**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА**

**ЕКОНОМІЧНИЙ ФАКУЛЬТЕТ**

**Кафедра економічної кібернетики**

****

**КУРСОВА РОБОТА**

на тему: «Прогнозування тенденцій і показників розвитку Біткоїна за допомогою методів згладжування, штучних нейронних мереж та економетричних моделей»

**Виконав**

студент 3 курсу

ОКР «Бакалавр»

спеціальності «Економіка»

ОП «Економічна кібернетика»

Шуба Даниїл Дмитрович

**Науковий керівник**

кандидат фізико-математичних наук, доцент

Кравець Тетяна Вікторівна

Київ – 2023

**ЗМІСТ**

[ВСТУП 4](#_Toc135915915)

[РОЗДІЛ 1. ЗАГАЛЬНІ ВІДОМОСТІ ПРО БІТКОЇН 6](#_Toc135915916)

[1.1. Характеристика складових елементів платіжної системи BTC 6](#_Toc135915917)

[1.2 Аналіз основних теорій щодо перспектив розвитку BTC 8](#_Toc135915918)

[РОЗДІЛ 2. АНАЛІЗ ТА ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ ЗА ДОПОМОГОЮ КЛАСИЧНИХ МАТЕМАТИЧНИХ МЕТОДІВ ТА НЕЙРОННИХ МЕРЕЖ 12](#_Toc135915919)

[2.1 Технологічний аналіз прогнозування 13](#_Toc135915920)

[2.2 Методи згладжування 14](#_Toc135915921)

[2.3 Методи авторегресії 17](#_Toc135915922)

[2.4 Методи машинного навчання прогнозування часових рядів за допомогою нейронних мереж 21](#_Toc135915923)

[2.6 Методи оцінки якості моделей 25](#_Toc135915924)

[РОЗДІЛ 3. МОДЕЛЮВАННЯ КУРСУ BTC ЗА ДОПОМОГОЮ МЕТОДІВ ЗГЛАДЖУВАННЯ, ЕКОНОМЕТРИЧНИХ МОДЕЛЕЙ ТА НЕЙРОННИХ МЕРЕЖ 29](#_Toc135915925)

[3.1 Збір та підготовка даних 29](#_Toc135915926)

[3.2 Реалізація методів згладжування 31](#_Toc135915927)

[3.2 Реалізація економетричних моделей 37](#_Toc135915928)

[3.4 Реалізація рекурентних нейронних мереж 41](#_Toc135915929)

[ВИСНОВКИ 43](#_Toc135915930)

[СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ 45](#_Toc135915931)

[ДОДАТОК А. РЕАЛІЗАЦІЯ МЕТОДІВ ЗГЛАДЖУВАННЯ 48](#_Toc135915932)

[ДОДАТОК Б. РЕАЛІЗАЦІЯ ЕКОНОМЕТРИЧНИХ МЕТОДІВ 53](#_Toc135915933)

[ДОДАТОК В. РЕАЛІЗАЦІЯ РЕКУРЕНТНИХ НЕЙРОННИХ МЕРЕЖ 56](#_Toc135915934)

[ДОДАТОК Г. ПОСИЛАННЯ НА БАЗУ ДАНИХ 60](#_Toc135915935)

ВСТУП

Фінансові системи окремих країн, разом з іншими секторами економіки, продовжують розвиватися та прогресувати в умовах глобалізації, поширення інформаційних технологій та загальної комп'ютеризації. Цей процес сприяє з'явленню нових фінансових інституцій, інструментів та форм взаємодії між людьми. Наприклад, на сучасному етапі з'явилася криптовалюта як аналог традиційних валют, а найбільш відомою з них є біткоїн. Виникнення постійної потреби в моніторингу руху грошових одиниць спричиняє сучасні тенденції грошово-валютних систем на світовому ринку.

Однією з найбільш актуальних тем на сьогоднішній день є можливість прогнозування курсу криптовалют, оскільки це дозволяє відстежувати тенденції розвитку цього сегменту валютних ринків. У сучасних дослідженнях широко використовуються моделі штучних нейронних мереж. Серед вітчизняних науковців, що займалися цією тематикою, можна відзначити Т. В. Кравець, О. І. Черняк, А.В. Ставицький, Н.М. Іванік, Затонацька Т.Г., Кочкіна Н.Ю., Хом’як В.Р., та інші [32].

Зазначені дослідники працюють над розробкою моделей та методів прогнозування курсу криптовалют, які базуються на аналізі великого обсягу даних і використанні штучних нейронних мереж. Ці методи можуть допомогти інвесторам, трейдерам та іншим зацікавленим сторонам у розумінні тенденцій ринку криптовалют та прийнятті обґрунтованих рішень.

Враховуючи швидкі зміни у галузі криптовалют, завжди важливо звертатися до останніх досліджень та робіт вчених, які активно працюють у цій галузі, оскільки нові відкриття та методи можуть постійно з'являтися.

У даній роботі проведено порівняльний аналіз трьох використаних підходів до прогнозування: методів згладжування, економетричних моделей та штучних нейронних мереж, що використовують алгоритми штучного інтелекту, з використанням відповідних бібліотек мови програмування Python.

**Об’єкт дослідження:** курс криптовалюти BTC.

**Предмет дослідження:** моделювання середньої щоденної ціни біткоіна за допомогою методів згладжування, економетричних моделей та штучних нейронних мереж, що використовують алгоритми штучного інтелекту.

**Мета дослідження:** порівняти прогностичні можливості досліджуваних методів прогнозування та визначити, який метод є найкращим для моделювання середньої щоденної ціни біткоіна.

**Основні завдання:**

* Дослідження платіжної системи Біткоін та технології її функціонування.
* Аналіз статистичних даних.
* Дослідження архітектури методів згладжування, економетричних моделей та методів штучних нейронних мереж.
* Визначення найкращого методу для моделювання середньої щоденної ціни біткоіна.

Інформаційною та аналітичною базою дослідження методів згладжування, економетричних моделей, методів нейронних мереж для моделювання середньої щоденної ціни біткоіна є наукові праці вітчизняних та зарубіжних вчених, а також Інтернет-ресурси.

**Структура роботи:** курсова робота складається зі вступу, першого розділу, у якому представлені теоретичні засади щодо платіжної системи біткоін, технологію її функціонування, а також аналіз гіпотез стосовно її подальшого розвитку. У другому розділі наведений математичний опис методів згладжування, економетричних моделей та нейронниз мереж, що використовують алгоритм штучного інтелекту. У третьому розділі наведені результати досліджуваних методів та обрано найкращу модель моделювання середньої ціни біткоіна. Також дана робота містить висновки, перелік використаної літератури та додатки з прогнами кодами на мові Python і посилання на базу даних.

РОЗДІЛ 1. ЗАГАЛЬНІ ВІДОМОСТІ ПРО БІТКОЇН

**1.1. Характеристика складових елементів платіжної системи BTC**

Криптовалюта - це цифровий актив, який дозволяє людям здійснювати транзакції між собою, оскільки вона може використовуватися як засіб обміну. Крім того, власники криптовалют зберігають свої активи в цифровому розподіленому реєстрі у системі децентралізованої мережі, яка реєструє транзакції різних користувачів. Криптовалюти не випускаються центральними органами, тому їхня вартість в основному залежить від масштабу участі на ринку. Однак утворення вартості та коливання криптовалют - це гарячі теми обговорення в літературі [2]. Біткоїн є найпоширенішою криптовалютою серед різних криптовалют на ринку. Його ринкова капіталізація перевищує 600 мільярдів доларів на січень 2021 року [2]. У 2009 році анонімний автор на ім'я Сатоші Накамото опублікував статтю під назвою "Біткоїн: електронна система готівки рівного до рівного", в якій обговорювалася можливість створення нової цифрової валюти (яка не контролюється або випускається центральним органом) шляхом створення децентралізованої мережі [2].

Англія та Фратрік (2018) стверджують, що існують чотири причини, які підштовхнули людей до широкого використання Біткоїна та збільшення інтересу до нього: (а) прогрес у криптографії, (б) порушення індивідуальної приватності через події, які сталися після 2001 року, (в) очікування інфляції через масштабні стимулюючі пакети та накачування резервів на ринки для зниження наслідків глобальної фінансової кризи (ГФК) у 2008 році, та (г) зростання незадоволення банками та іншими великими фінансовими установами через політику, яка призвела до кризи з підприємницькими іпотеками, а потім світовою кризою[2].

Крім того, Бурі та ін. обговорюють використання Біткоїна як захисту від глобальної невизначеності і порівнюють його з традиційними інвестиційними інструментами, такими як золото та сировина. Біткоїн відзначився значним стрибком ціни до рівня 20 000 доларів у 2017 році, однак його вартість почала знижатися нижче 10 000 доларів [13].

Концепція та принципи Біткоїна [2] ґрунтуються на кількох основних принципах, які надають йому унікальні характеристики. Основні з них включають децентралізацію, блокчейн технологію, шифрування та анонімність.

Децентралізація: Біткоїн є децентралізованою криптовалютою, що означає, що він не контролюється центральною установою, такою як уряд чи банк. Замість цього, мережа Біткоїна працює на основі технології peer-to-peer, де всі учасники мережі мають однакові права і можуть взаємодіяти напряму один з одним.

Блокчейн технологія: Біткоїн використовує блокчейн технологію, яка є розподіленою базою даних, що містить всі транзакції, здійснені в мережі. Блокчейн складається з послідовних блоків, які містять дані про транзакції та іншу інформацію. Кожен блок підтверджується мережею майнерів і додається до ланцюжка блоків, що робить систему надзвичайно безпечною та міцною.

Шифрування: Біткоїн використовує сильне шифрування для захисту транзакцій та збереження приватності користувачів. Кожна транзакція підписується приватним ключем власника, який відповідає його публічному ключу. Це забезпечує безпеку і автентифікацію транзакцій.

Анонімність: Біткоїн надає певний рівень анонімності своїм користувачам. Всі транзакції в мережі Біткоїна є публічно доступними, але ідентифікація власників гаманців залишається прихованою. Це дозволяє користувачам здійснювати транзакції без розкриття своєї особистої інформації.

Ці концепції та принципи роблять Біткоїн інноваційною та унікальною криптовалютою, яка змінює спосіб фінансових операцій і взаємодії у мережі. Вони забезпечують безпеку, приватність та контроль користувачів над своїми фінансовими активами, уникнення цензури та широкі можливості використання в різних сферах життя.

**1.2 Аналіз основних теорій щодо перспектив розвитку BTC**

Гіпотеза "Цифрове золото" [2] стверджує, що Біткоїн може стати цифровим еквівалентом золота і прийнятною альтернативою для збереження вартості. Основною ідеєю цієї гіпотези є те, що Біткоїн має властивості, які дозволяють йому функціонувати як безпечне, обмежене і децентралізоване сховище цінностей, схоже на те, як золото використовується у світі фінансів.

Основні аргументи, підтримуючі гіпотезу "Цифрове золото", включають наступне:

Обмеженість кількості: Схоже на золото, Біткоїн має обмежену загальну кількість. Згідно з його протоколом, кількість Біткоїнів, які можуть бути видобуті, обмежена до 21 мільйона монет. Це робить Біткоїн рідкісним активом, що може забезпечити йому потенційне зростання вартості з часом. [2]

Децентралізація: Біткоїн базується на технології блокчейн, що забезпечує децентралізовану природу мережі. Він не контролюється жодним центральним органом або урядом, що дає йому незалежність та невразливість до політичних або економічних факторів. Ця характеристика робить Біткоїн привабливим як захисний актив у ситуаціях економічної турбулентності чи нестабільності валют. [2]

Збереження вартості: За деякими оцінками, Біткоїн володіє властивістю збереження вартості. Значна частина прихильників Біткоїна стверджує, що його обмежена кількість, стійкість до інфляції та розподіленість дозволяють йому бути засобом збереження цінності протягом тривалого періоду.

Глобальна доступність: Біткоїн є цифровим активом, доступним для всіх з підключенням до Інтернету. Це робить його доступним для мільйонів людей, які не мають доступу до традиційних банківських послуг. Біткоїн може стати інструментом фінансової інклюзії для тих, хто має обмежений доступ до традиційної фінансової системи.

Важливо зазначити, що гіпотеза "Цифрове золото" має свої ризики і виклики. Наприклад, недостатня прийнятність Біткоїна в регульованому середовищі, відсутність легкодоступних інструментів для збереження і використання Біткоїна, а також волатильність його вартості можуть утруднити реалізацію цієї гіпотези.

Гіпотеза "Глобальна платіжна система" стверджує, що Біткоїн може стати масштабованою, ефективною та децентралізованою платіжною системою, яка може замінити традиційні методи платежів, такі як банківські перекази, кредитні картки або платіжні системи третіх сторін. Основна ідея полягає в тому, що Біткоїн може спростити та забезпечити безпеку глобальних трансакцій, зменшити витрати та зробити процес платежів швидким та доступним для всіх. Нижче наведено детальний опис основних аспектів цієї гіпотези:[2]

Масштабованість: Однією з головних переваг Біткоїна як платіжної системи є його потенціал для масштабування. Це означає, що Біткоїн може обробляти велику кількість трансакцій одночасно і забезпечувати швидку обробку платежів. Завдяки розвитку технологій, таких як мережеві розширення (наприклад, Lightning Network), можливості платіжних каналів та інші покращення, Біткоїн може досягти високого рівня масштабованості і стати готовим для масового використання.

Ефективність та низькі витрати: Використання Біткоїна для платежів може зменшити витрати, пов'язані з традиційними методами платежів. Традиційні банківські перекази та платіжні системи третіх сторін можуть вимагати високих комісій, особливо при міжнародних трансакціях. Біткоїн може забезпечити низькі комісії або навіть безкоштовні трансакції, залежно від обсягу та пріоритету транзакції.

Децентралізованість: Біткоїн базується на технології блокчейн, що забезпечує децентралізацію платіжної системи. Це означає, що немає центрального органу, який контролює та регулює всі трансакції. Кожен учасник мережі може бути вузлом, який підтримує та перевіряє трансакції. Децентралізованість дозволяє уникнути ризику цензури, блокування платежів або обмежень з боку урядів чи фінансових інституцій.

Швидкість та доступність: Біткоїн може забезпечити швидкі та доступні платежі в будь-який час та в будь-якому місці. Трансакції можуть бути здійснені безпосередньо між користувачами, без необхідності проміжних посередників або авторизації. Доступ до Біткоїну може бути забезпечений через смартфони, комп'ютери або інші пристрої з доступом до Інтернету.

Гіпотеза "Глобальна платіжна система" відображає потенціал Біткоїна стати новим стандартом для глобальних платежів, забезпечуючи швидкість, ефективність, безпеку та доступність для всіх користувачів незалежно від їх місця розташування чи фінансового статусу.

Гіпотеза "міжнародні перекази та фінансова інклюзія" відноситься до ролі криптовалют, зокрема Біткоїна, у поліпшенні міжнародних переказів та забезпеченні фінансової інклюзії для населення.

За традиційної фінансової системи, міжнародні перекази можуть бути складними, дорогими та часоємкими процесами. Банки, платіжні системи та посередники здійснюють трансакції, що супроводжуються високими комісіями, затримками та обмеженими годинами роботи. Крім того, існують мільйони людей по всьому світу, які не мають доступу до банківських послуг та основних фінансових інструментів. Ця проблема, відома як фінансова інклюзія, ставить під загрозу економічний розвиток та добробут цих людей.

Гіпотеза "міжнародні перекази та фінансова інклюзія" висуває тезу, що криптовалюти, зокрема Біткоїн, можуть відігравати ключову роль у вирішенні цих проблем. Завдяки своїй децентралізованій природі та використанню блокчейн технології, криптовалюти дозволяють людям здійснювати безпосередні перекази один одному, обходячи посередників і скорочуючи комісії та час очікування. Це може полегшити та здешевити процес міжнародних переказів, забезпечуючи швидкі, доступні та ефективні трансакції.

Крім того, криптовалюти можуть сприяти фінансовій інклюзії, оскільки вони надають можливість надсилаті та отримувати гроші без необхідності банківського рахунку. Люди, які не мають доступу до традиційних банківських послуг, можуть використовувати криптовалюти, щоб зберігати свої активи, здійснювати перекази та здійснювати платежі.

Гіпотеза "міжнародні перекази та фінансова інклюзія" визначає потенціал Біткоїна та інших криптовалют у покращенні фінансової доступності та забезпеченні ширшої фінансової інклюзії на глобальному рівні. Вона вказує на можливість зменшення вартості та складності міжнародних переказів, що особливо важливо для економічно вразливих груп населення. Крім того, вона пропонує, що використання криптовалют може допомогти розв'язати проблему фінансової інклюзії шляхом надання доступу до фінансових послуг без необхідності відкривати банківські рахунки.

Важливо зауважити, що гіпотеза "міжнародні перекази та фінансова інклюзія" залишається предметом досліджень та обговорень. Хоча криптовалюти, зокрема Біткоїн, мають потенціал поліпшити міжнародні перекази та забезпечити фінансову інклюзію, існують також виклики та ризики, пов'язані з цими технологіями, такі як волатильність цін, безпека та регуляторні аспекти.

РОЗДІЛ 2. АНАЛІЗ ТА ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ ЗА ДОПОМОГОЮ КЛАСИЧНИХ МАТЕМАТИЧНИХ МЕТОДІВ ТА НЕЙРОННИХ МЕРЕЖ

Прогнозування часових рядів - це процес аналізу часових рядів даних за допомогою статистики та моделювання для здійснення передбачень та підтримки стратегічного прийняття рішень. Це не завжди точний прогноз, ймовірность точності прогнозів може широко змінюватися, особливо коли маємо справу з постійно змінюваними змінними у часових рядах та факторами, над якими ми не маємо контролю. Проте прогнозування дає уявлення про те, які результати більш ймовірні або менш ймовірні, ніж інші можливі результати. Зазвичай, чим більше у нас повна інформація, тим точніші можуть бути прогнози. Хоча терміни "прогнозування" та "прогноз" загалом мають однакове значення, є помітна відмінність. У деяких галузях прогнозування може стосуватися даних на конкретний майбутній момент часу, тоді як прогнозування відноситься до майбутніх даних загалом. Прогнозування часових рядів часто використовується в поєднанні з аналізом часових рядів. Аналіз часових рядів передбачає розробку моделей для отримання розуміння даних та виявлення причин. Прогнозування потім переходить до наступного кроку - використання цих знань і передбачуваних екстраполяцій того, що може трапитися в майбутньому. [11]

* 1. **Технологічний аналіз прогнозування**

Технологічний аналіз прогнозування - це процес використання інформаційних технологій і аналітичних методів для прогнозування майбутніх подій, трендів і розвитку в галузі технологій. Цей аналіз спрямований на оцінку поточного стану технологій, виявлення нових тенденцій, ідентифікацію можливих перспективних напрямків розвитку і передбачення майбутнього використання технологій.

Технологічний аналіз прогнозування може включати в себе дослідження ринкових тенденцій, використання моделей та алгоритмів для аналізу даних, виявлення патернів та залежностей, а також оцінку впливу нових технологій на бізнес-середовище і суспільство.

Цей аналіз може бути застосований в різних галузях, включаючи інформаційні технології, медіа, енергетику, медицину, транспорт, фінанси та інші. Він допомагає організаціям і рішенням приймати обґрунтовані рішення щодо інвестицій в технології, розробки нових продуктів і послуг, стратегічного планування та інноваційного розвитку.

Технологічний аналіз прогнозування може використовувати різні методи, такі як експертні оцінки, сценарійний аналіз, моделювання та симуляція, дослідження трендів та статистичний аналіз даних. Він допомагає виявляти можливості для інновацій, адаптації до змін технологічного середовища і розробки стратегій, спрямованих на досягнення конкурентної переваги. [11]

* 1. **Методи згладжування**

Згладжування є методом прогнозування фінансових ринків, який має велике значення та широке застосування. Зазвичай використовуються різні методи згладжування, зокрема ковзні середні. Вони допомагають знизити вплив випадкових факторів у часовому ряді. В даному розділі будуть представлені 4 звичайних методів згладжування, які використовуються для прогнозування фінансових часових рядів. Основне припущення цих методів полягає в тому, що коливання попередніх значень є випадковими відхиленнями від деякої плавної кривої, яку можна використовувати для екстраполяції та створення прогнозу. [4]

Термін "ковзне середнє" використовується для позначення набору значень, які постійно змінюються в часі. Ковзне середнє відображає загальну тенденцію зміни цін і приглушує незначні коливання. Оскільки ковзне середнє представляє середнє значення цін в минулому, графіки ковзних середніх зазвичай "відстають" від поточних змін у часовому ряді. На ринках з вираженим трендом ковзні середні дають добрі результати, але на ринках без вираженої тенденції цей метод може призводити до значних похибок. Формально цей метод описується так:

Де N – число попередніх моментів, що було взято до уваги при побудові прогнозу,

– реальні значення показника в момент часу . [5]

Метод зваженого ковзного середнього розширює концепцію простого ковзного середнього шляхом призначення вагових коефіцієнтів для кожного компонента часового ряду. Це означає, що значення кожного компонента усереднюється залежно від його ваги. [5]:

,

де N – число попередніх моментів, що було взято до уваги при побудові прогнозу,

– реальні значення показника в момент часу ,

– ваговий коефіцієнт для i-го компоненту ряду.

Даний метод краще використовувати, оскільки в результаті отримана оцінка тренду є набагато гладшою. Кожне вхідне спостереження різко не змінює своє середнє значення за допомогою вагових коефіцієнтів.

Один з видів змінного середнього, що використовується, - експоненціальне ковзне середнє (EMA). Цей метод можна розглядати як зважене ковзне середнє, де кожен компонент часового ряду має свою вагу, яка зменшується залежно від віддаленості від поточного торгового періоду. Таке розподілення дозволяє зосередитися на поточних цінах і не пропустити важливі торговельні сигнали. Експоненціальне ковзне середнє зменшує затримку, приділяючи більше уваги останнім цінам порівняно з віддаленими цінами. Це дозволяє швидше реагувати на поточні зміни цін, порівняно з простим ковзним середнім. Вага, яку отримує остання ціна, залежить від періоду експоненціального ковзного середнього. З коротшим періодом EMA, більша вага надається останній ціні. Наприклад, 10-періодне EMA надає вазі 18,18% останній ціні, тоді як 20-періодне EMA надає вазі 9,25%. Однак розрахунок EMA виявляється складнішим, ніж простий ковзний середній.

Математична формула для розрахунку EMA є рекурсивною і при використанні коефіцієнта згладжування, що дорівнює n, має наступний вигляд [3]:

|  |
| --- |
|  |

де - значення експоненціального змінного середнього в точці t (останнє значення, в разі тимчасового ряду),

- значення експоненціального змінного середнього в точці ( попереднє значення в разі тимчасового ряду),

- значення вихідної функції в момент часу (останнє значення, в разі тимчасового ряду),

α (згладжуюча константа від англ. smoothing constant) - коефіцієнт що характеризує швидкість зменшення ваг.

Перше значення експоненціального змінного середнього, зазвичай приймається рівним першому значенню вихідної функції:

|  |  |  |
| --- | --- | --- |
|  |  |  |

Коефіцієнт , може бути обраний довільним чином, в межах від 0 до 1, наприклад, виражений через величину вікна усереднення:

|  |  |  |
| --- | --- | --- |
|  |  |  |

У звичайному експоненційному ковзному середньому згладжування піддаються значення вихідної функції, однак, згладжування можуть піддаватися і значення результуючої функції. Тому деякі автори визначають поняття експоненціальні ковзне середнє довільного порядку, які визначаються за формулою [3]:

|  |  |  |
| --- | --- | --- |
|  |  |  |

де - значення експоненціального змінного середнього n-го порядку в точці t (останнє значення, в разі тимчасового ряду), - значення експоненціального змінного середнього -го порядку в точці (попереднє значення в разі часового ряду), - значення експоненціального змінного середнього -го порядку в точці t (останнє значення, в разі тимчасового ряду), α - згладжує константа.

В даній роботі досліджувалося подвійне та потрійне (метод Холта-Вінтереса) експоненціальне ковзне середнє.

Експоненціально зважені ковзаючі середні другого і третього порядку позначають іноді як, відповідно (від англ. Double exponential moving average - подвійне (дворазове) експоненціальне ковзне середнє) і (від англ. triple exponential moving average - потрійне (триразове)) експоненціальне ковзне середнє:

|  |  |  |
| --- | --- | --- |
|  |  |  |

|  |  |  |
| --- | --- | --- |
|  |  |  |

* 1. **Методи авторегресії**

Методи авторегресії є статистичними моделями, які використовуються для аналізу та прогнозування часових рядів. Основна ідея за методами авторегресії полягає в тому, що значення змінної у поточний момент часу залежить від попередніх значень цієї змінної. [24]

Одна з основних моделей авторегресії - авторегресія AR(p), де "p" вказує на кількість попередніх значень, що використовуються для прогнозу. У моделі AR(p) кожне значення в часовому ряді представляється як лінійна комбінація попередніх значень цього ряду.

Позначення AR (p) відноситься до авторегресійної моделі порядку p. Модель AR (p) записують як: [21]

де – параметри авторегресії, с – постійна, а випадкова величина – білий шум. Деякі обмеження необхідні на значеннях параметрів, так що модель залишається стаціонарною. Наприклад, процеси в моделі AR(1) з не є стаціонарними. У статистиці та обробці сигналів модель авторегресії (AR) є поданням типу випадкового процесу; як таке, воно описує певні зміни в часі, процеси в природі, економіці тощо.

Додатково, існує розширена модель авторегресії ARMA(p, q), яка комбінує авторегресію з рухомим середнім. В цій моделі крім попередніх значень часового ряду також враховується вплив попередніх значень помилок моделі.

Модель - це просто злиття між моделями і :

Моделі намагаються пояснити ефекти імпульсу та середнього реверсії, які часто спостерігаються в стохастичних процесах.

Моделі намагаються захопити ударні ефекти, що спостерігаються в умовах білого шуму. Ці шокові ефекти можна розглядати як несподівані події, що впливають на процес спостереження[3].

Модель намагається охопити обидва ці аспекти при моделюванні часових рядів. [25]

Часто цю модель називають моделлю , де:

* - порядок полінома авторегресії,
* - порядок багаточлена ковзної середньої.

де - параметри моделі авторегресії,

- параметри моделі ковзної середньої.

- константа,

- терміни помилок (білий шум).

Алгоритм побудови моделі складається з наступних кроків:

1. Визначаємо порядок авторегресійної складової
2. Обчислюємо та
3. По значенням та визначаємо
4. Обчислюємо
5. Визначаємо порядок ковзного середнього –
6. Знаходимо та залишків моделі
7. По отриманним значенням знаходимо порядок

Більш складним варіантом є модель авторегресії з інтегрованим рухом ARIMA(p, d, q), де "d" вказує на кількість похідних, які потрібно взяти, щоб зробити ряд стаціонарним. Ця модель включає в себе як авторегресійну, так і рухому середню складові, а також може враховувати інтегровану складову для корекції тренду.

Різниця між -моделлю і . Ці дві моделі мають багато спільного. Насправді компоненти і MA ідентичні, об'єднуючи загальну авторегресивну модель і загальну ковзну середню модель . робить прогнози з використанням попередніх значень залежної змінної. MA (q) робить прогнози з використанням середніх і попередніх помилок.

Модель є стаціонарною моделлю; Якщо ваша модель не є стаціонарною, то можна досягти стаціонарності, взявши ряд відмінностей. в моделі означає інтегрований; Це міра того, як багато несезонних відмінностей потрібні для досягнення стаціонарності. Якщо ніякої диференціації не бере участь в моделі, то вона стає просто .

В основі методології, запропонованої Боксом і Дженкінсом, лежить ідея про те, що поведінка змінної в теперішньому і майбутньому часі з достатнім ступенем точності пояснюється її поведінкою в минулому.

Модель авторегресії з ковзним середнім є методологічний апарат для ідентифікації, оцінки та діагностики моделей стаціонарних часових рядів.

Подання про послідовні вигляді моделі змінного середнього і авторегресії в комбінації з послідовним взяттям різниць дає можливість моделювати нестаціонарні процеси, зводячи їх до стаціонарних. Модель авторегресії з проінтегрувати змінного середнім для нестаціонарних часових рядів отримала назву .

Короткий запис моделі має вигляд [25]:

,

Де - поліноми ступенів , - лаговий оператор, - порядок взяття послідовної різниці.

Завдання моделювання процесу за допомогою зводиться до знаходження оптимального значення параметрів , причому ці значення повинні бути мінімально можливими щоб уникнути надмірності моделі.

Алгоритм побудови моделі складається з наступних кроків:

1. Ідентифікація моделі - процес вибору моделі, яка найкращим чином відповідає досліджуваного тимчасового ряду.
2. Оцінювання моделі - використання різних методів для отримання оцінок параметрів, включених в модель.
3. Тестування моделі, зокрема, перевірка залишків на нормальність і незалежність розподілу.
4. Виконання прогнозу за допомогою отриманої моделі.

Методи авторегресії дозволяють аналізувати залежності та прогнозувати майбутні значення часових рядів на основі їх попередніх значень. Вони знаходять широке застосування у фінансовому аналізі, економічному прогнозуванні, прогнозуванні попиту та інших галузях, де важливо виявити та спрогнозувати патерни та тенденції в часових рядах.

* 1. **Методи машинного навчання прогнозування часових рядів за допомогою нейронних мереж**

Штучні нейронні мережі (Artificial Neural Networks - ANN) є корисним інструментом для прогнозування часових рядів, оскільки вони можуть ефективно вирішувати складні проблеми прогнозування. [18]

Однією з важливих особливостей ANN у контексті прогнозування часових рядів є їх здатність до нелінійного моделювання без потреби в будь-яких припущеннях про статистичний розподіл часового ряду. Кожна модель нейронної мережі адаптивно формується на основі доступних даних. Таким чином, штучні нейронні мережі керуються даними і є самоадаптивними за своєю природою.

Структура штучної нейронної мережі ґрунтується на з'єднаних вузлах, які називаються нейронами, подібно до біологічних нейронів у людському мозку.

Модель штучної нейронної мережі (ANN) може включати три типи шарів: вхідний шар, прихований шар та вихідний шар, які з'єднані без циклів. Деякі моделі можуть мати більше одного прихованого шару. Простою прикладом є тришарова архітектура ANN, яка може бути зображена схематично наступним чином.

Изображение выглядит как диаграмма, круг, линия, зарисовка

Автоматически созданное описание

Рис 1. Схематичне відображення штучної нейронної мережі

*Джерело:* [*https://www.datasciencecentral*](https://www.datasciencecentral.com/the-artificial-neural-networks-handbook-part-1/)

Рекурентна нейронна мережа (RNN) - це клас вдосконаленої штучної нейронної мережі (ANN), містять зворотні зв'язки і дозволяють зберігати інформацію (див. рис.2).

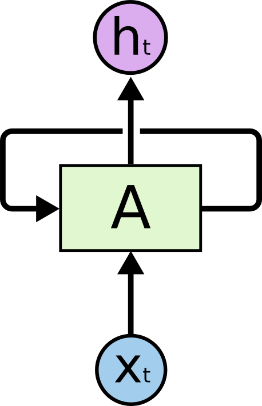


Рис. 2. Рекурентні нейронні мережі містять зворотні зв'язки

*Джерело:* [*https://www.datasciencecentral*](https://www.datasciencecentral.com/the-artificial-neural-networks-handbook-part-1/)

На схемі вище фрагмент нейронної мережі A приймає вхідний значення і повертає значення . Наявність зворотного зв'язку дозволяє передавати інформацію від одного кроку мережі до іншого.

Рекурентним нейронним мережам притаманний загадковий характер через наявність зворотних зв'язків. Проте, при уважному розгляді, вони не настільки суттєво відрізняються від звичайних нейронних мереж. Рекурентну мережу можна розглядати як кілька копій однієї мережі, кожна з яких передає інформацію наступній копії.

Нижче на рис. 3 показано, як зворотний зв'язок може бути розгорнутий для більшої ясності:

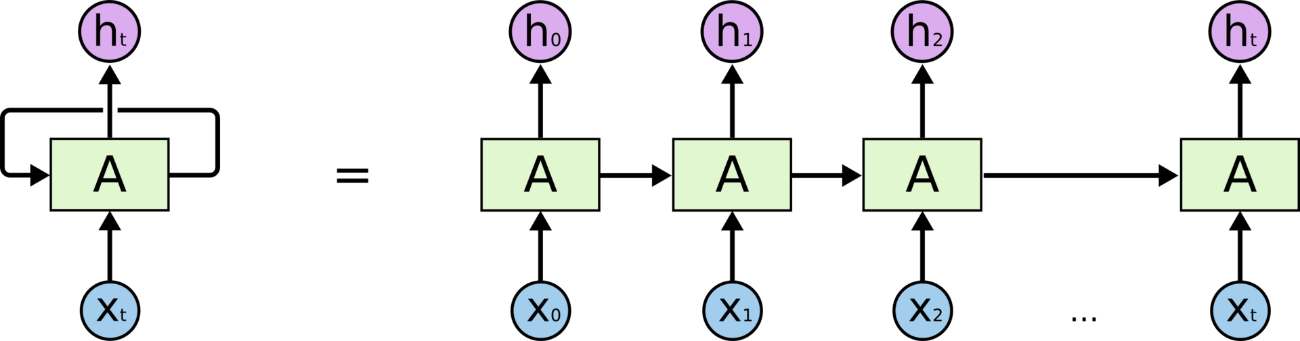


Рис.3. Рекурентна нейронна мережа в розгортці

*Джерело:* [*https://www.datasciencecentral*](https://www.datasciencecentral.com/the-artificial-neural-networks-handbook-part-1/)

Той факт, що рекурентні нейронні мережі (RNN) нагадують ланцюжок, свідчить про їх тісний зв'язок з послідовностями і списками. RNN є найбільш природною архітектурою нейронних мереж для роботи з такого типу даними і, звичайно ж, їх часто використовують для подібних завдань. Протягом останніх кількох років RNN з успіхом застосовувалися в різноманітних завданнях, таких як розпізнавання мови, мовне моделювання, переклад, розпізнавання зображень та інші. Цей список можна продовжувати [20].

Велика роль в цих успіхах належить LSTM, яка є особливою модифікацією рекурентної нейронної мережі і значно перевершує стандартну версію на багатьох завданнях. Майже всі вражаючі результати, досягнуті за допомогою RNN, були отримані завдяки використанню LSTM [20].

Одним з найсуттєвіших факторів для побудови нейронних мереж є визначення даних, на яких вони будуть навчатись. Основна мета більшості цих мереж полягає в точному прогнозуванні цін часового ряду, в нашому випадку – курсу криптовалюти, щоб приймати рішення про купівлю або продаж на основі попередніх ринкових даних. Одним з викликів полягає в тому, які індикатори та вхідні дані використовувати, а також в зборі достатньої кількості навчальних даних для ефективного тренування системи.

Вхідні дані можуть включати інформацію про обсяги, ціни або щоденні зміни цін, а також технічні показники (наприклад, рухомі середні, трендові лінії і т.д.) або фундаментальні показники (наприклад, внутрішня вартість акцій, економічні показники тощо). Один корисний приклад такої нейронної мережі - JSE-system, яка моделювала поведінку Йоганнесбурзької фондової біржі. Ця система використовувала 63 показники з різних категорій для отримання загального уявлення про ринкове середовище [20].

Проте аналіз показав, що не всі 63 вхідні показники є суттєвими, тому під час тренування автори цієї моделі відкинули 20 змінних, зменшили кількість прихованих вузлів з 21 до 14, що мало незначний вплив на продуктивність системи, а також зменшили час навчання.

Інші системи нейронних мереж можуть використовувати інші типи вхідних даних. Наприклад, простіші системи можуть використовувати лише минулі ціни акцій, тоді як деякі використовують ключові фрази, використані у звітах президента перед акціонерами. Дослідники також комбінують цифрові дані та інтелектуальний аналіз тексту новин для прогнозування вартості акцій.

Перший крок у тренуванні мережі - визначення відповідних вхідних даних. Другий крок - правильне подання цих даних, щоб мережа могла ефективно навчатись, уникаючи повторного тренування. У даній роботі використовувались попередні значення вартості акцій для забезпечення об'єктивної порівняльної характеристики класичних методів та методів машинного навчання.

**2.6 Методи оцінки якості моделей**

В цілому, для оцінки проблеми прогнозування цін використовуються різні метрики, як статистичні, так і нестатистичні.

Нестатистичні показники результативності включають річний прибуток, який отримується від конкретної моделі, а також коефіцієнт попадання або кількість правильних передбачень моделі, щодо руху ринку (зростання або падіння цін).

Для аналізу використовуються різноманітні статистичні показники, такі як середня абсолютна похибка (MAE), середньоквадратичне відхилення (RMSE), середня квадратична помилка передбачення (MSPE), коефіцієнт кореляції, прямокутний кореляційний коефіцієнт та мінімальна остаточна помилка передбачення Акайке.

В даній роботі було використано 5 статистичних метрик, таких, як середня абсолютна похибка (MAE), середньоквадратична похибка (MSE), середньоквадратична логарифмічна похибка (MSLE), відсоткова середня абсолютна похибка (MAPE) та коефіцієнт детермінації (R2 Score). [30]

MAE (Mean Absolute Error) - це одна з поширених метрик, яка використовується в задачах прогнозування для оцінки точності моделі. Вона вимірює середню абсолютну різницю між прогнозованими значеннями і фактичними значеннями. [30]

, де

MAE – середня абсолютна похибка,

N – кількість прикладів у наборі даних,

– фактичне значення

– прогнозоване значення

MAE вимірюється в одиницях вихідної змінної і дає загальну інформацію про середню величину похибки моделі. Чим нижче значення MAE, тим краще модель передбачає дані. Він особливо корисний, коли важлива точність прогнозування по всіх спостереженнях, без зваження на їхні значення.

MAE є простою в інтерпретації метрикою і не залежить від складності розподілу даних. Однак, він не враховує напрямок або розподіл похибки, тому в деяких випадках може бути вигідно використовувати інші метрики, такі як MSE (Mean Squared Error) або R2 (коефіцієнт детермінації), для повнішої оцінки моделі.

MSE (Mean Squared Error) - це метрика, яка використовується для вимірювання середнього квадратичного відхилення між прогнозованими значеннями і фактичними значеннями в задачах прогнозування. [15]

,

MSE – середньоквадратична похибка,

N – кількість прикладів у наборі даних,

– фактичне значення

– прогнозоване значення

MSE вимірюється в квадратних одиницях вихідної змінної. Ця метрика покаже середнє значення квадратів відхилень між прогнозованими і фактичними значеннями. Чим менше значення MSE, тим краще модель передбачає дані.

MSE часто використовується в багатьох алгоритмах оптимізації і навчання моделей, оскільки вона диференційована та неперервна функція. Вона приділяє більше уваги великим відхиленням між прогнозованими і фактичними значеннями, оскільки вони підносяться до квадрату.

Однак, якщо є викиди або значні похибки в даних, MSE може бути дуже чутливою до цих аномалій. Також, вона не дає прямої інтерпретації, як MAE. Тому в деяких випадках може бути корисною використовувати інші метрики, такі як MAE, R2 (коефіцієнт детермінації) або RMSE (квадратний корінь з середньоквадратичної помилки), для отримання більш повної інформації про точність моделі.

Середньоквадратична логарифмічна похибка (MSLE - Mean Squared Logarithmic Error) - це метрика, яка використовується для оцінки точності моделі в задачах прогнозування, особливо коли мають місце великі різниці між прогнозованими і фактичними значеннями.

MSLE вимірюється як середнє квадратичне відхилення між логарифмами прогнозованих і фактичних значень. Вона часто використовується, коли величина змінної має широкий діапазон і важко передбачити абсолютну похибку.

,

MSLE – середньоквадратична логарифмічна похибка,

N – кількість прикладів у наборі даних,

– фактичне значення

– прогнозоване значення

MSLE дозволяє зменшити вплив великих похибок, оскільки розрахунки проводяться на логарифмах значень, а не на самих значеннях. Вона особливо корисна в задачах, де важлива точність прогнозування змінної, яка має експоненційну природу або широкий діапазон значень, таких як приріст цін, кількість переглядів або доля в ринку. [15]

Аналогічно до інших метрик, менше значення MSLE вказує на більш точну модель. Проте, MSLE не має прямої інтерпретації, як MAE або MSE. Часто разом з MSLE використовуються інші метрики, такі як MAE або RMSLE (Root Mean Squared Logarithmic Error), для отримання більш повної оцінки точності моделі.

MAPE (Mean Absolute Percentage Error) - це метрика, яка використовується для вимірювання відсоткової похибки між прогнозованими значеннями і фактичними значеннями в задачах прогнозування. [15]

, де

MAPE – відсоткова середня абсолютна похибка,

N – кількість прикладів у наборі даних,

– фактичне значення

– прогнозоване значення

MAPE вимірюється у відсотках і показує середню відсоткову різницю між прогнозованими і фактичними значення.

Коефіцієнт детермінації, також відомий як R-squared (R2), є статистичною метрикою, що використовується для оцінки якості підгонки моделі до даних у задачах прогнозування.

R2 відображає пропорцію дисперсії відгуку, яка може бути пояснена моделлю, у відношенні до загальної дисперсії відгуку. Він виражається як значення від 0 до 1, де 0 означає, що модель не пояснює жодної дисперсії відгуку, а 1 означає, що модель повністю пояснює дисперсію відгуку. [15]

В контексті прогнозування, R2 оцінює, наскільки добре модель підганяється до фактичних даних. Він показує відсоток варіації відгуку, який може бути пояснений моделлю прогнозу.

R2 обчислюється шляхом порівняння суми квадратів відхилень прогнозованих значень відгуку від середнього значення відгуку з загальною сумою квадратів відхилень фактичних значень відгуку від середнього значення відгуку. Чим ближче значення R2 до 1, тим краще модель пояснює дисперсію відгуку і володіє кращою прогностичною здатністю.

Важливо пам'ятати, що R2 метрика має свої обмеження і не є безумовним показником якості моделі. Вона не враховує інші аспекти, такі як систематичні зміщення або незалежність залишків моделі. Тому, окрім R2, рекомендується також використовувати інші метрики для повної оцінки моделі прогнозування.

РОЗДІЛ 3. МОДЕЛЮВАННЯ КУРСУ BTC ЗА ДОПОМОГОЮ МЕТОДІВ ЗГЛАДЖУВАННЯ, ЕКОНОМЕТРИЧНИХ МОДЕЛЕЙ ТА НЕЙРОННИХ МЕРЕЖ

* 1. **Збір та підготовка даних**

Для реалізації методів прогнозування було взято дані про курс BTC/USD з офіційного сайту, який містить дані про фондові біржі та криптовалюту. Дані було взято за 5 років, починаючи з 1 квітня 2015 року по 1 квітня 2020 року.

База даних містить щоденну інформацію про курс Біткоїна на початку та в кінці дня, також максимальний та мінімальний курс за день, також кількість куплених монет. Зразок бази даних зображений у табл. 1

Таблиця 1

Зразок бази даних, що була використана для прогнозування курсу BTC

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Date | Open | High | Low | Close | Adj Close | Volume |
| 2015-04-01 | 244.223007 | 247.541000 | 241.160004 | 247.272003 | 247.272003 | 22877200 |
| 2015-04-02 | 247.089005 | 254.460999 | 245.416000 | 253.005005 | 253.005005 | 26272600 |
| 2015-04-03 | 253.074005 | 256.042999 | 251.878998 | 254.322006 | 254.322006 | 23146600 |
| 2015-04-04 | 254.291000 | 255.257996 | 251.100006 | 253.697006 | 253.697006 | 12493500 |
| 2015-04-05 | 253.761002 | 260.674988 | 251.942001 | 260.597992 | 260.597992 | 19649200 |
| 2015-04-06 | 260.721008 | 261.798004 | 254.574997 | 255.492004 | 255.492004 | 20034200 |
| 2015-04-07 | 255.274002 | 255.804993 | 252.205002 | 253.179993 | 253.179993 | 18467400 |
| 2015-04-08 | 253.063995 | 253.847000 | 244.214996 | 245.022003 | 245.022003 | 30086400 |
| 2015-04-09 | 244.751007 | 246.117996 | 239.399994 | 243.675995 | 243.675995 | 21643500 |
| 2015-04-10 | 243.694000 | 243.694000 | 232.770996 | 236.072006 | 236.072006 | 28882000 |
| 2015-04-11 | 236.016006 | 239.537003 | 234.175003 | 236.552002 | 236.552002 | 16365200 |

*Джерело:* [*https://www.investing.com/*](https://www.investing.com/)

Спочатку за допомогою відповідних модулей Python було «перевернуто» дані за спаданням дати. Потім було видалено стовпчики з непотрібними в аналізі даними (volume, adj close). Після цього було обраховано середнє значення за кожен день за відповідною формулою:

Середнє значення було додано до бази даних і на основі нього вже будувалися подальші моделі.

Для оптимізації роботи програми було нормалізувано середні значення. Потім було розбито значення середніх цін на два ряди: тренувальні та тестові у відношенні 70% до 30%. На рис. 4 можемо спостерігати графік цін протягом періодів. Можемо спостерігати наявність тренду, наявність сезонності (починаючи з середени 2017 року), а також циклічність.

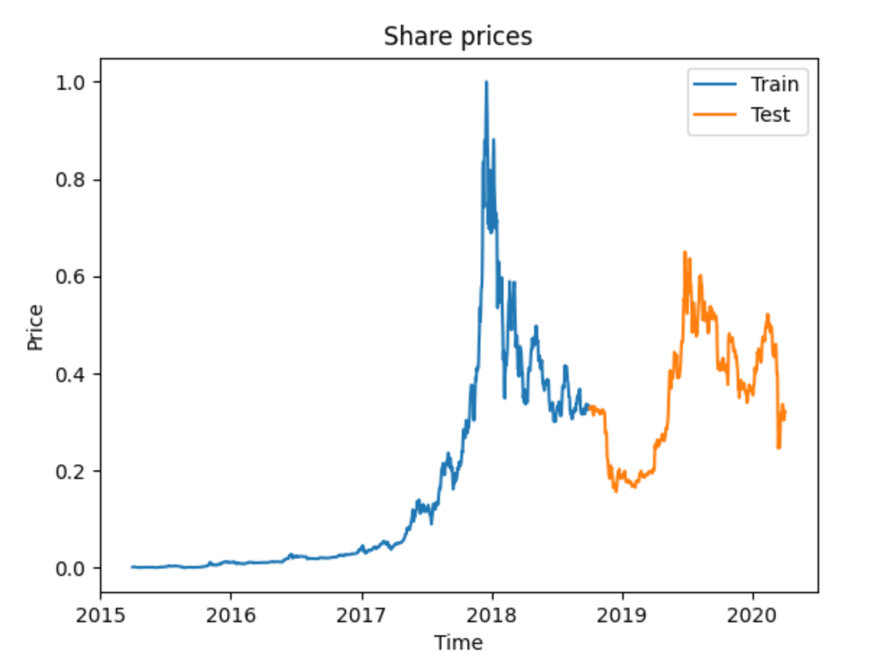


Рис. 4. Графік курсу BTC протягом 5 років на основі [27]

*Джерело: розрахунки автора*

**3.2 Реалізація методів згладжування**

В даній роботі було розглянуто 4 методи згладжування: метод простого ковзного середнього, метод експоненціального ковзного середнього, метод подвійного експоненціального ковзного середнього, а також метод Холта-Вінтереса (потрійне експоненціальне згладжування).

Метод простого ковзного середнього було реалізовано для декількох значень вікон (кроків назад). Очевидним є те, що при зменшенні вікна ми будемо мати більш точні результати, це було перевірено відповідним чином. Значення вікон були підібрані за допомогою періодичності у часовому ряді (середніх цін). Якщо ми поглянемо на рис. 4 (тренувальні дані), то можемо спостерігати загальні закономірності такі як зріст/падіння. Зауважимо, що тиждень вважається 5 днів лише на акційних біржах. В нашому випадку беремо тиждень 7 днів, оскільки крипто-біржі працюють весь тиждень. Однак варто зауважити, що більшість трейдерів не працюють у вихідні дні, а це в свою чергу призводить до нестабільності росту/падіння курсу. Тож було вирішено перевірити для наступних вікон: [30, 10, 5] та [42, 14, 7]. Маємо такі результати.

Изображение выглядит как График, снимок экрана, линия, диаграмма

Автоматически созданное описание

Рис. 5. Реалізація простого методу ковзного середнього з вікнами = [42, 14, 7]

Изображение выглядит как График, линия, диаграмма, снимок экрана

Автоматически созданное описание

Рис. 6. Реалізація методу простого ковзного середнього з вікнами = [30, 10, 5]

*Джерело: розрахунки автора*

Для того, щоб обрати кращу модель було обраховано чотири види похибок похибок, які відображені у табл. 2

Таблиця 2

Обрахунок похибок для двох моделей простого ковзного середнього

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Значення вікна | MSE | MAE | R2 | MSLE | MAPE |
| N = 42 | 0.003202 | 0.041659 | 0.803888 | 0.001672 | 0.123110 |
| N = 14 | 0.000999 | 0.021017 | 0.938810 | 0.000508 | 0.058175 |
| N = 7 | 0.000429 | 0.013479 | 0.973690 | 0.000215 | 0.037473 |
| N = 30 | 0.002311 | 0.034329 | 0.858412 | 0.001192 | 0.099962 |
| N = 10 | 0.000688 | 0.016976 | 0.957862 | 0.000348 | 0.047596 |
| N = 5 | 0.000258 | 0.010402 | 0.984210 | 0.000129 | 0.028627 |

*Джерело: розрахунки автора*

Зауважимо, що оцінка R2 чим ближча до 1, тим вона є кращою. Отже найркращою є модель зі значенням вікна = 5.

Для реалізацію методу екпоненціального ковзного середнього необхідно обрати параметр, в якому задається рівень згладжування, який названий ступенем зменшення зважування та лежить в межах від 0 до 1.

Було вирішено реалізувати дану модель для 9 ступенів зважування, а потім за результатами обчислених похибок обрати найкращий ступінь зважування (коефіцієнт α). Отже, результати відображені на рис. 7.

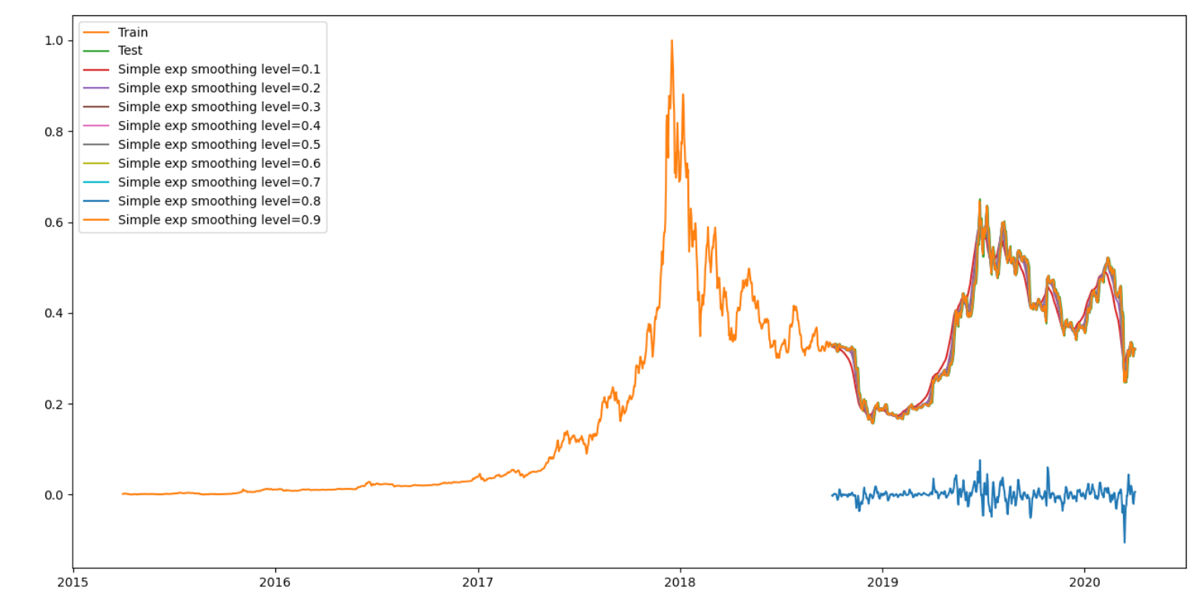


Рис. 7. Реалізація методів простого екпоненціального ковзного середнього

*Джерело: розрахунки автора*

Для вибору найкращого коефіцієнта згладжування було використано такий підхід, як і в попередньому методі. Було обраховано 4 види оцінки якості прогнозу та зроблено певні висновки щодо коефіцієнта α. Результати зображені у табл. 3.

Таблиця 3

Обрахунок показників якості для моделі простого експоненціального ковзного середнього

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| α | MSE | MAE | R2 | MSLE | MAPE |
| 0.2 | 0.000437 | 0.01755 | 0.96055 | 0.000323 | 0.048003 |
| 0.3 | 0.000437 | 0.01391 | 0.97326 | 0.000218 | 0.037803 |
| 0.9 | 0.000165 | 0.00788 | 0.98988 | 0.000018 | 0.021405 |

*Джерело: розрахунки автора*

Варто зауважити, що похибки були обраховані для кожного рівня зважування. У табл. 3 наведений приклад за якою логікою обирався даний коефіцієнт. Як бачимо чим вищий рівень згладжування, тим точнішим є прогноз.

Метод подвійного експоненціального згладжування ще передбачає параметр β, який відповідальний за згладжування тренду. Комбінація пари α та β дає ще кращі результати прогнозування. β підбирався так само. На рис. 8 бачимо результати реалізації даного методу.

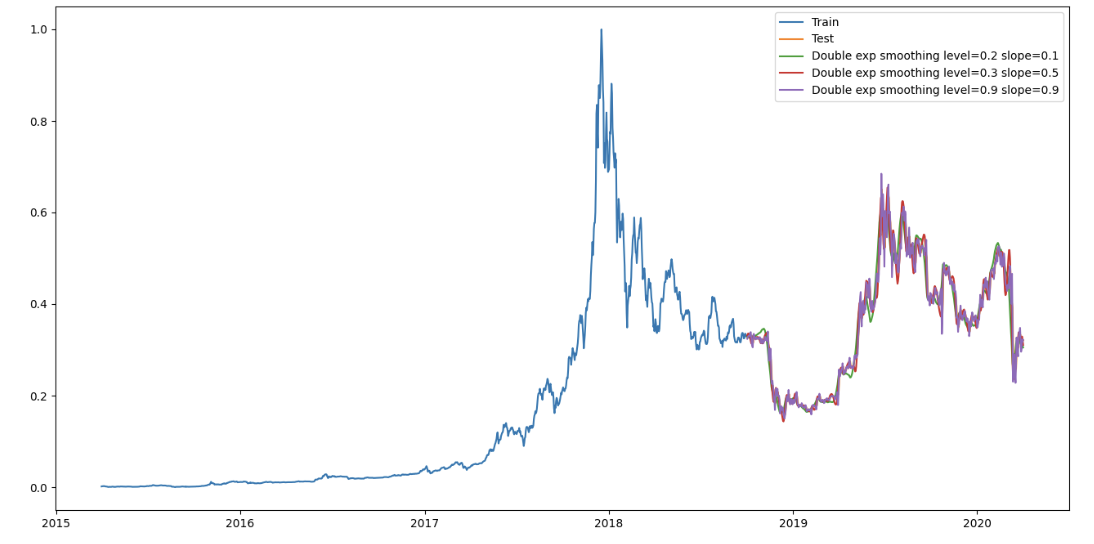


Рис. 8. Реалізація методу подвійного екпоненціального згладжування

*Джерело: розрахунки автора*

Таблиця 4

Обрахунок показників якості для моделі подвійного експоненціального ковзного середнього

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| α / β | MSE | MAE | R2 | MSLE | MAPE |
| 0.2 / 0.1 | 0.00065 | 0.01791 | 0.95994 | 0.000323 | 0.048936 |
| 0.3 / 0.5 | 0.00046 | 0.01475 | 0.97147 | 0.000226 | 0.039885 |
| 0.9 / 0.9 | 0.00020 | 0.00884 | 0.98753 | 0.000010 | 0.024024 |

*Джерело: розрахунки автора*

У розділі 2 було зазначено, що потрійне експоненційне ковзне середнє, відоме також як метод Холта-Вінтерса, є ефективним інструментом для аналізу даних, що містять як тренд, так і сезонність. Для моделювання таких даних використовується адитивна модель тренду та сезонної компоненти. У модулі tsa.holtwinters існує функція ExponentialSmoothing, яка дозволяє задати кількість сезонних періодів, що спостерігаються в даних. В наших даних присутня сезонність та тренд. Модель було перевірено для кількості сезонних періодів = 10. На рис. 9 видно результати.

Изображение выглядит как снимок экрана, График, линия, диаграмма

Автоматически созданное описание

Рис. 9. Реалізація методу Холта-Вінтереса

Джерело: розрахунки автора

Таблиця 5

Оцінка показників якості прогнозу методом потрійного експоненціального згладжування

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| level | MSE | MAE | R2 | MSLE | MAPE |
| 10 | 0.000153 | 0.00758 | 0.99050 | 0.00002 | 0.02062 |

*Джерело: розрахунки автора*

Для підсумку було підведено табл. 6, де зображені результати чотирьох методів згладжування.

Таблиця 6

Порівняння чотирьох методів згладжування

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Модель | MSE | MAE | R2 | MSLE | MAPE |
| Метод простого ковзного середнього (вікно = 5) | 0.000258 | 0.010402 | 0.984210 | 0.000129 | 0.028627 |
| Модель простого екпоненціального згладжування (α = 0.9) | 0.000165 | 0.00788 | 0.98988 | 0.000018 | 0.021405 |
| Модель подвійного експоненціального згладжування (α = 0.9,β = 0.9) | 0.00020 | 0.00884 | 0.98753 | 0.000010 | 0.024024 |
| Модель Холта-Вінтереса (сезонність 10) | 0.000153 | 0.00758 | 0.99050 | 0.00002 | 0.02062 |

*Джерело: розрахунки автора*

Отже, як ми можемо спостерігати, кожна наступна модель згладжування є кращою за попередню. Можемо зробити висновок, що з чотирьох реалізованих методів згладжування найкращою є модель потрійного експоненціального згладжування з додаванням сезонності та тренду.

* 1. **Реалізація економетричних моделей**

В даній роботі було розглянуто три моделі авторегресії: проста модель авторегресії, модель авторегресії ковзного середнього та модель авторегресії інтегрованого ковзного середнього. Відразу зауважимо, що кожний наступний метод показував кращі результати. Для нашого датасету був обрахований коефіцієнт Акайке, який допоміг нам визначити найкращу модель це модель ARIMA (4, 2, 2) та ARIMA (1, 1, 2).

Для реалізації простої моделі авторегресії було взято такі параметри: (4, 0, 0). На рис. 10 зображені результати даної моделі.

Изображение выглядит как График, линия, диаграмма, снимок экрана

Автоматически созданное описание

Рис. 10. Реалізація простої моделі авторегресії (4, 0, 0)

*Джерело: розрахунки автора*

Таблиця 7

Оцінка якості прогнозу методом простої авторегресії

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Параметри | MSE | MAE | R2 | MSLE | MAPE |
| (1, 0, 0) | 0.039285 | 0.181933 | -1.4063 | 0.02208 | 0.61867 |

*Джерело: розрахунки автора*

Для реалізації простої моделі авторегресії з ковзним середнім (ARMA) було взято такі параметри: (1, 0, 2), тому що модель з параметрами (4, 0, 2) показала неадекватні результати. На рис. 11 зображені результати даної моделі.

Изображение выглядит как График, диаграмма, линия, текст

Автоматически созданное описание

Рис. 11. Реалізація методу ковзного середнього (1, 0, 2)

Джерело: розрахунки автора

Таблиця 8

Оцінка якості прогнозу методом авторегресії ковзного середнього

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Параметри | MSE | MAE | R2 | MSLE | MAPE |
| (1, 0, 2) | 0.039281 | 0.181909 | -1.4061 | 0.02208 | 0.61860 |

*Джерело: розрахунки автора*

Для реалізації моделі авторегресії інтегрованого ковзного середнього було взято такі параметри: (1, 1, 2). На рис. 12 зображені результати даної моделі.

Изображение выглядит как График, диаграмма, линия, снимок экрана

Автоматически созданное описание

Рис. 12. Реалізація методу ARIMA(1, 1, 2)

*Джерело: розрахунки автора*

Таблиця 8

Оцінка якості прогнозу методом авторегресії ковзного середнього

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Параметри | MSE | MAE | R2 | MSLE | MAPE |
| (1, 1, 2) | 0.039281 | 0.181909 | -1.4084 | 0.02209 | 0.61860 |

*Джерело: розрахунки автора*

Отже, за допомогою коефіцієнта Айкайе було підібрано параметри для реалізації трьох методів авторегресії. Слід зауважити, що кожна модель є трошки кращою за попердню, однак якість прогнозу майже в кожній моделі висока. Результати реалізації трьох моделей бачимо у табл. 9

Таблиця 9

Порівняння результатів моделей авторегресії

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Параметри (p, q, d) | MSE | MAE | R2 | MSLE | MAPE |
| (1, 0, 0) | 0.0392850 | 0.1819330 | -1.4063 | 0.02208 | 0.61867 |
| (1, 0, 2) | 0.0392811 | 0.1819092 | -1.4061 | 0.02208 | 0.61861 |
| (1, 1, 2) | 0.0392810 | 0.1819091 | -1.4084 | 0.02207 | 0.61860 |

*Джерело: розрахунки автора*

**3.4 Реалізація рекурентних нейронних мереж**

Для реалізації прогнозування RNN було обрано модель LSTM (Long Short-Term Memory). Дана модель приймає наступні параметри: кількість кроків назад, кількість вузлів на першому, другому та третьому шарі LSTM та fully-connected шар з отриманим одним виходом з одного елементу.

В даному дослідженні кількість кроків назад підбиралася вручну, аналіз проводився за допомогою значень MSE, MAE, R2, MAPE, MSLGE. Було розроблено 3 архітектури програми, змінюючи параметр кількості кроків назад. Коли похибки зменшувалися, прогноз вважався кращим. В кінці цей параметр був взятий як 10 і ми отримали наступні результати. Результати ми бачимо на рис. 7 та у табл. 6.

Изображение выглядит как текст, снимок экрана, График, диаграмма

Автоматически созданное описание

Рис. 7. Реалізація моделі LSTM(10-30-30-30-1, epochs=20 000)

Таблиця 6

Навчання LSTM

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Модель | Кількість параметрів | R2 | MSLE | MAPE | TEST\_MAE | TEST\_  MSE | epochs |
| LSTM  (30-30-30-1) | 18,631 | 0.92275 | 0.00062 | 0.05917 | 0.03049 | 0.00173 | 100 |
| LSTM  (20-30-30-1) | 18,631 | 0.92276 | 0.00062 | 0.05923 | 0.03083 | 0.00185 | 100 |
| LSTM  (5-30-30-1) | 18,631 | 0.92278 | 0.00063 | 0.06795 | 0.02449 | 0.00128 | 100 |

*Джерело: розрахунки автора*

На табл. 43 зображено зразок начання LSTM з різним підбором параметрів: кількості кроків назад. Для навчання моделі було обрано 100 епох. Спочатку було взято кількість кроків назад як 30, програма показала досить високі резлуьтати. Потім, було зменшено кількість кроків назад до 20, програма показала ще кращі результати. На останок було взято 5 (за кожен тиждень) і мали найращі результати. Таким чином була первірена гіпотеза, що при зменшенні періодів якість прогнозу покращується.

Другий, третій та четвертий параметр даної моделі був незмінним. Кількість вузлів для кожного кроку була незмінна та обрана як 30 (відповідає за кількість параметрів, що даються на нвчання, тренування та вихід).

Як ми бачимо у табл. 6 кількість параметрів скрізь однакова. А при зменшенні кількості кроків похибка зменшувалася, що свідчить про кращий підбір параметра. Зауважимо, що у табл. 6 було наведено приклад тренування моделі з парметром epochs = 100. Збільшуючи кількість епох модель вдосконалюється. Однак незважаючи на не досить велику кількість ітерацій, модель показала дуже якісні результати. У табл. 7 зображена покращена модель, де було виконано 20 000 ітерацій, для того, щоб модель дала якомога точніші результати.

Таблиця 6

Навчання LSTM

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Модель | Кількість параметрів | R2 | MSLGE | MAPE | TEST\_MAE | TEST\_  MSE | epochs |
| LSTM  (30-30-30-1) | 18,631 | 0.99923 | 8e-05 | 13e-05 | 3e-05 | 2e-05 | 20 000 |
| LSTM  (20-30-30-1) | 18,631 | 0.99935 | 10e-05 | 15e-05 | 5e-05 | 4e-05 | 20 000 |
| LSTM  (5-30-30-1) | 18,631 | 0.99998 | 11e-05 | 16e-05 | 6e-05 | 5e-05 | 20 000 |

*Джерело: розрахунки автора*

Отже, при збільшенні кількості ітерацій прогноз стає майже ідеальним. Дана модель є найкращою серед усіх запропонованих.

ВИСНОВКИ

Для інвесторів управління капіталом набуває все більшої ваги в сучасному світі. Як професійні інвестиційні менеджери, так і індивідуальні інвестори, всі прагнуть мати ефективні інструменти для аналізу тенденцій на фондовому ринку, зменшення ризику та збільшення прибутку. І хоча деякі вважають прогнозування цін акцій складним завданням, насправді успішні трейдери щороку здійснюють тисячі транзакцій, які не є чисто спекулятивними.

З самого початку фінансових операцій на біржі люди розробляли методи прогнозування майбутньої вартості активів. З використанням обчислювальних технологій та штучного інтелекту точність цих методів постійно зростає.

У цьому дослідженні було розроблено три різних підходи до прогнозування: методи згладжування, економетричні методи та методи штучних нейронних мереж. Реалізувавши, ці методи найкращим виявився метод прогнозування за допомогою рекурентних нейронних мереж. У табл. 7 та табл. 8 підведено підсумки цих моделей прогнозування.

Таблиця 7

Порівняльна таблиця реалізованих методів

|  |  |  |
| --- | --- | --- |
| Модель | Переваги | Недоліки |
| Згладжування | Легка реалізація, досить якісні результати, обробка тенденцій змінних рівнів та компонентів сезонності. | Вразливі до екстремальних значень |
| Авторегресія | Автоматизований підбір параметрів. | Обмеження в приущеннях |
| Рекурентні нейронні мережі | Висока точність прогнозу | Довго проходить навчання моделі |

*Джерело: розрахунки автора*

Таблиця 8

Результати реалізованих методів

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Модель | MSE | MAE | R2 | MGLE | MAPE |
| Згладжування | 0.000153 | 0.00758 | 0.99050 | 0.00002 | 0.02062 |
| Авторегресія | 0.0392810 | 0.1819091 | -1.40840 | 0.02207 | 0.61860 |
| Рекурентні нейронні мережі | 5e-05 | 6e-05 | 0.99998 | 11e-05 | 16e-05 |

*Джерело: розрахунки автора*

Отже, підсумувавши дані можемо зробити висновок, що рекурентні нейронні мережі дають найточніші результати.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Mathematics Behind Random forest and XGBoost URL: <https://medium.com/analytics-vidhya/mathematics-behind-random-forest-andxgboost-ea8596657275> (дата звернення: 12.04.2021)
2. How may people own Bitcoin? URL: <https://www.buybitcoinworldwide.com/how-manybitcoinusers/#:~:text=With%20one%20study%20suggesting%20~25,the%20world's%20population%20owns%20bitcoin>.
3. Демківський Є.О., Демківська Т.І..Прогнозуванн нестаціонарних фінансово-економічних процесів. IV-а Міжнародна науково-практична конференція «Мехатронні системи: інновації та інжиніринг»: Київ, КНУТД, 22 жовтня 2020 року
4. Time Series Analysis Tutorial Using Financial Data URL: <https://towardsdatascience.com/time-series-analysis-tutorial-using-financial-data4d1b846489f9>
5. Rolling Window Regression: a Simple Approach for Time Series Next value Predictoins URL: <https://medium.com/making-sense-of-data/time-series-next-valueprediction-using-regression-over-a-rolling-window-228f0acae363>
6. Financial, Economic and Alternative Data. Quandl URL: <https://www.quandl.com/>
7. Python Release Python 3.9.2 URL: <https://www.python.org/downloads/release/python-392/>
8. Financial Forecasting Preparing a prediction of the future URL: <https://corporatefinanceinstitute.com/resources/knowledge/modeling/financialforecasting-guide/>
9. Predicing the Stock Market with Machine Learning.Introduction. URL: <https://towardsdatascience.com/predicting-the-stock-market-with-machine-learningintroduction-310cd6069ffa>
10. Predicting Stock Returns Using Financial Statement Information URL: <https://onlinelibrary.wiley.com/doi/abs/10.1111/1468-5957.t01-1-00205>
11. Financial Time Series Forecasting with Deep Learning:A Systematic Literature Review: 2005-2019 URL: <https://arxiv.org/pdf/1911.13288.pdf>
12. Introduction to interact Time Series Visualizations with Plotly in Python URL: [https://towardsdatascience.com/introduction-to-interactive-time-seriesvisualizations-with-plotly-in-python-d3219eb7a7af](file:///C:\Users\Tetyana%20Kravets\Downloads\Telegram%20Desktop\Introduction%20to%20interact%20Time%20Series%20Visualizations%20with%20Plotly%20in%20Python%20URL:%20https:\towardsdatascience.com\introduction-to-interactive-time-seriesvisualizations-with-plotly-in-python-d3219eb7a7af)
13. Machine Learning models for 100% better return in Algo-trading URL: <https://medium.datadriveninvestor.com/machine-learning-models-for-marketbeating-trading-strategies-c773ba46db66>
14. Hyper parameter optimization in Python URL: [https://towardsdatascience.com/hyperparameter-optimization-in-python-part-2- hyperopt-5f661db91324](https://towardsdatascience.com/hyperparameter-optimization-in-python-part-2-%20hyperopt-5f661db91324)
15. Error Analysis to Tour Rescue! URL: [https://towardsdatascience.com/erroranalysis-to-your-rescue-773b401380ef](file:///C:\Users\Tetyana%20Kravets\Downloads\Telegram%20Desktop\Error%20Analysis%20to%20Tour%20Rescue!%20URL:%20https:\towardsdatascience.com\erroranalysis-to-your-rescue-773b401380ef)
16. How to Improve my Ml algorithm? Lessons from Andew Ng’s experience - I URL: <https://medium.com/all-things-data-science/how-to-improve-my-mlalgorithm-lessons-from-andrew-ngs-experience-i-551ca1a32634>
17. Regularization: Ridge, Lasso and Elastic Net URL: <https://www.datacamp.com/community/tutorials/tutorial-ridge-lasso-elastic-net>
18. Deep Learning in Neural Networks: An Overview [Електроннийресурс] – URL: <https://arxiv.org/pdf/1404.7828.pdf>
19. Вибір статистичних методів прогнозування виробництва продукції <https://lib.chmnu.edu.ua/pdf/naukpraci/economy/2005/38-25-14.pdf>
20. Mathematics Behind Random forest and XGBoost URL: <https://medium.com/analytics-vidhya/mathematics-behind-random-forest-andxgboost-ea8596657275>
21. Cuthbertson K. Introductory econometrics for finance, Chris Brooks, Cambridge University Press, Cambridge, 2002. International Journal of Finance & Economics. 2004. Т. 9, № 1. С. 82–83. URL: <https://doi.org/10.1002/ijfe.223>
22. NumPy Tutorial. URL: <https://realpython.com/numpy-tutorial/>
23. Mean Absolute Error. URL: <https://www.statisticshowto.com/absoluteerror/>
24. Bollerslev T. Generalized autoregressive conditional heteroskedasticity. Journal of Econometrics. 1986. Т. 31, № 3. С. 307–327. URL: <https://doi.org/10.1016/0304-4076(86)90063-1>
25. Engle R. F. Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation. Econometrica. 1982. Т. 50, № 4. С. 987. URL: <https://doi.org/10.2307/1912773>
26. Cuthbertson K. Introductory econometrics for finance, Chris Brooks, Cambridge University Press, Cambridge, 2002. International Journal of Finance & Economics. 2004. Т. 9, № 1. С. 82–83. URL: <https://doi.org/10.1002/ijfe.223>
27. Офіційний сайт фондової біржі «Українська біржа». URL: [http://www.ux.ua](http://www.ux.ua/)
28. How-To Guide on Exploratory Data Analysis for Time Series Data URL: <https://medium.com/analytics-vidhya/how-to-guide-on-exploratory-dataanalysis-for-time-series-data-34250ff1d04f>
29. Hyper parameter optimization in Python URL: [https://towardsdatascience.com/hyperparameter-optimization-in-python-part-2- hyperopt-5f661db91324](https://towardsdatascience.com/hyperparameter-optimization-in-python-part-2-%20hyperopt-5f661db91324)
30. Error Analysis to Tour Rescue! URL: <https://towardsdatascience.com/erroranalysis-to-your-rescue-773b401380ef>
31. How to Improve my Ml algorithm? Lessons from Andew Ng’s experience - II URL: <https://medium.com/all-things-data-science/how-to-improve-my-mlalgorithm-lessons-from-andrew-ngs-experience-ii-f66926926f88>
32. Офіційний сайт Ставицького Андрія Володимировича. URL: <http://www.andriystav.cc.ua/S_Art_U.html>

ДОДАТОК А. РЕАЛІЗАЦІЯ МЕТОДІВ ЗГЛАДЖУВАННЯ

import pandas as pd

from sklearn import preprocessing

import matplotlib.pyplot as plt

from sklearn.metrics import mean\_squared\_error, mean\_absolute\_error, r2\_score, mean\_squared\_log\_error, \

mean\_absolute\_percentage\_error

from statsmodels.tsa.holtwinters import ExponentialSmoothing, SimpleExpSmoothing, Holt

from statsmodels.tsa.seasonal import seasonal\_decompose

def forecast\_errors(info, test\_data, real\_data):

mse = mean\_squared\_error(test\_data, real\_data)

mae = mean\_absolute\_error(test\_data, real\_data)

r2 = r2\_score(test\_data, real\_data)

mslge = mean\_squared\_log\_error(test\_data, real\_data)

mape = mean\_absolute\_percentage\_error(test\_data, real\_data)

print(info, '\nMSE = ', mse, '\nMAE =', mae,

'\nR2 = ', r2, '\nMSLGE = ', mslge, '\nMAPE = ', mape, '\n')

return None

def set\_plot(title='', axis='', y=''):

plt.title(title), plt.xlabel(axis), plt.ylabel(y)

plt.legend()

plt.show()

return None

stock\_data = pd.read\_csv('/Users/daniilsuba/Desktop/BTC-USD.csv', sep=',')

# print(stock\_data)

split = 0.3

stock\_data.index = pd.to\_datetime(stock\_data['Date'], format='%Y-%m-%d')

stock\_data = stock\_data.iloc[::-1]

# print(stock\_data)

# print(stock\_data.columns)

stock\_data = stock\_data.drop(columns='Adj Close', axis=1)

# print(stock\_data.columns)

stock\_data["average"] = (stock\_data["High"] + stock\_data["Low"]) / 2

stock\_data.head()

# print(stock\_data)

x = stock\_data.average.values

# print(x)

min\_max\_scaler = preprocessing.MinMaxScaler()

x\_scaled = min\_max\_scaler.fit\_transform(x.reshape(-1, 1))

stock\_data.average = x\_scaled

n\_test = int(split \* len(stock\_data))

test = stock\_data[:n\_test]

train = stock\_data[n\_test:]

plt.plot(train.average, label='Train')

plt.plot(test.average, label='Test')

set\_plot(title='Share prices', axis="Time", y='Price')

# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*moving\_average\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

window = [30, 10, 5]

plt.figure(figsize=(16, 8))

plt.plot(train['average'], label='Train')

plt.plot(test['average'], label='Test')

for w in window:

y\_hat = ((stock\_data['average'].iloc[::-1].rolling(w).mean()).iloc[::-1])

plt.plot(y\_hat[:n\_test], label='Moving average forecast window=' + str(w))

text = f'Moving Average: window = {w}'

forecast\_errors(text, test.average, y\_hat[:n\_test])

plt.legend(loc='best')

plt.show()

# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Simple\_Exp\_smoothing\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

plt.figure(figsize=(16, 8))

plt.plot(train['average'], label='Train')

plt.plot(test['average'], label='Test')

parameters = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]

for p in parameters:

fit1 = SimpleExpSmoothing(stock\_data.average).fit(smoothing\_level=p, optimized=False)

y\_hat = fit1.fittedvalues

plt.plot(y\_hat[:n\_test], label='Simple exp smoothing level=' + str(p))

text = f'Simple\_Exp\_Smoothing: level = {p}'

forecast\_errors(text, test.average, y\_hat[:n\_test])

plt.legend(loc='best')

plt.show()

# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*DOUBLE\_EXP\_SMOOTHING\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

plt.figure(figsize=(16, 8))

result = seasonal\_decompose(stock\_data.average, model='additive')

result.plot()

plt.show()

parameters = [[0.2, 0.1], [0.3, 0.5], [0.9, 0.9]]

plt.figure(figsize=(16, 8))

plt.plot(train['average'], label='Train')

plt.plot(test['average'], label='Test')

for p, s in parameters:

fit1 = Holt(stock\_data.average).fit(smoothing\_level=p, smoothing\_trend=s)

y\_hat = fit1.fittedvalues

plt.plot(y\_hat[0:n\_test], label='Double exp smoothing level=' + str(p) + ' slope=' + str(s))

text = f'Double exp smoothing: level = {p}, season = {s}'

forecast\_errors(text, test.average, y\_hat[:n\_test])

plt.legend(loc='best')

plt.show()

# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*HOLT\_WINTERES\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# look for seasonal periods, trend and seas

plt.figure(figsize=(16, 8))

plt.show()

parameters = [[10, 'add', 'add']]

plt.figure(figsize=(16, 8))

plt.plot(train['average'], label='Train')

plt.plot(test['average'], label='Test')

for p, tr, seas in parameters:

fit1 = ExponentialSmoothing(stock\_data.average, seasonal\_periods=p, trend=tr, seasonal=seas).fit()

y\_hat = fit1.fittedvalues

plt.plot(y\_hat[:n\_test], label=f'HOLT-WINTERES periods={p} trend={tr} seasonal={seas}')

text = f'Holt-Wi: level = {p}, trend = {tr}, season = {seas}'

forecast\_errors(text, test.average, y\_hat[:n\_test])

residual = test.average - y\_hat[0:n\_test]

plt.legend(loc='best')

plt.show()

ДОДАТОК Б. РЕАЛІЗАЦІЯ ЕКОНОМЕТРИЧНИХ МЕТОДІВ

# AR Methods

import pandas as pd

import matplotlib.pyplot as plt

from sklearn.metrics import mean\_squared\_error, mean\_absolute\_error, r2\_score, mean\_squared\_log\_error, \

mean\_absolute\_percentage\_error

import statsmodels.api as sm

from sklearn import preprocessing

from pmdarima.arima import auto\_arima

def forecast\_errors(info, test\_data, real\_data):

mse = mean\_squared\_error(test\_data, real\_data)

mae = mean\_absolute\_error(test\_data, real\_data)

r2 = r2\_score(test\_data, real\_data)

mslge = mean\_squared\_log\_error(test\_data, real\_data)

mape = mean\_absolute\_percentage\_error(test\_data, real\_data)

print(info, '\nMSE = ', mse, '\nMAE =', mae,

'\nR2 = ', r2, '\nMSLGE = ', mslge, '\nMAPE = ', mape, '\n')

return None

stock\_data = pd.read\_csv('/Users/daniilsuba/Desktop/BTC-USD.csv')

stock\_data = stock\_data.iloc[::-1]

stock\_data = stock\_data.drop(columns='Adj Close', axis=1)

stock\_data.index = pd.to\_datetime(stock\_data['Date'], format='%Y-%m-%d')

stock\_data["average"] = (stock\_data["High"] + stock\_data["Low"]) / 2

stock\_data.head()

x = stock\_data.average.values

min\_max\_scaler = preprocessing.MinMaxScaler()

x\_scaled = min\_max\_scaler.fit\_transform(x.reshape(-1, 1))

stock\_data.average = x\_scaled

n\_test = int(0.3\*len(stock\_data))

test = stock\_data[0:n\_test]

train = stock\_data[n\_test:]

plt.plot(train.average, label='Train')

plt.plot(test.average, label='Test')

plt.title("Share prices")

plt.legend()

plt.xlabel("Time")

plt.show()

y = stock\_data.average[::-1]

for el in range(1, 3):

stepwise\_model = auto\_arima(y, start\_p=3, start\_q=1,

max\_p=4, max\_q=5, m=12,

start\_P=0, seasonal=False,

d=el, D=0, trace=True,

error\_action='ignore',

suppress\_warnings=True,

stepwise=True)

# print(stepwise\_model.aic())

# (1, 1, 2)

def set\_arima(dataset, order, train\_data):

model = sm.tsa.statespace.SARIMAX(dataset,

order=tuple(order),

seasonal\_order=(0, 0, 0, 12),

enforce\_stationarity=False,

enforce\_invertibility=False)

results\_ar = model.fit()

pred\_ar = results\_ar.get\_prediction(start=len(train\_data), dynamic=False)

pred\_ci = pred\_ar.conf\_int()

plt.figure(figsize=(16, 8))

ax = train.average.plot(label='train', color='green')

ax = test.average.plot(label='test', color='blue')

pred\_ar.predicted\_mean.plot(ax=ax, label=f'AR {order}', alpha=0.7, color='red')

ax.set\_xlabel('Time check')

ax.set\_ylabel('Share price')

plt.legend()

plt.show()

y\_forecasted = pred\_ar.predicted\_mean

plt.figure(figsize=(16, 8))

forecast\_errors("", test.average, y\_forecasted)

return None

set\_arima(dataset=y, order=(1, 0, 0), train\_data=train)

set\_arima(dataset=y, order=(1, 0, 2), train\_data=train)

set\_arima(dataset=y, order=(1, 1, 2), train\_data=train)

ДОДАТОК В. РЕАЛІЗАЦІЯ РЕКУРЕНТНИХ НЕЙРОННИХ МЕРЕЖ

# RNN

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from keras import Sequential

from keras.layers import Dense, LSTM

from sklearn.metrics import mean\_squared\_error, mean\_absolute\_error, r2\_score, mean\_squared\_log\_error, \

mean\_absolute\_percentage\_error

from sklearn.preprocessing import MinMaxScaler

def forecast\_errors(info, test\_data, real\_data):

mse = mean\_squared\_error(test\_data, real\_data)

mae = mean\_absolute\_error(test\_data, real\_data)

r2 = r2\_score(test\_data, real\_data)

mslge = mean\_squared\_log\_error(test\_data, real\_data)

mape = mean\_absolute\_percentage\_error(test\_data, real\_data)

print(info, '\nMSE = ', mse, '\nMAE =', mae,

'\nR2 = ', r2, '\nMSLGE = ', mslge, '\nMAPE = ', mape, '\n')

return None

def set\_plot(title='', axis='', y=''):

plt.title(title), plt.xlabel(axis), plt.ylabel(y)

plt.legend()

plt.show()

return None

stock\_data = pd.read\_csv('/Users/daniilsuba/Desktop/BTC-USD.csv')

stock\_data = stock\_data.iloc[::-1]

plt.plot(stock\_data.Open[::-1], color='green', label='Open Price')

set\_plot(title='Stock Open Price', axis='Time', y='USD')

stock\_data = stock\_data.drop(columns='Adj Close', axis=1)

stock\_data.index = pd.to\_datetime(stock\_data['Date'], format='%Y-%m-%d')

stock\_data["average"] = (stock\_data["High"] + stock\_data["Low"]) / 2

stock\_data.head()

stock\_data.describe()

input\_feature = stock\_data.iloc[:, [2, 6]].values

input\_data = input\_feature

sc = MinMaxScaler(feature\_range=(0, 1))

input\_data[:, 0:2] = sc.fit\_transform(input\_feature[:, :])

lookback = 10

test\_size = int(0.3 \* len(stock\_data))

X = []

y = []

for i in range(len(stock\_data) - lookback - 1):

t = []

for j in range(lookback):

t.append(input\_data[[i + j], :])

X.append(t)

y.append(input\_data[i + lookback, 1])

X, y = np.array(X), np.array(y)

# print(X, y)

X\_test = X[:test\_size]

y\_test = y[:test\_size]

X\_train = X[test\_size:]

y\_train = y[test\_size:]

X = X.reshape(X.shape[0], lookback, 2)

X\_test = X\_test.reshape(X\_test.shape[0], lookback, 2)

X\_train = X\_train.reshape(X\_train.shape[0], lookback, 2)

# print(X.shape)

# print(X\_test.shape)

# print(X\_train.shape)

plt.plot(stock\_data.average[:test\_size], color='orange', label='Train')

plt.plot(stock\_data.average[test\_size:], color='blue', label='Test')

set\_plot(title='Stock Average Prices', axis='Time', y='Stock Opening Price')

model = Sequential()

model.add(LSTM(units=30, return\_sequences=True, input\_shape=(X.shape[1], 2)))

model.add(LSTM(units=30, return\_sequences=True))

model.add(LSTM(units=30))

model.add(Dense(units=1))

model.summary()

model.compile(optimizer='sgd', loss=['mean\_squared\_error', 'mean\_absolute\_error'])

history = model.fit(X\_train, y\_train, epochs=100, batch\_size=32)

predicted\_value\_test = model.predict(X\_test)

predicted\_value\_train = model.predict(X\_train)

score = model.evaluate(X\_test, y\_test, verbose=0)

forecast\_errors("", y\_test, predicted\_value\_test)

plt.plot(input\_data[:, 1][::-1], color='green', label='dataset')

plt.plot(np.concatenate((predicted\_value\_test, predicted\_value\_train))[::-1], color='yellow', label='predicted')

plt.plot(np.concatenate((predicted\_value\_test, predicted\_value\_train))[:test\_size:-1], color='red', label='train')

set\_plot(title='Price of stocks sold', axis='Time check', y='Stock Price')

ДОДАТОК Г. ПОСИЛАННЯ НА БАЗУ ДАНИХ

<https://github.com/danyashuba/BTC_prediction_models/blob/main/BTC-USD.csv>