнижчий, оперативний рівень. Інформаційні системи на цьому рівні слугують підтримкою працівника. Вони працюють з даними про угоди та події. Наприклад працівник может відкрити чи перевірити рахунок, виплатити зарплати, кредити, відслідковувати потік сировини на підприємство. Тож можна сказати що призначення інформаційних систем на цьому рівні – давати звіти про поточний стан будь-яких необхідних користувачу справ.

Усі задачі та джерела інформації, необхідні для їх вирішення, заздалегідь структуровані, а розв'язання цілком відповідає заданому алгоритму.

Як приклади інформаційних систем оперативного рівня можна навести бухгалтерські системи, системи депозитів, обробки замовлень будь-якої сфери

тощо. Також інформаційні системи рівня фахівців допомагають працівникам, за допомогою отриманих даних покращити якість та швидкість праці. Також серед завдань інформаційної системи цього рівня можна виділити інтеграцію відомостей та обробку документів для їх збереження. Ще у цьому класі можна виділити дві підгрупи інформаційних систем:

1. офісної автоматизації;
2. обробки знань.

Перші найчастіше застосовуються працівниками середньої ланки, наприклад бухгалтерами чи секретарями. Як основні завдання таких систем можна виділити обробку даних або полегшення рутинних підрахунків, у тому числі завдяки автоматизації та алгоритмізації розв'язку. Також інформаційні системи офісної автоматизації слугують засобом зв'язку між колегами, замовниками, покупцями та партнерами. Ще до основних задач можна віднести управління документацією, складання розкладу тощо.

В свою чергу інформаційні системи обробки знань необхідні для знаходження, обробки та зберігання знань, які можуть бути корисні інженерам, юристам, ученим для розроблення або створення нового програмного продукту. Їхнє завдання полягає у створенні нової інформації й нового знання.

Інформаційні системи рівня менеджменту використовуються працівниками середньої управлінської ланки для моніторингу актуальної інформації, контролю правильної роботи інформаційної системи, прийняття деяких рішень та адміністрування. До основних функцій можна віднести порівняння показників на різних часових відрізках, складання звітів за певний період, забезпечення доступу до архівної інформації працівників інших рівнів тощо. На цьому рівні можна виділити два типи інформаційних систем:

1. управлінські системи;
2. системи підтримки прийняття рішень.

Управлінські системи необхідні для керівників, наглядачів, які постійно моніторять інформацію про операції, проведені на підприємстві й періодично

формують звіти, яких викладають усі спостереження. Системи підтримки прийняття рішень необхідні для розв'язання частково структурованих задач, які важко спрогнозувати точно й заздалегідь. Такі системи мають дуже сильну аналітичну частину з використанням кількох моделей, а необхідну інформацію отримують з інноваційних систем інших рівнів. Основними користувачами інформаційних систем цього типу є ті, хто приймає рішення: аналітики, менеджери та інші фахівці.

Стратегічні інформаційні системи. Усі описані вище системи та методи роботи з даними не будуть мати сенсу без гарно розробленої стратегії. Стратегія — це комплекс методів та заходів знаходження перспективних ідей та їх розвиток у довгостроковій перспективі. Останнім часом з'являється все більше інформаційних систем, які оцінюються як стратегічно важливі. Їх робота, яка охоплює найповніший аналіз усіх наявних у компанії даних, може впливати на зміну вибору цілей підприємства, його завдань, методів, товарів, послуг. Такий підхід дає змогу оперативно керувати інформацією та випереджувати конкурентів, маючи більш вигідне місце на ринку. Через усі ці фактори й можна відокремити такий важливий тип інформаційних систем як стратегічний. Окрім внутрішніх даних на підприємство також дуже впливають зовнішні (див. рисунок 1.6). До цих факторів належать:

1. зовнішнє ринкове середовище, що спрямовує розвиток підприємства та визначає вимоги до конкурентоспроможного продукту;
2. нормативно-правове регулювання з боку держави, що накладає певні обмеження на діяльність підприємства;
3. споживачі, що мають різні можливості, потреби, мотивацію з вибору та придбання товарів та послуг;
4. постачальники, які здійснюють власну цінову політику.

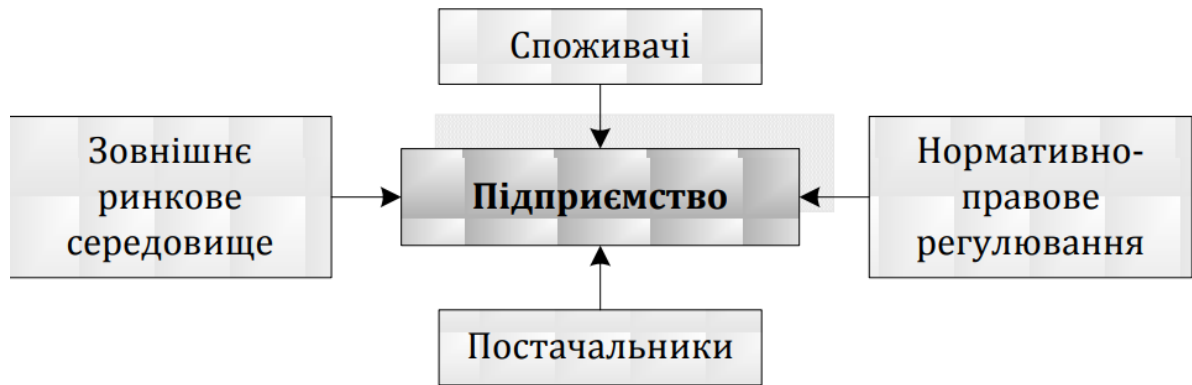


Рисунок 1.6 Зовнішні фактори, які впливають на підприємство

Підприємство може забезпечити високу конкурентоздатність, якщо правильно врахує усі фактори. А для ще більшого успіху виокремлюють декілька стратегій:

1. створення таких зв'язків, які закріплюють постачальників і покупців за даним підприємством, коли не вигідно звертатися до іншого;
2. знаходження ринків, де товари і послуги підприємства мають відмітні ознаки порівняно з наявними аналогами;
3. створення нових товарів і послуг, які вигідно відрізняються від аналогічних;
4. зниження вартості продукції без зниження якості.

Інформаційні системи стратегічного рівня використовуються керівниками вищої ланки за для полегшеного розв'язання неструктурованих або частково структурованих задач, а також розробки довгострокової стратегії підприємства. Основне їх призначення — порівняння змін, що відбуваються в зовнішньому середовищі, з наявним потенціалом підприємства.

2 ІНФОРМАЦІЙНІ СИСТЕМИ ДЛЯ ПРОГНОЗУВАННЯ ЧАСОВИХ

РЯДІВ

У цьому розділі розглянуто існуючі інформаційні системи, їх порівняння між собою та обрано підхід для аналізу часових рядів.

2.1 Постановка проблеми

У наш час жодна галузь, сфера діяльності чи система не стоїть на місці. Їде постійний розвиток, іноді різкий, або напроти спокійний, повільний. Й саме тому наразі дуже актуальною та одною з найбільш складних задач є задача прогнозування. Складність цього процесу обумовлена необхідністю обробки, зберігання та постійного аналізу неймовірно великих об'ємів інформації, необхідністю використовувати кілька методів обробки за для покращення результату й зменшення можливої похибки, постійною появою нових, більш сучасних методів тощо. Саме тому зараз розвиток методів прогнозування дуже тісно пов'язаний з розвитком інформаційних технологій.

Окрім складності усіх процесів безумовно виникає потреба їх зв'язку один до одного, що спричиняє необхідність розробки інформаційної системи прогнозування. Такі системи дуже широко використовуються у найрізноманітніших галузях: фінансова математика, економетрика, статистика тощо. Системи прогнозування є дуже важливою, майже невід'ємною частиною усіх сучасних процесів управління та прийняття рішень.

Під інформаційною системою прогнозування зазвичай розуміють якусь сукупність програм, кожна з яких виконує певну функцію: знаходження необхідної інформації, обробка, зберігання, аналіз, прогнозування та видача вихідних даних з прогнозом. Аналіз проводиться на основі тих методів й алгоритмів, яких потребує користувач та задача, надана їм. Системи прогнозування складаються з бази даних за великий проміжок часу (для

порівняння актуальних даних з минулими), комплекс аналітичних та прогнозних програм, моделей та комплексу методів оцінки якостя прогнозу. Усі складові обираються відповідно до суті поставленої задачі.

Під процесом прогнозування розуміється саме обробка та аналіз, на основі існуючих або спеціально розроблених підходів та методів, отриманих даних з показниками тих самих процесів минулого часу, тобто порівняння з ретроспективними даними. Зазвичай фіксуються лише якісь значущі значення та характеристики, які визначають перехід між станами та процесами. Тому більшість таких процесів описується саме часовими рядами, бо вони дозволяють послідовно розставити величини, які будуть використовуватися для порівняння та аналізу.

2.2 Часові ряди

Аналіз і прогнозування даних часових рядів стають все більш важливими через їх масове виробництво. Безперервний моніторинг і збір таких даних стають дуже поширеними, тому ми можемо очікувати, що кількість, якість і цінність даних часових рядів будуть експоненціально зростати в найближчі роки. Робота з часовими рядами, підготовка і обробка даних, прогнозування майбутніх значень і аналіз результатів стають частішими завданнями в повсякденному житті людей. Ці типи завдань стають широко поширеними для працівників спеціалізованих областей і в більш загальних сферах бізнесу. Будь-який часовий ряд визначається конкретним значенням або вимірюванням у конкретний момент часу. Ефективні інформаційні системи для прогнозування часових рядів невід'ємною частиною багатьох процесів прийняття рішення у різних галузях, а розробка таких систем стає все більш актуальною задачею. Необхідність їх застосування можна пояснити на прикладі економічної галузі. У цій сфері є багато факторів, таких як глобалізація, ріст темпів розвитку різних ринків та інше, які ведуть до непостійності розвитку, з невизначеністю, нестійкістю та різними ризиками. Така нестабільність дуже ускладнює використання звичайних методів аналізу

та прогнозу, бо вони матимуть малу ефективність на довгостроковій перспективі. Саме через усі ці фактори виникає необхідність розробки, модифікації та розвиненню методів та моделей аналізу та прогнозування процесів, а також створення більш оперативних систем, які б одразу реагували на будь які зміни в розвитку та швидко перераховували прогнози. Кінцевою ціллю є розробка швидкої, автоматизованої та як умога точнішої інформаційної системи.

2.3 Прогнозування часових рядів

Спочатку сформулюємо означення часового ряду. Нехай S – дискретна множина. Дискретним часовим рядом

$$\{z_i\}_{i=1}^n = \{z_1, z_2, \dots, z_n\} = \{z(t_1), z(t_2), \dots, z(t_n)\}$$

будемо називати послідовність n вимірювань, які проводяться в дискретні моменти часу $t_1 \in S, i = \overline{1, n}, n \in N$ і фіксують певні зміни показників процесу або явища, яке досліджується; t_1 – початковий момент часу. Будемо вважати, що вимірювання проводяться через встановленні часові інтервали: хвилина, година, день, тиждень, місяць, квартал тощо. Результати вимірювань дійсні $z_1 \in R, i = \overline{1, n}, n \in N$ і можуть являти собою однорідні економічні показники (рівень інфляції, об'єм випущеної продукції), фінансові показники тощо. В економіці та схожих галузях часовий ряд описують як часткову реалізацію стохастичного процесу, який представляється нескінченною випадковою послідовністю. Але значення часового ряду не є незалежними та однаково розподіленими, на відміну від елементів випадкової вибірки. Серед задач прогнозування часових рядів можна виділити основні три:

1. прогнозування майбутніх значень;
2. прогнозування знаків приростів часових рядів;

3. ідентифікація моментів локальних екстремумів.

Формулювання першої задачі має такий вигляд: на основі ретроспективних значень $z_n, z_{n-1}, \dots, z_{n-m+1}$, $m \leq n$ ряду $\{z_i\}_{i=1}^n$, найбільш точно оцінити його поведінку в майбутньому в моменти $t_{n+1}, t_{n+2}, \dots, t_{n+\theta}$, тобто побудувати послідовність прогнозних значень

$$\{\hat{z}_i\}_{i=n+1}^{n+\theta} = \{\hat{z}_{n+1}, \hat{z}_{n+2}, \dots, \hat{z}_{n+\theta}\}$$

де θ – горизонт прогнозування; m – об'єм ретроспективної вибірки.

Позначимо через $\hat{z}_\tau(n)$ прогноз, який розраховується в момент t_n (в точці n) на τ точок вперед, $\tau = \overline{1, \theta}$. Така функціональна залежність, яка дозволяє описати поведінку часового ряду, називається моделлю прогнозування. Позначимо модель для даної задачі через F_1 . Прогноз часового ряду $\{z_i\}_{i=1}^n$ на одну точку вперед на основі даної моделі можна формально записати у вигляді

$$\hat{z}_{n+1} = \hat{z}_1(n) = F_1(\hat{z}_{n-m+1}, \hat{z}_{n-m}, \dots, z_n).$$

У випадку прогнозування з горизонтом $\theta > 1$, застосовують ітераційний підхід:

$$\begin{aligned} \hat{z}_{n+1} &= \hat{z}_1(n) = F_1(z_{n-m+1}, z_{n-m+2}, \dots, z_n), \\ \hat{z}_{n+2} &= \hat{z}_2(n) = F_1(z_{n-m+2}, z_{n-m+3}, \dots, z_n, \hat{z}_{n+1}), \\ &\vdots \\ \hat{z}_{n+\theta} &= \hat{z}_\theta(n) = F_1(z_{n-m+\theta}, \dots, \hat{z}_{n+\theta-2}, \hat{z}_{n+\theta-1}), \quad m \leq n \end{aligned}$$

Вона може приймати різні значення, все залежить від типу обраної моделі. Прикладом моделей зазначеного вище підходу можуть слугувати адаптивні поліноміальні моделі. Точність оцінки побудованої моделі визначається тим, чи задовольняє відповідний критерій оцінки якості прогнозування. Зазвичай такими критеріями виступають середнє абсолютне відхилення, середня квадратична похибка, стандартне відхилення тощо. В економіці зазвичай

досить розрахувати та спрогнозувати короточасні зміни часового ряду та немає необхідності робити надто точні розрахунки безпосередньо прогнозних значень. Основною складністю є побудова моделі, яка б прогнозувала приріст значення ряду на одну точку вперед та була як умога точнішою. Зазвичай задачі такого роду використовуються для визначення майбутнього напрямку руху лінії тренду та точок відскоку. Тобто основна ціль це визначення ключових значень. В теорії ця модель може застосовуватися як частковий випадок попередньої задачі. Але для більш точного результату рекомендується застосовувати специфічні моделі та їх модифікації. Позначимо моделі цього типу через F_2 . Далі розглянемо формальний вигляд задачі прогнозування часових рядів. На основі ряду $\{z_i\}_{i=1}^n$, побудуємо ряд, який складається з перших різниць $\{\Delta z_i\}_{i=2}^n$, де $\Delta z_i = z_i - z_{i-1}, i = \overline{2, n}$. Позначимо через $\{\chi_i\}_{i=2}^n$ знаковий ряд, кожен елемент якого задається формулою: $\chi_i = \text{sgn}(\Delta z_i)$ або

$$\chi_i = \begin{cases} +1, & \Delta z_i > 0 \\ 0, & \Delta z_i = 0 \\ -1, & \Delta z_i < 0 \end{cases}, \text{ при } i = \overline{2, n},$$

а прогноз знаку приросту, який розраховується в точці n на τ точок вперед позначимо через $\hat{\chi}_\tau(n)$. Тоді модель F_2 у випадку прогнозування на одну точку вперед визначається так:

$$\hat{\chi}_{n+1} = \hat{\chi}_1(n) = F_2(z_{n-m+1}, z_{n-m+2}, \dots, z_n).$$

Також, у випадку прогнозування для $\theta > 1$ можна застосовувати ітераційний підхід, але на практиці його не часто використовують через велику кількість розрахунків. На відміну від попередньої в описаній вище задачі використовуються специфічні критерії оцінки якості. Зазвичай такі критерії будуються на основі функції Хевісайда або шляхом підрахунку кількості правильно передбачених знаків приростів та с подальшим урахуванням вагових коефіцієнтів для кожного з них.

Однією з найважливіших задач економіки та фінансів є прогнозування локальних екстремумів лінії тренду. Зазвичай для вирішення цієї задачі будуються спеціальні методи та алгоритми, основані на існуючих математичних моделях та методах аналізу. Також варто зазначити, що окрім розглянутих задач також мають місце випадки з урахуванням зовнішніх факторів, які мають вплив на точність прогнозу.

2.4 Огляд моделей прогнозування

У наш час часові ряди мають дуже велику популярність у використанні через їх зручність до опису зміни станів процесу. Також через постійне розвинення інформаційних технологій постійно з'являються все нові, більш потужні комп'ютери, які дозволяють швидко та точно робити усі необхідні підрахунки для аналізу та прогнозування. За останні 30 років біло розроблено чимало різних моделей, методів, та їх модифікацій. Та їх кількість все одно постійно зростає. Проте зазвичай використовуються лише декілька класичних, основних моделей через їх достатньо велику точність та доступність. Класичними моделями прогнозування задач першого розглянутого нами типу F_1 , які застосовуються впродовж багатьох років та добре показують свою ефективність для прийняття управлінських рішень, є:

1. моделі експоненціального згладжування;
2. регресійні моделі;
3. авторегресійні моделі.

Моделі експоненціального згладжування є дуже популярними серед аналітиків, тому що прості у використанні й мають наочний показник якості результату. Прикладами таких моделей можуть слугувати адаптивна поліноміальна модель Брауна, модель Хольта, яка добре показує трендову складову тощо.

Регресійні моделі використовуються, коли в нас є ряд незалежних змінних та одна шукана залежна. Суть методів полягає саме у знаходженні цієї залежності. Регресію поділяють на парну, в якій на основі однієї незалежної змінної робиться оцінка залежної, та множинна, в якій розглядають ряд незалежних змінних. Також виокремлюють лінійну та нелінійну регресії.

Авторегресійні моделі. Суть цих моделей полягає у тому, щоб визначити залежність між значеннями часового ряду. Робиться це за допомогою дослідження автокореляції між спостереженнями на кінцях певного інтервалу. Класичною моделлю цього класу є авторегресійна модель плинного середнього порядків p та q — ARMA (p, q).

Для коректного використання моделей, описаних вище, необхідно щоб вхідний часовий ряд був стаціонарним, тобто мав сталі математичне сподівання та дисперсію. Модель ARMA (p, q) має декілька широко поширених модифікацій ARIMA (p, d, q) і ARIMAX. Вони використовуються, коли ряд не стаціонарний та направлені на те, щоб шляхом деяких маніпуляцій привести його до стаціонарного виду. Надалі розвиток цього типу моделей пішов у напрямку ряду випадкових величин з різною дисперсією, так звана гетероскедастичність. Ці методи та модифікації знайшли своє застосування в першу чергу в економіці та фінансах, а самі для прогнозування часових рядів фондових ринків. Прикладами таких модифікацій класичних моделей слугують модель ARCH, яка спирається на залежність умовної дисперсії від попередніх значень часового ряду та модель GARCH (узагальнена ARCH), яка розглядає залежність не тільки від попередніх значень, а й від попередніх значень дисперсії. Але слід зазначити, що усе має як свої сильні сторони, так і недоліки. Наприклад роздивимось більш детально моделі експоненціального згладжування. Їх популярність обумовлена простотою у використанні та відкритістю, тобто користувач завжди знає як саме алгоритм аналізує дані. Але в той же час велика кількість обмежень до початкових даних не дозволяє використовувати широко цей метод. Регресійні моделі мають ті ж переваги,

що й попередній метод, але кожний з підтипів має свої недоліки. Наприклад модель, яка використовує лінійну регресію, не зможе обробити та описати нелінійні дані. Але, як відомо, більшість часових рядів мають саме нелінійну структуру. З іншого боку, при використанні нелінійної регресійної моделі, ми матимемо складність з визначенням залежностей та розрахунку параметрів, що також сильно обмежує поле можливого використання цього методу. Наразі дуже велику популярність набирають авторегресійні моделі. Найвідомішою та класичною моделлю цього класу є модель ARIMA, або, як її же називають, модель Бокса-Дженкінса. Її перевага над іншими методами полягає в тому, що усі оцінки параметрів розраховуються виключно для конкретного заданного часового ряду, що робить її більш гнучкою та розширює галузі використання. Недоліком цієї моделі є складність та рутинність процесу розрахунку численних параметрів. Також додатково обмежує використання вимога стаціонарності ряду, нормальності та незалежності залишків. Але в ході аналізу часових рядів за допомогою цих методів не є складною задачею позбутися багатьох недоліків, автоматизувавши деякі процеси за звести відхилення до мінімуму.

3 КРИПТОВАЛЮТИ ТА ЇХ ПРОГНОЗУВАННЯ

3.1 Постановка проблеми

В наш час криптовалюти мають неймовірну популярність та перспективність розвитку. Така популярність обумовлена тим, що на курс майже не впливають зовнішні фактори, бо криптовалюти є майже незалежним від усього матеріального ресурсом. Єдине що впливає на коливання лінії тренду – попит. Наприклад у 2017 році приблизна капіталізація ринку криптовалют становила 15,6 мільярдів доларів. В цей момент почав рости попит, а з ним й ціна. Й чим більше був попит, тим більше біла ціна. То ж у 2018 капіталізація складала вже 467 мільярдів доларів. Різні експерти підтверджують, що вартість криптовалют визначається переважно динамікою попиту. Але в цьому є й недолік, бо спекулянти та "великі" гравці, які мають великі суми грошей та можливості до покупки можуть власноруч змінювати попит, а з ним й ціну. Наприклад для найбільш популярної криптовалюти Bitcoin коливання курсу могли досягати десятикратної різниці протягом одного місяця. З прикладної точки зору питання доцільності використання такої валюти як довгостроковий інвестиційний актив пов'язане з ефективністю використання статистичних методів та їх модифікацій для прогнозування курсу.

На даний момент існують такі три найпопулярніші підходи до технічного прогнозування курсу на криптовалютному ринку:

1. способи із використанням машинного навчання (MLP, BNN, RBF, тощо);
2. способи із використанням нейронних мереж (RNN, LSTM);
3. класичні способи (Naive, ARIMA, тощо).

3.2 Способи із використанням машинного навчання

Розглянемо цей спосіб на конкретному прикладі.

MLP – Multi-Layer Perceptron. Дана модель має об'ємну складну структуру, яку можна представити у вигляді шарів:

1. початковий рівень, коли надходять початкові дані;
2. середній рівень, прихований від користувача, на якому виконується основна аналітична робота;
3. останній рівень, на якому виконується саме прогнозування.

Схема описаної вище структури дивитися на рисунку 3.1.

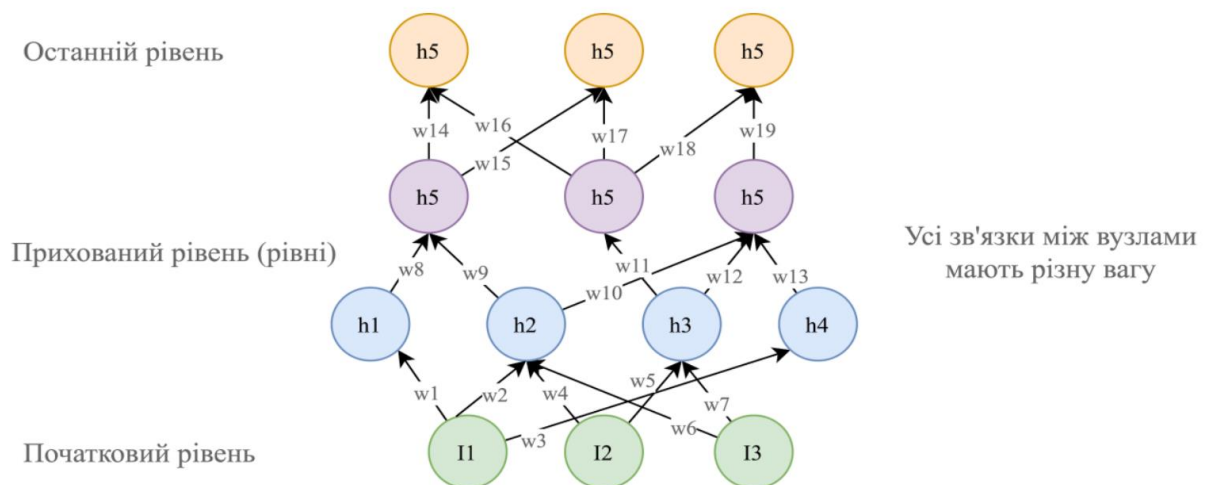


Рисунок 3.1. Схема MLP

Переваги використання машинного навчання:

1. достатньо мале відхилення від реальних результатів. Чим більше даних надається для обробки, тим точніше працює спосіб;
2. вміння змінюватися під конкретні дані;
3. наявність великої кількості методів, можливих для використання та порівняння між собою.

Недоліки використання машинного навчання:

1. для якісного прогнозування необхідна велика кількість даних;
2. потребує великих обчислювальних потужностей.

3.3 Способи із використанням нейронних мереж

Цей спосіб також розглянемо на конкретних існуючих методах:

1. RNN – Recurrent Neural Network – рекурентна нейронна мережа;
2. LSTM – Long Short-Term Memory – довга короткострокова пам'ять.

RNN – різновид нейронних мереж, в основі якого лежить принцип запам'ятовування попереднього стану для використання його при прогнозуванні майбутніх значень. LSTM — модифікація RNN зі спеціальними функціями, які використовуються моделлю для відкидання незначущих значень, що збільшує ймовірність збігу отриманого прогнозу з реальними даними.

Переваги використання нейронних мереж:

1. найсучасніші способи прогнозування, що показують високі результати;
2. наявність великої кількості способів, популярність на ринку.

Недоліки використання нейронних мереж:

1. потребують великих обчислювальних потужностей;
2. «чорна коробка» всередині, неможливо пояснити прогнозування;
3. для якісного прогнозування необхідна велика кількість даних.

3.4 Класичні способи

Криптовалютний ринок також має багато методів аналізу та прогнозування. Найпопулярнішим вважається клас моделей ARIMA. Також можливо використання його модифікацій, наприклад модифікація SARIMA, яка враховує сезонність тренду, або ARMA, для якої необхідною умовою є стаціонарність ряду.

Переваги використання класичних способів (ARIMA):

1. висока точність у прогнозуванні;
2. велика кількість модифікацій, а отже можливість адаптування до різних ситуацій та типів часових рядів;
3. не потребують великих обчислювальних потужностей;

4. для ефективної роботи не потрібен занадто великий об'єм даних, на відміну від інших способів.

Але, як було зазначено у минулому розділі, модель ARIMA також має свій недолік: менш добре працює у випадку різкої зміни напрямку руху курсу на криптовалютному ринку.

Для роботи було обрано класичний спосіб аналізу ARIMA за його доступність, швидкість роботи та невеликі системні вимоги. Тобто інформаційна система, розроблена за таким способом буде доступна для будь якого криптотрейдера, а не лише для великих трейдових компаній, яким доступні більші обчислювальні потужності та об'єми зберігаємих даних.

4 РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ

Для розробки програми була обрана мова програмування Python та встановлений пакет statsmodels. Даний модуль надає широкий набір засобів та методів для проведення статистичного аналізу та економетрики. Повний код програми наведений у Додатку А.

4.1 Завантаження та попередня обробка даних

Для початку завантажимо дані та подивимося на них. У цій роботі ми використовуємо дані курсу криптовалюти Bitcoin за період з 2020-01-01 по 2021-10-27.

Для початку завантажимо дані та подивимося на них (див. рисунок 4.1). Також на першому рисунку бачимо підключення усіх необхідних нам

| Date | Open | High | Low | Close | Adj Close | Volume |
|------------|-------------|-------------|-------------|-------------|-------------|--------------|
| 2020-01-01 | 7194.892090 | 7254.330566 | 7174.944336 | 7200.174316 | 7200.174316 | 1.856566e+10 |
| 2020-01-02 | 7202.551270 | 7212.155273 | 6935.270020 | 6985.470215 | 6985.470215 | 2.080208e+10 |
| 2020-01-03 | 6984.428711 | 7413.715332 | 6914.996094 | 7344.884277 | 7344.884277 | 2.811148e+10 |
| 2020-01-04 | 7345.375488 | 7427.385742 | 7309.514160 | 7410.656738 | 7410.656738 | 1.844427e+10 |
| 2020-01-05 | 7410.451660 | 7544.497070 | 7400.535645 | 7411.317383 | 7411.317383 | 1.972507e+10 |

Рисунок 4.1 Перші рядки даних

Отже, як можна помітити функція `read_csv()`, в даному випадку крім вказівки основних параметрів, які задають колонки та індекс, що використовуються, можна помітити ще 2 параметри для роботи з датою. Зупинимося на них детальніше. `parse_dates` задає імена стовпців, які будуть перетворені на тип `DateTime`. Завершальний параметр `dayfirst` вказує функції парсингу, у рядку першим йде день, потім місяць. Якщо не вказати цей параметр, то функція може не правильно перетворювати дати та плутати місяць та день місцями. Наприклад `01.02.2013` буде перетворено на `02-01-2013`, що буде неправильно.

Виділимо в окрему серію часовий ряд зі значеннями найвищих значень за кожен день. (див. рисунок 4.2)

```

Date
2020-01-01    7174.944336
2020-01-02    6935.270020
2020-01-03    6914.996094
2020-01-04    7309.514160
2020-01-05    7400.535645
Name: Low, dtype: float64

```

Рисунок 4.2 Перші рядки обраних для аналізу даних

Тепер в нас є часовий ряд, тож перейдемо до його аналізу.

4.2 Аналіз часового ряду

З початку подивимось на графік нашого ряду (див. рисунок 4.3).

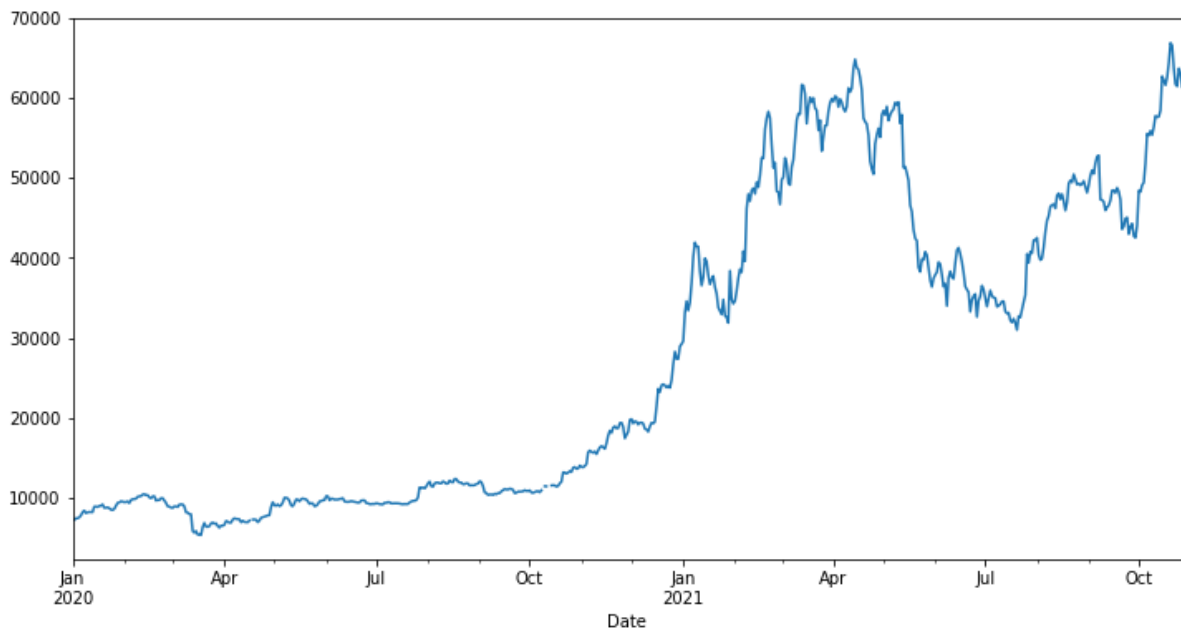


Рисунок 4.3 Графік часового ряду з обраних даних

Як можна помітити, графік не має яскравих викидів та має помітний тренд. З цих даних ми робимо висновок, що ряд не стаціонарен.

Задля більш детального аналізу виведемо деяку корисну інформацію та гістограму нашого ряду (див. рисунки 4.4.1 та 4.4.2).

```

count      662.000000
mean       27505.954832
std        19214.432721
min         5331.833984
25%        9698.759521
50%        18928.085938
75%        46222.665039
max        66930.390625
Name: High, dtype: float64

```

Рисунок 4.4.1 Опис даних

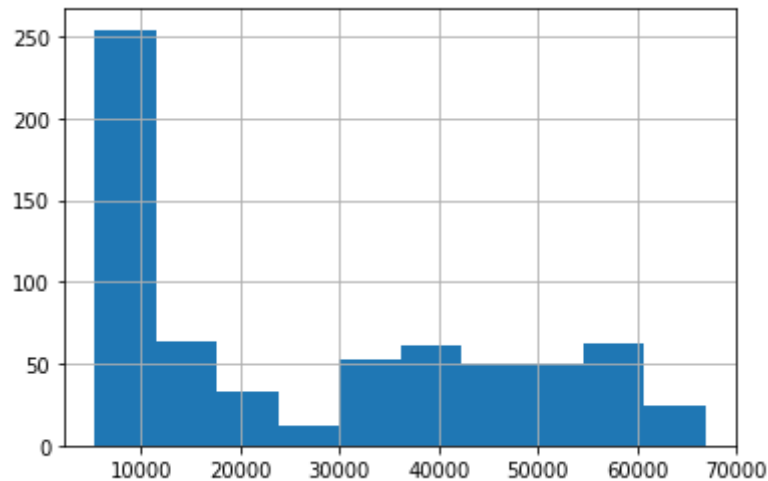


Рисунок 4.4.2 гістограма даних

Дивлячись на графік складно говорити про його однорідність. Тому підрахуємо коефіцієнт варіації $V = \frac{\sigma}{\bar{x}}$, де σ – середньоквадратичне відхилення, \bar{x} – середнє арифметичної вибірки. Значення беремо зі змінної describe.

$$V = 0.6985553796754184$$

З таким значенням можна казати що наша вибірка має відносно невеликий розкид, тож можна брати її для подальшого розгляду.

Проведемо тест Харки-Бера для визначення нормальності розподілу, щоб підтвердити припущення про однорідність. Тест Харки-Бера – це статистичний тест, який перевіряє помилки спостережень на нормальність. Для цього існує функція `jarque_bera()`, яка повертає значення даної статистики (див. рисунок 4.5)

| JB | p-value | skew | kurtosis |
|-------------------|-----------------------|--------------------|--------------------|
| 74.01107756721876 | 8.485915569041652e-17 | 0.4508941867104553 | 1.6384407512557444 |

Рисунок 4.5 Результат тесту Харки-Бера

Значення даної статистики свідчить про те, що нульова гіпотеза про нормальність розподілу відкидається з малою ймовірністю ($p\text{-value} > 0.05$), і, отже, наш ряд має нормальний розподіл.

Функція `SimpleTable()` служить оформленням виводу. У нашому випадку на вхід їй подається масив значень (розмірність не більше 2) та список із назвами стовпців або рядків. Багато методів і моделей засновані на припущеннях про стаціонарність ряду, але як було помічено вище, обраний нами ряд таким не є. Для перевірки стаціонарності проведемо узагальнений тест Діккі-Фуллера на наявність одиничного коріння. Для цього в модулі `statsmodels` є функція `adfuller()` (див рисунок 4.6).

```
adf: -0.48979281024740684
p-value: 0.8939930168732189
Critical values: {'1%': -3.4402817347322583, '5%': -2.8659224464259823, '10%': -2.5691038118332603}
Є одиничні корні, ряд не стаціонарен.
```

Рисунок 4.6 Результат перевірки стаціонарності ряду

Проведений тест підтвердив припущення про нестаціонарність ряду. У багатьох випадках взяття різниці рядів дозволяє це виправити. Якщо, наприклад, перші різниці ряду стаціонарні, він називається інтегрованим рядом першого порядку.

Отже, визначимо порядок інтегрованого ряду нашого ряду. У кодї функція `diff()` обчислює різницю вихідного ряду із рядом заданим зсувом періоду. Період усунення передається як параметр `period`. Так як в різниці перше значення буде невизначеним, то нам треба позбутися його, для цього і використовується метод `dropna()`.

Перевіримо ряд на стаціонарність (див. рисунок 4.7).

```
adf: -10.92175304596596
p-value: 1.0338141628118691e-19
Critical values: {'1%': -3.4402817347322583, '5%': -2.8659224464259823, '10%': -2.5691038118332603}
Одиничних корнів немає, ряд стаціонарен.
```

Рисунок 4.7 Результат повторної перевірки стаціонарності ряду

Як видно з коду ряд перших різниць, що вийшов, наблизився до стаціонарного. Для повної впевненості розіб'ємо його на кілька проміжків і переконаємося мат. очікування на різних інтервалах.

p-value: 0.50279387002780156

Високе p-value дає можливість стверджувати, що нульова гіпотеза про рівність середніх вірна, що свідчить про стаціонарності ряду. Залишилося переконатися у відсутності тренду (основної зміни), для цього збудуємо графік нашого нового ряду (див. рисунок 4.8):

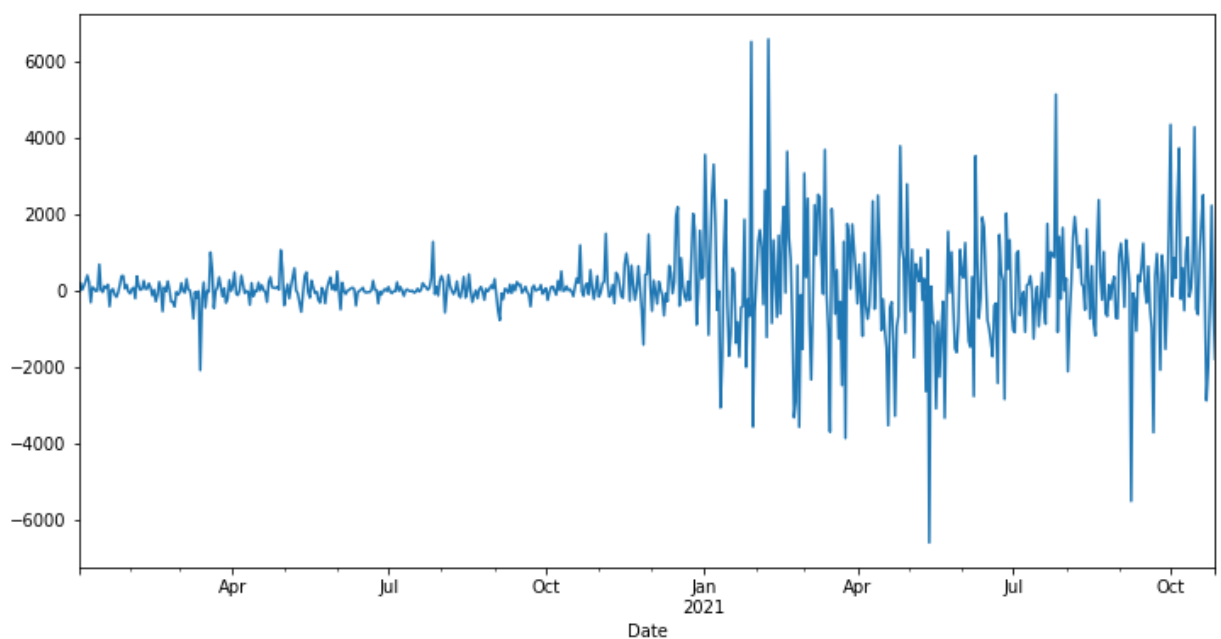


Рисунок 4.8 Графік ряду після приведення до стаціонарного

Тренд дійсно відсутній, таким чином ряд перших різниць є стаціонарним, а наш вихідний ряд — інтегрованим рядом першого порядку.

4.3 Побудова моделі часового ряду

Для моделювання використовуватимемо модель ARIMA, побудовану для ряду перших різниць. Отже, щоб побудувати модель, нам потрібно знати її порядок, що складається з 3-х параметрів:

p - порядок компоненти AR;

d - порядок інтегрованого ряду;

q - порядок компоненти MA.

Параметр d є і він дорівнює 1, залишилося визначити p і q . Для їх визначення нам треба вивчити авторкореляційну (ACF) та частково авторкореляційну (PACF) функції для низки перших різниць. ACF допоможе нам визначити q . За корелограмою можна подивитися кількість коефіцієнтів, які сильно відмінні від 0 моделі MA.

PACF допоможе нам визначити p , тому що по її корелограмі можна визначити максимальний номер коефіцієнта, що дуже відрізняється від 0 у моделі AR. Щоб побудувати відповідні корелограми, у пакеті `statsmodels` є такі функції: `plot_acf()` та `plot_pacf()`. Вони виводять графіки ACF і PACF, у яких на осі X відкладаються номери лагів, а на осі Y значення відповідних функцій. Слід зазначити, що кількість лагів у функціях і визначає кількість значних коефіцієнтів. Отже, наші функції мають наступний вигляд (див. рисунок 4.9).

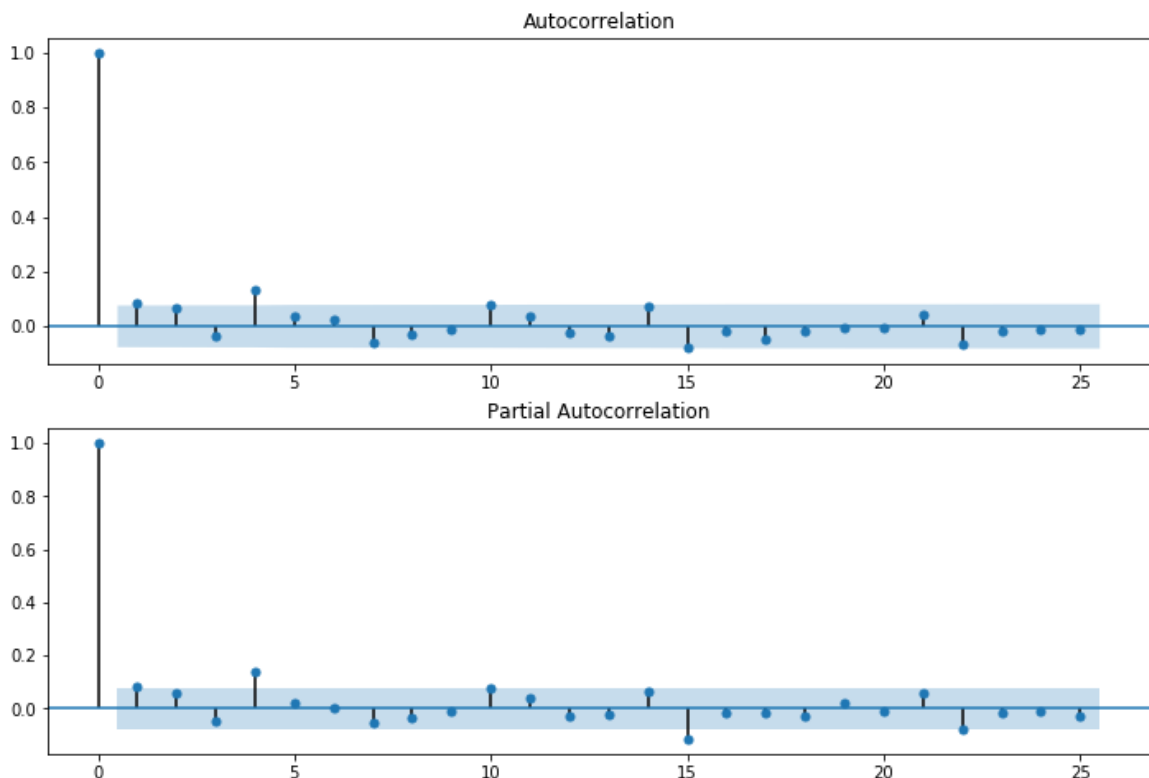


Рисунок 4.9 Графіки ACF і PACF

Після вивчення корелограми PACF можна дійти висновку, що $p = 3$, так як на ній тільки 3 лаги сильно відрізняються від нуля. По корелограмі ACF можна побачити, що $q = 1$, так як після лага 1 значення функцій різко падають.

Отже, коли відомі всі параметри можна побудувати модель, але для її побудови ми візьмемо не всі дані, лише частину. Дані з частини, що не потрапили в модель, ми залишимо для перевірки точності прогнозу нашої моделі. Як видно з отриманої інформації (див. рисунок 4.10) в нашій моделі всі коефіцієнти значущі (значення стовпчика real за модулем більше за 0.1 або 10%) і можна перейти до оцінки моделі.

| ARIMA Model Results | | | | | | |
|---------------------|------------------|---------------------|----------|-----------|----------|---------|
| Dep. Variable: | D.Low | No. Observations: | 46 | | | |
| Model: | ARIMA(3, 1, 1) | Log Likelihood | -368.712 | | | |
| Method: | css-mle | S.D. of innovations | 729.498 | | | |
| Date: | Mon, 06 Jun 2022 | AIC | 749.424 | | | |
| Time: | 16:33:04 | BIC | 760.396 | | | |
| Sample: | 01-12-2020 | HQIC | 753.534 | | | |
| | - 11-22-2020 | | | | | |
| ===== | | | | | | |
| | coef | std err | z | P> z | [0.025 | 0.975] |
| ----- | | | | | | |
| const | 269.2435 | 194.345 | 1.385 | 0.173 | -111.666 | 650.153 |
| ar.L1.D.Low | -0.6188 | 0.329 | -1.879 | 0.067 | -1.264 | 0.027 |
| ar.L2.D.Low | 0.4559 | 0.170 | 2.677 | 0.011 | 0.122 | 0.790 |
| ar.L3.D.Low | 0.1969 | 0.209 | 0.943 | 0.351 | -0.212 | 0.606 |
| ma.L1.D.Low | 0.7666 | 0.283 | 2.708 | 0.010 | 0.212 | 1.322 |
| Roots | | | | | | |
| ===== | | | | | | |
| | Real | Imaginary | Modulus | Frequency | | |
| ----- | | | | | | |
| AR.1 | 1.6069 | +0.0000j | 1.6069 | 0.0000 | | |
| AR.2 | -1.1331 | +0.0000j | 1.1331 | 0.5000 | | |
| AR.3 | -2.7893 | +0.0000j | 2.7893 | 0.5000 | | |
| MA.1 | -1.3044 | +0.0000j | 1.3044 | 0.5000 | | |
| ----- | | | | | | |

Рисунок 4.10 Модель ARIMA для обраних даних

4.4 Аналіз та оцінка моделі

Перевіримо залишки даної моделі на відповідність «білому шуму», а також проаналізуємо корелограму залишків, оскільки це може допомогти нам у визначенні важливих для включення і прогнозування елементів регресії.

Отже, перше, що ми зробимо, це проведемо Q-тест Льюнга — Бокса. Він використовується для перевірки, чи є залишки випадковими. Проведемо тест на залишках моделі ARIMA. Таким чином, нам треба спочатку отримати залишки моделі, а потім до коефіцієнтів, що вийшли, запустити тест. Це можна зробити за допомогою statsmodels (див. рисунок 4.11).

| | Q-stat | p-value |
|----|-----------|----------|
| 0 | 0.003781 | 0.950968 |
| 1 | 0.017173 | 0.991450 |
| 2 | 0.265431 | 0.966393 |
| 3 | 0.997320 | 0.910202 |
| 4 | 1.685721 | 0.890694 |
| 5 | 2.445649 | 0.874500 |
| 6 | 2.483977 | 0.928299 |
| 7 | 3.498010 | 0.899344 |
| 8 | 4.150688 | 0.901217 |
| 9 | 4.190087 | 0.938365 |
| 10 | 5.975531 | 0.874992 |
| 11 | 7.825221 | 0.798635 |
| 12 | 9.595444 | 0.726634 |
| 13 | 9.959105 | 0.765167 |
| 14 | 10.015919 | 0.818737 |
| 15 | 12.118665 | 0.735771 |
| 16 | 12.194896 | 0.788194 |
| 17 | 12.263973 | 0.833314 |
| 18 | 12.838382 | 0.846741 |
| 19 | 14.702677 | 0.793160 |
| 20 | 14.756747 | 0.834957 |
| 21 | 15.827013 | 0.824377 |
| 22 | 16.240715 | 0.844772 |
| 23 | 16.244589 | 0.879046 |
| 24 | 24.859381 | 0.470276 |
| 25 | 25.043943 | 0.516489 |
| 26 | 26.435009 | 0.494572 |
| 27 | 29.857189 | 0.370077 |
| 28 | 31.700732 | 0.333162 |
| 29 | 32.132643 | 0.361384 |
| 30 | 32.185048 | 0.407821 |
| 31 | 32.429776 | 0.445576 |
| 32 | 32.617492 | 0.486044 |
| 33 | 37.151000 | 0.325900 |
| 34 | 37.458440 | 0.356972 |
| 35 | 46.614446 | 0.110762 |
| 36 | 46.770739 | 0.130313 |
| 37 | 47.547497 | 0.137855 |
| 38 | 48.184549 | 0.148668 |
| 39 | 48.188163 | 0.175418 |

Рисунок 4.11 Результат Q-тесту

Значення даної статистики і p-values мають достатній розбіг щоб дійти висновку, що швидше за все залишкові дані є «білим шумом».

Залишилось намалювати наш прогноз на графіку (див. рисунок 4.12).

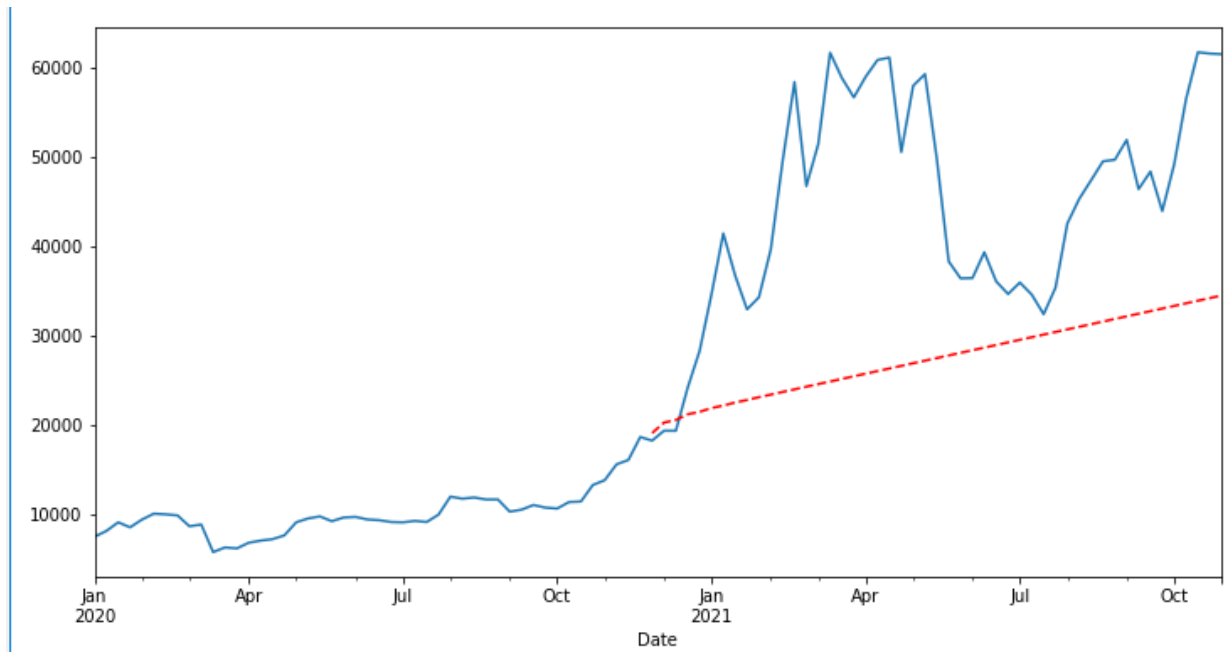


Рисунок 4.12 Графік отриманого прогнозу

ВИСНОВКИ

Отже, у роботі були розглянуті методи аналізу часових рядів та розроблена інформаційна система для аналізу курсу Bitcoin за допомогою вищезазначених методів. Як можна побачити на графіку отриманого прогнозу (див. рисунок 4.14) використана модель ARIMA та функції її реалізації у мові програмування python не дають стовідсотково точного прогнозу. Це може бути викликано неточністю деяких коефіцієнтів, функцій або не повнотою використаних даних. Так як у цьому аналізі використовувалися дані мінімумів за кожен проміжок часу, то наша модель добре описує майбутні мінімуми руху лінії тренду, що є не менш важливим у праці з криптобіржами та криптовалютами.

Список використаної літератури

1. Павленко П. М., Філоненко С. Ф., Бабіч К. С., Гавриленко О. В., Логачов Є. Г. – Інформаційні системи та технології, Навчальний посібник
2. Зелінська О.В., Потапова Н.А., Волонтир Л.О. – Інформаційні системи та технології в галузі, Навчальний посібник
3. Берзлев О.Ю. – Сучасний стан інформаційних систем прогнозування часових рядів.
4. Петренко В.О., Молоканова В.М., Дорожко Г.К. – Збірник наукових праць за матеріалами III Міжнародної науково-практичної інтернет-конференції.
5. Лукашин Ю.П. – Адаптивные методы краткосрочного прогнозирования временных рядов: Учеб. пособие.
6. Bollerslev Tim. – Generalized autoregressive conditional heteroskedasticity
7. Draper N., Smith H. – Applied regression analysis.
8. Тихонов Э.Е. – Методы прогнозирования в условиях рынка: учеб. пособие.
9. Brown Robert G. – Statistical forecasting for inventory control.
10. Holt Charles C. – Forecasting trends and seasonal by exponentially weighted averages.
11. Box G.E.P., Jenkins G.M., – Time series analysis: forecasting and control.
12. Vercellis Carlo. – Business intelligence: data mining and optimization for decision making.
13. Jaideep Kaur, Kamaljit Kaur. – Internet of Things: A Review on Technologies, Architecture, Challenges, Applications, Future Trends.
14. Aroosa Umair, Ureeba Ashfaq, Muhammad Gufran Khan. – Recent Trends, Applications, and Challenges of Brain-Computer Interfacing.

15. Investors Underground: Introduction to Technical Analysis
[Электронный ресурс] // Investors Underground — Режим доступа:
<https://www.investorsunderground.com/technical-analysis/>
16. Thuy Nguyen Thi Thu, Vuong Dang Xuan. – Supervised Support Vector Machine in Predicting Foreign Exchange Trading.
17. Hamza, A. – Forecasting Future Prices of Cryptocurrency using Historical Data.
18. Catania L., Grassi S., Ravazzolo F. – Forecasting Cryptocurrencies Financial Time Series.
19. Фомин А.Н. – Криптовалюты в цифровой экономике: финансово-экономические аргументы «за» и «против» Аналитический доклад.
20. Cheah E. T., Fry J. – Speculative bubbles in Bitcoin markets? An empirical investigation into the fundamental value of bitcoin.

Додаток А

Інформаційна система (код програми)

```
from pandas import read_csv, DataFrame
import statsmodels.api as sm
from statsmodels.iolib.table import SimpleTable
from sklearn.metrics import r2_score
import numpy as np
import matplotlib.pyplot as plt

dataset = read_csv('BTC-USD-1-DAY.csv',
encoding='utf-8', index_col='Date', parse_dates=['Date'], dayfirst=True)
print(dataset.head())

Low = dataset.Low
print(Low.head())

Low.plot(figsize=(12,6))

describe = Low.describe()
print(describe)
Low.hist()

V = describe['std']/describe['mean']
print (f'V = {V}')

row = [u'JB', u'p-value', u'skew', u'kurtosis']
jb_test = sm.stats.stattools.jarque_bera(Low)
a = np.vstack([jb_test])
```

```
describe = SimpleTable(a, row)
print (describe)

test = sm.tsa.adfuller(Low)
print ('adf: ', test[0])
print ('p-value: ', test[1])
print ('Critical values: ', test[4])
if test[0] > test[4]['5%']:
    print ('Є одиничні корні, ряд не стаціонарен.')
else:
    print ('Одиничних корнів немає, ряд стаціонарен.')

Low1diff = Low.diff( periods=1 ).dropna()

test = sm.tsa.adfuller(Low1diff)
print ('adf: ', test[0])
print ('p-value: ', test[1])
print ('Critical values: ', test[4])
if test[0] > test[4]['5%']:
    print ('Є одиничні корні, ряд не стаціонарен.')
else:
    print ('Одиничних корнів немає, ряд стаціонарен.')

m = Low1diff.index[ len(Low1diff.index)//2+1 ]
r1 = sm.stats.DescrStatsW(Low1diff[m:])
r2 = sm.stats.DescrStatsW(Low1diff[:m])
print ('p-value: ', sm.stats.CompareMeans(r1,r2).ttest_ind()[1])

Low1diff.plot(figsize=(12,6))
```

```
fig = plt.figure(figsize=(12,8))
ax1 = fig.add_subplot(211)
fig = sm.graphics.tsa.plot_acf(Low1diff.values.squeeze(), lags=25, ax=ax1)
ax2 = fig.add_subplot(212)
fig = sm.graphics.tsa.plot_pacf(Low1diff, lags=25, ax=ax2)

Low = Low.resample('W').last()
train_data = Low[:'2020-11-23']
model = sm.tsa.ARIMA(train_data, order=(2,1,1), freq='W').fit(full_output=False,
disp=0)
print (model.summary())

q_test = sm.tsa.stattools.acf(model.resid, qstat=True)
print (DataFrame({'Q-stat':q_test[1], 'p-value':q_test[2]}))

test_data = Low['2020-11-23:']
pred = model.predict(start=train_data.shape[0],
                    end=(train_data.shape[0]+test_data.shape[0]-1),
                    typ='levels')

Low.plot(figsize=(12,6))
pred.plot(style='r--')
```