

УДК 004.94; 536.2

Я.І. Соколовський¹, М. Бордун², М.В. Левкович¹¹Національний університет «Львівська політехніка»²Львівський національний університет імені І.Франка

РОЗРОБЛЕННЯ ПРОГРАМНОГО ТА АЛГОРИТМІЧНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ПРОГНОЗУВАННЯ КУРСУ КРИПТОВАЛЮТ З ВИКОРИСТАННЯМ МЕТОДІВ ФРАКТАЛЬНОГО АНАЛІЗУ

У роботі створено програмне та алгоритмічне забезпечення для моделювання та прогнозування криптовалюти Bitcoin з використанням фрактальної моделі ARFIMA (AutoRegressive Fractionally Integrated Moving Average). Проведено аналіз моделей прогнозування часових рядів (авторегресійні, фрактальні). Також проведено підбір найбільш відповідних параметрів обраної фрактальної моделі для максимізації точності з огляду на метрику RMSE. Проаналізовані ряди на наявність тренду, сезонності, білого шуму, нестационарності та довготривалої пам'яті. Досліджено показники Херста та адаптовано алгоритм вибору оптимального параметра d фрактального диференціювання моделі ARFIMA. Обґрунтовано вибір програмних засобів для реалізації алгоритмів та моделей прогнозування за допомогою мови програмування Python версії 3.6.5 з використанням бібліотек pandas версії 1.1.3 та numpy версії 1.19.2. Для того, щоб прогнозувати часові ряди, була використана мова програмування R версії 4.1.3, разом з бібліотеками forecast версії 8.16 та arfima версії 1.8.0. Здійснено програмну реалізацію фрактальної моделі ARFIMA. Перенесено застосунок на хмарний сервіс Google Colab із використанням сховища Google Drive для зберігання даних та результатів прогнозування. Отримано результати порівняння ефективності створеної фрактальної моделі з такою ж моделлю, що має автоматичний підбір параметрів, а також з найбільш відповідною авторегресійною моделлю на різних розмірах тренувальних та тестових даних. Встановлено, що більша кількість даних як тренувальних, так і тестових однозначно посилює перевагу фрактальних моделей, оскільки в даному випадку наявний тривалий ефект, тобто яскраво виражена довга пам'ять в розглянутому часовому ряді.

Ключові слова: Криптовалюта; часовий ряд; авторегресійна модель; довга пам'ять; фрактальна модель; ARFIMA.

ВСТУП

Протягом останніх років криптовалюти набувають все більшої популярності, особливо це стосується Bitcoin, який є ключовим рушієм для всього крипторинку. Криптовалюта є більш безпечною альтернативою фізичній валюті, яку ми на даний час використовуємо. Це було здобуто завдяки шифруванню передачі конфіденційних даних за допомогою криптографічних протоколів, які є надзвичайно складними кодовими системами.

Для того, щоб проаналізувати та прогнозувати історичні дані, а особливо фінансові, найчастіше використовують авторегресійні моделі. Однак, аналіз наукових публікацій свідчить про те, що фрактальні моделі користуються популярністю, оскільки дозволяють врахувати наявність довготривалої пам'яті у процесах.

Об'єктом дослідження є дані курсів криптовалют, які представлені у вигляді впорядкованих послідовностей точок даних, розподілених на певний період часу.

Предметом дослідження є власне програмне та алгоритмічне забезпечення для аналізу динаміки курсів криптовалют та їх прогнозування на підставі адаптованих алгоритмів фрактального аналізу.

Отже, **метою роботи** є розроблення програмного та алгоритмічного забезпечення інформаційної системи, достатньо надійної для того, щоб допомогти інвестору в аналізі та прогнозуванні динаміки обраної ним криптовалюти.

Для досягнення поставленої мети у роботі виконано такі **завдання**:

- Підібрано набір даних найбільш популярних криптовалют та проаналізовано обмежену вибірку для правильної фільтрації даних;
- Синтезовано математичну модель фрактального часового ряду криптовалют, проведено декомпозицію ряду, проаналізовано його на білий шум, стаціонарність і наявність довгої пам'яті;
- Проведено підбір відповідних параметрів фрактальної моделі для максимізації точності з використанням метрики RMSE;
- Адаптовано алгоритм для ідентифікації фрактальних параметрів моделі ARFIMA;
- Порівняно ефективність створеної фрактальної моделі з такою ж моделлю з автоматичним підбором параметрів і авторегресійною моделлю на різних розмірах тренувальних та тестових даних. Виконано виміри з використанням хмарних технологій і порівняно час моделювання фрактальними методами.

Аналіз літературних джерел

Фрактальний аналіз — це відносно молода і швидко прогресуюча галузь сучасного комп'ютерного моделювання. Вона вивчає властивості математичних об'єктів зі складною локальною будовою за допомогою засобів теорії мір дробових порядків, метричних розмірностей, операторів дробового інтегрування та диференціювання. Нехтувати мікроструктурами і мікрофлуктуаціями реальних об'єктів, процесів і явищ — це означає спотворювати істинну природу речей. Особливо це стосується фінансового сектору, зокрема ринку криптовалют, що є хаотичний, і стандартні методи не можуть зафіксувати всі закономірності для подальшого ефективного прогнозування [1]. Термін «криптовалюта» вперше почали використовувати після появи платіжної системи Bitcoin. Надзвичайно важливим є те, що система ціни криптовалют забезпечена інформаційними потужностями, які використовуються для її видобутку і всією іншою інфраструктурою, що створена для обслуговування транзакцій. Щодо кількості криптовалют на ринку, то ця сума вже перевищує 17800 станом на березень 2022, базуючись на біржу. Збільшення кількості криптовалют було викликано зростанням популярності Ethereum з еко-системою, яка основана на смарт-контрактах DeFi (Decentralized finance). Це, звісно, спонукало до того, щоб створити різні типи криптовалют для вирішення вже існуючих проблем блокчейну, включаючи конфіденційність, масштабованість, швидкість транзакцій.

Найбільш відомими методами прогнозування закономірностей ринку криптовалют є моделі часових рядів. Найкращим представником серед яких виступає ARIMA (AutoRegressive Integrated Moving Average) [2]. Autoregressive (AR) порядку p , що описує число значущих затримок часового ряду, порядок інтегрованості ряду d та Moving Average (MA) порядку p , який описує кількість значущих помилок прогнозу. Варто зазначити, що узагальненням цієї моделі є фрактальна модель ARFIMA (AutoRegressive Fractionally Integrated Moving Average). Саме вона дає можливість моделювання часових рядів з довгою пам'яттю, тобто показник d може набувати нецілих значень. Отже, таке узагальнення сприяє здійсненню необхідного аналізу різних часових рядів із врахуванням тривалої пам'яті в часовому ряді, оскільки модель ARIMA дає можливість фіксувати процеси або з короткою пам'яттю з $d=0$, або з нескінченною при $d=1$. Зокрема, порівняння моделювання часових рядів за допомогою процесів ARMA і ARFIMA, використовуючи симуляції Монте Карло [3]-[4], доводить, що ігнорування довгої пам'яті може зумовити погіршення результатів. Таке спостереження є надзвичайно важливим, тому що на практиці досліднику ніколи не відомо, який процес у дійсності лежить в основі динаміки цін фінансових активів. Виходячи з вище наведених міркувань, можна розглядати використання процесів ARFIMA як один із найбільш сучасних та актуальних підходів для вивчення фінансових часових рядів.

Дослідження, які пов'язані з аналізом часових рядів та побудови фрактального диференціювання, були проведені за допомогою мови програмування Python версії 3.6.5 з використанням бібліотек pandas версії 1.1.3 та numpy версії 1.19.2. Для прогнозування часового ряду використана мова програмування R версії 4.1.3, разом з бібліотеками forecast версії 8.16 та arfima версії 1.8.0. Розділення на дві мови програмування для реалізації цієї роботи було вимушеним, бо на даний час немає програмних бібліотек, що пов'язані з роботою такої моделі як ARFIMA чи іншими фрактальними моделями.

Слід зазначити, що були використані також хмарні технології, а саме: Google Colab, разом зі сховищем даних Google Drive. Colab. Однак, є можливість вибрати мову програмування як і Python, так і R. В даному випадку було створено R-середовище на базі версії R 4.1.3. У ньому використовувалися дані, що були розміщені на Google Drive, і здійснювалися кроки, які пов'язані з моделюванням та прогнозуванням, що попередньо виконані на власних потужностях ПК.

Результати дослідження та їх обговорення

Датасет для прогнозування динаміки курсів криптовалют.

Датасет було взято з платформи Kaggle [5], а дані для його створення було отримано між 28 квітня 2013 аж до 7 липня 2021 року та взято з біржі <https://coinmarketcap.com/>. Для кожної криптовалюти датасет має по одному csv-файлу. За версією <https://www.bankrate.com/investing/types-of-cryptocurrency/> було обрано 5 найбільш популярних криптовалют, а саме: Bitcoin, Ethereum, Binance Coin, Cardano. Відомості стосовно кожної валюти брали щодня. Їх описували декількома характеристиками, такими як: дата, відкриваюча ціна на ринку, максимальна, мінімальна, закриваюча ціна, об'єм транзакцій, а також капіталізація ринку.

Дослідження даних для найпопулярніших криптовалют.

Для того, щоб більш детально зрозуміти загальні патерни крипторинку, потрібно визначити певні залежності між його елементами. Тому було проаналізовано кореляцію Пірсона між обраними криптовалютами (рис.1). Можна зробити висновок, що ринок є дуже зв'язний, а найбільше це стосується найбільш відомих криптовалют, для яких кореляція не була нижче за показника 0.86.

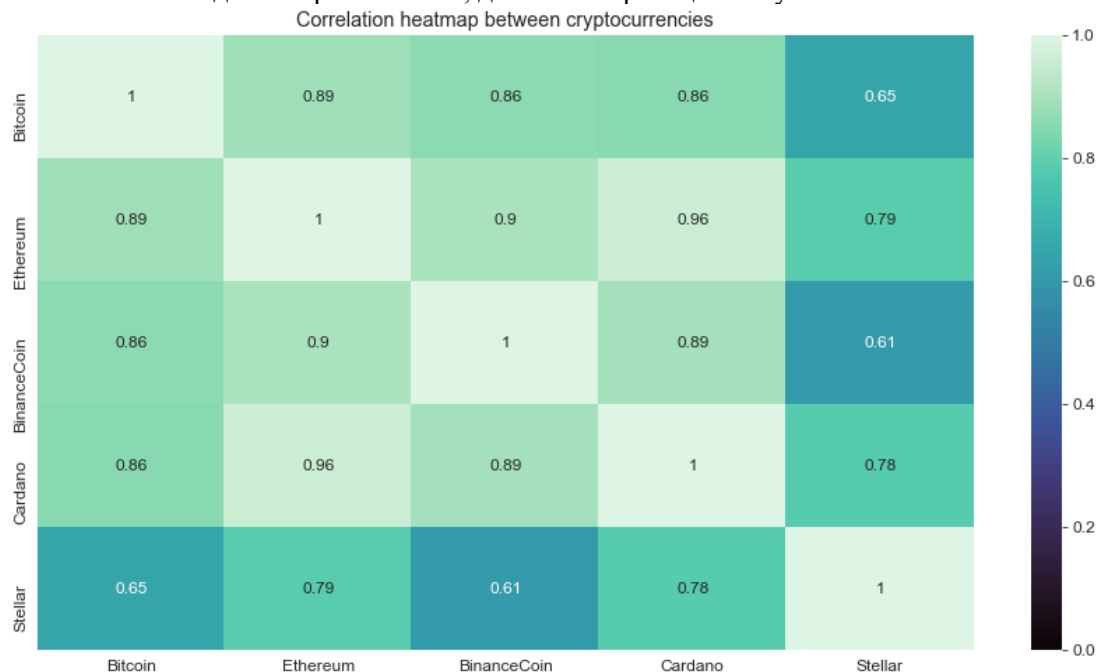


Рис. 1. Кореляція Пірсона для часових рядів криптовалют

Дослідження наявності тренду, сезонності та білого шуму в часових рядах.

Для того, щоб почати моделювання та прогнозування, потрібно визначити певні властивості часових рядів. Тож першочерговим кроком виступає декомпозиція часового ряду ціни Bitcoin (рис.2). Вона створена завдяки модулю statsmodels.apі мови програмування Python. Таким чином, з неї можна визначити наявність тренду та чітко виражену сезонність.

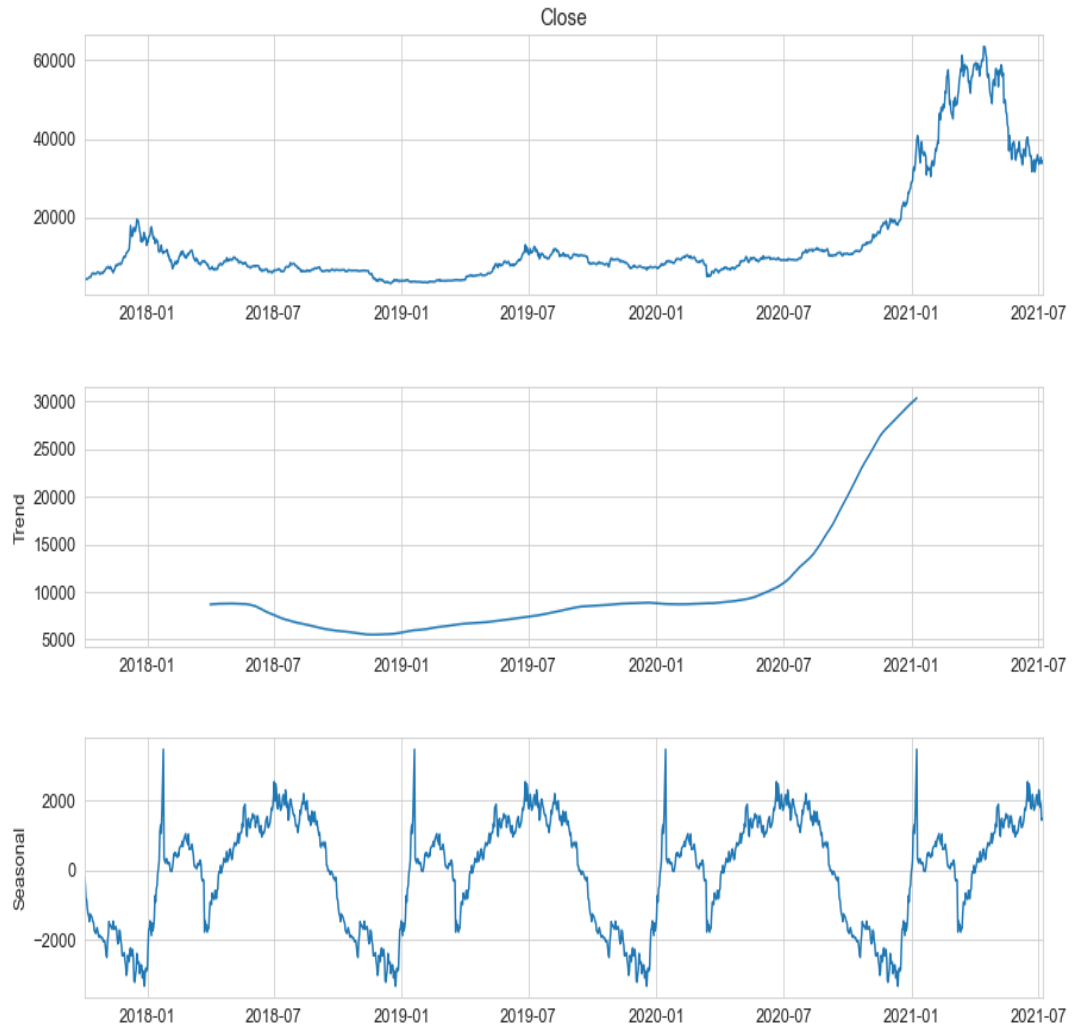


Рис.2. Декомпозиція часового ряду ціни Bitcoin

Що стосується останнього графіка з рис. 2, то він характеризує надлишки, тобто показує шум в нашому часовому ряді. В основному він повинен мати патерни білого шуму, що характеризується нульовим середнім значенням, сталою дисперсією і незалежністю змінних [6]. Проте, в даному випадку можна побачити, що хоч і середнє значення є близьке до 0, дисперсія не є стабільною і тому можна відкинути гіпотезу про наявність білого шуму в графіку надлишків.

Дослідження показників Херста, стаціонарності часових рядів та довгої пам'яті.

Зазначимо, що подальшим етапом аналізу є знаходження показника Херста для оцінки довготривалої пам'яті в часовому ряді. Значення експоненти Херста для часового ряду ціни Bitcoin було визначено завдяки R/S аналізу [7]. В цьому випадку отримали значення 0.9347. Воно вказує, що часовий ряд характеризується довгостроковою позитивною автокореляцією. Отже, це свідчить про наявність тренду та довгої тривалої пам'яті.

На рис. 3 наведено графік ACF, тобто графік автокореляцій. Він показує значення кореляції між часовим рядом і його версією з відставанням при кожному значенні затримки. Чітко виражена гіпербола на графіку підтверджує наявність довготривалої пам'яті. Варто зауважити, що потрібно близько 300 часових затримок, щоб повністю нівелювати ефект залежності.

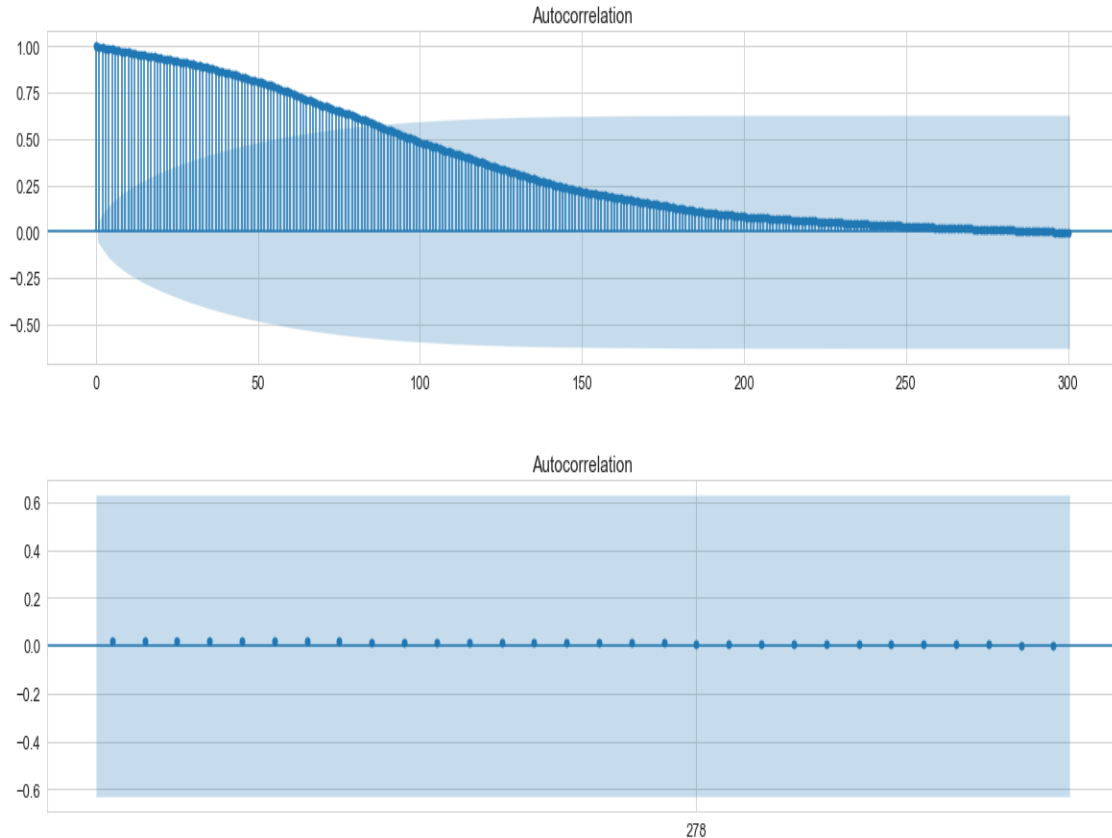


Рис.3. Функція ACF для часового ряду ціни Bitcoin/ ACF function for Bitcoin price time series

Використані два статистичні тести ADF та KPSS, які перевіряють протилежні гіпотези щодо визначення стаціонарності. Це є важливим індикатором наявності одиничного кореня, тобто процесу $I(1)$, який показує нестационарність часового ряду. Цей результат отриманий з використанням модуля `statsmodels.tsa.stattools` мови програмування Python, який показав p -значення статистики ADF 0.832468, що є більшим за поріг 0.05, і служить підтвердженням нульової гіпотези. І p -значення статистики KPSS становить менше ніж 0.01, що є менше за поріг 0.05.

Математичний опис фрактальної моделі ARFIMA.

Autoregressive (AR) модель [2] порядку p має вигляд:

$$X_t = c + \sum_{i=1}^p \phi_i X_{t-i} + \varepsilon_t, \quad (1)$$

де ϕ_1, \dots, ϕ_p – авторегресійні параметри, c – константа, випадкова змінна, ε_t – білий шум.

Moving Average (MA) модель [2] порядку q :

$$X_t = \mu + \sum_{i=1}^q \theta_i \varepsilon_{t-i} + \varepsilon_t, \quad (2)$$

де $\theta_1, \dots, \theta_q$ – параметри рухомого середнього, μ – константа.

Узагальненням перелічених вище моделей [2] виступає:

$$\left(1 - \sum_{i=1}^p \phi_i B^i\right) (1-B)^d (X_t - \mu) = \left(1 + \sum_{i=1}^q \theta_i B^i\right) \varepsilon_t, \quad (3)$$

де $(1-B)^d$ називають оператором диференціювання. Моделі ARMA та ARIMA можуть фіксувати тільки процеси з короткою пам'яттю, оскільки параметр d набуває тільки цілочислових значень. Для того, щоб фіксувати процеси з довгою пам'яттю, необхідно використати фрактальну модель ARFIMA(p, d, q), де показник d набуває вже дробових значень.

Можемо також розкласти оператор диференціювання, використовуючи біноміальний розклад для будь-якого дійсного числа d :

$$(1-B)^d = \sum_{k=0}^{\infty} \binom{d}{k} (-B)^k = 1 - dB + \frac{d(d-1)}{2!} B^2 - \frac{d(d-1)(d-2)}{3!} B^3 + \dots \quad (4)$$

Оператор диференціювання акиж можна представити через Гамма-функцію $\Gamma(\cdot)$ [8]:

$$(1-B)^d = \sum_{k=0}^{\infty} \left(\frac{\Gamma(k-d)}{\Gamma(k+1)\Gamma(-d)} \right) (-B)^k. \quad (5)$$

При $d = 0$, X_t є білим шумом, а його автокореляційна функція дорівнює 0. Коли $d = 1$, X_t є випадковим блуканням, значення автокореляційної функції якого дорівнює 1, і його можна розглядати як білий шум після диференціації першого порядку. Коли d є вже дійсним числом,

$$X_t = - \sum_{k=1}^{\infty} \left(\frac{\Gamma(k-d)}{\Gamma(k+1)\Gamma(-d)} \right) X_{t-k} + \varepsilon_t. \text{ Отже, на } X_t \text{ впливають усі історичні дані } (X_{t-1}, X_{t-2}, \dots).$$

Обґрунтування вибору оптимального параметра d фрактального диференціювання моделі ARFIMA.

Різноматні методи оцінки параметра фрактального диференціювання можуть класифікуватися в дві групи – параметричні та напівпараметричні [10]. У першій групі методи включають в собі функцію правдоподібності. В другій найбільш використовуваним є метод GPH (автори Geweke та Porter-Hudak) [8]. Суть напівпараметричного підходу полягає в оцінці параметрів моделі в два етапи: окремо оцінюється параметр d , а вже потім відбувається оцінка інших параметрів. Оскільки модель ARFIMA з бібліотеки `forecast` мови програмування R з автоматичним підбором параметрів використовує напівпараметричний підхід, то детальніше опишемо метод GPH. Він починається з оцінки параметру d і заснований на регресії найменших квадратів в спектральній області.

Щоб продемонструвати цей метод, використаємо функцію спектральної щільності стаціонарної моделі X_t , $t = 1, \dots, T$:

$$f_X(\lambda) = \left[4 \sin^2 \left(\frac{\lambda}{2} \right) \right]^d f_\varepsilon(\lambda). \quad (6)$$

де $f_\varepsilon(\lambda)$ є спектральною щільністю ε_t .

Логарифм з функції спектральної щільності можна представити наступним чином:

$$\log \{ f_X(\lambda) \} = \log \{ f_\varepsilon(0) \} - d \log \left\{ 4 \sin^2 \left(\frac{\lambda}{2} \right) \right\} + \log \left\{ \frac{f_\varepsilon(\lambda)}{f_\varepsilon(0)} \right\}. \quad (7)$$

Нехай $I_X(\lambda_j)$ є періодограмою, побудованою на частотах Фур'є $\lambda_j = \frac{2\pi j}{T}$, $j = 1, 2, \dots, m, T$

[11]. Величина m є числом розглянутих частот Фур'є, що є кількістю ординат періодграми [12]. Тоді отримаємо:

$$\log \{ I_X(\lambda_j) \} = \log \{ f_\varepsilon(0) \} - d \log \left\{ 4 \sin^2 \left(\frac{\lambda_j}{2} \right) \right\} + \log \left\{ \frac{f_\varepsilon(\lambda_j)}{f_\varepsilon(0)} \right\} + \log \left\{ \frac{I_X(\lambda_j)}{f_X(\lambda_j)} \right\}. \quad (8)$$

де $\log \{ f_\varepsilon(0) \}$ – константа, $Y_j = \log \left\{ 4 \sin^2 \left(\frac{\lambda_j}{2} \right) \right\}$ – змінна, $\log \left\{ \frac{I_X(\lambda_j)}{f_X(\lambda_j)} \right\}$ – помилка.

Вибір величини m є важливим питанням, оскільки це впливає на результати оцінювання. З одного боку, m має бути достатньо малою величиною, щоб розглядати лише частоти, близькі до нуля. З іншого боку, m має бути достатньо великою, щоб забезпечити збіжність оцінки методу

найменших квадратів. Оцінка GPH потребує два головних припущення пов'язаних з асимптотичною поведінкою рівняння (8):

H1: для низьких частот, ми припускаємо, що $\log \left\{ \frac{f_\varepsilon(\lambda_j)}{f_\varepsilon(0)} \right\}$ є незначною;

H2: випадкові змінні $\log \left\{ \frac{I_X(\lambda_j)}{f_X(\lambda_j)} \right\}, j=1, 2, \dots, m$ є асимптотично незалежні однаково розподілені випадкові величини.

Тоді оцінка методу GPH є оцінкою методу найменших квадратів для регресії $\log \{I_X(\lambda_j)\}$ на константах α та Y_j . Оцінка показника d , позначена як \hat{d}_{GPH} визначається наступним чином:

$$\hat{d}_{GPH} = \frac{\sum_{j=1}^m (Y_j - \bar{Y}) \log \{I_X(\lambda_j)\}}{\sum_{j=1}^m (Y_j - \bar{Y})^2}, \quad (9)$$

де $\bar{Y} = m^{-1} \sum_{j=1}^m (Y_j)$ та $m = g(T)$ з $\lim_{T \rightarrow \infty} g(T) = \infty$ та $\lim_{T \rightarrow \infty} \frac{g(T)}{T} = 0$.

У роботі [6] показали, що якщо $T \rightarrow \infty$ та $|d| < \frac{1}{2}$, то маємо:

$$\sqrt{m}(\hat{d}_{GPH} - d) \square N \left[0, \frac{\pi^2}{6} \left\{ \sum_{j=1}^m (Y_j - \bar{Y})^2 \right\}^{-1} \right]. \quad (10)$$

Згідно [9] параметр m повинен бути вибраний так, що $m = T^\nu$, $\nu = 0.5, 0.6, 0.7$. Згідно з припущенням нормального закону для X_t , було доведено, що результуюча оцінка є узгодженою. Тому оцінене середнє квадратичне відхилення можна використовувати для висновку.

На підставі використання цього методу з тренувальною вибіркою часового ряду ціни криптовалюти Bitcoin у таблиці 1 наведено порівняння оцінок та їх середніх квадратичних відхилень. Це тільки ще раз підтверджує факт нестационарності часового ряду і високої волатильності ринку, оскільки $d \geq 1/2$, а також те що необхідно брати достатньо велику вибірку для аналізу, щоб отримати прийнятну похибку.

Табл. 1. Порівняння оцінок параметру d та їх точності для різного розміру вибірки часового ряду ціни Bitcoin

Розмір вибірки	Оцінка параметру d	Середнє квадратичне відхилення
1845	1.3232	0.1123
1344	1.3048	0.1120
500	1.1757	0.1951

Програмна реалізація фрактальної моделі ARFIMA.

Для моделювання часового ряду з необхідними для моделі ARFIMA властивостями використано алгоритм фрактального диференціювання [9], який включає два етапи: визначення ваг та застосування їх у добутку з нашим часовим рядом, використовуючи метод фіксованого вікна. Розроблений програмний код. Зокрема, метод get weights, який приймає показники параметру інтегрованості часового ряду, довжини часового ряду, а також порогового значення, та повертає ваги, необхідні для визначення фрактального диференціювання. Також запрограмований метод fracdiff_FFD, який приймає часовий ряд, параметр інтегрованості часового ряду, а також порогове значення, та повертає фрактально диференційований ряд отриманий алгоритмом фіксованого вікна.

Також необхідно описати Autoregressive (AR) та Moving Average (MA) моделі. Для реалізації симуляції фрактального процесу ARFIMA початковий часовий ряд генерується випадковим чином з використанням нормального розподілу. Метод `ma_model` приймає показники MA-предикторів, кількість необхідних для генерування елементів часового ряду, середнє квадратичне відхилення та альфа значення при застосуванні стійкого розподілу, та повертає згенерований ряд за моделлю MA. Метод `arma_model` приймає показники AR-предикторів та показники шуму, які визначаються фрактальним диференціюванням часового ряду за моделлю MA, та повертає згенерований ряд за моделлю ARMA. Також враховуються впливи MA коефіцієнтів, виконується фрактальне диференціювання і в підсумку враховуються AR коефіцієнти. Послідовність виконання цієї логіки, що зображено на рисунку 4.

```
def arfima_sim(
    ar_params,
    d,
    ma_params,
    n_points,
    noise_std = 1,
    noise_alpha = 2,
    warmup = 0):
    """Generate discrete series using ARFIMA process"""
    ma_list = ma_model(
        ma_params, n_points + warmup, noise_std=noise_std, noise_alpha=noise_alpha
    )
    ma_df = pd.DataFrame(ma_list, columns=["MA Series"])
    frac_ma = fracdiff_FFD(ma_df, d)
    arfima_series = arma_model(ar_params, frac_ma)
    return arfima_series[-n_points:]
```

Рис. 4. Функція симуляції ARFIMA, для якої необхідно вказати значення коефіцієнтів $AR(p)$, d , $MA(q)$ та кількість елементів часового ряду/ ARFIMA simulation function, for which it is necessary to specify the values of the coefficients $AR(p)$, d , $MA(q)$ and the number of elements of the time series

Приклад виконання генерування часового ряду із симуляцією процесу ARFIMA представлено на рис. 5.

Часовий ряд ціни Bitcoin є нестационарний. Це було підтверджено двома статистичними тестами ADF та KPSS. Для того, щоб визначити параметри $AR(p)$ та $MA(q)$ моделі ARFIMA, необхідно отримати стаціонарний ряд. Тому в алгоритмі фрактального диференціювання було підібрано мінімальний параметр інтегрованості ряду d , так, щоб відхилялася нульова гіпотеза тестом ADF. Таким чином можна отримати стаціонарний ряд (рис. 6).

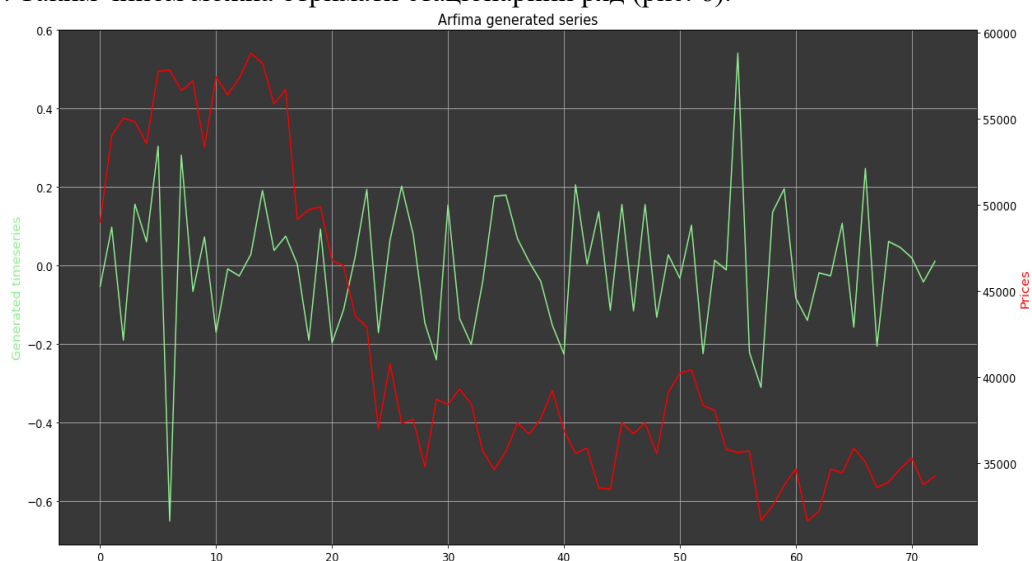


Рис. 5. Симульований часовий ряд процесом ARFIMA (салатовий колір) та порівняння з вихідним часовим рядом (червоний колір)

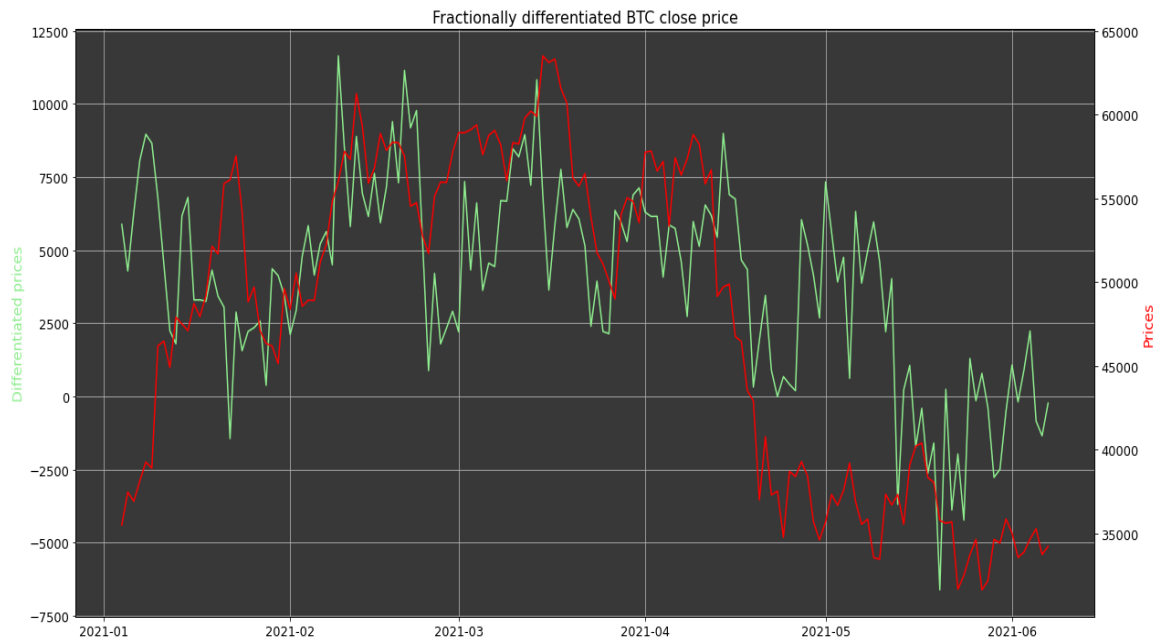
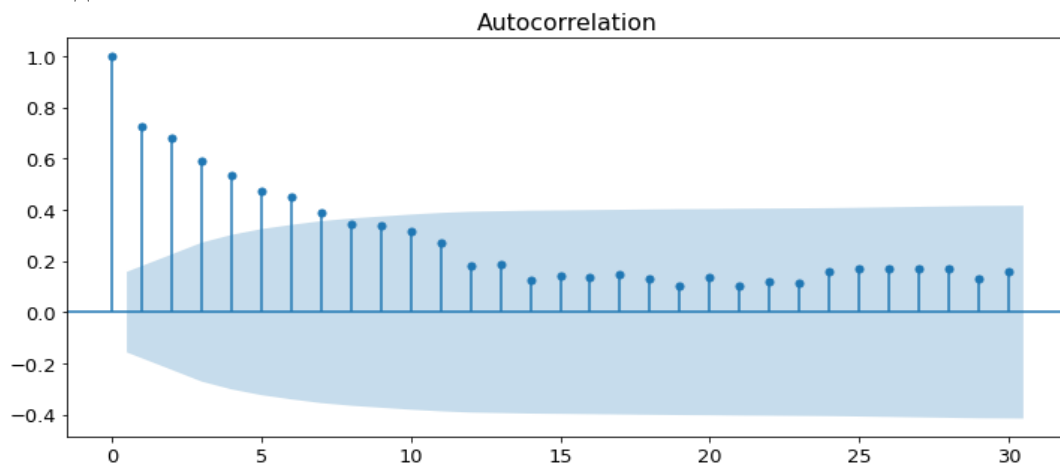


Рис.6. Диференційований часовий ряд ціни Bitcoin (салатовий колір) та порівняння з вихідним часовим рядом (червоний колір)/ Differentiated time series of Bitcoin price (green color) and comparison with the original time series (red color)

Значення функцій ACF та PACF для нашого диференційованого ряду і кількість тих значущих затримок, які більші за діапазон інтервалу довіри (голуба область на рис. 7), буде визначати максимальне число для наших параметрів $AR(p)$ з графіку ACF та $MA(q)$ з графіку PACF відповідно. Також визначався показник d інтегрованості ряду за допомогою функції `fracdiff` з бібліотеки `fracdiff` мови програмування R. Вибір оптимальної моделі визначався по оцінці RMSE з огляду на різницю між тестовими даними. А також для порівняння виконано моделювання за допомогою моделей ARFIMA та ARIMA з автоматичним підбором параметрів. Було також проведено дослідження зі зміною розмірів тестової і тренувальної вибірки для кожної з вище згаданих моделей.



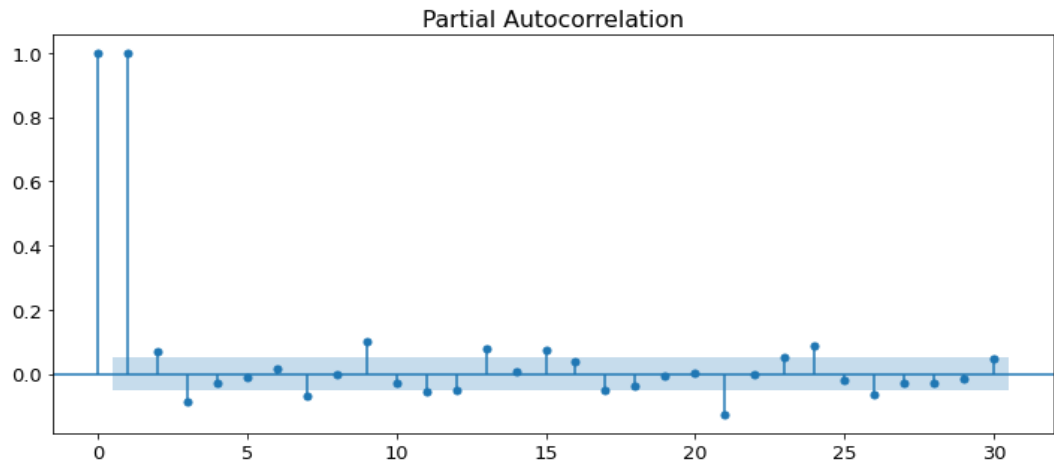


Рис. 7. Функції ACF та PACF для диференційованого часового ряду ціни Bitcoin/ ACF and PACF functions for differentiated Bitcoin price time series

Отже, прогнозовані дані візуально представлені на рис. 8. Також ще показується 95% інтервал довіри. Він позначений голубою областю та означає інтервал, у межах якого з заданою довірчою імовірністю можна очікувати значення оцінюваної випадкової величини.

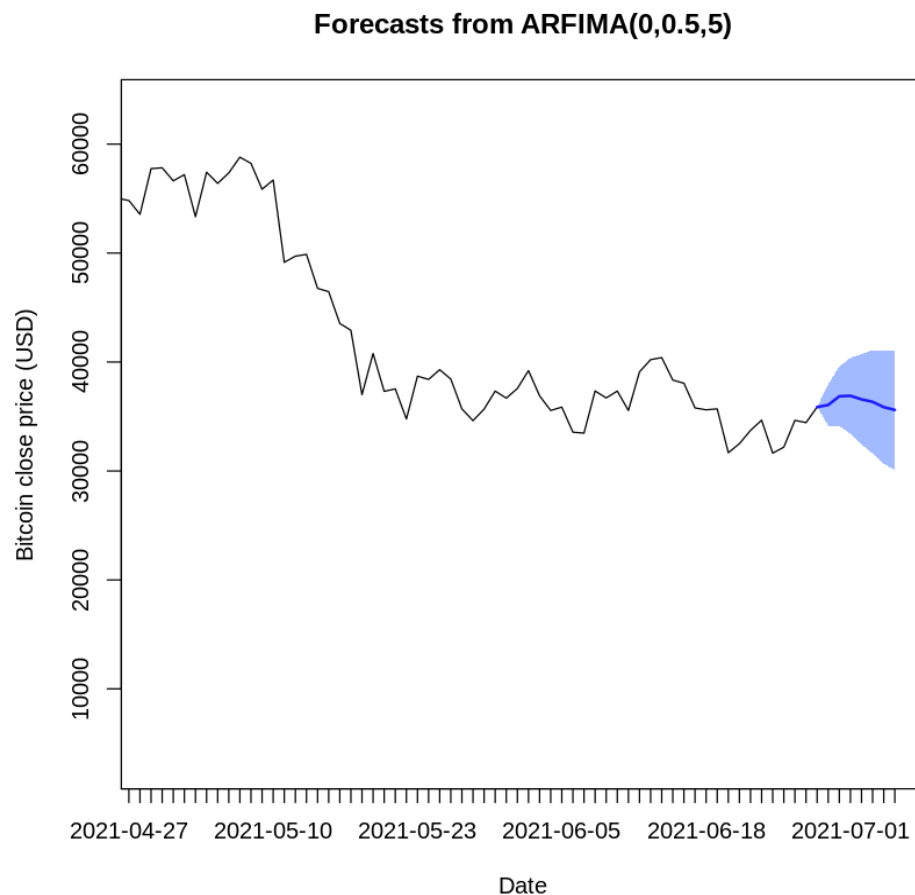


Рис.8. Тижневе прогнозування Bitcoin моделлю ARFIMA(0, 0.4994476, 5)

Порівняння між вище згаданими моделями проведено з використанням двох метрик для тестової вибірки, а саме: RMSE та MAPE. Метрика RMSE характеризує середнє квадратичне відхилення між прогнозованими та реальними значеннями, а MAPE описує середнє абсолютне відсоткове відхилення. Результати подано у табл. 2.

Табл. 2. Порівняння точності прогнозування даних між тренувальними моделями та тестовими даними

1344 тренувальних, 30 тестових даних	Модель	RMSE	MAPE (%)
	(auto)ARFIMA	2109.5	4.81
	(auto)ARIMA	2317.6	5.35
	(manually fitted) ARFIMA	2140.78	4.53
1367 тренувальних, 7 тестових даних	(auto)ARFIMA	2136	5.76
	(auto)ARIMA	1636.34	4.43
	(manually fitted) ARFIMA	654.14	1.71
500 тренувальних, 30 тестових даних	(auto)ARFIMA	2741.5	6.92
	(auto)ARIMA	2308.63	5.3
	(manually fitted) ARFIMA	2205.05	4.67

Для дослідження використаний хмарний сервіс Google Colab. Він забезпечує необхідну модель надання хмарних сервісів PaaS. У цьому випадку хмарний провайдер надає доступ до всієї інфраструктури, включаючи засоби розробки і тестування. З позиції користувача, необхідно тільки подбати про свій програмний продукт, який можна почати створювати у зручному інтерактивному режимі, як з використанням мови програмування Python, так і з R language. Таким чином, використані бібліотеки віртуальної машини, а власні програмні застосунки можна завантажити командою `!pip install`, в будь-якій вільній комірці для Python-коду та `install.packages()` для R-коду.

Висновки та рекомендації

У роботі розглянута інформаційна технологія прогнозування курсу криптовалют за наявності довготривалої пам'яті в нестационарних часових рядах.

Обґрунтовано вибір програмних засобів для реалізації алгоритмів та моделей прогнозування за допомогою мови програмування Python версії 3.6.5 з використанням бібліотек pandas версії 1.1.3 та numpy версії 1.19.2. Для прогнозування часових рядів використана мова програмування R версії 4.1.3, разом з бібліотеками forecast версії 8.16 та arfima версії 1.8.0. Проведено дослідження показників Херста, стаціонарності часових рядів та довгої пам'яті.

На зразку криптовалюти Bitcoin результати виконання прогнозування на повному наборі тренувальних даних також підтверджують доцільність використання моделі ARFIMA в порівнянні з моделлю ARIMA з огляду на різноманітні оцінки тестової похибки такі як: RMSE, MAPE. Адаптовано алгоритм для вибору оптимального параметра d фрактального диференціювання моделі ARFIMA.

В загальному слід зазначити, що більша кількість тренувальних та тестових даних однозначно посилює перевагу фрактальних моделей, адже в цьому випадку наявний тривалий ефект, яскраво виражена довга пам'ять в розглянутому часовому ряді. Проте, потрібно завжди перевіряти дані й очищувати аномальні відхилення, що спричиняють похибку в оцінці прогнозування.

Зокрема, було успішно проведено дослідження моделювання та прогнозування з використанням хмарного сервісу Google Colab, а також сховища даних Google Drive. В майбутньому можна використати хмарну інфраструктуру на сервісі Microsoft Azure із використанням Telegram-бота на Python, що давав би результати прогнозування для даних із біржі в

реальному часі з використанням функціоналу, який вже був створений з використанням R language.

Література

1. Shah, V., Shroff, G. (2021). *Forecasting Market Prices using DL with Data Augmentation and Meta-learning: ARIMA still wins!* <https://doi.org/10.48550/arXiv.2110.10233>
2. Liu, K., Chen, Y., Zhang, X. (2017). *An Evaluation of ARFIMA (Autoregressive Fractional Integral Moving Average) Programs*. *Axioms*, Vol. 6, 1-16. <https://doi.org/10.3390/axioms6020016>
3. Andersson, M., (1998). *On Effects of Imposing or Ignoring Long Memory When Forecasting*. *Working Paper Series in Economics and Finance*, 225.
4. Ndong, M., Diongue, A, Diop, A., Hili, O. (2015). *Estimation of Fractional ARIMA Process with Stable Innovations: A Monte Carlo Study*. Retrieved from: <https://hal.archives-ouvertes.fr/hal-01121443/document>
5. *Cryptocurrency Historical Prices.* (2021). Retrieved from: <https://www.kaggle.com/datasets/sudalairajkumar/cryptocurrencypricehistory>
6. *How to Detect Random Walk and White Noise in Time Series Forecasting.* (2021). Medium. Retrieved from: <https://towardsdatascience.com/how-to-detect-random-walk-and-white-noise-in-time-series-forecasting-bdb5bbd4ef81>
7. Safitri, D., Mustafid, D. (2019). *Gold price modeling in Indonesia using ARFIMA method*. *Ispriyanti, Sugito, IOP Conf. Series: Journal of Physics: Conf. Series* 1217, 7-9. [DOI:10.1088/1742-6596/1217/1/012087](https://doi.org/10.1088/1742-6596/1217/1/012087)
8. Mohamed, B., Khalfaoui, R. (2011). *Estimation of the long memory parameter in non stationary models: A Simulation Study*. Retrieved from: <https://halshs.archives-ouvertes.fr/halshs-00595057/document>
9. M. L. de Prado (2018). *Advances in Financial Machine Learning*. John Wiley & Sons, Inc., Hoboken, New Jersey, 75-91. <https://doi.org/10.1080/14697688.2019.1703030>
10. Reisen, V., Abraham, B., Lopes, S. (2006). *Estimation of Parameters in ARFIMA Processes: A Simulation Study*. [DOI:10.1081/SAC-100107781](https://doi.org/10.1081/SAC-100107781)
11. Kokoszka, P., Mikosch, T. (2000). *The periodogram at the Fourier frequencies Stochastic Processes and their Applications*. Volume 86, Issue 1, 49-79. [https://doi.org/10.1016/S0304-4149\(99\)00086-1](https://doi.org/10.1016/S0304-4149(99)00086-1)
12. Cerovecki, C., Characiejus, V., Hormann, S. (2022). *The Maximum of the Periodogram of a Sequence of Functional Data*. <https://doi.org/10.1080/01621459.2022.2071720>

Ya. I. Sokolovskyy ¹, M. Bordun ², M.V. Levkovich ¹

¹ Lviv Polytechnic National University

² Ivan Franko Lviv National University of Lviv

DEVELOPMENT OF SOFTWARE AND ALGORITHMIC SECURITY FOR FORECASTING THE CRYPTOCURRENCY COURSE USING FRACTAL ANALYSIS METHODS

The work created software and algorithmic support for modeling and forecasting the Bitcoin cryptocurrency using the ARFIMA (AutoRegressive Fractionally Integrated Moving Average) fractal model. Time series forecasting models (autoregressive, fractal) were analyzed. The selection of the most appropriate parameters of the selected fractal model was also carried out to maximize accuracy in view of the RMSE metric. The series were analyzed for trend, seasonality, white noise, non-stationarity and long-term memory. The Hurst indicators were studied and the algorithm for choosing the optimal parameter d of fractal differentiation of the ARFIMA model was adapted. The choice of software tools for implementing algorithms and forecasting models using the Python programming language version 3.6.5 using the pandas version 1.1.3 and numpy version 1.19.2 libraries is justified. In order to forecast the time series, the

programming language R version 4.1.3 was used, along with the forecast version 8.16 and arfima version 1.8.0 libraries. The software implementation of the ARFIMA fractal model was carried out. Transferred the application to the Google Colab cloud service using Google Drive storage for storing data and forecasting results. The results of comparing the effectiveness of the created fractal model with the same model with automatic selection of parameters, as well as with the most appropriate autoregression model on different sizes of training and test data were obtained. It was established that a larger amount of both training and test data clearly favors fractal models, since in this case there is a long-lasting effect, that is, a pronounced long memory in the second time series. The result is a software system that can be used by investors and ordinary people to analyze and forecast their chosen cryptocurrency using a modern fractal modeling approach. It is important to always check the data and clean up anomalous deviations that cause error in the prediction estimate.

Key words: Cryptocurrency; time series; autoregressive model; long memory; fractal model; ARFIMA.

