УКРАЇНСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ НАУКИ І ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ «ДНІПРОВСЬКИЙ ІНСТИТУТ ІНФРАСТРУКТУРИ І ТРАНСПОРТУ»
МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

УКРАЇНСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ НАУКИ І ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ «ДНІПРОВСЬКИЙ ІНСТИТУТ ІНФРАСТРУКТУРИ І ТРАНСПОРТУ»
МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Кваліфікаційна наукова
праця на правах рукопису

ЖАДАН АРТЕМ АНАТОЛІЙОВИЧ

УДК 004.42:[004.94:519.76]+519.246.8

ДИСЕРТАЦІЯ

КОНСТРУКТИВНО-ПРОДУКЦІЙНЕ МОДЕЛЮВАННЯ ФРАКТАЛЬНИХ СКЛАДОВИХ ЧАСОВИХ РЯДІВ

122 – Комп'ютерні науки
12 – Інформаційні технології

Подається на здобуття наукового ступеня доктора філософії

Дисертація містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

_____ А. А. Жадан

Науковий керівник: Шинкаренко Віктор Іванович, д.т.н., професор

Дніпро – 2026

АНОТАЦІЯ

Жадан А. А. Конструктивно-продукційне моделювання фрактальних складових часових рядів.

Дисертація на здобуття наукового ступеня доктора філософії за спеціальністю 122 – Комп'ютерні науки (галузь знань 12 – Інформаційні технології). – Український державний університет науки і технологій, Дніпро, 2026.

Дисертаційна робота присвячена дослідженню використання та автоматизації методів конструктивно-продукційного моделювання для відновлення параметричної конструктивної моделі наявного часового ряду. В межах цього дослідження розглядається розширення застосування конструктивізму для вирішення оберненої задачі моделювання, а саме визначення конструктивних моделей для вже відомих значень часового ряду.

Часовий ряд являє собою набір послідовних спостережень значень певної величини або станів системи, розташованих у хронологічному порядку відповідно до визначеного інтервалу часу. Часові ряди широко використовуються у різних прикладних сферах, таких як економіка, медицина, інженерія та інші. Вони дозволяють виявляти наявні залежності між значеннями спостережень на відповідному проміжку часу, що дає змогу визначати поточні тенденції, сезонні коливання та циклічні процеси.

Популярність досліджень часових рядів зумовлена насамперед зростаючими потребами в аналізі даних у реальному часі та прогнозуванні їх подальших значень. Залежно від сфери застосування кінцева мета аналізу може відрізнятися, проте загальна користь полягає у можливості виявлення ключових тенденцій зміни даних та передбачення їх розвитку. Це дозволяє заздалегідь обґрунтовувати рішення, оптимізувати поточні процеси та підвищувати ефективність управлінських чи виробничих рішень.

Сучасні методи дослідження часових рядів дозволяють виявляти як лінійні, так і нелінійні закономірності, аналізувати тренди даних та здійснювати прогнозування на різних часових інтервалах. Розвиток обчислювальних технологій

спрощує етапи дослідження та впровадження методів, а збільшення обсягів даних розширює можливості для верифікації моделей та підвищення точності прогнозів.

Методи конструктивно-продукційного моделювання вже були успішно випробувані у сфері моделювання фрактальних часових рядів різної структури та поведінки, зокрема детермінованих і стохастичних. Втім, розв'язання оберненої задачі, а саме визначення конструктивної моделі та її набору параметрів для заданого часового ряду – як синтетичного, так і реального – залишається відкритим і малодослідженим питанням.

У цьому дослідженні розглядаються та пропонуються підходи до організації процесу відновлення конструктивної моделі та набору її параметрів для визначеного часового ряду. Зокрема, впроваджується поступовий ітеративний процес контрольованого підбору параметрів моделі з метою побудови ряду, максимально наближеного до вхідного. Досягнення рішення поставленої задачі включає розширення, уніфікацію та адаптацію методів конструктивно-продукційного моделювання для опису процесу відновлення.

Метою дисертаційної роботи є розробка, теоретичне обґрунтування та дослідження методу відновлення параметрів конструктивно-продукційних моделей часових рядів різної природи. Підхід базується на створенні універсального інструментарію, що дозволяє працювати як з детермінованими, так і зі стохастичними даними, використовуючи математичний апарат конструктивізму для виявлення прихованих закономірностей відповідно до внутрішніх фрактальних властивостей даних.

Об'єктом дослідження виступають процеси моделювання та відновлення параметрів часових рядів різної природи, а предметом – методи, моделі та алгоритмічні засоби реконструкції конструктивно-продукційних моделей. У межах поставлених задач формалізовано обернений процес моделювання, розроблено нові компоненти для аналізу стохастичних рядів, створено масштабоване програмне забезпечення для автоматизації експериментів та проведено оцінку адекватності отриманих моделей на реальних даних.

Наукова новизна результатів полягає у вперше запропонованому ітеративному підході до відновлення параметрів, що дозволяє отримати структурну математичну модель замість класичної апроксимації. Формалізовано систему конструкторів для відтворення логіки генерації ряду за його значеннями. Удосконалено метод оцінки близькості (фітнес-функцію) для стохастичних даних на основі ансамблевого зіставлення реалізацій, що підвищує точність реконструкції, а також оптимізовано архітектуру генетичного алгоритму через механізм ізольованого масштабування обчислень.

Подальшого розвитку набули методи гібридного кодування складних хромосомних структур, які поєднують символьні правила L-систем із безперервними числовими параметрами. Також розширено апарат формальних граматик для опису взаємодії фрактальних та стохастичних компонентів, що дозволило інтегрувати засоби генетичного пошуку як базові модулі у загальну систему імітаційного моделювання.

У першому розділі проведено комплексний аналіз сучасних підходів до дослідження часових рядів різної природи. Розглянуто як класичні лінійні та нелінійні моделі, що базуються на аналізі внутрішніх взаємозв'язків між відліками, так і специфічні підходи – зокрема, рекурентний аналіз та конструктивно-продукційне моделювання. Досліджено їхні ключові особливості, переваги, недоліки та специфіку практичного застосування. Результатом цього етапу стало формування актуального аналітичного інструментарію та окреслення фундаментальних обмежень існуючих методів.

У другому розділі викладено теоретичні засади дослідження. Основну увагу приділено формалізації комплексу спеціалізованих конструкторів, їхнього функціоналу та правил взаємодії для забезпечення процедури відновлення параметрів конструктивної моделі. Відповідно до обраних підходів детально представлено процеси обробки детермінованих та стохастичних синтетичних рядів із виділенням їхніх ключових відмінностей та необхідних алгоритмічних модифікацій. Результатом розділу є узагальнена методика роботи з реальними наборами даних для ідентифікації параметрів їхніх конструктивних моделей.

У третьому розділі представлено експериментальні дослідження, що базуються на викладеному раніше теоретичному базисі. Окремо досліджено процеси моделювання синтетичних детермінованих, синтетичних стохастичних та реальних часових рядів. Для кожного типу даних було визначено специфічні особливості обробки, наведено репрезентативні приклади вибірок та узагальнено отримані результати.

У четвертому розділі розглянуто різні підходи до архітектурної реалізації програмного рішення, призначеного для автоматизації експериментальних випробувань та подальшого впровадження системи. Відповідно до еволюції завдань дослідження було розроблено та протестовано три парадигми побудови архітектури: монолітну однопотокову, мультиагентну та мікросервісну (безсерверну). Кожен із цих підходів було послідовно впроваджено для досягнення максимальної обчислювальної та часової ефективності системи.

Відповідно, у дисертаційній роботі представлено завершений цикл наукового дослідження – від теоретичного обґрунтування концепції до її практичного втілення у вигляді програмного комплексу. Отримані результати визначають нові можливості для застосування конструктивно-продукційного моделювання для аналізу синтетичних та реальних часових рядів. Практичну цінність отриманих висновків та створених програмних модулів підтверджено їхнім використанням у межах зазначених науково-дослідних робіт.

Ключові слова: фрактальні часові ряди, конструктивно-продукційне моделювання, L-система, конструктор, формальні граматики, генетичний алгоритм, програмне забезпечення, інформаційні технології, хмарні обчислення, показник Герста.

СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА ЗА ТЕМОЮ ДИСЕРТАЦІЇ

Праці у фахових виданнях МОН України

1. Шинкаренко, В. И., ***Жадан, А. А.*** (2025). Використання методів конструктивно-продукційного моделювання для прогнозування детермінованих часових рядів. Системи і технології, 6, с. 124-137. DOI: 10.34185/1562-9945-5-161-2025-12.

2. Шинкаренко, В. И., ***Жадан, А. А.*** (2025). Сучасні методи та засоби роботи з часовими рядами. Наука і прогрес транспорту. 4(112), 142–154. DOI: 10.15802/stp2025/341773

3. Шинкаренко, В. И., ***Жадан, А. А.*** (2025). Реалізація процесу відновлення конструктивно-продукційних моделей фрактальних часових рядів. 4, с. 3-11. DOI: 10.15407/pp2025.04

Праці міжнародних конференцій, включені до міжнародної наукометричної бази Scopus

4. Shynkarenko, V., Zhadan, A. (2020). Modeling of the deterministic fractal time series by one rule constructors. In: Proceedings of the 2020 IEEE 15th International Conference on Computer Sciences and Information Technologies (CSIT), Zbarazh, Ukraine, 23-26 September. pp. 336–339. DOI: 10.1109/CSIT49958.2020.9321923.

5. Shinkarenko, V., Zhadan, A. (2024). Multiservice Architecture of Software for Stochastic Fractal Time Series Forecasting. 2024 IEEE 19th International Conference on Computer Science and Information Technologies (CSIT), Lviv, Ukraine, 14-17 October. pp. 1-4. DOI: 10.1109/CSIT65290.2024.10982626.

6. Shynkarenko, V., Zhadan, A., & Halushka, O. (2024). Multi-Agent System for Reconstruction Constructive Models of Stochastic Fractal Time Series. Proceedings of the Information Technology and Implementation (IT&I) Workshop: IT Infrastructure and Applied Solutions (IT&I-WS 2024: ITIAS), Kyiv, Ukraine, November 20 – 21, pp. 66-77.

Матеріали міжнародних наукових конференцій

7. Шинкаренко, В. І., Чигир, Р. Р., Жадан, А. А. (2017). Моделювання взаємопов’язаних часових рядів та зображень L-подібних конструктивно-продукційних структур. В «Інформаційні технології в металургії та машинобудуванні». Дніпро. с. 98.

8. Шинкаренко, В. И., Литвиненко, К. В., Чигирь, Р. Р., Жадан, А. А. (2018). Констуктивно-продукционное моделирование фракталов. В «Інтелектуальні сис-

теми прийняття рішень і проблеми обчислювального інтелекту». Матеріали міжнародної наукової конференції. Херсон: ПП Вишемирський В.С. 289–291.

9. Шинкаренко, В. І., Литвиненко, К. В., Чигір, Р. Р., Жадан, А. А. (2017). Вариативность уточняющих преобразований конструктивно-продукционного моделирования. В «Теоретичні та прикладні аспекти побудови програмних систем» (TAAPSD'2017) Київ. 225–230.

10. Шинкаренко, В. І., Жадан, А. А. (2022). Конструктивно-продукційне моделювання фрактальних складових часових рядів. В «Сучасні інформаційні та комунікаційні технології на транспорті, в промисловості та освіті». Дніпро. с. 71.

11. Шинкаренко, В. І., Жадан, А. А. (2023). Конструктивно-продукційне моделювання детермінованих часових рядів з фрактальними властивостями. В «Інформаційні технології в металургії та машинобудуванні». Дніпро. с. 222–226. DOI: 10.34185/1991-7848.itmm.2023.01.060.

12. Шинкаренко, В. І., Жадан, А. А., Галушко, О. В. (2023). Мультиагентне конструктивне моделювання часових рядів. В «Сучасні інформаційні та комунікаційні технології на транспорті, в промисловості та освіті». Дніпро. с. 64

13. Шинкаренко, В. І., Жадан, А. А. (2024). Мультиагентне середовище відновлення фрактальних часових рядів. В «Інформаційні технології в металургії та машинобудуванні», Дніпро. с. 232–236. DOI: 10.34185/1991-7848.itmm.2024.01.041.

14. Шинкаренко, В. І., Жадан, А. А. (2024). Проблеми синхронізації розподілених обчислень на прикладі мікросервісної реалізації генетичного алгоритму. В «Сучасні інформаційні та комунікаційні технології на транспорті, в промисловості та освіті». Дніпро. с. 93.

15. Шинкаренко, В. І., Жадан, А. А. (2025). Реалізація розподілення процесу роботи генетичного алгоритму шляхом впровадження каналів та вузлів обробки повідомлень. В «Інформаційні технології в металургії та машинобудуванні». Дніпро. с. 215–219. DOI: 10.34185/1991-7848.itmm.2025.01.037.

16. Шинкаренко, В. І., Жадан, А. А. (2025). Проблеми синхронізації розподілених обчислень на прикладі мікросервісної реалізації генетичного алгоритму. В

«Сучасні інформаційні та комунікаційні технології на транспорті, в промисловості та освіті». Дніпро. с. 98.

Матеріали наукових конференцій

17. Шинкаренко, В. И., Литвиненко, К. В., Чигирь, Р. Р., Жадан, А. А. (2018). Констуктивно-продукционное моделирование фрактальных временных рядов на основе L-систем. В «Проблеми математичного моделювання». Матеріали Всеукраїнської науково-методичної конференції. Кам'янське: ДДТУ. 161–163.

Свідоцтва про реєстрацію авторського права

18. Шинкаренко, В. І., Чигір, Р. Р., Жадан, А. А. (2019). Комп'ютерна програма «Рекурентний аналіз часових рядів породжуваних в L-системах». Рішення про реєстрацію договору, який стосується права автора на твір № 4367 від 29.05.2019.

19. Шинкаренко, В. І., Чигір, Р. Р., Жадан, А. А. (2019). Комп'ютерна програма «Моделювання взаємопов'язаних часових рядів й геометричних фракталів породжуваних в L-системах». Рішення про реєстрацію договору, який стосується права автора на твір № 4369 від 31.05.2019.

20. Шинкаренко, В. І., Чигір, Р. Р., Жадан, А. А. (2017). Комп'ютерна програма «Моделювання часових рядів із заданими фрактальними властивостями». Свідоцтво про реєстрацію авторського права на твір № 72579 від 27.06.2017.

ABSTRACT

Zhadan A. A. Constructive-synthesizing modeling of time series fractal components.

Dissertation for the degree of PhD in specialty 122 "Computer Science" - Ukrainian State University of Science and Technology, Dnipro, 2026.

The dissertation is devoted to the study of the application and automation of constructive–synthesizing modeling methods for reconstructing a parametric constructive model of a given time series. Within this research, the extension of constructivist approaches to solving the inverse modeling problem is considered, namely, the determination of constructive models for already known values of a time series.

A time series represents a sequence of observations of a certain variable or system states arranged in chronological order according to a specified time interval. Time series are widely used in various applied domains such as economics, medicine, engineering, and others. They enable the identification of dependencies between observed values over time, allowing for the detection of current trends, seasonal fluctuations, and cyclic processes.

The growing interest in time series research is primarily driven by the increasing demand for real-time data analysis and forecasting. Depending on the application domain, the goal of analysis may differ; however, the overall benefit lies in the ability to identify key patterns in data evolution and predict their future behavior. This makes it possible to justify decisions in advance, optimize ongoing processes, and improve the efficiency of managerial and operational decisions.

Modern methods for time series analysis allow the identification of both linear and nonlinear patterns, the analysis of trends, and forecasting across different time horizons. Advances in computational technologies simplify the research and implementation processes, while the growth of available data enhances model validation capabilities and improves forecasting accuracy.

Constructive–synthesizing modeling methods have already been successfully applied to modeling fractal time series of various structures and behaviors, including both deterministic and stochastic ones. However, solving the inverse problem – namely, determining a constructive model and its parameter set for a given time series, whether synthetic or real – remains an open and insufficiently studied issue.

This research proposes approaches to organizing the process of reconstructing a constructive model and its parameter set for a given time series. A gradual iterative process of controlled parameter selection is introduced to generate a series that closely approximates the input data. Solving this problem involves the extension, unification, and adaptation of constructive–synthesizing modeling methods to describe the reconstruction process.

The dissertation aims to develop, theoretically substantiate, and investigate a method for reconstructing the parameters of constructive–synthesizing models for time series of various nature. The approach is based on creating a universal toolkit capable of handling both deterministic and stochastic data by employing the mathematical framework of constructivism to identify hidden patterns in accordance with the intrinsic fractal properties of the data.

The object of research is the processes of modeling and parameter reconstruction of time series of different nature, while the subject is the methods, models, and algorithmic tools for reconstructing constructive–synthesizing models. Within the scope of the study, the inverse modeling process is formalized, new components for analyzing stochastic series are developed, scalable software for automating experiments is created, and the adequacy of the obtained models is evaluated using real data.

The scientific novelty of the results lies in the proposed iterative approach to parameter reconstruction, which enables obtaining a structural mathematical model instead of a classical approximation. A system of constructors is formalized to reproduce the logic of series generation based on its values. The fitness evaluation method for stochastic data is improved through ensemble-based comparison of realizations, increasing reconstruction accuracy. Additionally, the architecture of the genetic algorithm is optimized through a mechanism of isolated computational scaling.

Further development is achieved in hybrid encoding methods for complex chromosomal structures that combine symbolic L-system rules with continuous numerical parameters. The formal grammar framework is also extended to describe the interaction between fractal and stochastic components, allowing the integration of genetic search mechanisms as core modules within a general simulation modeling system.

The first chapter provides a comprehensive analysis of modern approaches to time series research. Both classical linear and nonlinear models based on internal dependencies between observations and specialized approaches such as recurrence analysis and constructive–synthesizing modeling are considered. Their key features, advantages, disadvantages, and application specifics are examined. The result of this stage is the formation of a relevant analytical toolkit and the identification of fundamental limitations of existing methods.

The second chapter presents the theoretical foundations of the research. Special attention is paid to the formalization of a set of specialized constructors, their functionality, and interaction rules to support the parameter reconstruction procedure. According to the chosen methodology, the processes of handling deterministic and stochastic synthetic series are described in detail, highlighting their key differences and required algorithmic modifications. The result of this chapter is a generalized methodology for working with real datasets to identify parameters of their constructive models.

The third chapter presents experimental studies based on the proposed theoretical framework. The processes of modeling synthetic deterministic, synthetic stochastic, and real-world time series are examined separately. For each type of data, specific processing features are identified, representative samples are provided, and the obtained results are summarized.

The fourth chapter discusses various approaches to the architectural implementation of the software system designed to automate experimental studies and support further deployment. In accordance with the evolution of research tasks, three architectural paradigms are developed and tested: monolithic single-threaded, multi-

agent, and microservice (serverless). Each approach is successively implemented to achieve maximum computational and time efficiency.

Thus, the dissertation presents a complete cycle of scientific research – from theoretical justification of the concept to its practical implementation in the form of a software system. The obtained results define new possibilities for applying constructive–synthesizing modeling to the analysis of both synthetic and real time series. The practical value of the findings and developed software modules is confirmed by their application within the framework of the specified research projects.

Keywords: fractal time series, constructive-synthesizing modeling, L-system, constructor, formal grammars, genetic algorithm, software, information technologies, cloud computing, Hurst exponent.

ПЕРЕЛІК АБРЕВІАТУР ТА УМОВНИХ ПОЗНАЧЕНЬ

Абревіатури

API – Application Programming Interface (Інтерфейс програмування додатків).

AWS – Amazon Web Services (Хмарна платформа Amazon).

DDD – Domain-Driven Design (Проєктування на основі предметної області).

FIFO – First In, First Out (Першим прийшов – першим пішов; принцип черги).

gRPC – Google Remote Procedure Call (Протокол віддаленого виклику процедур від Google).

HTTP/2 – Hypertext Transfer Protocol Version 2 (Друга версія протоколу передачі гіпертексту).

JSON – JavaScript Object Notation (Текстовий формат обміну даними).

L-система – Lindenmayer System (Математичний апарат для моделювання фракталів).

OOD – Object-Oriented Design (Об'єктно-орієнтоване проєктування).

PK – Partition Key (Ключ партиціонування в базах даних).

SK – Sort Key (Ключ сортування в NoSQL базах даних).

SQS – Simple Queue Service (Сервіс черг повідомлень від AWS).

TTL – Time to Live (Час життя даних або пакета).

Умовні позначки

C – Конструктор.

C_A – Алгоритмічний конструктор.

Ξ_{TS} – Множина термінальних символів, відповідальних за розгортку та генерацію значень часового ряду.

Φ_{TS} – Множина термінальних символів, відповідальних за зміну поточного значення точки часового ряду.

$B|_X^Y$ – Термінальний алгоритм з вхідною множиною X та вихідною Y.

M – Поновлюваний неоднорідний носій.

Σ – Сигнатура відношень та операцій.

Λ – Набір тверджень для забезпечення процесу конструювання.

ЗМІСТ

ВСТУП 18

РОЗДІЛ 1 ОГЛЯД ІСНУЮЧИХ ПІДХОДІВ ДЛЯ АНАЛІЗУ ТА ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ РІЗНОЇ ПРИРОДИ 28

1.1 Загальна класифікація сучасних методів для аналізу часових рядів 28

1.2 Класичні лінійні методі прогнозування та аналізу часових рядів 29

1.3 Нелінійні методі прогнозування значень часових рядів 31

1.4 Методи штучного інтелекту для прогнозування значень часових рядів 31

1.4.1 Модель Long Short-Term Memory 33

1.4.2 Моделі трансформери для аналізу часових рядів 34

1.4.3 Гібридні моделі нейронних мереж 35

1.5 Рекурентний аналіз та використання рекурентних діаграм 35

1.6 Засоби аналізу та прогнозування фрактальних часових рядів 37

1.6.1 Визначення фрактальної розмірності ряду на основі показника Герста 37

1.6.2 Нелінійні моделі прогнозування та обробки фрактальними часовими рядами 38

1.6.3 Використання фрактальних складових часових рядів 39

1.7 Конструктивно-продукційне моделювання складових часових рядів 40

Висновки до першого розділу 42

РОЗДІЛ 2 КОНСТРУКТИВНО-ПРОДУКЦІЙНЕ ВІДНОВЛЕННЯ МОДЕЛЕЙ ЧАСОВИХ РЯДІВ 44

2.1 Використання конструктивізму для моделювання складових часових рядів 45

2.2 Конструктори і мультиконструктори 46

2.2.1 Перелік параметрів конструктивної моделі часового ряду 47

2.2.2 Конструктори мультиконструктора та їх елементи 48

2.3 Взаємодія конструкторів під час виконання процесу відновлення відповідно до обраного режиму виконання 52

2.4 Обгрунтування фрактальності часового ряду породжених конструктивною моделлю 54

2.5 Модифікація процесу відновлення для охоплення синтетичних стохастичних часових рядів 56

2.6 Визначення підходу до відновлення параметрів моделі реальних часових рядів 59

2.6.1 Опис ітеративного процесу відновлення параметрів моделі 59

2.6.2 Обгрунтування відмови від використання стохастичного підходу для реальних рядів 60

2.6.3 Узагальнення особливостей процесу роботи з реальними часовими рядами 61

Висновки до другого розділу 62

РОЗДІЛ 3 ЕКСПЕРЕМЕНТАЛЬНІ ДОСЛІДЖЕННЯ З МОДЕЛЮВАННЯ ФРАКТАЛЬНИХ СКЛАДОВИХ ЧАСОВИХ РЯДІВ РІЗНОЇ ПРИРОДИ 65

3.1 Експеримент 1: Відновлення параметрів конструктивної моделі синтетичних детермінованих рядів 66

3.1.1 Експеримент 1.1: Відновлення параметрів для одного простого правила підстановки 66

3.1.2 Експеримент 1.2: Відновлення параметрів для одного складного правила підстановки 67

3.1.3 Експеримент 1.3: Відновлення параметрів для кількох простих правил підстановки 69

3.1.4 Експеримент 1.4: Відновлення параметрів для кількох простих правил підстановки та складної структури аксіоми 71

3.1.5 Узагальнення результатів першого етапу експерементальних досліджень 72

3.2 Експеримент 2: Предбачення подальших значень синтетичного детермінованого ряду 74

3.2.1 Визначення списку існуючих моделей для порівняння результатів прогнозування 75

3.2.2 Проведення експерементальної частини та порівняння отриманих результатів прогнозування до обраного набору моделей 77

3.3 Експеримент 3: Відновлення параметрів моделі на основі синтетичних стохастичних часових рядів 80

3.3.1 Процес відновлення та визначення критерії близькості вхідних та отриманних часових рядів 82

3.3.2 Аналіз отриманих результатів експериментів 83

3.4 Експеримент 4: Відновлення параметрів моделі для реальних часових рядів 86

3.4.1 Обрані метрики оцінювання точності та структурної відповідності між вхідним та відновленим часовими рядами 87

3.4.2 Спроба адаптації підходу до відновлення складових моделей стохастичних рядів до обробки реальних даних 88

3.4.3 Експеримент 4.1: Випробування на основі покрокового ітеративного процесу 89

3.4.4 Експеримент 4.2: Дослідження можливостей прогнозування та визначення проблеми її реалізації 94

3.4.5 Експеримент 4.3: Дослідження фрактальних властивостей відновлених стохастичних часових рядів 96

Висновки до третього розділу 97

РОЗДІЛ 4 РЕАЛІЗАЦІЯ ПРОГРАМНОГО РІШЕННЯ ДЛЯ АВТОМАТИЗАЦІЇ ВІДНОВЛЕННЯ КОНСТРУКТИВНОЇ МОДЕЛІ ЧАСОВИХ РЯДІВ 100

4.1 Постановка завдання для автоматизації процесу відновлення 100

4.2 Визначення загального підходу до проектування 101

4.2.1 Перелік основних логічних сутностей 102

4.3 Монолітна реалізація програмного додатку 102

4.4 Мульти-агентний підхід для обробки стохастичних часових рядів 104

4.4.1 Визначення топології зв'язків обчислювальних сутностей 106

4.4.2 Визначення підходу до обміну інформацією між агентами 108

4.5 Вирішення проблеми масштабування окремих етапів ітерації 110

4.5.1 Стратегія роздільного масштабування та модифікація генетичних операцій 110

4.5.2 Загальний архітектурний підхід та мультисервісна парадигма 112

4.5.3 Реалізація програмного додатку на базі провайдеру хмарного середовища AWS 114
4.5.4 Проблеми синхронізації агрегації 116
4.5.5 Модернізація основних сутностей та їх логічна реалізація 120
4.5.6 Реалізація неблокуючих конвеєрів та керування потоком даних .. 124
4.6 Модифікація программного додатку для покриття відновлення параметрів реальних рядів 127
Висновки до четвертого розділу 129
ЗАГАЛЬНІ ВИСНОВКИ 132
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 134
Додаток А Список публікацій здобувача та відомості про апробацію результатів дисертації 146
Додаток Б Акт впровадження 151

ВСТУП

Актуальність теми. Складні системи є невід'ємною складовою сучасного світу, оскільки характеризуються наявністю великої кількості взаємопов'язаних компонентів і динамічними змінами їхніх станів у часі. Для опису та аналізу таких змін широко застосовують часові ряди – один із найпоширеніших способів подання даних, що відображає еволюцію поведінки, станів або окремих показників у хронологічному порядку на заданому інтервалі часу або в режимі реального спостереження. Методи моделювання, аналізу та прогнозування часових рядів відіграють важливу роль у різних галузях, оскільки дозволяють виявляти закономірності розвитку процесів і оцінювати їхні можливі подальші зміни.

Сфери застосування часових рядів є надзвичайно широкими та охоплюють різноманітні галузі – від фінансового аналізу й економіки до геодезії, дослідження біологічних показників організму та аналізу споживання електроенергії. Така універсальність зумовлює необхідність розвитку ефективних методів їх моделювання.

Для розв'язання задач моделювання часових рядів застосовують різноманітні підходи. Зокрема, широко використовуються авторегресійні моделі (AR, ARMA), які описують залежність поточних значень від попередніх і дозволяють ефективно враховувати автокореляційні властивості даних. Значного поширення набули також моделі простору станів із використанням фільтра Калмана, що дають змогу виділяти приховані компоненти ряду та виконувати згладжування сигналу. Окрему групу становлять компонентні методи на основі експоненційного згладжування (ETS, Holt–Winters), у яких часовий ряд розглядається як сукупність рівня, тренду та сезонності. Для більш гнучкого опису структури даних застосовують регресійні моделі з часовими ознаками, такими як лаги, сезонні індикатори та ковзні статистики. У випадках складних нелінійних залежностей використовуються методи машинного навчання, зокрема градієнтний бустинг і глибокі нейронні мережі, здатні відтворювати складні закономірності у часових рядах.

Одним із перспективних напрямів дослідження є фрактальні часові ряди, які описують процеси зі складною нелінійною динамікою, властивостями самоподіб-

ності та масштабної інваріантності. Це означає, що структура ряду зберігає подібність при аналізі на різних часових масштабах. На відміну від класичних моделей, що часто базуються на припущеннях лінійності та стаціонарності, фрактальний підхід дозволяє враховувати довготривалі залежності між спостереженнями та нерегулярний характер поведінки процесів.

Відповідно до фрактальних властивостей формуються спеціалізовані методи аналізу й моделювання, які ґрунтуються на дослідженні нелінійних зв'язків між значеннями ряду та оцінюванні параметрів, що характеризують його структуру. Застосування таких методів дає змогу більш точно описувати внутрішні закономірності процесів, виявляти приховані особливості динаміки та будувати адекватні моделі складних систем.

Фундаментальні основи дослідження фрактальних часових рядів були закладені у працях H. Hurst. Подальший розвиток цього напряму пов'язаний із роботами J. Kantelhardt, D. C. Caccia, J. B. Bassingthwaighte, G. M. Raymond, A. Eke, P. Herman, M. Li, V. V. Kulish, I. R. Pilgrim, R. P. Taylor, C. Porcaro, O. Koval, J. Nystrom, A. Akhrif, M.-A. Paun, Л. Кіріченко та В. Булаха.

Значущість та актуальність досліджень фракталів і часових рядів в Україні зумовили створення відділу динамічних систем та фрактального аналізу в Інституті математики НАН України. Вагомий внесок у розвиток цього напряму здійснили такі науковці, як М. В. Працьовитий, Г. М. Торбін та П. Ратушняк.

Одним із підходів до дослідження та генерації складних систем є конструктивно-продукційне моделювання, яке ґрунтується на використанні формальних граматик і забезпечує можливість гнучкого керування вхідними параметрами. Це дозволяє генерувати структури різної складності та конфігурації, що є важливим для моделювання фрактальних процесів.

Теоретичні засади конструктивізму розглядаються у працях В. Шинкаренка, В. Ільмана та В. Скалозуба. Практичне застосування цього підходу представлено у роботах О. Куроп'ятник (системи виявлення запозичень у текстах), Л. Жучій (онтологічна гармонізація інформаційних систем), О. Жеваго (аналіз процесів налагодження програмного забезпечення), Р. Чигіря та О. Летучих (моделювання

фрактальних кристалічних решіток), а також Г. В. Забули (оптимізація розміщення структур даних у пам'яті).

У межах дослідження часових рядів конструктивно-продукційний підхід застосовується для розв'язання двох взаємопов'язаних задач. Перша задача полягає у моделюванні фрактальних часових рядів на основі заданого набору параметрів. Такий підхід дозволяє відтворювати динаміку та фрактальні властивості ряду відповідно до L-систем, запропонованих A. Lindenmayer забезпечує можливість моделювання як детермінованих, так і стохастичних процесів.

Друга задача є оберненою та полягає у відновленні конструктивної моделі та її параметрів за заданим часовим рядом. У початкових дослідженнях використовувалися синтетичні ряди, сформовані на основі відомих параметрів. Для реалізації процесу відновлення застосовувався генетичний алгоритм, який забезпечує ефективний пошук параметрів, що мінімізують відхилення між змодельованим і вхідним рядом.

Разом із тим, для узагальнення підходу виникає необхідність у впровадженні модифікацій для роботи зі стохастичними часовими рядами. Основною метою є забезпечення можливості відновлення параметрів конструктивних моделей для реальних даних. Важливим аспектом є також створення програмного інструментарію для автоматизації цього процесу з урахуванням обчислювальної складності та забезпечення масштабованості.

У даній роботі розв'язано задачу відновлення параметрів конструктивних моделей для часових рядів різної природи. Дослідження охоплювали як синтетичні детерміновані та стохастичні ряди, так і реальні часові дані. За результатами роботи визначено необхідні операції та особливості обробки різних типів даних. Подальших досліджень потребує практичне застосування побудованих моделей, зокрема для прогнозування шляхом екстраполяції часових рядів на основі відновлених конструктивних структур.

Зв'язок роботи з науковими програмами, темами, планами. Результати досліджень використання конструктивно-продукційного моделювання для роботи з фрактальними часовими рядами були використані при розробці програмних до-

датків у науково-дослідних роботах кафедри «Комп'ютерні інформаційні технології» Дніпровського національного університету залізничного транспорту імені академіка Всеволода Лазаряна (Українського державного університету науки і технологій), які включають роботи «Конструктивно-продукційне моделювання фракталів» (2018 р. № держреєстрації 0118U004215) та «Підвищення конкурентоспроможності залізничного транспорту на основі уніфікованих інтелектуальних технологій процесів перевезень та експлуатації парків технічних систем» (2018 р. № держреєстрації 0117U004392).

Мета та задачі дослідження. Метою дисертаційної роботи є розробка, теоретичне обґрунтування та дослідження методу відновлення параметрів конструктивно-продукційних моделей часових рядів різної природи (детермінованих і стохастичних) із використанням підходів конструктивно-продукційного моделювання.

Досягнення поставленої мети передбачає вирішення таких взаємопов'язаних задач:

- виконати аналіз існуючих результатів та підходів при роботі з синтетичними детермінованими часовими рядами, а також дослідження можливостей застосування конструктивно-продукційного підходу для їх моделювання;
- розробити методи відновлення конструктивних моделей та їх параметрів для синтетичних детермінованих часових рядів з використанням конструктивно-продукційного моделювання та генетичного алгоритму;
- розробити моделі процесу відновлення параметрів конструктивно-продукційної моделі для стохастичних синтетичних та реальних часових рядів і визначення необхідних етапів та компонентів цього процесу;
- розробити нові та вдосконалити існуючі підходи до прикладної реалізації, що забезпечують автоматизацію процесу експериментальних досліджень і подальше впровадження програмного рішення;
- експериментально перевірити ефективність запропонованого підходу на синтетичних і реальних часових рядах, оцінювання точності відновлення параметрів та аналіз адекватності побудованих моделей.

Об’єктом дослідження є процеси моделювання та відновлення параметрів часових рядів різної природи.

Предметом дослідження є методи, моделі та алгоритмічні засоби відновлення параметрів конструктивно-продукційних моделей часових рядів із урахуванням їх фрактальних властивостей.

Методи дослідження. Для досягнення поставленої мети та вирішення визначених задач у дисертаційній роботі використано комплекс загальнонаукових, спеціальних і експериментальних методів дослідження. До загальнонаукових методів належать аналіз і синтез, який застосовувався при дослідженні існуючих підходів до моделювання та аналізу часових рядів, узагальнення та систематизація, що використовувалася для формування структури процесу відновлення параметрів конструктивно-продукційних моделей, а також порівняння, яке застосовувалося для оцінювання ефективності запропонованого підходу відносно існуючих рішень.

До спеціальних методів відносяться конструктивно-продукційне моделювання, що забезпечує формалізацію та опис фрактальних часових рядів і розробку відповідних моделей; методи аналізу фрактальних властивостей часових рядів для врахування самоподібності та складної структури залежностей при відновленні параметрів; ітеративні методи оптимізації, зокрема генетичні алгоритми, для відновлення параметрів конструктивно-продукційних моделей; об’єктно-орієнтований аналіз для визначення функціональних компонентів і структурної організації програмної системи; підхід Domain-Driven Design (DDD) для моделювання предметної області та узгодження її з архітектурою програмного забезпечення; мультиагентний та мікросервісний підходи для побудови масштабованої та модульної архітектури програмного додатку; а також методи аналізу продуктивності, включаючи оцінку швидкодії, навантаження та масштабованості, що дозволяють оцінити ефективність реалізації програмної системи.

До експериментальних методів належать комп’ютерне моделювання та обчислювальні експерименти, що забезпечують оцінку точності відновлення параметрів моделей на синтетичних детермінованих і стохастичних рядах. Тестування

на реальних часових рядах проводилося для оцінювання якості відновлених моделей та прогнозів за допомогою стандартних метрик: середньої абсолютної помилки (MAE), середньоквадратичної помилки (MSE), кореня середньоквадратичної помилки (RMSE), середньої абсолютної відносної помилки (MAPE), симетричної середньої абсолютної відносної помилки (SMAPE), коефіцієнта детермінації (R^2), а також кореляційних показників Пірсона та Спірмена для аналізу ступеня лінійної та рангової залежності між відновленими та фактичними значеннями.

Наукова новизна отриманих результатів.

Наукова новизна роботи полягає у розв'язанні задачі відновлення параметрів конструктивно-продукційних моделей для часових рядів різної природи. В результаті виконаних досліджень одержано такі наукові результати.

Вперше:

- розроблено методи відновлення конструктивних моделей та їх відповідних параметрів для синтетичних детермінованих рядів, що на відміну від існуючих засобів, дозволяє відновити фрактальну структуру вже існуючого синтетичного ряду;
- запропоновано ітеративний підхід до відновлення параметрів конструктивних моделей стохастичних синтетичних та реальних часових рядів шляхом їх розбиття на складові. На відміну від традиційних методів , розроблений підхід дозволяє отримати структурну математичну модель та відповідний набір параметрів, на основі якого визначаються існуючі закономірності змін значень сладових часового ряду відповідно до його фрактальних властивостей;
- формалізовано комплекс спеціалізованих конструкторів, характер їхньої взаємодії та систему відповідних алгоритмів. На відміну від існуючих методів конструктивізму, які орієнтовані виключно на пряме виведення (генерацію), запропонована формалізація дозволяє повноцінно представити обернений процес моделювання – відтворення моделі на основі аналізу існуючих значень.

Удосконалено:

- удосконалено метод оцінки близькості (фітнес-функції) для стохастичних часових рядів у процесі відновлення параметрів конструктивних моделей, який,

на відміну від традиційного попарного порівняння, базується на перехресному зіставленні ансамблів згенерованих та еталонних вхідних реалізацій. Використання критерію суми найменших відхилень між цими множинами дозволяє коректно врахувати нескінченну варіативність форм, зумовлену нормальним розподілом, що суттєво підвищує точність та керованість процесу реконструкції;

- архітектурний підхід до програмної реалізації ітеративних процесів та генетичного алгоритму. На відміну від класичних монолітних підходів, у роботі запропоновано та реалізовано механізм ізольованого масштабування на рівні окремих складових операцій відповідно до міри їхньої обчислювальної складності.

Отримали подальший розвиток:

- метод кодування складних хромосомних структур у генетичному алгоритмі, що дозволило поєднати в межах однієї популяції символьну частину (правила підстановки L-системи) та математичні параметри;
- базові генетичні оператори (кросинговер, мутація та селекція), які було адаптовано для коректної обробки специфічної гібридної структури хромосом без втрати фізичного змісту параметрів;
- архітектурні засоби генетичного пошуку, шляхом їх інтеграції як базових модулів (конструкторів) у загальну систему моделювання;
- апарат формальних граматик, який було розширено для опису взаємодії фрактальних компонентів та закономірностей між значеннями часового ряду.

Практичне значення отриманих результатів полягає у розробці та впровадженні високоефективного програмно-алгоритмічного комплексу для автоматизації процесів ідентифікації та відновлення параметрів конструктивних моделей часових рядів різної природи.

Основні результати практичного спрямування:

- розширення сфери застосування конструктивізму: доведено можливість ефективного застосування конструктивно-продукційного моделювання не лише для прямого генерування фрактальних рядів, а й для розв'язання оберненої задачі – відновлення породжуючої моделі (аксіом та правил L-систем) та набору параметрів на основі аналізу вже існуючих реальних дискретних спостережень;

- створення інструментарію для аналізу: розроблений підхід дозволяє використовувати отримані L-системи для глибинного структурного аналізу складних процесів, а також здійснювати довгострокове прогнозування майбутніх станів шляхом генераційного продовження часового ряду;
- підвищення обчислювальної ефективності: запропонована безсерверна хмарна архітектура з ізольованим масштабуванням дозволяє суттєво скоротити витрати часу та обчислювальних ресурсів при обробці великих масивів стохастичних даних.

Результати роботи безпосередньо використані та впроваджені під час виконання таких науково-дослідних робіт:

- «Підвищення конкурентоспроможності залізничного транспорту на основі уніфікованих інтелектуальних технологій процесів перевезень та експлуатації парків технічних систем» (№ держреєстрації 0117U004392) – у частині аналізу та прогнозування часових рядів експлуатаційних показників залізничної інфраструктури з використанням розроблених еволюційних алгоритмів;
- «Конструктивно-продукційне моделювання фракталів» (№ держреєстрації 0118U004215) – у частині алгоритмізації оберненого процесу моделювання фрактальних структур та розширення апарату формальних граматик для опису стохастичних компонентів.

Особистий внесок здобувача. Отримані результати та проміжковий прогрес досліджень був висвітлений у наступних роботах:

[115] – тестування та верифікація результатів;

[111] – проведення дослідження з фрактальних часових рядів;

[109] – розробка частини програми, обговорення результатів;

[110] – планування та проведення дослідження;

[107] – проведено комплексний аналіз існуючих методів та моделей для роботи з часовими рядами на основі їх структурних особливостей;

[76, 100] – визначення процесу відновлення на основі синтетичних детермінованих часових рядів та розробка відповідного програмного додатку для автоматизації експерементального процесу;

[99] – структуризація основних засад процесу відновлення на приклады синтетичних детермінованих часових рядів;

[103] – дослідження ефективності використання відновленної конструктивної моделі та її параметрів для пердбачення подальших значень із використанням синтетичних часових рядів.

[76, 101, 108] – аналіз та визначення підходів до реалізації, написання програмного дотатку та розширення мульти-агентного підходу для роботи зі синтетичними стохастичними часовими рядами.

[77, 102, 104, 105, 106] – визначення процесу роботи з реальними наборами даних та мікро-сервісна реалізація програмного додатку із вирішенням проблем синхронізації.

Окрім наукових праць під час роботи над дослідженнями були отримані наступні свідоцтва авторського права на комп‘ютерні програми – [112, 113, 114].

Апробація результатів дисертації.

Результати роботи щодо визначення методу та його компонентів були представлені на міжнародних і всеукраїнських наукових конференціях:

- «Теоретичні та прикладні аспекти побудови програмних систем» (TAAPSD) (м. Київ, 2017);
- Міжнародна наукова конференція «Інтелектуальні системи прийняття рішень і проблеми обчислювального інтелекту» (м. Херсон, 2018);
- Всеукраїнська науково-методична конференція «Проблеми математичного моделювання» (м. Кам’янське, 2018);
- IEEE 15th, 19th International Conference on Computer Sciences and Information Technologies (CSIT) (2020, 2024);
- Міжнародна науково-практична конференція «Сучасні інформаційні та комунікаційні технології на транспорті, в промисловості та освіті» (м. Дніпро, 2018, 2022, 2023, 2024, 2025);
- Міжнародній науково-технічній конференції «Інформаційні технології в металургії та машинобудуванні ІТММ» (м. Дніпро, 2023, 2024, 2025).

Публікації. Результати, отримані під час досліджень у рамках дисертаційної роботи, були опубліковані в 15 наукових працях.

У фахових та рекомендованих Міністерством освіти і науки України для публікації результатів дисертації – 3.

Матеріали міжнародних конференцій, що індексуються НМБД Scopus – 3.

У тезах доповідей міжнародних та всеукраїнських конференцій – 7.

Структура та обсяг дисертації. Дисертаційна робота складається з вступу, 4 розділів, висновків, списку використаних джерел і додатків. Загальний обсяг дисертації становить 151 сторінок, у тому числі 116 сторінок основної частини, 39 рисунків, 12 таблиць, 2 додатки на 5 сторінках та список використаних джерел із 115 найменувань на 11 сторінках.

РОЗДІЛ 1
ОГЛЯД ІСНУЮЧИХ ПІДХОДІВ ДЛЯ АНАЛІЗУ ТА ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ РІЗНОЇ ПРИРОДИ

1.1 Загальна класифікація сучасних методів для аналізу часових рядів

Часові ряди є одним із найпоширеніших способів опису зміни стану складних систем у часі та становлять фундаментальний інструмент для аналізу та інтерпретації результатів спостережень. Вони дають змогу відстежувати розвиток процесів поведінки, визначати їхні характерні властивості та, за наявності достатнього обсягу даних, оцінювати ймовірні тенденції та можливі майбутні значення. Завдяки універсальності такого підходу часові ряди широко застосовуються у фінансових, економічних, технічних, природничих і медичних дослідженнях, де важливо отримати цілісне уявлення про поведінку системи у динаміці.

Із розвитком математичного апарату та сучасних інформаційних технологій суттєво розширився спектр методів обробки часових рядів, що включає їх аналіз, моделювання та прогнозування [91]. Кожен з методів оперує відповідно до власного базису інтерпретації та визначеної області застосування, що зумовлює потребу у виборі адекватного підходу залежно від вхідних даних.

Традиційні математичні методи зазвичай поділяють на лінійні та нелінійні моделі [53]. Перші виходять з припущення, що зв’язки між елементами ряду можуть бути представлені через лінійні комбінації попередніх значень [84]. Підходи даного типу забезпечують простоту реалізації та використання, проте виявляються недостатньо ефективними у випадках, коли дані містять складні, нелінійні залежності.

Нелінійні моделі у свою чергу дають змогу враховувати складну внутрішню структуру часових рядів, включно з ефектами пам’яті, взаємозв’язками на різних часових масштабах і фрактальними характеристиками [39]. Це дозволяє значно точніше описувати реальні процеси, хоча такі методи є обчислювально складнішими та мають потребу у параметризації та подальшій інтерпретації отриманих результатів.

Поряд із класичними підходами в наукових дослідженнях зростає інтерес до альтернативних, нетипових методів аналізу часових рядів, які виходять за межі традиційних статистичних і машинних моделей. Одним із таких напрямів є конструктивно-продукційне моделювання [78], що спирається на генеративні підходи та дозволяє формувати часові ряди різних структур і властивостей [70]. Іншим важливим прикладом є рекурентний аналіз [10], який дає змогу досліджувати приховані закономірності в нелінійних і хаотичних системах через реконструкцію фазового простору та аналізу повторюваних станів, що забезпечує глибше розуміння внутрішньої організації ряду та сприяє побудові точніших моделей або виявленню критичних змін у динаміці системи.

1.2 Класичні лінійні методі прогнозування та аналізу часових рядів

Лінійні методи аналізу часових рядів ґрунтуються на припущенні, що поточні значення можуть бути представлені як лінійна комбінація попередніх спостережень. Це означає, що структура ряду розглядається як така, що визначається лінійними співвідношеннями між її теперішніми і минулими значеннями. Подібне припущення значно спрощує моделювання та обчислення, що, своєю чергою, забезпечує використання лінійних методів у якості базової основи або складової частини більш складних моделей. До таких фундаментальних підходів належать авторегресійні моделі (AR) та моделі ковзного середнього (MA) [32, 25].

Авторегресійна модель (AR) формально описує залежність поточного значення ряду як лінійну комбінацію кількох попередніх значень. Це дозволяє визначити короткострокові закономірності, напрямки руху та локальні тенденції у даних. AR-моделі добре працюють у випадках, коли значення ряду мають виражені внутрішні взаємозв’язки, а залежності між сусідніми точками є статистично значущими.

Модель ковзного середнього (MA) натомість орієнтована не на самі значення ряду, а на попередні помилки моделювання або випадкові коливання. Вона формує результат як лінійну комбінацію попередніх залишкових значень (помилок), що дозволяє згладжувати шуми і зменшувати вплив випадкових викидів значень.

Таким чином, MA-моделі є ефективними у випадках, коли на значення ряду суттєво впливають стохастичні фактори.

Ще одним важливим лінійним методом є експоненційне згладжування, яке застосовується для рядів із трендами та сезонними компонентами. Його ключовою особливістю є надання визначального значення останнім спостереженням, що дозволяє швидко реагувати на зміни динаміки. Існують кілька варіантів цього методу – просте, подвійне та потрійне згладжування – кожен із яких розширює можливість моделі враховувати тренди та сезонність. Через свою простоту та гнучкість експоненційне згладжування широко використовується у практичних задачах, особливо коли потрібні оперативні результати без складного калібрування.

Для системного поєднання ідей авторегресії, інтегрування та ковзного середнього використовуються моделі ARIMA (авторегресійна інтегрована модель ковзного середнього) [40], що є узагальненням моделі ARMA [16]. Модель ARIMA включає три ключові компоненти:

- AR (авторегресія) – описує залежність поточного значення від попередніх;
- I (інтегрування) – забезпечує приведення ряду до стаціонарної форми шляхом різницювання;
- MA (ковзне середнє) – враховує випадкові коливання помилок моделі.

Поєднання цих елементів робить ARIMA достатньо гнучким інструментом для роботи з широким спектром часових рядів. Однак одне з обмежень ARIMA полягає в тому, що модель не враховує сезонні коливання.

Для вирішення цієї проблеми використовується SARIMA (сезонна ARIMA) [58], яка розширює ARIMA за рахунок додаткових сезонних компонентів. SARIMA дозволяє моделі аналізувати як короткострокові закономірності, так і повторювані циклічні структури, характерні для певних часових інтервалів, наприклад, місяця, квартала або року. Завдяки цьому SARIMA більш точно відображає сезонні зміни та закономірності порівняно з ARIMA [19, 56].

У практичному застосуванні ARIMA зазвичай демонструє високу ефективність у роботі з часовими рядами без явно вираженої сезонності або при аналізі коротких часових інтервалів [41, 18]. У випадках, коли дані мають сезонний хара-

ктер, SARIMA забезпечує значно точніше відображення циклічних коливань за рахунок урахування багаторівневої сезонної структури.

1.3 Нелінійні методі прогнозування значень часових рядів

Нелінійні методи аналізу часових рядів становлять окремий важливий напрям сучасної аналітики даних, оскільки вони ґрунтуються не лише на лінійних зв'язках між значеннями часового ряду, але й враховують складніші, багаторівневі та нетривіальні залежності. Це дозволяє описувати ряди з більш динамічною структурою, у яких поведінка змінних визначається не просто сукупністю попередніх значень, а значно ширшим спектром взаємодій.

До класичних представників нелінійних підходів належать моделі ARCH та GARCH [50], які базуються на припущенні, що дисперсія часового ряду на кожному проміжку не є сталою величиною, а змінюється залежно від набору параметрів. Ключова відмінність моделі GARCH полягає у тому, що вона враховує не лише вплив попередніх помилок, але й залежність від попередніх значень дисперсії, що забезпечує більш гнучку формалізацію процесів зі змінною волатильністю. Завдяки цій властивості ARCH та GARCH залишаються одними з найбільш поширених підходів у аналізі часових рядів із високою мінливістю, зокрема у фінансовій сфері, де їх результати нерідко порівнюють з ARIMA-моделями [80].

Перевагою нелінійних підходів є їх універсальність та здатність враховувати складні залежності, які залишаються недоступними для лінійних моделей. Водночас їхнім недоліком виступає підвищена обчислювальна складність і необхідність ретельного налаштування параметрів, що може ускладнювати практичне застосування без достатньої експертної підтримки.

1.4 Методи штучного інтелекту для прогнозування значень часових рядів

З розвитком інформаційних технологій, методів штучного інтелекту та синтетичних нейронних мереж у задачах прогнозування часових рядів дедалі частіше застосовують саме ці підходи. Їхня популярність зумовлена низкою суттєвих переваг порівняно з класичними лінійними та нелінійними статистичними методами. Зокрема, моделі штучного інтелекту вирізняються високою гнучкістю та зда-

тністю адаптуватися до складних структур і взаємозв’язків у даних, автоматизують процес навчання та, як правило, забезпечують вищу точність у задачах аналізу послідовностей. Такі властивості роблять їх ефективним інструментом для роботи з динамічними системами, поведінка яких не завжди піддається опису традиційними методами.

Водночас одним із найбільш істотних недоліків цих моделей є необхідність проходження фази навчання, що потребує попередньої підготовки й перетворення вхідних даних з метою підвищення ефективності обчислень. Після цього починається процес навчання моделі на основі наявних фрагментів часового ряду. Зазвичай процес організовано у вигляді поетапної подачі фіксованих вікон даних разом з очікуваним результатом. На кожному з таких кроків мережа формує й коригує внутрішні зв’язки між елементами, поступово визначаючи оптимальні приховані параметри. Процес повторюється протягом кількох епох, доки модель не досягає збалансованого рівня узгодженості між даними та власними параметрами. Важливо забезпечити достатню кількість навчальних спостережень і визначити оптимальну кількість епох, щоб уникнути як недонавчання, так і перенавчання нейронної мережі.

Найпоширенішою архітектурою нейронних мереж для прогнозування подальших значень часових рядів є рекурентні нейронні мережі (Recurrent Neural Networks, RNN) [61]. Їхня ключова особливість полягає у наявності внутрішньої пам’яті, яка формується та вдосконалюється в процесі навчання. Завдяки цьому інформація з попередніх кроків передається далі в часі, що дає змогу моделі зберігати та використовувати залежності між значеннями ряду на різних часових масштабах. Така властивість робить RNN ефективним інструментом для роботи з послідовними даними, де порядок та контекст мають вирішальне значення.

Попри свою гнучкість і здатність моделювати складні залежності, класичні RNN можуть стикатися з проблемами затухання або вибуху градієнта при роботі з довгими послідовностями. Саме ці обмеження зумовили появу модифікованих архітектур, спрямованих на стабілізацію процесу навчання та підвищення якості результатів.

Завдяки своїм властивостям RNN та їх варіації знаходять широке застосування не лише у прогнозуванні фінансових та економічних часових рядів [81], але й у задачах обробки природної мови [62], аналізі біомедичних сигналів [69], дослідженнях енергетичних систем [8] та опрацюванні метеорологічних даних [52]. Завдяки здатності працювати з послідовностями різної природи, ці моделі успішно описують складні нелінійні залежності, які важко піддаються аналізу класичними підходами.

1.4.1 Модель Long Short-Term Memory

Серед різних варіантів рекурентних нейронних мереж найбільш поширеною є архітектура LSTM (Long Short-Term Memory). Її ключовою особливістю виступає наявність каналу довготривалої пам’яті, який здатний поетапно накопичувати інформацію, а також системи з трьох типів вентилів, що регулюють потік даних до каналу і з нього. Кожен із цих вентилів виконує окрему функцію та забезпечує контроль над збереженням, оновленням і передаванням інформації, зокрема:

- вхідний вентиль визначає, які нові фрагменти інформації потрібно додати до поточного стану каналу, забезпечуючи фіксацію релевантних даних з поточного кроку;
- вентиль забуття вирішує, які елементи інформації необхідно видалити, порівнюючи попередній стан із поточним вхідним рядом, що дозволяє запобігати накопиченню надлишкових або застарілих даних;
- вихідний вентиль контролює, які частини поточного стану пам’яті будуть передані на вихід, інтегруючи інформацію як із попередніх станів, так і з нового вхідного ряду, тим самим забезпечуючи коректне поширення інформації на наступні кроки.

Завдяки цим механізмам LSTM ефективно працює з часовими рядами, що характеризуються довготривалими залежностями та складними закономірностями між значеннями. Типовим прикладом застосування є аналіз змін вартості акцій [79], де важливо враховувати взаємозв’язки на значних часових інтервалах.

Разом із тим, LSTM має низку обмежень. До основних недоліків відносять схильність до перевантаження, суттєві обчислювальні витрати та високу чутливість до налаштувань гіперпараметрів, що потребує ретельного вибору параметрів навчання [94].

1.4.2 Моделі трансформери для аналізу часових рядів

У 2020-х роках значної популярності набули моделі трансформери, які стали домінуючим архітектурним підходом у сучасному машинному навчанні [86]. Головною відмінністю трансформерів від класичних RNN-підходів є використання механізму самоуваги (self-attention). Принцип його роботи полягає у одночасному аналізі всього набору вхідних даних без необхідності їх послідовної обробки. Для кожного елемента обчислюється оцінка його впливу на всі інші елементи, що дозволяє моделі захоплювати як локальні, так і глобальні залежності в ряді даних.

Архітектура трансформерів зробила революцію у сфері обробки природної мови (NLP) та генеративних моделей [82]. На її основі було також розроблено підходи для аналізу часових рядів. Одним із таких рішень є Autoformer [92], який поєднує принципи декомпозиції ряду та обчислення автокореляції для виявлення закономірностей у часовій послідовності.

Процес роботи Autoformer складається з кількох етапів. На першому етапі вхідний часовий ряд розкладається на компоненти, що відображають тренд та сезонність, із використанням ковзного середнього. Наступним кроком обчислюється показник автокореляції між отриманими компонентами, що дозволяє моделі визначити часові затримки та взаємозв’язки між значеннями ряду. Використовуючи ці виявлені закономірності, Autoformer формує прогноз подальших значень, одночасно враховуючи довгострокові та короткострокові залежності.

Завдяки такому підходу трансформерні моделі здатні ефективно працювати з часовими рядами, де існують складні та багаторівневі взаємозв’язки, що важко моделювати класичними RNN-архітектурами.

1.4.3 Гібридні моделі нейронних мереж

Сучасні моделі машинного навчання, подібно до класичних статистичних підходів, активно розширюються шляхом гібридизації архітектур та інтеграції додаткових компонентів, що підвищують ефективність і точність моделювання. На прикладі моделей типу LSTM можна виділити два основні підходи до гібридизації: (1) розширення за допомогою додаткових статистичних операцій та (2) повне поєднання з іншими сучасними ML-моделями.

Перший підхід реалізується, наприклад, у моделі LSTM-ARIMA [33]. Як випливає з назви, модель поєднує роботу двох компонентів: ARIMA відповідає за побудову прогнозу лінійної частини часового ряду, а різниця між фактичними даними та прогнозом передається до LSTM для моделювання нелінійних компонентів. Такий підхід дозволяє ефективно комбінувати переваги класичних статистичних методів і рекурентних нейронних мереж.

Другий підхід передбачає повне гібридне поєднання LSTM з іншими сучасними моделями машинного навчання, такими як Autoformer-LSTM [88] та CNN-LSTM [96].

У моделі Autoformer-LSTM перший етап включає декомпозицію часового ряду та обчислення автокореляції для виявлення довгострокових залежностей. На основі отриманих закономірностей Autoformer формує довгостроковий прогноз, після чого LSTM аналізує короткострокові залежності та фіналізує прогноз, забезпечуючи точне відтворення як глобальних, так і локальних патернів.

У моделі CNN-LSTM спочатку застосовується CNN-шар для виділення локальних ознак та зменшення шуму у вхідних даних. Далі LSTM-шар аналізує часові залежності між отриманими ознаками, що дозволяє моделі одночасно враховувати просторові та тимчасові закономірності ряду.

1.5 Рекурентний аналіз та використання рекурентних діаграм

Рекурентний аналіз є методом дослідження динаміки часових рядів, який ґрунтується на ідеї повторюваності станів системи. Основна мета цього підходу полягає у визначенні моментів часу, коли система повертається до стану, близько-

го до попереднього. Такий аналіз дозволяє виявляти закономірності, періодичність, хаотичну поведінку та структурні зміни в ряді.

Для візуального дослідження повторюваних станів застосовуються рекурентні діаграми (Recurrence Plots, RP). Вони наочно відображають моменти, коли значення ряду наближаються до попередніх станів. На діаграмі чорні точки позначають наявність повторюваного стану, а білі – його відсутність (рис. 1.1). Завдяки цьому можна легко виділяти періодичні патерни, повторювані закономірності, структурні зміни, а також виявляти хаотичні елементи або аномалії у динаміці ряду.

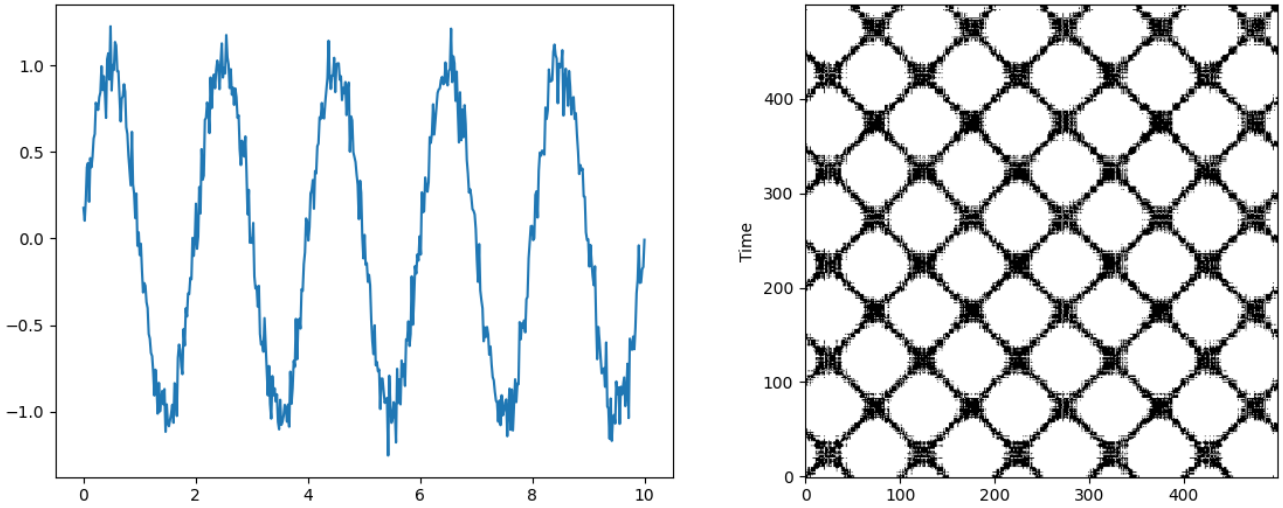


Рисунок 1.1 – Графік часового ряду та відповідна йому рекурентна діаграма

Окрім візуального аналізу, на основі рекурентних діаграм розроблені кількісні показники, що дозволяють більш детально оцінювати поведінку часового ряду [35]. До них належать:

- частка точок, що відображає загальну повторюваність;
- відсоток точок, що формують діагональні лінії, які свідчать про регулярність поведінки;
- вертикальні лінії, що інформують про застійні або «заблоковані» стани.

Додатково спеціальні метрики дозволяють оцінювати складність і непередбачуваність ряду [10, 67].

Рекурентний аналіз та діаграми особливо ефективні при дослідженні складних, нелінійних та фрактальних часових рядів. До таких відносять фінансові індекси, кліматичні та метеорологічні дані, біомедичні сигнали, енергетичні та геофі-

зичні ряди. Використання цього підходу дозволяє оцінити структуру ряду, виділити закономірності, виявити довгострокові залежності та підготувати дані для подальшого моделювання або аналізу.

Таким чином, рекурентний аналіз у поєднанні з рекурентними діаграмами є потужним інструментом для вивчення динаміки часових рядів, особливо в тих випадках, коли класичні лінійні або прості статистичні методи виявляються недостатньо ефективними.

1.6 Засоби аналізу та прогнозування фрактальних часових рядів

У межах дослідження нелінійних підходів до аналізу часових рядів важливе місце займає вивчення їхніх фрактальних характеристик. Фрактальність відкриває можливості для опису складних властивостей рядів даних, зокрема їх багаторівневої організації, внутрішньої структурної складності та наявності нетривіальних залежностей між значеннями, що розташовані на значній відстані одне від одного у часовій послідовності [3, 97, 98]. Такий підхід дозволяє не лише глибше зрозуміти природу формування часових рядів, але й виявити закономірності, які залишаються прихованими за межами класичних лінійних моделей.

У контексті фрактального аналізу зазвичай виокремлюють три основні характеристики, що визначають фрактальну природу часового ряду:

- самоподібність – здатність часового ряду зберігати характерний патерн незалежно від масштабу розгляду;
- фрактальна розмірність – величина, що може набувати дробових значень і характеризує ступінь структурної складності;
- довготривала залежність (Long-Range Dependence, LRD) – властивість, за якої вплив попередніх значень зберігається навіть на великих часових інтервалах.

1.6.1 Визначення фрактальної розмірності ряду на основі показника Герста

Одним із основних параметрів, за допомогою яких визначається ступінь фрактальності часового ряду, є показник Герста (або експонента Герста) [66]. Цей показник відображає рівень довготривалої пам’яті процесу, представленого часо-

вим рядом, тобто вплив минулих станів на поведінку за поточні та майбутні метрики. Експонента Герста дозволяє кількісно оцінити, наскільки зміни в часовому ряді залежать від попередніх значень, а також визначити наявність і силу трендових або зворотних зв'язків. Вона тісно пов'язана з автокореляційними властивостями часового ряду та швидкістю зміни його значень у часі.

Показник Герста визначається на основі асимптотичної поведінки діапазону зміненого масштабу як функції часового проміжку часового ряду [63]. Відповідний математичний вираз має вигляд:

$$\mathbb{E}\left[\frac{R(n)}{S(n)}\right] = C_{n^H}, n \to \infty, \quad (1)$$

де R(n) – діапазон перших n накопичених відхилень від середнього значення, S(n) – це сума перших n стандартних відхилень, $\mathbb{E}[x]$ – математичне очікування, n – діапазон часового проміжку спостережень, C – константне значення.

Відповідно до отриманого значення, проводиться класифікація типу залежності між значеннями часовго ряду. Показник у межах $0 < H < 0{,}5$ свідчить про антиперсистентність та тенденцію до повернення до середнього значення. При H дорівнює 0,5 можна стверджувати, що ряд не має довготривалих зв'язків та здебільшого є хаотичним без явного тренду. При значеннях $0{,}5 < H < 1$ демонструється персистентність та тренд до збереження напрямку подальших значень ряду.

На основі значення експонента Герста можливе подальше обчислення показника фрактальної розмірності за Гаусдорфом [85] за допомогою наступного математичного виразу:

$$D = 2 - H, \quad (2)$$

де D – значення фрактальної розмірності. Відповідно до можливих коливань параметра Герста, чим вищим є його значення, тим більш плавним є тренд та тим меньшим є шумність та волатильність структури ряду.

1.6.2 Нелінійні моделі прогнозування та обробки фрактальними часовими рядами

Показник Герста широко використовується у нелінійних мультифрактальних моделях часових рядів, таких як MMAR (Multifractal Model of Asset Returns) та MFDFA (Multifractal Detrended Fluctuation Analysis) [45], а також у фрактальних

модифікаціях класичних моделей, зокрема у ARFIMA (Autoregressive Fractionally Integrated Moving Average) [47]. Використання цього показника дозволяє більш точно описувати складні нелінійні процеси, що характеризуються наявністю довготривалих залежностей і фрактальної структури.

Сфера застосування фрактальних моделей є надзвичайно широкою, оскільки вони ефективно працюють із даними, що мають складну структуру, нестабільну динаміку та ознаки довготривалої пам'яті. Найчастіше такі методи використовують для аналізу фінансових часових рядів [27], кліматичних і метеорологічних показників [44], біомедичних сигналів [7], даних енергетичних та телекомунікаційних систем [15, 25], а також у геофізичних і сейсмологічних дослідженнях.

1.6.3 Використання фрактальних складових часових рядів

У роботах [12, 13, 36] розглядаються різні підходи до класифікації та аналізу фрактальних часових рядів.

Застосування мультифрактального аналізу дозволяє виявляти приховані закономірності, що недоступні для традиційних лінійних чи стаціонарних моделей, і забезпечує ефективне моделювання, класифікацію та прогнозування часових рядів із нестандартною динамікою. Такий підхід дозволяє глибше зрозуміти внутрішні закономірності поведінкових та соціальних процесів у цифрових мережах і створює основу для подальшого застосування алгоритмів машинного навчання у комплексному аналізі часових рядів [37].

Враховуючи специфічні властивості фрактальних часових рядів, зокрема їх самоподібність, багатомасштабність та довготривалі кореляції, у сучасних дослідженнях активно розробляються моделі, здатні ефективно опрацьовувати такі структури. Одним із прикладів підходів, які інтегрують фрактальні характеристики у процес обробки даних, є модель FractalNet-LSTM [68]. Вона поєднує фрактальні згорткові структури для виділення самоподібних патернів і рекурентний механізм LSTM, що забезпечує аналіз часової послідовності цих ознак.

Процес обробки даних у цій моделі включає два основні етапи.

Фрактальні шари – вхідні дані проходять через кілька рівнів фрактальної згортки, що дає змогу виділити складні структурні патерни та ознаки самоподібності, які важко виявити традиційними методами.

LSTM-шар – отримані ознаки передаються до рекурентної мережі LSTM, яка моделює часові залежності між послідовними значеннями для прогнозування подальших станів ряду.

Водночас у випадку рядів із високою волатильністю, зокрема фінансових показників, експериментальні результати показали, що модель потребує додаткового доопрацювання [68]. Це свідчить про те, що хоча комбінування фрактальних та рекурентних механізмів забезпечує суттєві переваги при моделюванні складних структурних залежностей, адаптація до швидкозмінних динамік може вимагати розширення або модифікації архітектури.

1.7 Конструктивно-продукційне моделювання складових часових рядів

Методи роботи з часовими рядами, орієнтовані виключно на прогнозування майбутніх значень, мають важливе обмеження – відсутність описувальної моделі як частини результату. Більшість таких підходів зосереджується на генерації наступних точок ряду, не забезпечуючи формалізованого представлення внутрішньої структури, що визначає поведінку даних.

У результаті формується лише числовий прогноз без можливості глибокого аналізу часових рядів. Відсутність інтерпретованої моделі обмежує дослідження патернів поведінки значень, оцінку стабільності динаміки, виявлення ключових чинників варіативності та виконання структурної діагностики. Таким чином, зазначені методи виконують функцію прогнозування, а не реконструкції або пояснення поведінки часових даних.

Методи конструктивно-продукційного моделювання забезпечують формалізований опис структур і алгоритмів, що лежать в основі моделі та її параметрів. Такий опис слугує структурною нотацією часового ряду та дозволяє інтерпретувати його внутрішню організацію [78, 111]. Подібний підхід при вирішенні прик-

ладних задач був розглянутий у роботах [70, 109, 110]. Відповідна програмна автоматизація процесу моделювання розглядається у роботі [72].

На основі визначених конструкторів та відповідних моделей реалізовано підходи до моделювання часових рядів різної структури. Залежно від набору та значень вхідних параметрів формальні структури забезпечують генерацію як детермінованих (рис. 1.3), так і стохастичних рядів (рис. 1.2).

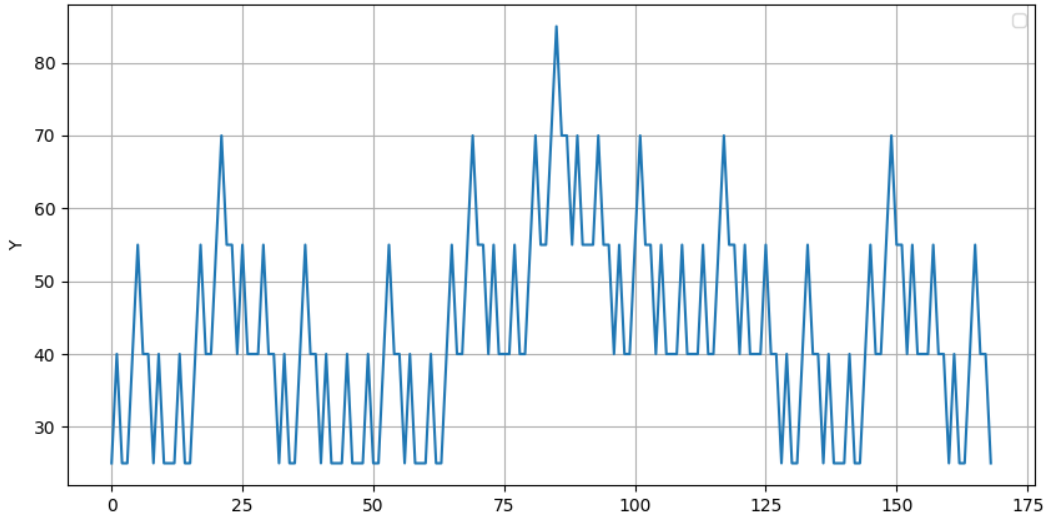


Рисунок 1.2 – Синтетичний детермінований часовий ряд

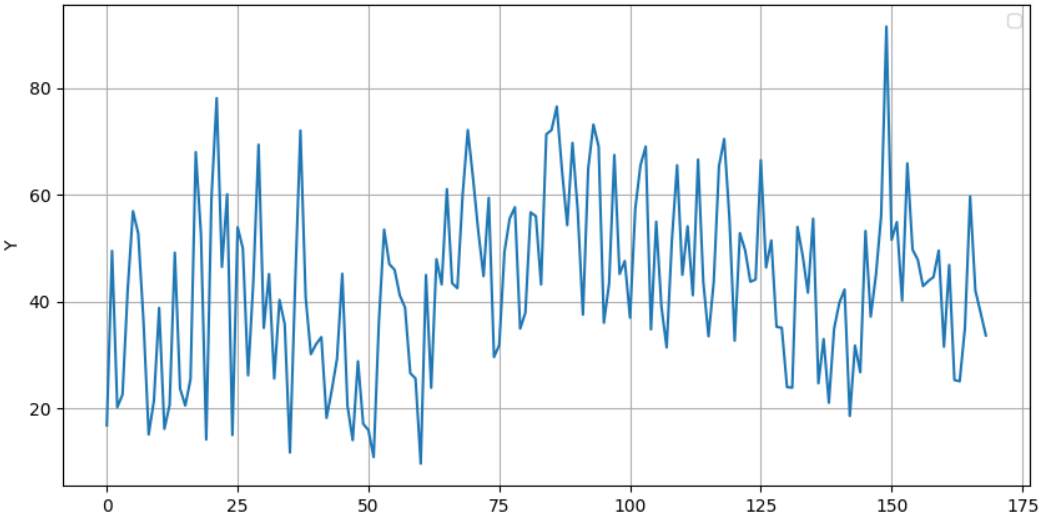


Рисунок 1.3 – Синтетичний стохістичний часовий ряд

Окрім процесу моделювання, у роботі розглянуто зворотний процес – відновлення конструктивної моделі та її параметрів на основі заданого часового ряду із використанням генетичного алгоритму. Для тестування підходу використовувалися синтетичні детерміновані часові ряди різної структурної складності, що дозволило сформувати репрезентативну експериментальну вибірку та оцінити ефективність запропонованого методу.

Використання генетичного алгоритму як складової конструктивізму раніше розглядалося при дослідженні моделювання окремих структур і часових рядів [71]. В межах роботи зі синтетичними часовими рядами було сформовано базис процесу відновлення, який включає:

- визначення структури хромосоми, що відтворює набір параметрів конструктивної моделі;
- опис правил реалізації операцій кросинговеру та мутації відповідно до структурних особливостей хромосоми;
- визначення фітнес-функції та особливостей етапу селекції.

Отримані результати демонструють позитивну динаміку та свідчать про загальну, хоча й частково обмежену, ефективність запропонованого підходу [76]. Водночас встановлено необхідність подальших досліджень із використанням синтетичних стохастичних рядів, адаптації методу до їх аналізу, а також його впровадження для дослідження часових рядів реальної природи.

Окремим напрямом дослідження є розробка підходів до програмної реалізації запропонованого методу, що має забезпечити підвищення часової ефективності обчислювального процесу. На початкових етапах експериментальної частини програмний інструментарій не вирізнявся складною архітектурою, що було зумовлено відносною простотою роботи з детермінованими рядами [76, 99].

Проте при переході до аналізу часових рядів складнішої структури програмна реалізація повинна передбачати можливості масштабування обчислень, зокрема шляхом розширення кількості хромосом у популяції та оптимізації виконання окремих етапів генетичного алгоритму.

Висновки до першого розділу

Широке застосування часових рядів для опису та аналізу складних систем призвело до появи численних підходів роботи з ними. Найпоширенішим завданням у цій сфері є прогнозування подальших значень ряду. Популярність цього завдання забезпечила розвиток великої кількості інструментів різного походження – від класичних математичних моделей до методів машинного навчання. Виходячи

з природи та структури обраного часового ряду, існує можливість використати оптимальний інструмент, який забезпечить максимальну ефективність при вирішенні задачі прогнозування наступних значень.

Втім, одним із основних недоліків зазначених методів є відсутність можливості аналізу внутрішньої структури часового ряду на основі описуючої моделі як частини результатів обробки. Одним із можливих інструментів для такого аналізу є використання рекурентних діаграм, які дозволяють візуалізувати динамічні закономірності, повторювані патерни та переходи між різними станами системи. Водночас застосування рекурентних діаграм не забезпечує повного формалізованого опису моделі та її параметрів, що обмежує можливість кількісної інтерпретації структури ряду та прогнозування на основі виявлених патернів.

Методи конструктивно-продукційного моделювання дозволяють використання їх інструментарію та специфікації для вирішення великого обсягу задач, до яких також входить робота із часовими рядами.

Були представлені необхідні засади для використання даного підходу при моделюванні часових рядів різної структури та характеру змін їх значень як детермінованих, так і стохастичних. Для кожного типу сформовано відповідний набір параметрів конструктивної моделі, що забезпечує генерацію рядів відповідного типу.

На основі представленого процесу були визначені базові правила побудови зворотного процесу, а саме відновлення параметрів конструктивної моделі на основі вхідного часового ряду. Дослідження проводилися з використанням синтетичних детермінованих рядів та генетичного алгоритму як основи для ітеративного процесу відновлення моделі.

Було доведено ефективність підходу при роботі з рядами зазначеної природи. Визначено завдання для подальшого розвитку, зокрема: розширення методу для роботи зі стохастичними моделями, адаптація підходу для цього процесу та впровадження його для аналізу часових рядів реальної природи походження.

Матеріали розділу опубліковані у роботах: [76, 99, 100, 104 107]

РОЗДІЛ 2
КОНСТРУКТИВНО-ПРОДУКЦІЙНЕ ВІДНОВЛЕННЯ МОДЕЛЕЙ ЧАСОВИХ РЯДІВ

У наукових дослідженнях виділяють три основні завдання роботи з часовими рядами: аналіз, моделювання та прогнозування [91]. Аналіз часових рядів передбачає дослідження їхніх статистичних характеристик, виявлення тенденцій, сезонних коливань, а також інших закономірностей. Моделювання спрямоване на побудову математичної інтерпретації процесу, яка дозволяє відтворити його основні властивості та структуру залежностей. Прогнозування є застосуванням побудованих моделей для оцінювання можливих майбутніх значень ряду.

У сучасній практиці методи дослідження часових рядів значно еволюціонували завдяки розвитку математичних підходів та широкому впровадженню інформаційних технологій. Сучасні обчислювальні засоби дозволяють ефективно обробляти значні обсяги даних, автоматизувати процедури аналізу та реалізовувати складні алгоритми моделювання. Це значно розширює можливості дослідників щодо виявлення прихованих закономірностей у часових рядах та побудови більш точних моделей складних процесів.

Однією з важливих характеристик складних часових рядів є наявність фрактальних властивостей. Це проявляється у структурних патернах, форма яких зберігається при розгляді на різних масштабах. Така властивість свідчить про наявність складної внутрішньої структури процесу. Крім того, для фрактальних часових рядів характерна наявність довготривалих залежностей, коли попередні значення ряду можуть впливати на поточні через значні часові інтервали [63].

Конструктивно-продукційний підхід дозволяє працювати не лише із загальними статистичними характеристиками та значеннями часового ряду, але й визначати конструктивні моделі, які безпосередньо описують механізм формування досліджуваних послідовностей [70]. Саме ця особливість відрізняє конструктивізм від класичних методів аналізу та моделювання, які переважно зосереджені на статистичному описі даних.

2.1 Використання конструктивізму для моделювання складових часових рядів

Конструктивно-продукційний підхід був успішно застосований для моделювання фрактальних часових рядів на основі визначених початкових даних [109, 110, 115]. Також, був розглянутий обернений процесс для відновлення набору необхідних параметрів конструктивної моделі на основі заданого часового ряду [76]. Такий підхід дозволяє не лише генерувати ряди із заданими властивостями, а й реконструювати механізм їхнього формування.

Основним поняттям у конструктивно-продукційному моделюванні є конструктор [78]. Під час розробки конструктивної моделі було введено поняття породжуючих конструкторів, призначених для створення набору конструкцій або конструктивних процесів із елементів несучої структури. Робота таких конструкторів визначається початковими умовами, умовами закінчення процесу та правилами підстановки.

Для побудови складніших моделей використовуються мультиконструктори, які об’єднують кілька конструкторів. Вони виконують перетворення від однієї структури до іншої у визначеному порядку, забезпечуючи ієрархічну та масштабовану організацію моделі [109].

Взаємодія між конструкторами реалізується різними способами. По-перше, це може здійснюватися через зовнішнього виконавця, який координує роботу конструкторів. По-друге, один із конструкторів може виконувати роль зовнішнього виконавця щодо інших. У цьому випадку «зовнішній» конструктор запускає алгоритми «внутрішніх», забезпечуючи їх повне функціонування, та повинен мати відповідні операції для управління процесом.

Таким чином, конструктивно-продукційний підхід забезпечує гнучкий та ефективний механізм моделювання фрактальних часових рядів, дозволяючи не лише відтворювати їхню структуру, а й досліджувати процеси їх формування та взаємодії складових елементів моделі.

2.2 Конструктори і мультиконструктори

Поняття узагальненої конструктивно-продукційної структури [73, 74], або узагальненого конструктора (GC), є основою для конструктивно-продукційного моделювання. У його основі 3 визначних параметри, який описують його як:

$$C = \langle M, \Sigma, \Lambda \rangle \tag{2.1}$$

де основні елементи, це:

- M – поновлюваний неоднорідний носій, на якому здійснюється конструювання;
- Σ – сигнатура відносин та пов'язаних з ними операцій, що визначає правила взаємодії елементів конструкції;
- Λ – набір тверджень, що містить інформацію для підтримки процесу конструювання (CSIS), а саме:
    - умови початку та завершення будівництва;
    - правила та обмеження;
    - онтологію та призначення об'єктів [78].

Характеристики конструктора включають:

- поновлюваний гетерогенний носій;
- атрибути елементів, що визначають їхню функціональну роль та властивості;
- проміжні форми, які виникають у процесі конструювання;
- відношення та операції, що описують взаємодію елементів;
- модель виконавця, яка реалізує операції над елементами;
- взаємозв'язок операцій з алгоритмами їх реалізації, що забезпечує формальну коректність процесу.

Онтологія узагальненого конструктора у його неформальному представленні детально розглядається в [73, 74].

Отримані конструкції, сформовані за допомогою конструктора, позначаються як $\Omega(C)$.

Низка перетворень здійснюється з метою побудови: спеціалізація ($_S\!\rightarrow$), інтерпретація ($_I\!\rightarrow$), конкретизація ($_K\!\rightarrow$), реалізація ($_R\!\rightarrow$), як у [70, 73].

Метою породжуючих конструкторів є створення набору структур або конструктивних процесів з елементів носія на основі початкових умов, умов завершення та правил підстановки.

Мульти-конструктори складаються з декількох конструкторів, що виконують перетворення від однієї структури до іншої у заданому порядку. Зв'язок між конструкторами може бути реалізований різними способами: за допомогою зовнішнього виконавця або через один із конструкторів, що діє як зовнішній виконавець стосовно інших. У другому випадку «зовнішній» має запустити алгоритм для повного виконання «внутрішнього» і повинен мати відповідну операцію. У будь-якому разі зовнішній виконавець створює конструктори та розпочинає процес побудови.

2.2.1 Перелік параметрів конструктивної моделі часового ряду

Відповідно до вже зазначеного використання L-системи як основного компоненту для опису структури, параметри конструктивної моделі часового ряду також включають: f – аксіома L-системи, P – визначені правила підстановки, V_f – початкове значення часового ряду та dV_f – його приріст. Визначений алфавіт системи включає термінальні символи, кожному з яких відповідає операція інтерпретації (табл. 2.3) при перетворені фінального мультисимвольного рядка у часовий ряд.

Таблиця 2.2.1 – Опис набору терміналів та їх відповідних операції

Термінальний символ	Відповідна операція
f	Додає нову точку до часового ряду, значення якої дорівнює поточному значенню V_f
+	Збільшує поточне значення параметра V_f відповідно до значення dV_f.
-	Зменшує поточне значення параметра V_f відповідно до значення dV_f.

2.2.2 Конструктори мультиконструктора та їх елементи

Для відновлення конструктивної моделі часового ряду були виділені три ключових конструктора, кожен з яких виконує специфічну функцію:

- C_{MS} – формує фрактальну мультисимвольну послідовність відповідно по зазначених правил;
- C_{TS} – перетворює вищезазначену символьну послідовність у фрактальний часовий ряд відповідно за визначених початкового значення та його перетворення;
- C_{RS} – відновлює похідну модель наданого часового ряду.

Кожен з зазначених конструкторів відповідає за відповідну операцію в рамках мультиконструктора C_M, який у свою чергу має два визначених варіанти виконання: контроль якості перетворення часових рядів у конструктивну модель (*MQ*) та екстраполяція часових рядів для прогнозування (*MF*) [76].

Основною метою режиму *MQ* є визначення процесу верифікації роботи побудованого процесу за рахунок випробувань на синтетичних часових рядах, згенерованих за заздалегідь визначеною конструктивною моделлю. Він включає наступні етапи:

- випадково генерує правило підстановки;
- послідовно ініціює виконання конструкторів C_{MS}, C_{TS} та C_{RS};
- передає параметри та отримує згенеровані структури (правило підстановки, знайдене конструктором C_{RS});
- порівнює дані, передані конструктору C_{MS}, та отримані від C_{RS} правила підстановки, та визначає різницю між результатами;
- неодноразово виконує попередні кроки.

Виконання в режимі *MF* описує загальний процес обробки наданого ряду з генерацією його конструктивної моделі та на її основі продовження його значень на визначений проміжок. Порядок виконання включає наступні етапи:

- отримує часовий ряд;
- послідовно ініціює виконання конструкторів C_{RS}, C_{MS} та C_{TS};
- отримує сформовані структури та часовий ряд з продовженням.

Ключовим елементом, що визначає ітеративну поведінку системи, є генетичний алгоритм. Відповідно до його класичної реалізації з урахуванням необхідних модифікацій відповідно до області використання, процес включає фази: генерацію початкової популяції хромосом, кросинговер та мутацію, оцінку життєздатності кожного представника та подальшу селекцію.

На етапі селекції з кожної хромосоми формується часовий ряд, який порівнюється з вхідним за допомогою МНК. Отримане значення виступає у ролі фітнес-функції, оцінюючи точність відповідності моделі та визначаючи її життєздатність у популяції [76].

Конкретизація параметричних мультисимвольних конструкторів:

$$C_L = \langle M_L, \Sigma_L, \Lambda_L \rangle \ {}_K\!\rightarrow C_{MS}(P, n) = \langle M_{MS}, \Sigma_{MS}, \Lambda_{MS} \rangle, \tag{2.2}$$

де P – правило підстановки, n – кількість терміналів f у вихідному рядку, що був сформований згідно з P, $M_{MS} \supset \{f, +, -\}$, $\Sigma_{MS} = \{\,.\,\}$, "." – операція конкатенації, $\Lambda_{MS} = \Lambda_L \cup \Lambda_1$. Спеціалізація породжуючих конструкторів для L-систем C_L розглядається в [70].

CSIS Λ_1 включає:

- призначення *f*-конструкції – формування рядка з фрактальними структурами – мета конструкції;
- параметр *P* – визначає набір правил підстановки;
- операції над атрибутами – відсутні;
- аксіома – *f*;
- коли кількість терміналів *f* у вихідному рядку задовольняє умову $\geq n$, процес побудови завершується.

Результат інтерпретації формує конструктивну систему, який складається з двох частин: внутрішнього виконавця та конструктора, розглядається у [76].

Наступні конструктори перетворюють рядки у часові ряди:

$$C_{TS}\big(\Omega_i(C_{MS}), V_f, dV_f, m\big) = \langle M_{TS}, \Sigma_{TS}, \Lambda_{TS} \rangle, \tag{2.3}$$

де $\Omega_i(C_{MS})$ – результат реалізації конструктора C_{MS} (рядок фрактальних структур); V_f та dV_f – початкове значення та значення приросту точки побудованого часового ряду, m – загальна кількість точок побудованого часового ряду, $M_{TS} \supset (T \cup N)$,

$T = \{f, v, +, -\}$ – множина терміналів, $N = \{\sigma\}$ – множина нетерміналів, $\Sigma_{TS} = \Xi_{TS} \cup \Phi_{TS}$, $\Xi_{TS} = \{\circ, f\}$, $\Phi_{TS} = \{+, -\}$, $\Lambda_{TS} = \Lambda \cup \Lambda_2$.

Нижче наведено операції над атрибутами:

- $+(c, a, b)$, $-(c, a, b)$ – додавання, віднімання a та b з результатом c;
- $=: (c, a)$ – присвоєння значення a змінній c.

CSIS Λ_2 включає наступне:

- формування статистичного часового ряду $v(t)$ є метою побудови;
- правила підстановки:

  $\{\langle\{\sigma \to f\sigma, \sigma \to v\sigma\}, \{=: \big(v(t), tV_f\big), +(t, t, dt)\}\rangle$

  $\langle\{\sigma \to +\sigma\}, \{+\big(tV_f, tV_f, dV_f\big)\}\rangle$

  $\langle\{\sigma \to -\sigma\}, \{-\big(tV_f, tV_f, dV_f\big)\}\rangle$

  $\langle\{\sigma \to \varepsilon\}, \{\varepsilon\}\rangle\}$
- правило $\langle \sigma \to \varepsilon, \langle \varepsilon \rangle \rangle$ використовується, якщо інші правила незастосовні;
- початкові умови – рядок $\Omega_i(C_{MS})$; початковий час $t = 0$, його крок $dt = 1$, поточне значення $tV_f = V_f$;
- використання порожнього правила є умовою завершення процесу побудови.

Наступні конструктори знаходять правило підстановки для даного детермінованого часового ряду:

$$C_{RS}(TS_{In}, N) = \langle M_{RS}, \Sigma_{RS}, \Lambda_{RS} \rangle, \tag{2.4}$$

де TS_{In} – вектор значень початкового часового ряду розмірності N. M_{RS} включає термінали $T = \{f, +, -, B_i\}$ (B_i – алгоритми, наведені нижче), нетермінали $N = \{\sigma, \alpha, \beta, \gamma\}$ та конструктивні форми, $\Sigma_{TS} = \Xi_{TS} \cup \Phi_{TS}$, $\Xi_{TS} = \{\cdot\}$, $\Phi_{TS} = \{+, -, =\}$, $\Lambda_{RS} = \Lambda \cup \Lambda_3$.

Операції над атрибутами як у C_{TS}, а також наступні:

- порівняння $< (c, a, b)$ – якщо $a \leq b$, то $c = true$, інакше $c = false$;
- порівняння $= (c, a, b)$ та $\downarrow (c, a, b)$ – $c = \min(a, b)$.

Відповідно до використання генетичного алгоритму для побудови ітеративного процесу відновлення, були визначені відповідні термінальні алгоритми, які безпосередньо відповідають за:

- генерація частин хромосоми;
- створення часових рядів на основі властивостей хромосом;
- управління популяцією.

Нижче приведений перелік усіх розроблених алгоритмів з описом принципу їх роботи:

- $B_1|_{f,+,-}^{r,V_f,dV_f}$ – випадковим чином формуються праві частини правил із символами з попередньо визначеного алфавіту $f, +, -$, значення V_f та приріст dV_f точок часового ряду;
- $B_2|_{P,r,V_f,dV_f}^{X,P}$ – випадковим чином генерує хромосому X та додає її до популяції P;
- $B_3|_{||\Rightarrow(C_{MS}),r(X_n)}^{\Omega(C_{MS})}$ – ініціює вивід мулитисимвольної ланцюжка $\Omega(C_{MS})$ через конструктора C_{MS}, передавши йому правило виводу r хромосоми X_n як параметр;
- $B_4|_{||\Rightarrow(C_{TS}),\Omega(C_{MS})}^{\Omega(C_{TS})}$ – ініціює виведення часового ряду конструктором C_{TS} з мультисимвольним ланцюжком $\Omega(C_{MS})$;
- $B_5|_{\Omega(C_{TS}),TS_{In}}^{q(X_n)}$ – визначає показник живучості q хромосоми X_n через розбіжність між джерелом TS_{In} і побудований (за допомогою хромосоми) X_n часовий ряд;
- $B_6|_P^P, B_7|_P^P, B_8|_P^P$ – сортує хромосоми в популяції і видаляє найгіршу з них. Також додають нові хромосоми, які утворилися під час фази схрещування та мутації.

CSIS Λ_3 включає наступне:

- мета – пошук правила підстановки, за якого конструктори C_{MS} та C_{TS} генерують часовий ряд $TS(C_{MS}, C_{TS})$, що є найближчим до TS_{In};
- правила підстановки:

$$\{\langle\langle\sigma \to \alpha\rangle, \langle =: (n,1), =: (k,1), =: (N, N_1)\rangle\rangle,$$
$$\left\langle\begin{matrix}\left\langle\alpha_v \to B_1|_{\{z,+,-\}}^{r,V_z,dV_z} \cdot B_2|_{P,r,V_z,dV_z}^{X_n,P} \cdot \alpha\right\rangle, \\ \langle <(v,n,N), +(n,n,1)\rangle\end{matrix}\right\rangle,$$

$$\left\langle \left\langle \alpha_{\xi} \to \beta \right\rangle, \langle <(\xi, n, N), =:(n, 1), =:(N, N_1)\rangle \right\rangle,$$

$$\left\langle \left\langle \beta_{\theta} \to B_3|^{\Omega(C_{MS})}_{\vDash(C_{MS}),r(X_n)} \cdot B_4|^{\Omega(C_{RS})}_{\vDash(C_{RS}),\Omega(C_{MS})} \cdot B_5|^{q(X_n)}_{\Omega(C_{RS}),TS_{In}} \cdot \beta \right\rangle, \right.$$
$$\left. \langle <(\theta, n, N), +(n, n, 1)\rangle \right\rangle$$

$$\left\langle \begin{array}{c} \langle \gamma_{\eta} \to B_6|^{P}_{P} \cdot B_7|^{P}_{P} \cdot B_8|^{P}_{P} \cdot \alpha \rangle, \\ \left\langle \begin{array}{c} =:(N,N_2),+(k,k,1),=:(a,q(X_1),0),<(b,k,K), \\ c{\downarrow}(\eta,a,b) \end{array} \right\rangle \end{array} \right\rangle$$

$$\langle \langle \gamma \to \varepsilon \rangle, \langle \varepsilon \rangle \rangle\}$$

- обмеження – правило $\langle \gamma \to \varepsilon, \langle \varepsilon \rangle \rangle$ дотримується, якщо інші правила незастосовні;
- відношення $\to_c$ доступне для використання, якщо $c = true$;
- використання порожнього правила є умовою завершення процесу побудови.

У свою чергу, структура хромосоми повністю відтворює набір параметрів конструктивної моделі:

$$X = \langle f, P, V_f, dV_f \rangle, \tag{2.5}$$

Для визначення показника фітнесу кожної з хромосом, використовується класичний підрахунок МНК із використанням значень вхідного ряду та ряду, згенерованного на основі даних поточної хромосоми:

$$q(X_n) = \sum_{k=1}^{K} (TS(C_{MS}, C_{TS})_k - TS_{In_k})^2 \tag{2.6}$$

2.3 Взаємодія конструкторів під час виконання процесу відновлення відповідно до обраного режиму виконання

Відповідно до визначених підходів, процес взаємодії між елементами мультиконструктора відповідно до обраного режиму виконання.

Для режиму MQ, беспосереднє виконання починається з конструкторів C_{MS} та C_{TS} для отримання відповідного фрактального синтетичного часового ряду для подальшого виконання процесу відновлення.

Відповідно до сигнатури конструктора C_{MS}, він отримає відповідні параметри, а саме:

- аксіому f та правила підстановки P_{In} L-системи, при чому структура правої частини правил підстановки відповідно до визначених лімітів довжини;
- обрану довжину n часовго ряду та відповідна кількість термінальних символів f.

Результат виконання являє собою мультисимвольний ланцюжок розгортку L-системи із відповідною довжиною:

$$P_{In} = \{f : "f + f -"\}; m = 4;$$
$$\Omega(\mathrm{C}_{MS}) = \mathrm{C}_{MS}(P_{In}, m) = "f + f - + f + f -".$$

Відповідно, базуючись на значенні $\Omega(\mathrm{C}_{MS})$ та визначених значеннях вхідних параметрів $V_{f_{In}}$ та $dV_{f_{In}}$ вивід алгоритмічного конструктора C_{TS} виглядатиме наступним чином:

$$\Omega(\mathrm{C}_{MS}) = "f + f - + f + f -";\ m = 4;\ V_{f_{In}} = 4;\ dV_{f_{In}} = 2;$$
$$\Omega(\mathrm{C}_{TS}) = \mathrm{C}_{TS}(\Omega(\mathrm{C}_{MS}), V_{f_{In}}, dV_{f_{In}}, m) = [4, 6, 6, 8].$$

Було отримано синтетичний модельний детермінований часовий ряд. Відповідно до обраного рижиму виконання, результат виконання $\Omega(\mathrm{C}_{TS})$ виступатиме у якості вхідного ряду TS_{In} конструктора C_{RS}:

$$TS_{In} = \Omega(\mathrm{C}_{TS}) = [4, 6, 6, 8];\ N = m = 4;$$
$$\Omega(\mathrm{C}_{RS}) = \mathrm{C}_{RS}(TS_{In}, N) = P_{Out}, V_{f_{Out}}, dV_{f_{Out}}.$$

Очікуваним результатом подальшого виконання при $N = m$ відповідає набору значень, використаних для ініціалізації роботи конструкторів C_{MS} та C_{TS} та відповідно відповідати набору $\langle P, V_f, dV_f \rangle$:

$$\left\langle \begin{matrix} P_{In} = P_{Out} = \{f : "f + f -"\} \\ V_{f_{In}} = V_{f_{Out}} = 4; \\ dV_{f_{In}} = dV_{f_{Out}} = 2 \end{matrix} \right\rangle.$$

У свою чергу, відповідно до порядку виконання у режимі MQ, процесс починається з виконання конструктора C_{RS}. Відповідно, значення вхідного часового ряду TS_{In} не маєть передвизначене синтетичне походженняб яке б потребувало передвчасне виконання конструкторів C_{MS} та C_{TS}, які в даному випадку, виконується лише у контекстах виклику термінальних алгоритмів B_3, B_4 та B_5.

Вхідні параметри P_{In}, $V_{f\,In}$ та $dV_{f\,In}$ безпосередньо визначаються в межах відповідної хромосоми на основі результатів роботи відповідного термінального алгоритму B_1. Значення визначаються на підставі визначених принципах генерації структури правих частин правил підстановки та відповідних граничних значень параметрів $V_{f\,In}$ та $dV_{f\,In}$.

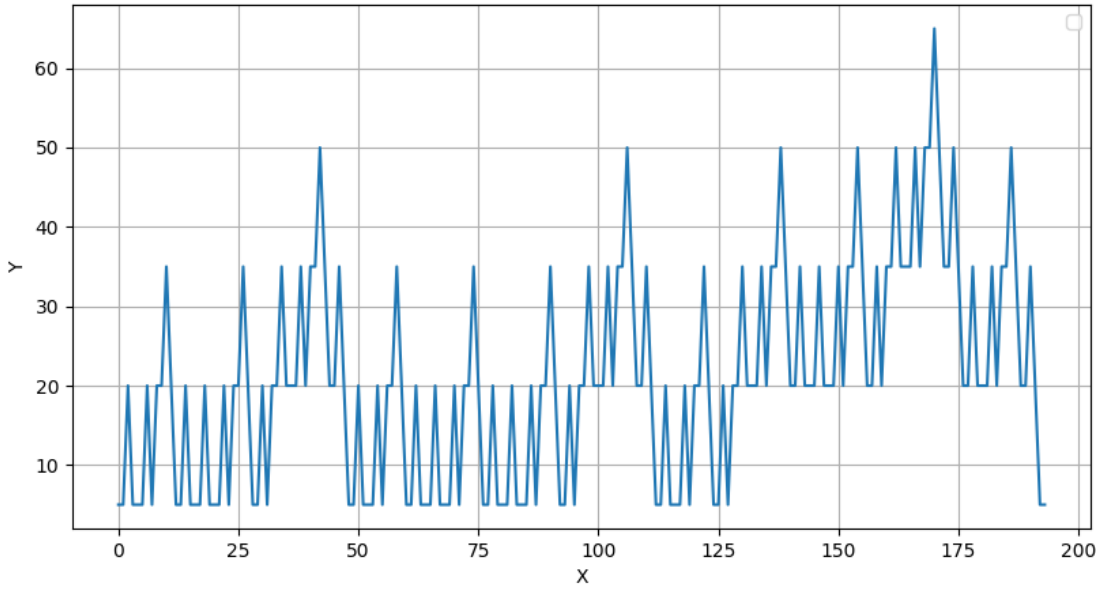


Рисунок 2.1 – Приклад синтетичного детермінованого ряду

2.4 Обгрунтування фрактальності часового ряду породжених конструктивною моделлю

Аналіз властивостей рядів, породжених конструктивною моделлю, свідчить про їх фрактальну природу. Це підтверджується на основі загальновідомих показників, серед яких ключове значення мають самоподібність та фрактальна розмірність, оцінювана за допомогою показника Герста.

Однією зі складових параметрів конструктивно-продукційної моделі є L-система (Lindenmayer system, L-система) [57]. L-система виступає інструментом, який дозволяє реалізувати породжуючі конструктори у формалізованому вигляді. Завдяки правилам підстановки терміналів L-система імітує процес формування складної структури часового ряду. Послідовна ітеративна заміна елементів відповідними правими частинами правил дозволяє відтворити глобальний патерн, який повторюється на різних масштабах та забезпечує самоподібність ряду (рис. 2.2).

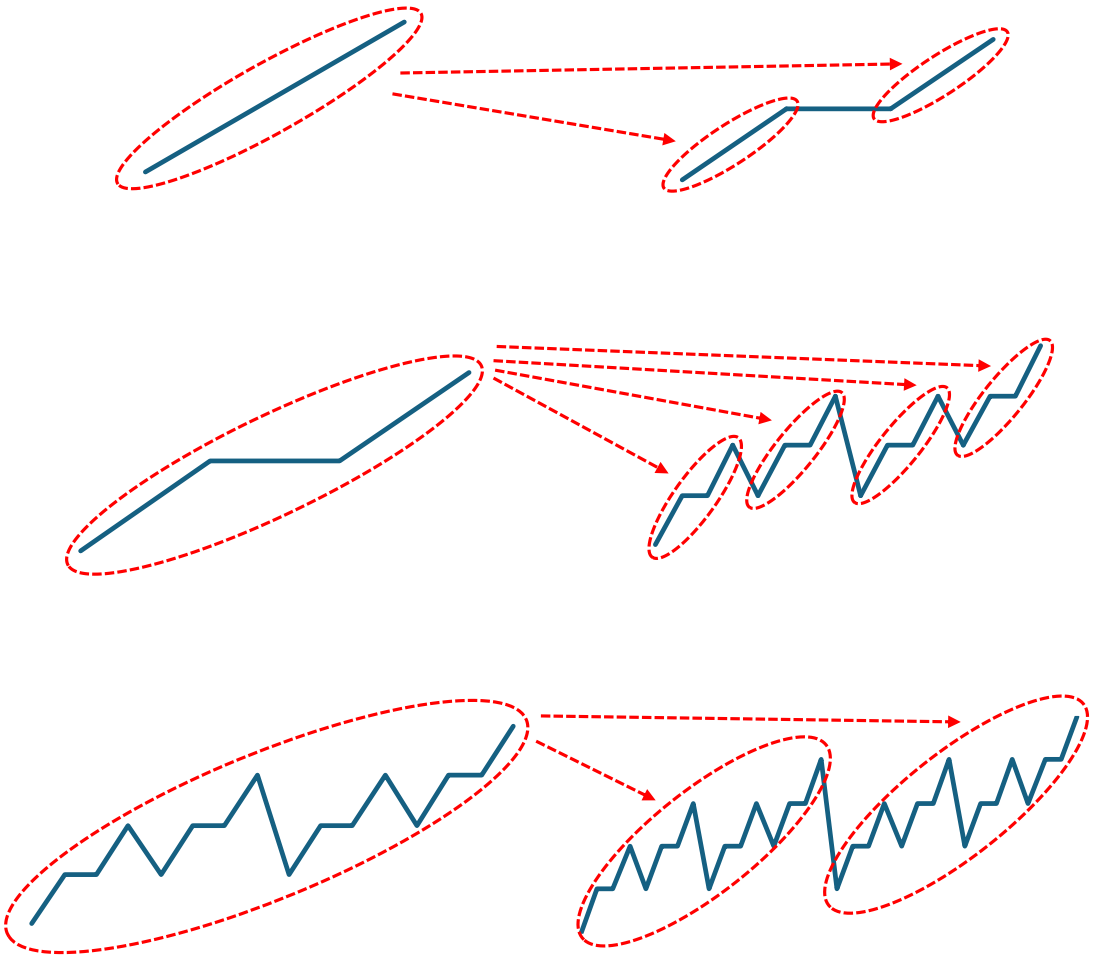


Рисунок 2.2 – Покрокова розгортка ряду заданого моделлю з параметрами $\{f, P: f \rightarrow f+f-;\ V_f = 15, dV_f = 10\}$

Властивість досягається відповідно до використання розгортки L-системи як фінального мультисимвольного ланцюжка термінальних та нетермінальних символів. Принципи самоподоби [71, 110, 111] прослідкуються й на її власній структурі при розгляді процесу розгортки (рис. 2.3).

Додаткове підтвердження фрактальної природи було отримано шляхом розрахунку показника Херста. Для часових рядів, породжених конструктивною моделлю, значення цього показника стабільно перевищують 0,8. Це свідчить про персистентність рядів та виражену тенденцію до продовження поточного руху їхніх значень, а також означає, що поведінка системи в майбутньому суттєво залежить від її попередніх станів.

Крім того, розрахована фрактальна розмірність Гаусдорфа [85] для таких рядів становить приблизно $D \approx 1.2$. Це значення додатково підтверджує їх фрактальну природу та узгоджується з виявленою властивістю самоподібності.

$$n = 0;\ f$$

$$n = 1;\ \overbrace{f + f -}^{f}$$

$$n = 2;\ \overbrace{f + f -}^{f} + \overbrace{f + f -}^{f} -$$

$$n = 3;\ \overbrace{f + f -}^{f} + \overbrace{f + f -}^{f} - + \overbrace{f + f -}^{f} + \overbrace{f + f -}^{f} - -$$

Рисунок 2.3 – Покрокова розгортка L-системи відповідно до аксіоми та правила підстановки: $\{ f, P: f \rightarrow f + f -; \}$

2.5 Модифікація процесу відновлення для охоплення синтетичних стохастичних часових рядів

Головним недоліком детермінованих часових рядів є простота їхньої структури (рис. 2.1). На основі їхніх значень процес реконструкції є тривіальним. Відсутність варіативності значень дозволяє побудувати чіткий і однозначний алгоритм оцінювання. У свою чергу, стохастичні часові ряди мають складнішу структуру та варіативність значень. Завдяки цьому їх можна використовувати для опису більш складних природних процесів (рис 2.4).

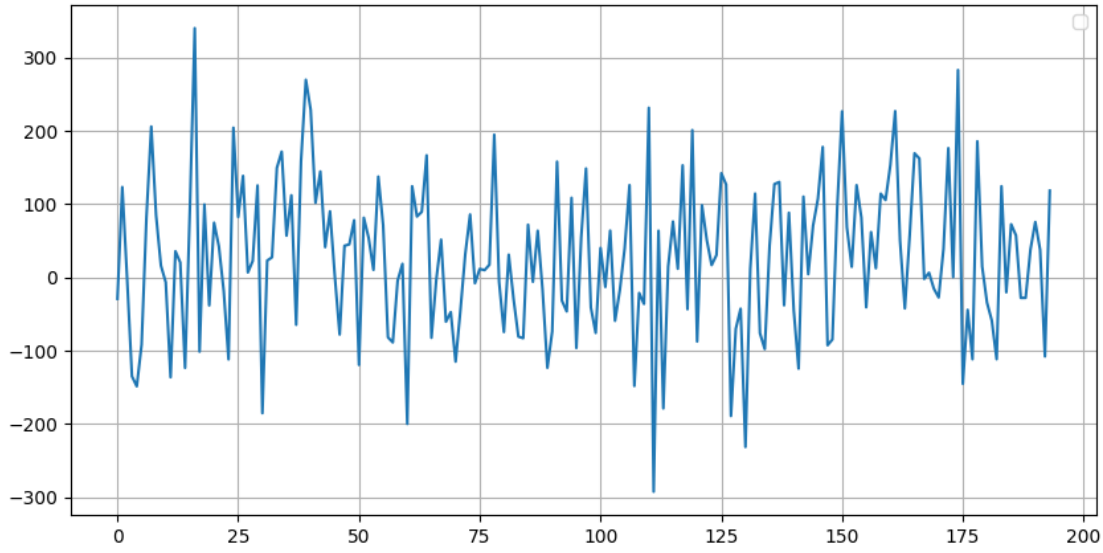


Рисунок 2.4 – Приклад синтетичного стохістичного ряду

Щоб забезпечити можливість обробки синтетичного стохастичного часового ряду, процес отримання окремих значень ряду було ускладнено впровадженням операції нормального розподілу, де значення tV_f розглядається як математичне очікування, а для значення дисперсії вводиться окремий параметр D_f конструктивної моделі. Відповідно, отримання окремих значень часового ряду $v(t)$ матиме наступний вигляд:

$$\begin{aligned}
&\{\left\langle \begin{matrix} \langle \sigma \to f\sigma, \sigma \to v\sigma \rangle, \\ \langle =: (v(t), N(tV_f, \sqrt{D_f})), +(t, t, dt) \rangle \end{matrix} \right\rangle, \\
&\left\langle \langle \sigma \to +\sigma \rangle, \langle +(tV_f, tV_f, dV_f) \rangle \right\rangle, \\
&\left\langle \langle \sigma \to -\sigma \rangle, \langle -(tV_f, tV_f, dV_f) \rangle \right\rangle, \\
&\langle \langle \sigma \to \varepsilon \rangle, \langle \varepsilon \rangle \rangle\}
\end{aligned} \tag{2.7}$$

де N – функція нормального розподілу з математичним очікуванням tV_f та дисперсією D_f. Зміни в процесі моделювання ряду ускладнюють його структуру і, відповідно, процес реконструкції конструктивної моделі. Збільшення складності процесу безпосередньо впливає на його часову ефективність, але цей побічний ефект нівелюється вже впровадженим мультиагентним підходом до обчислень. Введення додаткового параметра в процес призвело до розширення структури хромосоми та додавання параметра дисперсії з його інтеграцією в процеси кросинговеру та мутації.

При модифікації фітнес-функції враховувався стохастичний характер часових рядів. Щоб відобразити цей фактор, при розрахунку значення придатності хромосоми на основі її параметрів генерується визначенна множина з N часових рядів. Через свою стохастичну природу вхідний ряд може мати нескінченну кількість різних форм, які залежать від параметрів моделі його проектування.

Також, один набір параметрів конструктивної моделі може відповідати за породження нескінченної кількості різних часових рядів відповідно до використання нормального розподілу під час генерації їх значень (рис. 2.5).

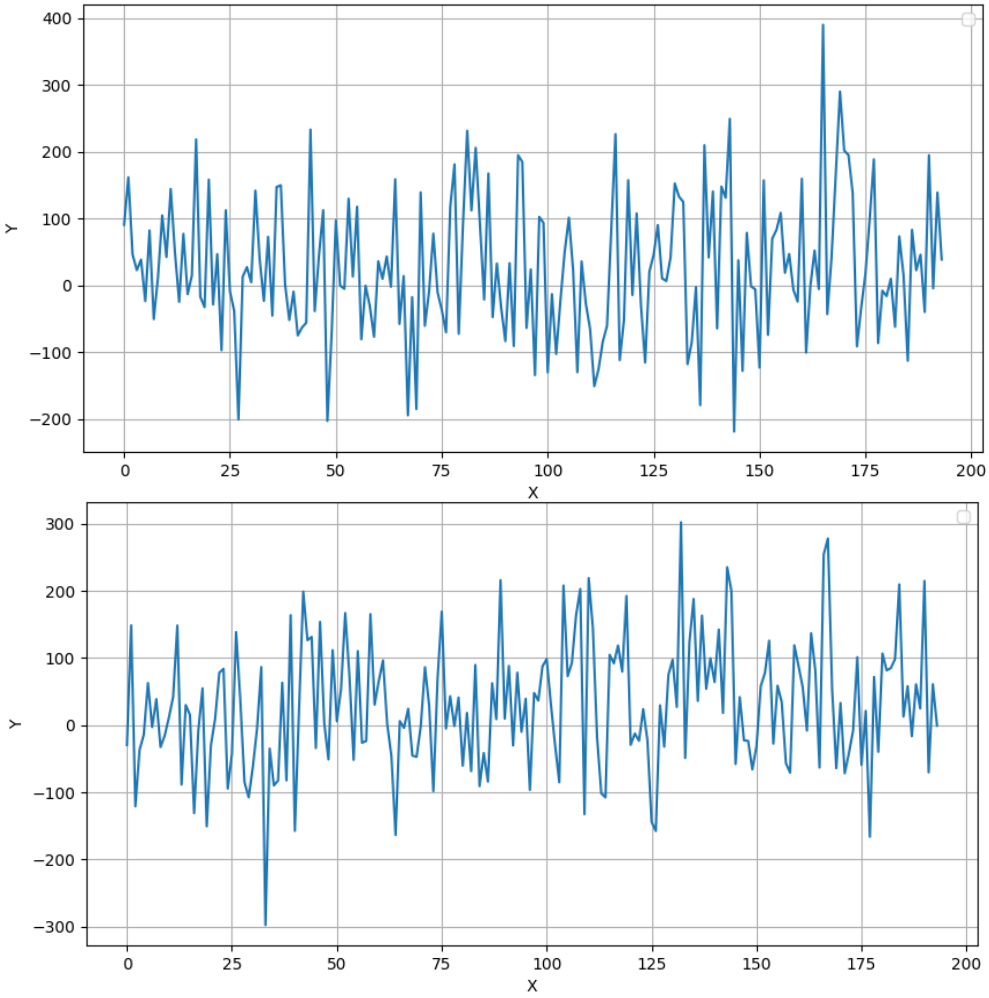


Рисунок 2.5 – Синтетичні стохастичні ряди породжені одним набором параметрів конструктивної моделі

Виходячи з цього твердження, було вирішено включити у вхідні параметри не один синтетичний ряд, а визначенна кількість з M рядів однакової довжини та породжених однаковими параметрами конструктивної моделі. Під час розрахунку придатності кожен із N згенерованих рядів в межах однієї хромосоми зіставляється з кожним із M входів. Результатом розрахунку є сума найменших різниць між одним згенерованим рядом та M входами:

$$q(X_n) = \sum_{i=0}^{N} \min_j \left\{ \sum_{k=1}^{K} \left(TS_{In_{j,k}} - TS_{i,k}(C_{MS}, C_{TS}) \right)^2 \right\}_{j=1}^{M}, \quad (2.8)$$

де $q(X_n)$ – показник придатності (фітнес-індикатор) хромосоми X_n, K – визначена довжина часового ряду, $TS_{In_{j,k}}, TS_{i,k}(C_{MS}, C_{TS})$ – значення k-ї точки j-го вхідного TS_{In} та i-го згенерованого $TS_i(C_{MS}, C_{TS})$ часових рядів відповідно.

Такий підхід охоплює стохастичну природу часових рядів і підвищує точність та керованість процесу реконструкції. Єдиним недоліком цього підходу є велика кількість операцій порівняння рядів, що дорівнює $N \times M$ і призводить до погіршення часової ефективності роботи. Для пом'якшення цього впливу було прийнято рішення зменшити кількість хромосом у популяції, щоб обмежити кількість обчислень.

2.6 Визначення підходу до відновлення параметрів моделі реальних часових рядів

Обидва підходи продемонстрували високу ефективність при аналізі синтетичних часових рядів. Водночас їх застосування до часових рядів реальної природи виявило низку обмежень та загальну неефективність. Детермінований підхід є недостатнім для відтворення стохастичних елементів реального часового ряду, що проявляється у недостатній близькості між вхідним рядом та рядом, згенерованим на основі відновлених параметрів конструктивної моделі.

У свою чергу, підхід, орієнтований на стохастичні синтетичні часові ряди, потребує складної попередньої обробки вхідних даних. Така обробка передбачає початкове розбиття часового ряду на декілька взаємно близьких рядів, що вимагає визначення параметра близькості. На основі цього параметра здійснюється виділення окремих сегментів часового ряду, які повинні мати достатню довжину для подальших обчислень. Необхідність виконання таких операцій формує окремий напрям досліджень і значно ускладнює практичне застосування підходу.

З огляду на зазначені обмеження було запропоновано новий підхід до побудови процесу відновлення параметрів конструктивної моделі для реальних часових рядів. Він характеризується підвищеною складністю, але зберігає чітко виражену ітеративну структуру.

2.6.1 Опис ітеративного процесу відновлення параметрів моделі

У попередніх реалізаціях, орієнтованих на синтетичні часові ряди, процес відновлення зводився до одноразового застосування генетичного алгоритму з

отриманням одного набору параметрів конструктивної моделі. Такий підхід є достатнім у контрольованих умовах та при роботі із синтетичними даними, однак виявляється обмеженим при роботі з реальними часовими рядами. На відміну від цього, запропонований метод змінює як організацію загального процесу відновлення, так і формат кінцевого результату.

2.6.2 Обгрунтування відмови від використання стохастичного підходу для реальних рядів

Окремо слід зазначити, що з метою усунення можливої варіативності результатів із процесу відновлення було виключено використання показника дисперсії D_f, який раніше застосовувався під час моделювання стохастичних часових рядів. Його використання призводило до того, що одна й та сама конструктивна модель могла породжувати декілька різних часових рядів, відмінних за числовими значеннями, унаслідок варіативності нормального розподілу під час генерації окремих точок ряду.

Попри те, що під час процесу підбору використовується виключно детермінований механізм моделювання, застосування сукупності конструктивних моделей дозволяє істотно збільшити структурну складність результуючого ряду. Кожний окремий набір параметрів описує лише певний аспект або часткову складову вхідного часового ряду, проте їх сукупна дія у вигляді сумування породжених рядів створює ефект складної композиції.

У цьому сенсі стохастичні властивості вхідного ряду відтворюються не за рахунок внутрішньої випадковості окремого набору параметрів, а внаслідок багатокомпонентної структури, сформованої у процесі ітеративного додавання незалежних детермінованих складових. Таке поєднання надає можливість досягти збільшення різноманітності та деталізації, що у підсумку дозволяє точніше наближати форму результуючого ряду до структури оригінального ряду значень.

Таким чином, відмова від використання показника дисперсії у поєднанні з ітеративною декомпозицією часового ряду дозволяє підвищити стабільність результатів, зменшити неоднозначність процесу відновлення та забезпечити більшу

керованість усього етапу побудови конструктивної моделі. Відсутність стохастичної варіативності робить процес відновлення більш передбачуваним і формально визначеним, що суттєво полегшує аналіз проміжних результатів.

2.6.3 Узагальнення особливостей процесу роботи з реальними часовими рядами

Даний підхід дозволяє ітеративно зменшувати різницю між вхідним та генерованими рядами, забезпечуючи покрокове наближення до найбільш узгодженого рішення. На кожному етапі аналізу залишковий ряд зменшується, що свідчить про поступове врахування новими моделями тих структурних елементів, які не були відтворені попередніми. У підсумку формується багаторівнева система, у якій кожна ітерація уточнює результат попередньої, а кінцевий результат досягається за рахунок сумарної дії всіх відновлених конструктивних моделей, що надає можливість покроково наближати результат до вхідного ряду (рис. 2.6).

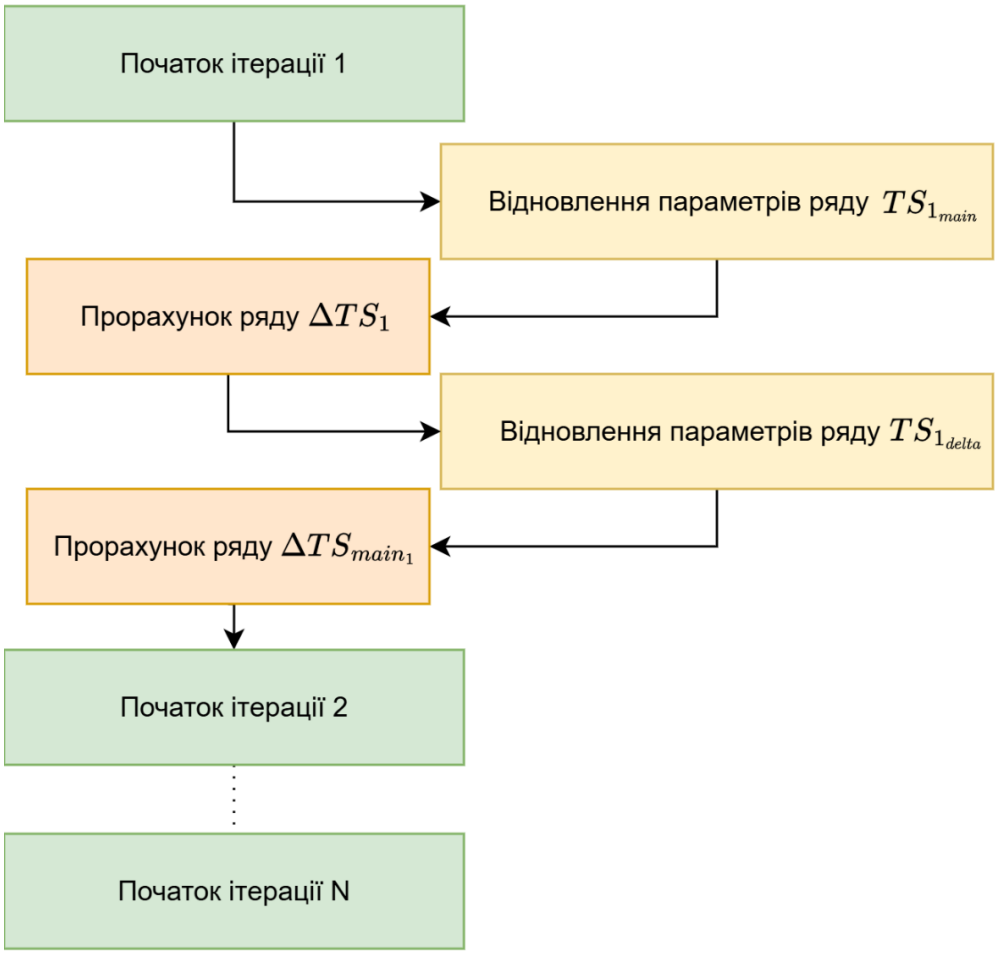


Рисунок 2.6 – Схема ітеративного процесу відновлення

Крім того, він не потребує будь-якої попередньої обробки вхідного часового ряду, яка була необхідною, наприклад, під час роботи із синтетичними стохастичними рядами. У тих випадках потрібне було початкове розбиття ряду на декілька взаємно близьких підрядностей, що вимагало визначення параметра близькості та подальшої перевірки коректності такого розбиття. Відсутність цієї стадії у новому підході зменшує загальну складність процесу, усуває залежність від додаткових евристичних критеріїв та виключає необхідність у паралельних дослідженнях для забезпечення коректності подальших етапів моделювання. Як наслідок, усувається потреба у нетривіальних технічних рішеннях для розблокування процесу відновлення, що робить метод практично застосовним без значних модифікацій та додаткових припущень.

Висновки до другого розділу

Сучасний стан наукових досліджень у сфері обробки часових рядів визначає три пріоритетні напрями, які включають аналіз статистичних характеристик, побудову математичних інтерпретацій процесів та прогнозування їхніх майбутніх значень. Еволюція цих методів безпосередньо пов’язана з розвитком обчислювальних засобів, що дозволяють ефективно виявляти приховані закономірності та складні внутрішні структури даних. Однією з найбільш вагомих характеристик таких процесів є їхня фрактальна природа, яка проявляється у збереженні структурних патернів на різних масштабах та наявності довготривалих залежностей між значеннями.

Застосування конструктивно-продукційного підходу до роботи з часовими рядами забезпечує можливість переходу від простого статистичного опису до визначення конструктивних моделей з відповідними наборами параметрів, які безпосередньо відтворюють механізм формування послідовностей.

Основою такого моделювання виступає поняття узагальненого конструктора, який на базі поновлюваного носія, сигнатури відносин та набору тверджень дозволяє реалізовувати складні ієрархічні структури. Важливим інструментом у цьому процесі є системи Ліндемаєра, які через ітеративні правила підстановки

термінальних символів дозволяють імітувати формування самоподібних патернів та глобальних структур часового ряду.

Для практичної реалізації моделювання було розроблено архітектуру мультиконструктора, що включає окремі модулі для формування символьних послідовностей, їхнього перетворення у числові ряди та виконання оберненого процесу відновлення параметрів. Використання генетичного алгоритму як основи ітеративного пошуку дозволило автоматизувати підбір правил підстановки та початкових умов, орієнтуючись на мінімізацію розбіжностей між вхідними та генерованими даними за допомогою методу найменших квадратів. Такий підхід продемонстрував високу ефективність при роботі з детермінованими моделями та створив підґрунтя для подальшої обробки складніших структур.

Була розглянута адаптація методу для роботи зі стохастичними часовими рядами, які мають складнішу структуру порівняно з детермінованими аналогами і краще описують природні, соціальні чи фінансові процеси. Для врахування стохастичної природи в процес моделювання було впроваджено функцію нормального розподілу, де значення кожної точки ряду визначається на основі математичного очікування та параметра дисперсії конструктивної моделі. Відновлення параметрів таких моделей здійснюється за допомогою генетичного алгоритму з використанням модифікованої цільової функції, яка передбачає порівняння множини згенерованих послідовностей із набором вхідних даних для охоплення варіативності стохастичного процесу.

Особливу увагу було приділено адаптації методу для аналізу реальних часових рядів шляхом впровадження процесу ітеративної декомпозиції. Замість використання стохастичної варіативності було запропоновано механізм послідовного виділення основної та залишкової складових ряду на кожній ітерації. Це дозволило досягти стабільності результатів та усунути неоднозначність моделювання, забезпечуючи при цьому точне наближення результуючого сумарного ряду до оригінального.

Запропонована методика виключає потребу у складній попередній обробці вхідних даних, що робить конструктивно-продукційний підхід універсальним та практично застосовним інструментом для дослідження динаміки складних систем.

Матеріали розділу опубліковані у роботах: [75, 76, 105, 108]

РОЗДІЛ 3
ЕКСПЕРЕМЕНТАЛЬНІ ДОСЛІДЖЕННЯ З МОДЕЛЮВАННЯ ФРАКТАЛЬНИХ СКЛАДОВИХ ЧАСОВИХ РЯДІВ РІЗНОЇ ПРИРОДИ

Відповідно до визначеного процесу верифікації запропонованого підходу на різних наборах даних було сформовано послідовність прикладних досліджень, яка забезпечує поетапну перевірку його ефективності та коректності. Така організація експериментів дозволяє поступово ускладнювати вхідні дані та оцінювати здатність методу адаптуватися до різних умов із впровадження подальших модифікацій для охоплення більш широкого спектра.

Початкові експериментальні дослідження безпосередньо пов’язані з випробуваннями на синтетичних часових рядах. Використання саме такого типу даних обумовлено можливістю повного контролю їх структури та параметрів, що, у свою чергу, створює сприятливі умови для детального аналізу поведінки алгоритму. Синтетичні часові ряди було поділено на дві основні категорії: детерміновані та стохастичні. Детерміновані ряди дозволяють оцінити точність відновлення чітко визначених залежностей, тоді як стохастичні – перевірити стійкість підходу та розширити його охоплення.

Під час дослідження синтетичних часових рядів процес верифікації здійснювався передусім шляхом порівняння початкових параметрів конструктивної моделі з параметрами її відновленого аналога. Такий підхід дозволяє безпосередньо оцінити точність відновлення структури моделі, а також виявити відхилення, що виникають у процесі роботи алгоритму.

Після отримання та аналізу результатів на синтетичних даних наступні експерименти були спрямовані на роботу з реальними часовими рядами. На цьому етапі особливу увагу було приділено верифікації результатів із використанням обраних кількісних метрик, що дозволяють оцінити загальну якість відновлення.

3.1 Експеримент 1: Відновлення параметрів конструктивної моделі синтетичних детермінованих рядів

У межах першого етапу досліджень, відповідно до визначеного процесу верифікації, було проведено експериментальне дослідження процесу відновлення параметрів, що задають часові ряди, з використанням конструктивно-продукційного підходу. Основною метою проведених випробувань було доведення можливості відновлення параметрів, які визначають модель генерації вхідного часового ряду. При цьому основний акцент робився на досягненні коректного відновлення параметрів, тоді як питання оптимізації обчислювальної ефективності не розглядалося як пріоритетне.

У процесі дослідження виконувалося поступове ускладнення параметрів шляхом збільшення кількості символів у правилах підстановки, збільшення кількості правил, а також ускладнення структури аксіоми. Відповідно до цього всі випробування були поділені на дві основні категорії: відновлення параметрів для часових рядів, породжених одним правилом підстановки (Експерименти 1.1 та 1.2), та відновлення параметрів для часових рядів, породжених кількома правилами (Експерименти 1.3 та 1.4).

3.1.1 Експеримент 1.1: Відновлення параметрів для одного простого правила підстановки

На початковому етапі дослідження використовувалася найпростіша конфігурація параметрів. Аксіома та ліва частина правила залишалися фіксованими та задавалися символом f. Права частина правила формувалася як послідовність із чотирьох символів, кожен з яких належав множині $\{f, +, -\}$, що в свою чергу дозволяло генерацію рядів із простою фрактальною структурою (рис. 3.1).

У межах даного експерименту було проведено 100 дослідницьких випробувань. Для кожного випробування генерувався часовий ряд за допомогою конструктора, після чого здійснювалося відновлення параметрів правила підстановки із застосуванням розробленого підходу.

За результатами дослідження встановлено, що максимальна кількість ітерацій, необхідна для відновлення параметрів, становила 207. Отримані результати свідчать про відносну простоту даного типу параметрів, а також про ефективність їх відновлення при використанні базової конфігурації алгоритму.

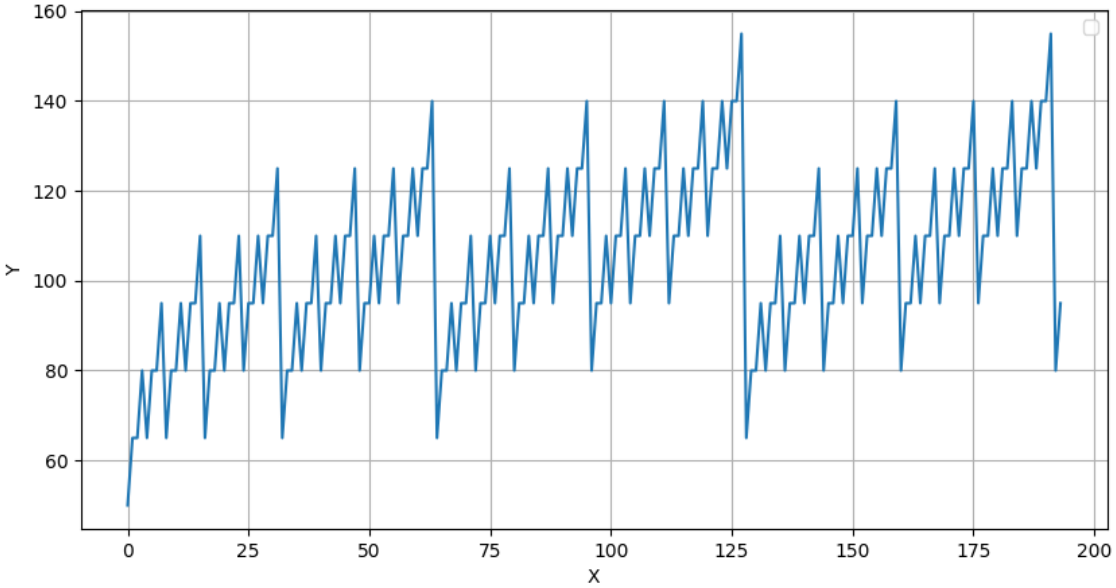


Рисунок 3.1 – Приклад синтетичного детермінованого ряду породженного моделлю з одним правилом замішення

Особливістю реалізації на даному етапі експериментів було використання значної початкової популяції хромосом та обмеження їх кількості на кожній ітерації. Операція мутації виконувалася не на кожному кроці, а лише у випадках, коли показники новоствореної популяції були гіршими за початкові, що дозволяло частково контролювати процес пошуку.

3.1.2 Експеримент 1.2: Відновлення параметрів для одного складного правила підстановки

Подальшим етапом стало ускладнення структури параметрів шляхом збільшення довжини правої частини правила підстановки до десяти символів. При цьому були введені додаткові умови генерації: перші символи повинні належати визначеному алфавіту; для наступних позицій допускається використання пробільного символу; ймовірність появи пробілу зростає ближче до кінця правила; під час генерації часового ряду пробільні символи ігноруються.

Таке розширення призвело до суттєвого збільшення складності структур часового породжених часових рядів (рис. 3.2) та ускладнення процесу відновлення параметрів.

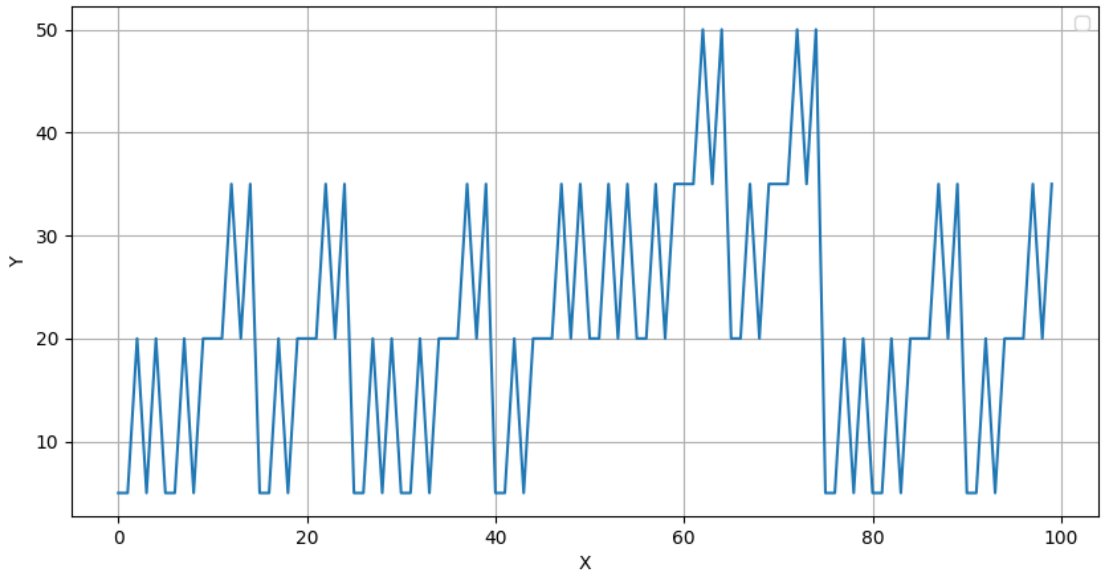


Рисунок 3.2 Приклад синтетичного детермінованого ряду породженного моделлю з одним правилом замішення складної структури

У ході дослідів було виявлено низку проблем, пов’язаних як із зростанням складності обчислень, так і з технічними обмеженнями реалізації. Зокрема, збільшилася кількість операцій у фазах схрещування та мутації, а також виникали ситуації переповнення числових типів при обробці значень.

Для вирішення зазначених проблем було виконано модифікацію алгоритму. Основні зміни полягали у наступному: операція мутації почала виконуватися на кожній ітерації; було зменшено розмір початкової популяції та кількість особин у наступних ітераціях; на початку кожної ітерації відбиралися найкращі особини; змінено підхід до формування нових хромосом під час схрещування; мутація лівої частини правила виконувалася шляхом повної регенерації.

Після внесення зазначених змін було проведено повторну серію зі 100 випробувань. У результаті максимальна кількість ітерацій для відновлення параметрів склала 22705 (рис. 5.6). Це підтверджує значне ускладнення задачі, однак також демонструє ефективність адаптованого алгоритму.

3.1.3 Експеримент 1.3: Відновлення параметрів для кількох простих правил підстановки

Після аналізу випадків із одним правилом було здійснено перехід до більш складних конфігурацій параметрів, які включають кілька правил підстановки. На даному етапі використовувалася модель із трьома правилами, кожне з яких мало унікальний символ у лівій частині, що забезпечувало детермінованість.

Додатково було введено умову жвавості, відповідно до якої всі правила повинні брати участь у формуванні кінцевого результату. Права частина кожного правила обмежувалася чотирма символами, що дозволяло зосередитися саме на впливі кількості правил.

Аксіома формувалася таким чином, щоб включати всі символи лівих частин правил, що гарантувало їх використання (рис. 3.4).

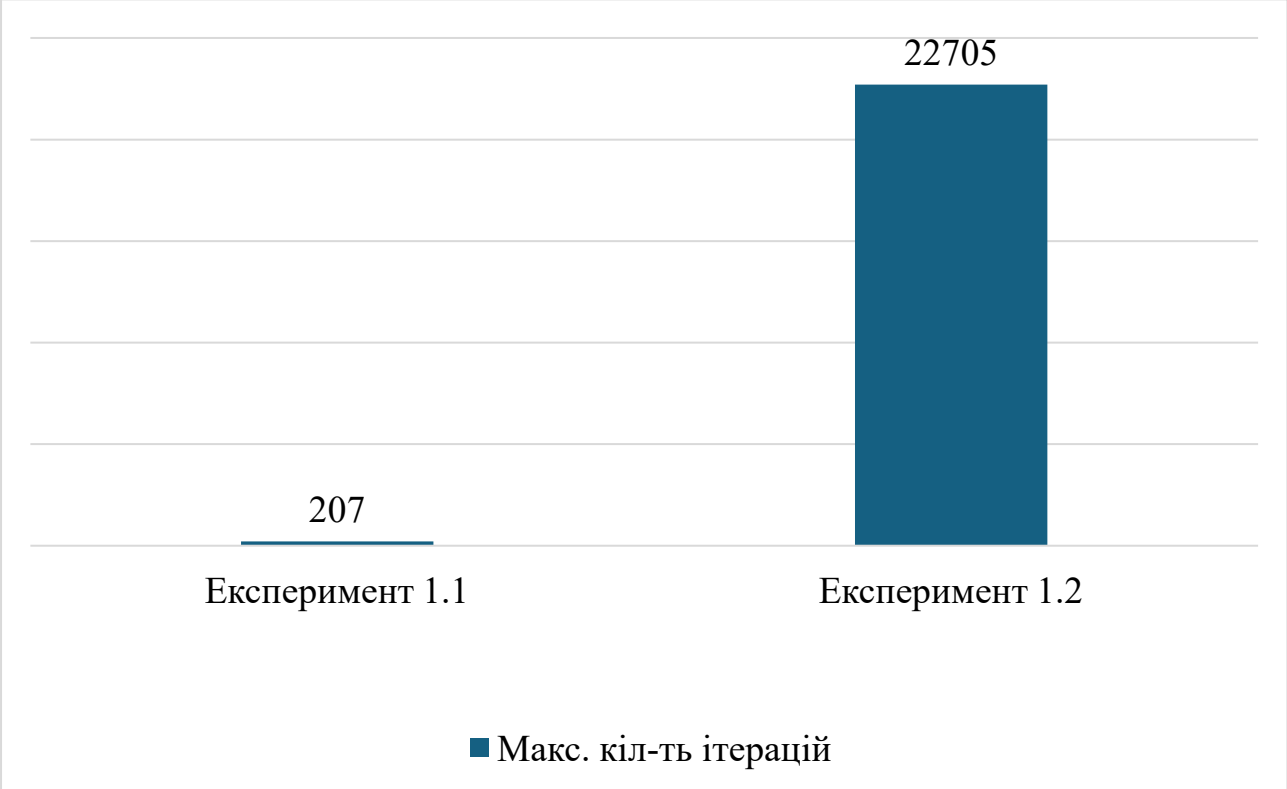


Рисунок 3.3 – Порівняння максимальної кількості ітерацій для етапів 1.1 та 1.2 випробувань на основі синтетичних детермінованих часових рядів

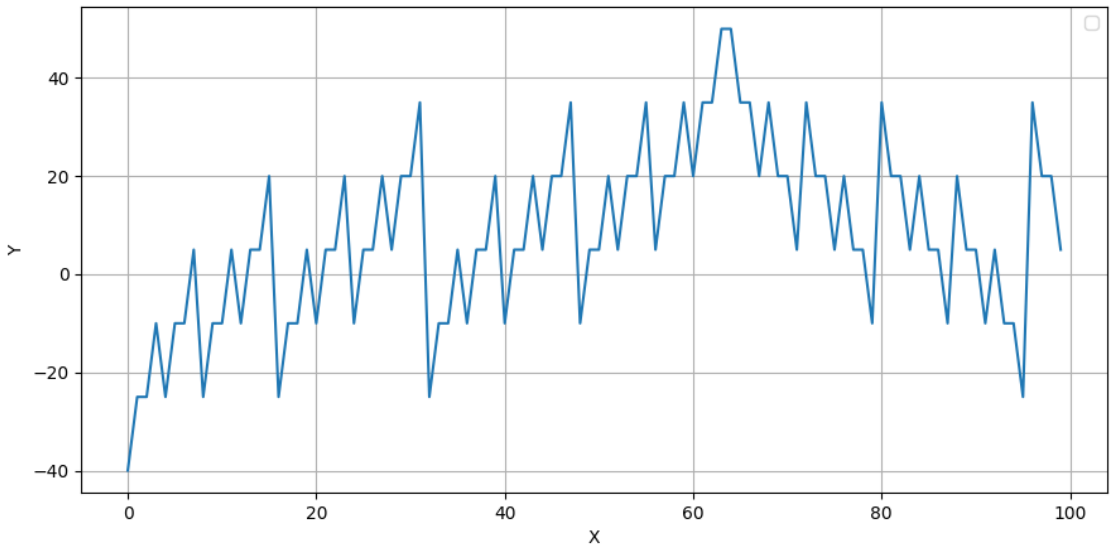


Рисунок 3.4 – Приклад синтетичного детермінованого ряду породженного моделлю з декількома правилами замішення

Для реалізації даного підходу було змінено процес генерації хромосом, а також механізми схрещування та мутації, які тепер враховували структуру множини правил.

Було проведено 100 дослідів, у результаті яких встановлено, що максимальна кількість ітерацій досягла 79462 (рис 7.1). Отримані результати свідчать про суттєве зростання складності задачі при переході до кількох правил, однак підтверджують можливість відновлення параметрів.

Рисунок 3.5 – Порівняння максимальної кількості ітерацій для етапів 1.1, 1.2 та 1.3 випробувань на основі синтетичних детермінованих рядів

3.1.4 Експеримент 1.4: Відновлення параметрів для кількох простих правил підстановки та складної структури аксіоми

Подальше ускладнення полягало у включенні аксіоми до процесів схрещування та мутації. Аксіома генерувалася випадковим чином на основі заданого алфавіту та кількості правил і мала довжину від 5 до 8 символів (рис. 3.6).

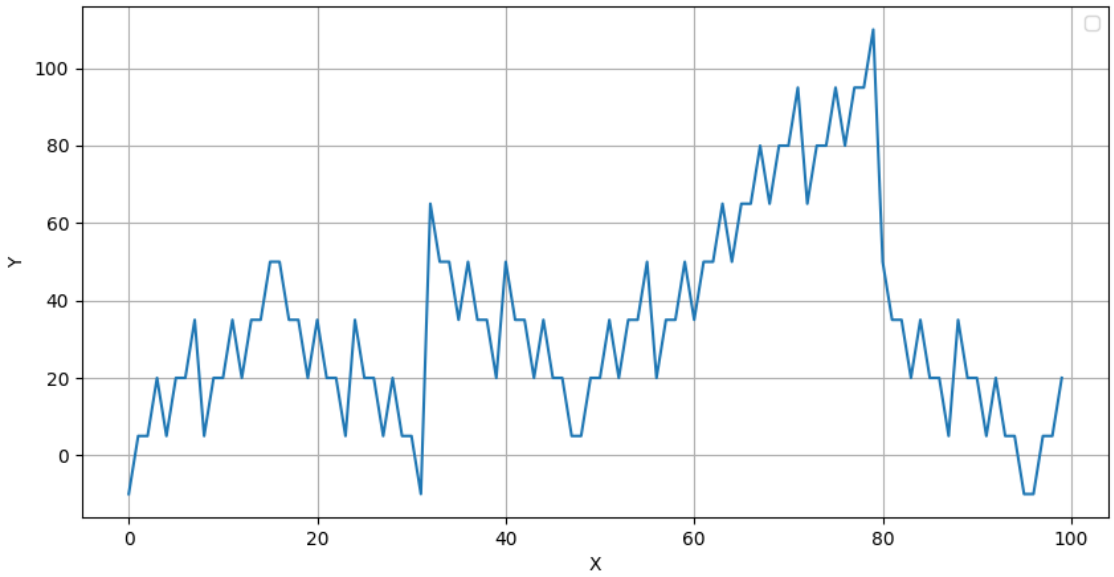


Рисунок 3.6 – Приклад синтетичного детермінованого ряду породженного моделлю з декількома правилами замішення та складною аксиомою

Однак така модифікація призвела до появи нової проблеми – порушення умови жвавості. У результаті схрещування могли утворюватися хромосоми, в яких деякі правила ставали недосяжними, оскільки відповідні символи були відсутні в аксіомі.

Першим підходом до вирішення цієї проблеми стало ребалансування аксіоми шляхом додавання або видалення символів відповідно до структури правил. Проте дана стратегія не дала очікуваних результатів під час дослідів.

У зв’язку з цим було запропоновано альтернативне рішення, яке полягало у зміні механізму схрещування аксіом: замість комбінування частин здійснювався їх обмін між хромосомами. У випадку порушення умов жвавості аксіома генерувалася повторно з урахуванням кількості правил.

Після впровадження зазначеного підходу було проведено 60 випробувань. За результатами дослідження максимальна кількість ітерацій склала 39346 (рис. 3.7). Водночас частина випробувань не завершилася успішно через досягнення

граничного значення кількості ітерацій, що свідчить про високу складність даної конфігурації параметрів.

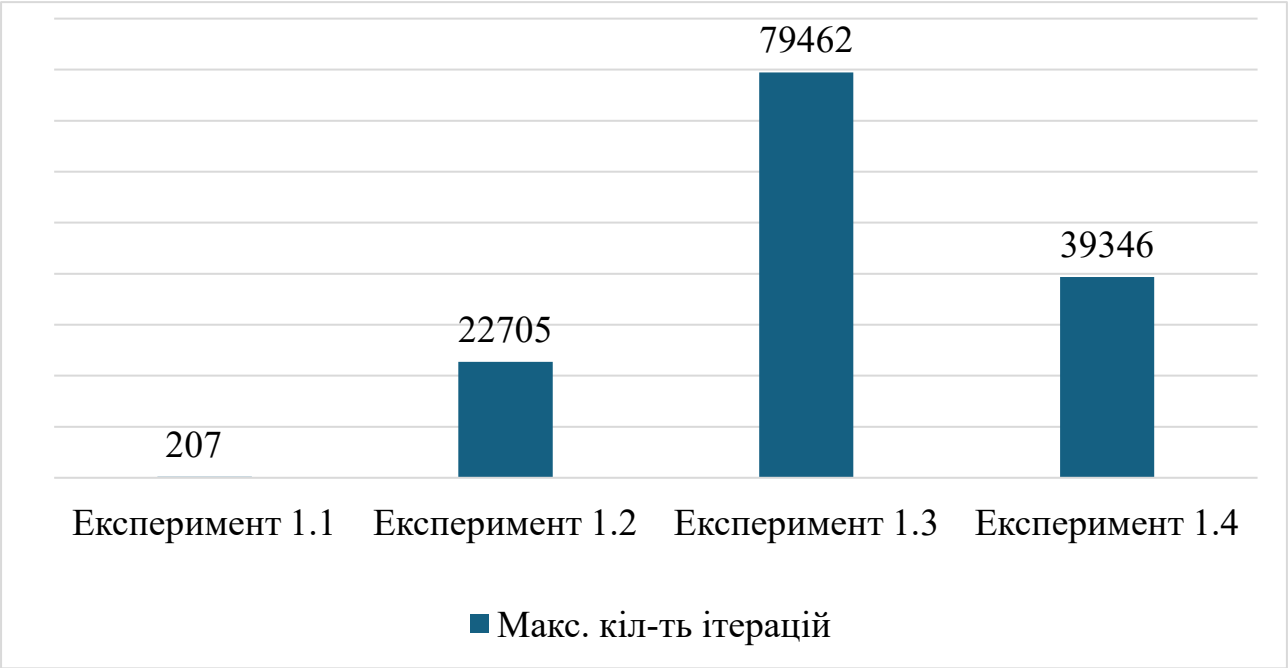


Рисунок 3.7 – Фінальне порівняння максимальної кількості ітерацій для першого етапу досліджень із використанням синтетичних детермінованих рядів

3.1.5 Узагальнення результатів першого етапу експерементальних досліджень

У результаті проведених випробувань першого етапу досліджень було визначено процес відновлення параметрів для різних конфігурацій, зокрема, для одного правила підстановки фіксованої та змінної довжини, а також для кількох правил із різною структурою аксіоми.

Встановлено, що зі зростанням складності параметрів суттєво збільшується кількість ітерацій, необхідних для їх відновлення. Найбільш складними виявилися конфігурації з довгими правилами та зі складною аксіомою, яка бере участь у генетичних операціях, відповідно до результатів експериментів 1.3 та 1.4.

У процесі дослідження було виконано модифікацію структури генетичного алгоритму, що дозволило адаптувати його до роботи з більш складними конфігураціями параметрів. Отримані результати підтверджують можливість застосування запропонованого підходу для задачі відновлення параметрів часових рядів та визначають напрями подальших досліджень.

Також відповідно до кожного з експериментальних етапів для відповідних наборів параметрів конструктивних моделей були визначені відповідні значення показників експоненти Герста *H* та фрактальної розмірності *D* на основі згенерованих часових рядів.

Для Експерименту 1.1 в ході випробувань було отримано результати, у яких показник *H* зафіксовано на рівні 0.226, а фрактальна розмірність *D* становить 1.774. Ці показники відображають специфіку структури часових рядів, що мають виражену антиперсистентну природу.

Для Експериметну 1.2 в ході випробувань було зафіксовано чітке розмежування отриманих результатів на два полярні кластери, що відображають різні структурні властивості даних.

Перша частина результатів характеризується високими значеннями показника *H* (0.970-0.998) та мінімальними значеннями фрактальної розмірності *D* (1.002-1.030). Такі значення притаманні рядам з максимальною інертністю: кожен наступний крок у даних з великою ймовірністю повторює напрямок попереднього. Геометрично ці послідовності наближаються до гладких ліній, що підтверджується значенням розмірності *D*, яке майже дорівнює одиниці.

Друга група результатів демонструє низькі значення *H* (0.187-0.231) за одночасного зростання фрактальної розмірності *D* до рівнів 1.769-1.813. У цьому випадку дані мають зворотну залежність, де будь-яке відхилення швидко компенсується рухом у протилежному напрямку. Висока фрактальна розмірність вказує на значну щільність коливань, через що графік стає складнішим і прагне заповнити площину.

Для Експерименту 1.3 отримані результати демонструють зміщення характеристик у бік більшої варіативності та появу перехідних станів.

На відміну від попереднього етапу, у цьому масиві спостерігається розмиття чітких меж між кластерами. Хоча значна частина значень показника *H* залишається в зоні антиперсистентності (близько 0.226 при *D* близько 1.774), з’явилася група результатів із середніми значеннями *H* у діапазоні 0.342-0.509 та відповідними показниками розмірності *D* (1.491-1.658). Поява значень поблизу відмітки 0.5 вка-

зує на наближення структури окремих часових рядів до характеристик випадкового шуму, що є прямим наслідком ускладнення засад генерації.

Водночас група персистентних результатів також зазнала змін: показники *H* тепер охоплюють ширший спектр – від 0.834 до 0.984, а значення фрактальної розмірності *D* коливаються від 1.016 до 1.166. Це свідчить про те, що стійкі тренди в цих даних стали менш ідеальними та містять більше внутрішніх коливань.

Для Експерименту 1.4 характерним є найбільш рівномірний розподіл показників по всьому теоретичному діапазону. Отримані результати демонструють значне кількісне зростання часових рядів із перехідними характеристиками, що свідчить про високу складність сформованих структур.

У цьому масиві суттєво розширився спектр значень у зоні антиперсистентності та хаотичності. Поряд із базовими показниками *H* на рівні 0.218-0.226, значна частина результатів зосередилася в діапазоні 0.276-0.488. Особливу увагу привертає група значень, що максимально наближені до відмітки 0.509 при розмірності 1.491. Така концентрація даних поблизу критичної межі вказує на те, що значна кількість згенерованих рядів за своїми властивостями майже ідентична випадковим процесам, де пам’ять про попередні стани практично відсутня.

Персистентний сегмент даних також представлений широким набором показників: від помірно стійких значень *H* на рівні 0.838-0.851 до екстремально високих показників 0.982-0.996 із розмірністю *D*, що прямує до одиниці (1.004-1.018). Це підтверджує, що в межах одного експерименту вдалося згенерувати повний спектр динамічних режимів.

3.2 Експеримент 2: Предбачення подальших значень синтетичного детермінованого ряду

Отримані результати досліджень в межах експерименту 1 показали високу точність відновлення параметрів конструктивних моделей для детермінованих часових рядів. Коректна реконструкція числових значень та структури L-системи дозволяє не лише відтворювати внутрішню структуру рядів, а й створювати підг-

рунтя для подальшого аналізу їх динаміки. Встановлено, що навіть при складних конфігураціях правих частин правил підстановки та наявності шуму в даних адаптована методика забезпечує відтворення ключових характеристик рядів у межах допустимої похибки, що підтверджує її стабільність і надійність.

На основі відновлених параметрів конструктивних моделей виникає природне продовження дослідження – прогнозування подальших значень часового ряду. Перевага такого підходу полягає у можливості використовувати внутрішні закономірності, закодовані у відновлених параметрах, для побудови точних прогнозів поза межами навчальної вибірки. Це створює основу для порівняння з класичними методами прогнозування, дозволяючи оцінити ефективність конструктивно-продукційного підходу не лише у відтворенні минулих даних, а й у передбаченні майбутніх значень ряду. Цей напрям став логічним наступним етапом досліджень.

3.2.1 Визначення списку існуючих моделей для порівняння результатів прогнозування

У противагу підходу конструктивізму, для оцінки ефективності операції передбачення подальших значень ряду були обрані загальновідомі лінійні та нелінійні моделі. Одними з найпопулярніших лінійних моделей прогнозування часових рядів є моделі типу ARIMA (Autoregressive Integrated Moving Average) [51]. Вона являє собою узагальнення моделі авторегресійної ковзної середньої (ARMA) й стала класичним підходом для роботи з часовими рядами.

Загальне позначення описується як $ARIMA(p, d, q)$, де параметр p визначає порядок (кількість часових затримок) компоненту авторегресії, параметр d – кількість разів, коли з даних віднімали попередні значення (порядок диференціювання), а параметр q – порядок моделі ковзного середнього.

Найчастіше ARIMA використовується з часовими рядами обмеженої довжини [83], які не мають чіткої сезонності значень [9] та мають явну тенденцію. Також дана модель може виступати як один с компонентів більш складного підходу, як ARIMA-LSTM, де використовується для визначення трендів та кореляції [59].

Основними недоліками зазначеної моделі є саме вузькість області використання. У противагу цьому вона пропагує низьку складність та ефективність у застосуванні з відповідними наборами даних.

Для прогнозування рядів, які мають складніші зв’язки між своїми значеннями на поточний момент, досить широко застосовують засоби машинного навчання. Найбільш популярним є використання рекурентних нейронних мереж [61]. Найпопулярнішими серед них визначають LSTM моделі.

Їх суть полягає у наявності каналу довготривалої пам‘яті, керування яким визначається трьома основними затворами:

- вхідний затвор, який визначає набір даних поточного кроку, який буде записаний у канал пам‘яті;
- затвор забування, який визначає частину збережених даних яку потрібно видалити;
- вихідний затвор, який контролює селекцію даних з каналу пам’яті, які необхідно передати на вихід.

За допомогою зазначених компонентів модель крок за кроком зчитує значення часового ряду, формуючи стан у каналі пам’яті. Протягом цього процесу зберігаються короткострокові та довгострокові зв’язки, на основі чого зберігається загальний тренд разом з окремими коливаннями значень часового ряду.

Сфера використання моделей типу LSTM досить широка. Її класична реалізація найчастіше використовується при прогнозуванні фінансових часових рядів, таких як індекси фондових бірж [43] та інших часових рядів зі складними та довготривалими залежностями [90]

Окрім класичного підходу до реалізації даної моделі, існує велика кількість її варіацій, які включають к себе додаткові шари нейронної мережі разом із статистичними моделями, такі як CNN-LSTM [22] та вже зазначена ARIMA-LSTM.

Особливістю першої є наявність додаткового шару згорткової мережі (CNN). Завдяки ньому модель обробляє локальні патерни, які у випадку часових рядів являють собою аномальні стани системи. На їх основі формується відповідний набір даних, який відображає ознаки часового ряду. На основі нього вже

LSTM-модель формує вихідний набір даних, виходячи з визначення довготривалих зв’язків між ознаками часового ряду.

Основною перевагою даної моделі є її можливість працювати з наборами даних, які мають велику шумність значень. Завдяки цьому вона має більш широку область використання, починаючи з вже зазначеного прогнозування показників фондових бірж [86], та продовжуючи транспортним трафіком [89] та показниками навколишнього середовища [96].

У свою чергу, модель ARIMA-LSTM працює за принципом, подібним до CNN-LSTM. На першому етапі ARIMA будує лінійний прогноз можливих значень часового ряду. На основі цих даних формується набір залишків, який формується на основі різниці реального значення та лінійного прогнозу.

На наступному етапі LSTM-частина будує свій прогноз на основі саме залишкових значень. Результуючий прогноз формується як сума двох отриманих наборів двох компонентів моделі. Це дозволяє отримати точніший прогноз за рахунок охоплення лінійних показників та обробку можливих шумів.

Дана модель показала ефективність прогнозування цінових показників з явним проявом сезонності [59] та показників захворюваності інфекційними захворюваннями [31].

3.2.2 Проведення експерементальної частини та порівняння отриманих результатів прогнозування до обраного набору моделей

Для підтвердження сформованої теорії та отримання статистично значущих результатів було сформовано початковий набір експериментальних синтетичних детермінованих часових рядів, що генеруються конструктивними моделями різного рівня складності. На поточному етапі дослідження складність структури конструктивної моделі визначається за довжиною правої частини правила підстановки породжуючої L-системи. Поряд із цим, кількість правил підстановки встановлюється рівною одному. Ця умова обумовлена методом генерації кінцевої послідовності символів.

Загалом було визначено 15 конструктивних моделей відповідної структури. На основі кожної з них був побудований часовий ряд довжиною у 500 точок.

На основі кожного сформованого часового ряду було проведено процес відновлення конструктивної моделі із застосуванням спеціалізованого автоматизованого програмного додатку [77]. На цьому етапі випробувань для побудови моделі використовувалися лише перші 400 точок ряду, що розглядалися як навчальна вибірка. Решта 100 точок відкладалася для подальшої верифікації прогнозної здатності моделі. Перевірка здійснювалася шляхом оцінки відповідності прогнозованих значень фактичним даним із використанням обраних метрик точності, що дозволяло кількісно оцінити рівень наближення моделі до реальних результатів.

Відповідно до поставленої мети дослідження були проведені паралельні випробування з використанням зазначених класичних підходів прогнозування. Для забезпечення коректності порівняння експерименти здійснювалися за однаковою методикою:

- перші 400 точок часового ряду використовувалися для навчання моделей;
- останні 100 точок застосовувалися для побудови прогнозу та подальшої оцінки точності, що дозволяло визначити близькість результатів до еталонних значень.

Такий підхід забезпечує можливість безпосереднього порівняння ефективності конструктивно-продукційного моделювання із традиційними методами прогнозування та дозволяє зробити обґрунтовані висновки щодо переваг або обмежень кожного з підходів.

Після завершення першої частини випробувань для всіх вхідних часових рядів було успішно відновлено породжуючі конструктивні моделі (табл. 3.1). Це свідчить про коректність застосованої методики реконструкції та її здатність відтворювати внутрішню структуру рядів різної складності.

Під час другої частини було проведено прогнозування подальших значень часових рядів із використанням усіх обраних методів із подальших порівнянням отриманих значень із тими, які були визначенні за допомогою операціїї подов-

ження часового ряду за рахунок визначення відповідної довжини під час його генераціїї отиманною конструктивною моделлю (табл. 3.2).

Таблиця 3.1 – Порівняння наданих та відновлених параметрів рядів

Параметри синтетичного ряду	Параметри відновленого ряду
$P: f \rightarrow +ff - ff$ $V_f = 54, dV_f = 43$	$P: f \rightarrow ff - ff + ff$ $V_f = 24.95, dV_f = 34.96$
$P: f \rightarrow +ff - ff + f -$ $V_f = 42, dV_f = 31$	$P: f \rightarrow + + ff - -ff + +f - -$ $V_f = 42.9, dV_f = 15.49$
$P: f \rightarrow +f - f + f -$ $V_f = 78, dV_f = 29$	$P: f \rightarrow +f - f + f -$ $V_f = 75.81, dV_f = 28.98$
$P: f \rightarrow ff - f + +ff -$ $V_f = 24, dV_f = 15$	$P: f \rightarrow ff - f + +ff -$ $V_f = 23.97, dV_f = 14.56$
$P: f \rightarrow f + f + f - f -$ $V_f = 36, dV_f = 45$	$P: f \rightarrow f + f + f - f -$ $V_f = 36.69, dV_f = 44.79$

Таблиця 3.2 – Порівняння точності передбачення подальших значень ряду

№	Метрика	КПМ	ARIMA	LSTM	ARIMA-LSTM	CNN-LSTM
1	MAE	0.24	29.9	2.33	15.3122	5.2323
	MAPE	0.14%	22.08%	1.49%	9.71%	3.30%
2	MAE	0.87	49.6	1.77	17.112	2.6026
	MAPE	1.02%	67.34%	2.32%	15.44%	3.37%
3	MAE	2.27	31.18	3.61	12.4109	4.0443
	MAPE	1.23%	15.57%	1.86%	5.39%	2.28%
4	MAE	0.82	22.49	2.91	9.677	5.9627
	MAPE	1.72%	51.71%	6.02%	15.46%	15.77%
5	MAE	0.5	69.87	8.77	19.3636	9.0058
	MAPE	0.16%	22.98%	3.73%	6.57%	3.94%

Аналіз значень метрик свідчить, що прогнозування на основі конструктивно-продукційних моделей забезпечує найвищу точність при роботі з синтетичними детермінованими часовими рядами.

Разом із тим, результати також демонструють, що нелінійні методи прогнозування (наприклад, нейронні мережі чи інші алгоритми машинного навчання) здатні забезпечувати досить високий рівень точності відтворення часових рядів.

Це свідчить про їхню адаптивність та здатність якісно відображати складні внутрішні залежності між значеннями.

Найгірші результати були отримані у випадку застосування ARIMA-моделі, за винятком тих ситуацій, коли вхідний часовий ряд наближався до майже повністю лінійного. Такий результат є закономірним, оскільки область застосування лінійних моделей обмежена рядами з простою структурою та слабко вираженими нелінійними залежностями. У випадку складних фрактальних систем можливості ARIMA є недостатніми для відтворення їх динаміки.

3.3 Експеримент 3: Відновлення параметрів моделі на основі синтетичних стохастичних часових рядів

При попередніх етапах досліджень (Експеримент 1 та 2) було продемонстровано ефективність запропонованого методу відновлення параметрів конструктивних моделей для детермінованих часових рядів. Відновлені параметри дозволяли відтворювати внутрішню структуру рядів із високою точністю (Експеримент 1.1-1.4) та прогнозувати подальші значення з мінімальними відхиленнями (Експеримент 2). Проте реальні системи часто характеризуються наявністю випадкових впливів та шуму, що значно ускладнює завдання відновлення та прогнозування. Для оцінки стійкості та адаптивності запропонованого підходу необхідно перевірити його роботу в умовах стохастичних коливань, а також визначити межі ефективності алгоритму у випадках нестабільних структур рядів.

Такий підхід дозволяє комплексно оцінити не лише точність відтворення середніх тенденцій, а й здатність алгоритму підтримувати узгодженість структурних залежностей, характерних для вихідних часових даних.

Перехід від детермінованих до стохастичних часових рядів (рис. 3.8) передбачає врахування додаткового параметра дисперсії, який визначає величину випадкових відхилень у ряді. Цей параметр ускладнює однозначне відтворення ряду за параметрами конструктивних моделей, оскільки окремі значення більше не залежать лише від логіки L-системи, а містять випадкові компоненти.

Тому алгоритм відновлення потребував модифікації структури хромосоми та функції оцінки фітнесу. Дослідження дозволили оцінити стійкість методу та здатність виділяти приховані закономірності, що лежать в основі значень рядів.

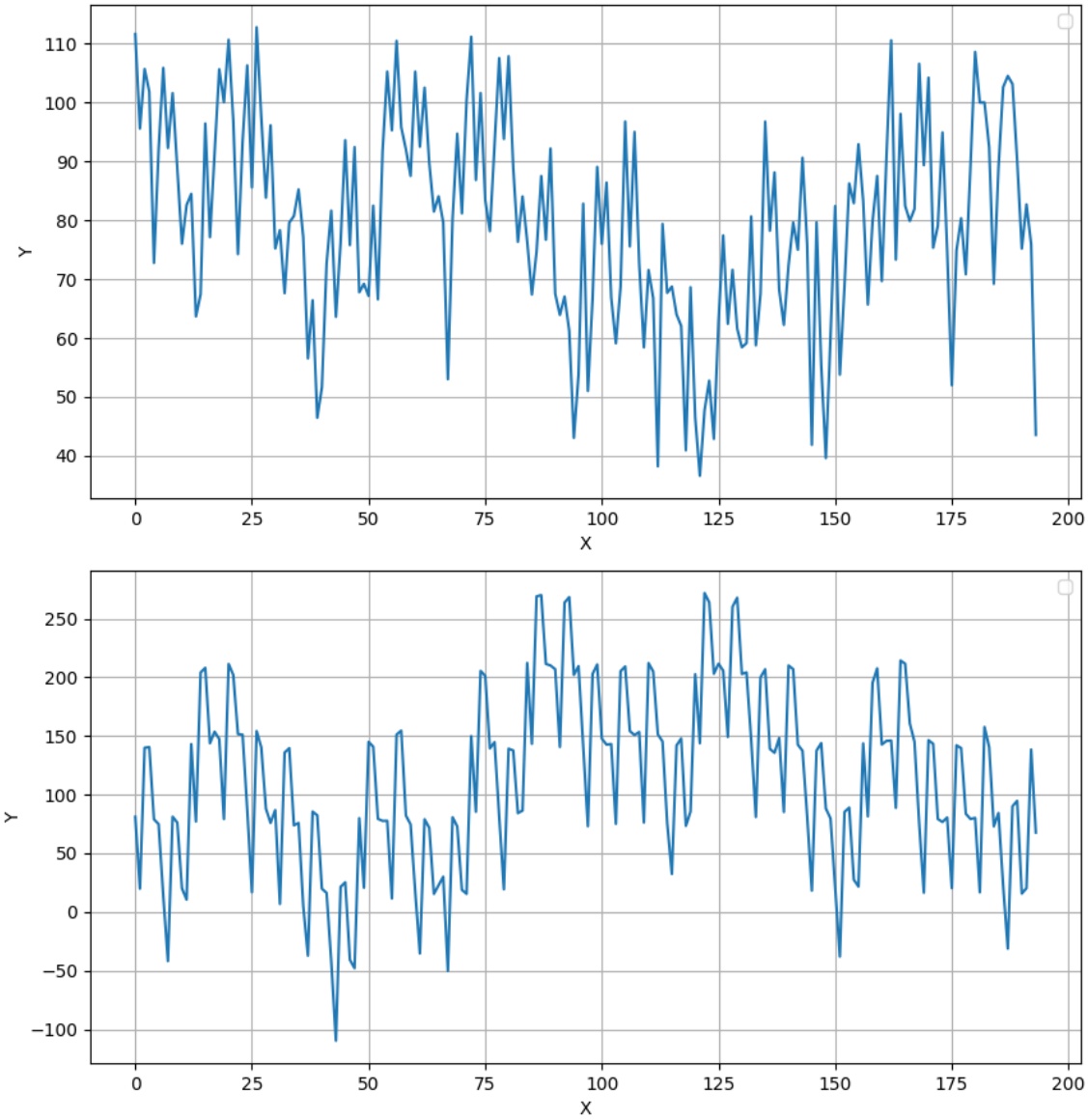


Рисунок 3.8 – Приклади синтетичних стохастичних рядів використаних при виконанні експерементальної частини

Процедура експериментальної оцінки та верифікації здійснювалася на основі спеціально згенерованих синтетичних стохастичних часових рядів.

В межах випробувань L-система містила лише одне правило підстановки, права частина якого варіювалася від чотирьох до десяти символів. Для кожного ряду структурні компоненти формувалися випадковим чином із суворим дотриманням попередньо визначених правил, що забезпечувало контроль за довжиною

правої частини правила та граничними значеннями ключових параметрів конструктивних моделей.

При використанні двох і більше правил підстановки з простою аксіомою виникала специфічна проблема підбору розгортки відповідно до заданої довжини часового ряду. Основна складність полягала у тому, що співвідношення кількості розгорток L-системи до довжини ряду могло використовувати лише частину повного ланцюжка символів. У випадку простої аксіоми, яка є конкатенацією символів лівих частин правил підстановки, деякі правила могли не брати участі в формуванні остаточного ряду, що потенційно зменшувало повноту представлення параметрів

Враховуючи цю особливість, фокус дослідження було зміщено на роботу зі стохастичними синтетичними рядами, породженими L-системою з одним правилом підстановки, щоб більш глибоко проаналізувати саме вплив несталої структури рядів на процес відновлення параметрів.

3.3.1 Процес відновлення та визначення критерії близькості вхідних та отриманних часових рядів

Безпосередній процес кожного експерименту тривав до повної ідентифікації та успішного відновлення параметрів конструктивних моделей, включно з числовими значеннями та лінгвістичною структурою L-системи. Було встановлено аналітичний критерій точності: похибка відновлених числових параметрів не перевищувала 1.5% від базового математичного очікування, а лінгвістичні компоненти відтворювалися абсолютно точно.

Для параметрів $M = 50$ та $N = 50$ було проведено серію з 50 незалежних випробувань. У кожній ітерації здійснювалося повне відновлення параметрів конструктивних моделей в межах встановлених критеріїв. Характерною особливістю процесу стало прагнення фінальних значень параметрів до граничних показників, зафіксованих на початковому етапі конфігурування та калібрування експерименту.

Особливо важливим було спостереження за параметром дисперсії, який у ході роботи алгоритму систематично зменшувався до найменшого можливого невід'ємного значення. Це відбувається через механізм фітнес-оцінки: алгоритм усереднює значення параметрів популяції хромосом, поступово знижуючи дисперсію для формування більш стабільної структури рядів. Завдяки цьому фінальний ряд зберігає математичну коректність і фізично обґрунтовану поведінку навіть у разі високого стохастичного шуму.

У стохастичних рядах класичну структуру хромосоми та базову модель розрахунку фітнес-функції суттєво модифікували під специфіку випадкового середовища. Розрахунок поточної фітнес-функції враховував ступінь близькості кількох незалежних часових рядів, згенерованих на основі параметрів конструктивних моделей досліджуваної хромосоми, до відповідних еталонних рядів. Такий багатовимірний підхід дозволяв зменшити вплив випадкової дисперсії та забезпечити високу стійкість алгоритму, одночасно контролюючи коректність і повноту відтворення параметрів.

3.3.2 Аналіз отриманих результатів експериментів

Загалом на даному етапі досліджень було виконано 50 досліджень із використанням заздалегідь підготовлених синтетичних стохастичних часових рядів, породжених за допомогою конструктивних моделей із різними значеннями вхідних параметрів.

У результаті проведених випробувань на базі синтетичних стохастичних часових рядів, згенерованих параметрами конструктивних моделей з одним правилом підстановки, було експериментально підтверджено та математично доведено здатність системи відновлювати параметри з високою точністю. Відхилення синтезованих показників математичного очікування та дискретного кроку від початкових еталонних значень утримувалися в межах допустимого 1.5%.

Основний акцент був поставлений саме на відновленні структури L-системи, значення якої повністю збігалося з початковими вхідними синтетичними рядами у всіх успішних експериментах. Це підтверджує високу ефективність запропонованого підходу, його здатність відтворювати спільні елементи структур фрактальних часових рядів (рис. 3.9, 3.10).

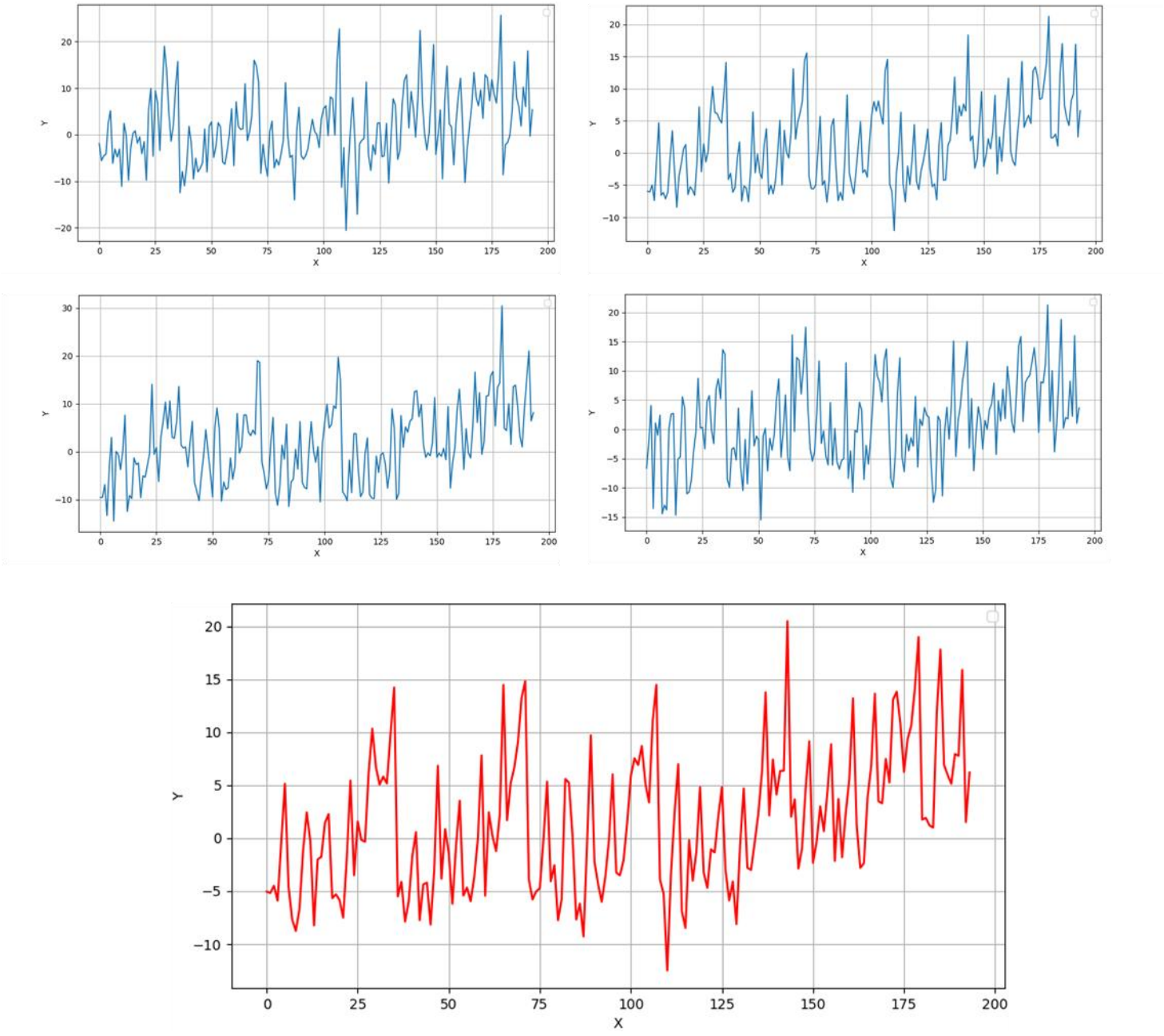


Рисунок 3.9 – Усереднення 4 синтетичних стохастичних рядів породжених одним набором параметрів конструктивної моделі

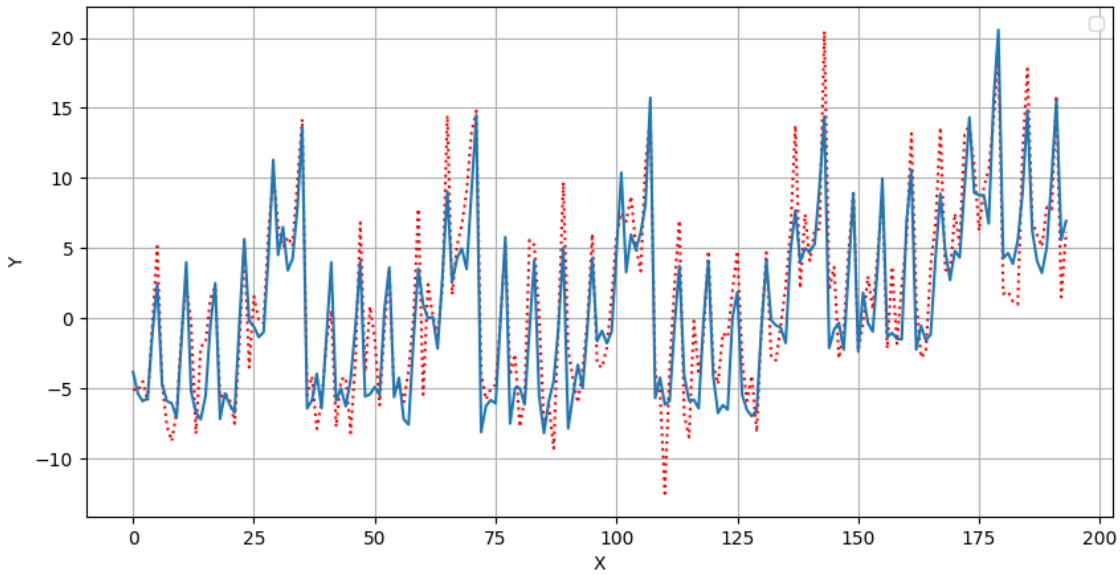


Рисунок 3.10 – Порівняння усередненого ряду з рис. 3.10 та побудованого на основі відновлених параметрів для відповідних наданих рядів

Варто відзначити, що для стохастичних синтетичних рядів спостерігається закономірність у відновленні параметрів: зі збільшенням складності початкових параметрів зростає і необхідна кількість ітерацій генетичного алгоритму для повного відновлення (табл. 3.3). Однак ця закономірність не є настільки чітко вираженою, як у випадку детермінованих часових рядів (рис. 3.7) і проявляється значно менш однозначно через вплив випадкових коливань та шумових компонентів у даних.

Таблиця 3.3 – Приклади результатів процесу відновлення параметрів

Параметри синтетичного ряду	Параметри відновленого ряду	Кіл-сть ітерацій
$P: f \rightarrow f+f-f$ $V_f = 55, dV_f = 21, D_f = 45$	$P: f \rightarrow f+f-f$ $V_f = 55.02, dV_f = 21.1, D_f = 1.15$	2656
$P: f \rightarrow ff-ff+$ $V_f = 45, dV_f = 15, D_f = 20$	$P: f \rightarrow ff-ff+$ $V_f = 45.09, dV_f = 14.14, D_f = 3.08$	25
$P: f \rightarrow ff-ff+f-f$ $V_f = 45, dV_f = 15, D_f = 20$	$P: f \rightarrow ff-ff+f-f$ $V_f = 44.86, dV_f = 14.98, D_f = 0.94$	8872
$P: f \rightarrow f-f++ff-ff$ $V_f = 82, dV_f = 63, D_f = 25$	$P: f \rightarrow f-f++ff-ff$ $V_f = 81.37, dV_f = 62.62, D_f = 1.76$	919
$P: f \rightarrow +f-f+f-$ $V_f = 30, dV_f = 15, D_f = 45$	$P: +f-f+f-$ $V_f = 29.23, dV_f = 15.12, D_f = 3.64$	9387

Відповідно до Експерименту 1, для Експерименту 3 також були визначені значення показника Герста та відповідної фрактальної розмірності для синтетичних стохастичних часових рядів.

Перший кластер результатів представлений рядами з граничною персистентністю. Показники H тут варіюються від 0.928 до 1.0, а фрактальна розмірність D знаходиться в межах 1.0–1.072. Наявність значень $H = 1$ та $D = 1$ свідчить про створення послідовностей, що мають ідеальну пам'ять та лінійну структуру. Ці ряди характеризуються абсолютною стабільністю тренду, де випадкові коливання фактично відсутні.

Друга група значень охоплює зону сильної антиперсистентності. Тут показник H коливається у межах 0.141–0.238, а розмірність D сягає рівнів 1.762–1.859. Такі параметри притаманні стохастичним рядам із надзвичайно високою інтенсивністю змін: кожне значення у послідовності демонструє різке відштовхування від попереднього, що створює ефект максимальної "порізаності" графіка та його значної щільності на площині.

Окремо слід зазначити, що для окремої конструктивної моделі та набору її параметрів для всіх відповідно згенерованих часових рядів спостерігається сталість відповідних значень H та D, що доводить збереження фрактальної структури, незважаючи на різні значення самого часового ряду.

3.4 Експеримент 4: Відновлення параметрів моделі для реальних часових рядів

На попередніх етапах досліджень було детально розглянуто процес відновлення параметрів конструктивних моделей для детермінованих (Експерименти 1-2) та стохастичних (Експерименти 3) часових рядів. Для детермінованих рядів встановлено, що відновлені параметри дозволяють із високою точністю відтворювати внутрішню структуру ряду (Експерименти 1.1-1.4), а також забезпечують можливість подальшого прогнозування значень із мінімальними відхиленнями (Експерименти 2). При цьому процес відновлення характеризується відносно стабільною поведінкою, де складність параметрів безпосередньо впливає на кількість ітерацій, необхідних для досягнення результату.

У випадку стохастичних часових рядів було продемонстровано, що запропонований підхід зберігає свою ефективність навіть за наявності випадкових збурень та шуму. Разом із тим, процес відновлення набуває менш передбачуваного характеру, оскільки на нього впливають додаткові параметри, зокрема дисперсія. Це призводить до варіативності кількості ітерацій та певної нестабільності проміжних результатів, однак загальна здатність алгоритму до відновлення параметрів залишається на високому рівні.

Таким чином, отримані результати підтверджують універсальність підходу та його придатність до роботи з різними типами часових рядів. Водночас виникає необхідність у використанні узагальнених критеріїв оцінювання, які дозволяють формалізувати процес аналізу якості відновлення та забезпечити можливість порівняння результатів для різних типів даних у єдиному метричному просторі.

3.4.1 Обрані метрики оцінювання точності та структурної відповідності між вхідним та відновленим часовими рядами

Для комплексної оцінки якості відновлення застосовуються метрики, що дозволяють оцінити різні аспекти відхилень між вихідним і відновленим рядом:

- MAE (Mean Absolute Error) – середня абсолютна похибка, що характеризує загальний рівень помилки між відповідними значеннями рядів [6, 17, 29].
- MSE (Mean Squared Error) – середньоквадратична похибка, яка підсилює вплив значних відхилень і дозволяє більш критично оцінювати великі відхилення від вихідних значень [6, 17, 29].
- RMSE (Root Mean Squared Error) – коренева середньоквадратична похибка, яка інтерпретується у тих самих одиницях, що й вихідні дані, і підкреслює вплив найбільших відхилень [17, 29].
- MAPE (Mean Absolute Percentage Error) – середня відносна похибка, що дозволяє оцінити точність у відсотках і коректно порівнювати ряди різного масштабу [6, 17, 42].

- SMAPE (Symmetric Mean Absolute Percentage Error) – симетрична відносна похибка, яка забезпечує збалансовану оцінку навіть для малих значень ряду [17, 26].
- R^2 (Coefficient of Determination) – відображає, яку частку варіації вихідного ряду пояснює відновлений ряд. Це дозволяє оцінити структурну адекватність моделі [17, 49].
- Коефіцієнт кореляції Пірсона – визначає лінійну залежність між рядами, що показує узгодженість змін [48].
- Коефіцієнт кореляції Спірмена – оцінює монотонний зв'язок між рядами, що важливо для рядів із нелінійними залежностями [48].

Використання сукупності цих показників дозволяє не лише оцінити локальні відхилення, але й проаналізувати загальні закономірності та форму ряду, що робить оцінку максимально комплексною.

3.4.2 Спроба адаптації підходу до відновлення складових моделей стохастичних рядів до обробки реальних даних

На початку поточного етапу досліджень було розглянуто можливість перенесення методу, застосованого до стохастичних рядів під час 3-го етапу експериментальних випробувань, на реальні часові дані. Основна ідея полягала у розбитті вихідного ряду на підряди, які могли б виступати як незалежні реалізації процесу. Це дозволяло б застосувати підхід оцінювання близькості у форматі множинних реалізацій, аналогічно схемі M на N.

Однак реалізація такого підходу виявила суттєві труднощі. Ключовою проблемою стала відсутність універсального методу коректного розбиття ряду на підряди без втрати структурних залежностей. Це призвело до необхідності складної передобробки, яка включає сегментацію та аналіз однорідності даних. У результаті було прийнято рішення відмовитися від цього підходу на користь більш узагальнених методів оцінювання.

3.4.3 Експеримент 4.1: Випробування на основі покрокового ітеративного процесу

Відповідно до результатів, отриманих у ході випробування попередньо запропонованого підходу, було прийнято рішення про суттєве переосмислення загальної організації процесу відновлення параметрів конструктивних моделей. Аналіз проміжних результатів показав, що використання прямого підходу без урахування внутрішньої структури ряду та його можливого декомпозиційного представлення не дозволяє досягти стабільно високої точності, особливо у випадках складних або зашумлених часових залежностей.

У зв’язку з цим було впроваджено покроковий ітеративний підхід, який базується на поступовому підборі параметрів шляхом розкладання вихідного часового ряду на складові компоненти. Такий підхід дозволяє на кожному етапі процесу концентруватися на окремій частині структури ряду, послідовно уточнюючи параметри конструктивних моделей та зменшуючи загальну похибку відновлення. Фактично, процес набуває характеру багаторівневої ітеративної декомпозиції, де кожен наступний крок базується на результатах попереднього.

Під ітерацією в межах даного підходу слід розуміти не окрему ітерацію генетичного алгоритму, а завершений цикл відновлення параметрів для двох взаємопов’язаних компонентів: основного та залишкового рядів.

Застосування такого підходу дозволяє значно підвищити гнучкість алгоритму та адаптувати його до складних структур часових рядів, оскільки кожна наступна ітерація враховує невідтворені раніше компоненти часового ряду. У результаті формується послідовність уточнених параметрів конструктивних моделей, що в сукупності забезпечують більш точне відтворення як глобальних тенденцій, так і локальних особливостей ряду.

У таблиці 3.4 наведено результати обчислень для окремих ітерацій. Як демонструють представлені дані, збільшення кількості ітерацій безпосередньо впливає на підвищення точності відновлення. Кожна наступна ітерація забезпечує поступове зменшення відхилення між вхідним та згенерованим часовими рядами, що

свідчить про ефективність ітеративної декомпозиції та здатність моделі наближатися до цільової форми з покроковим зниженням похибок.

Таблиця 3.4 – Порівняння метрик результату відповідно до номера ітерації

Номер ітерації	Показник	Значення
1	MAE	4.665
	MSE	39.561
	RMSE	6.290
	MAPE	4.45%
	SMAPE	4.37%
	R^2	0.766
	Коефіцієнт кореляції Пірсона	0.879
	Коефіцієнт кореляції рангу Спірмена	0.834
2	MAE	4.083
	MSE	29.643
	RMSE	5.445
	MAPE	3.87%
	SMAPE	3.86%
	R^2	0.824
	Коефіцієнт кореляції Пірсона	0.909
	Коефіцієнт кореляції рангу Спірмена	0.886
4	MAE	3.606
	MSE	24.176
	RMSE	4.917
	MAPE	3.45%
	SMAPE	3.43%
	R^2	0.857
	Коефіцієнт кореляції Пірсона	0.926
	Коефіцієнт кореляції рангу Спірмена	0.909

Загалом, експериментальна частина дослідження передбачала використання реальних часових рядів, що відображають динаміку вартості акцій компаній згідно з даними біржі NASDAQ. До досліджуваних рядів входили цінові дані таких компаній, як CISCO, Netflix, Uber та низки інших (рис. 3.11).

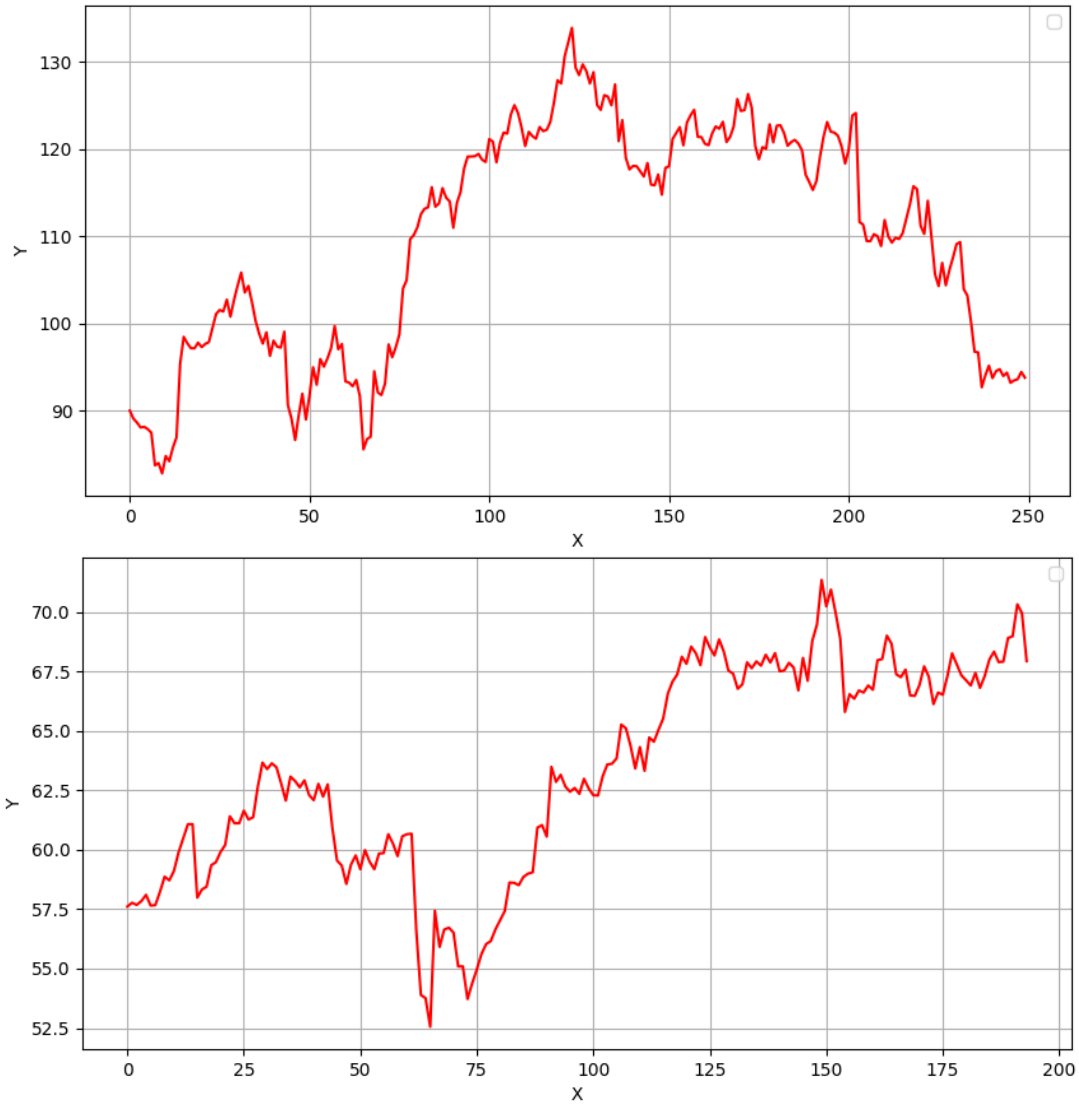
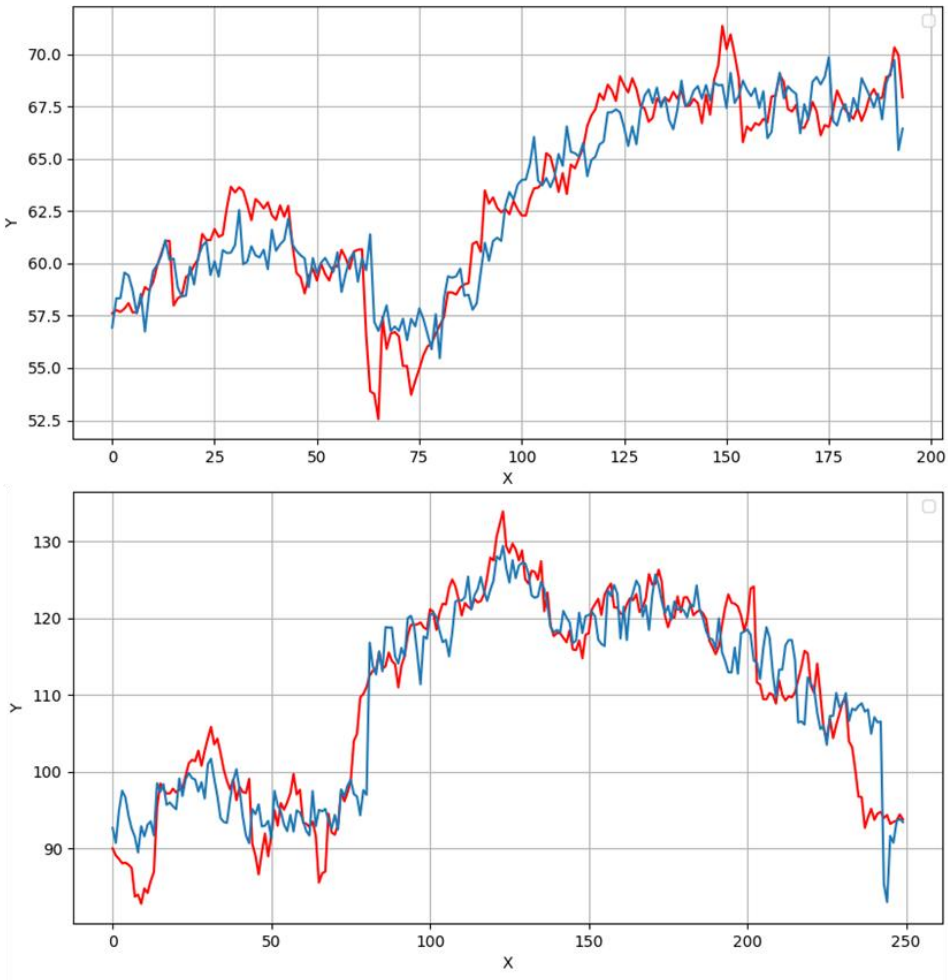


Рисунок 3.11 – Приклади часових рядів обраних для експерементальної частини

У підсумку було опрацьовано 20 часових рядів зазначеної природи походження та їх сегментованих частин. Для кожного з них було отримано позитивні результати, що проявлялося у значеннях обраних метрик (табл. 3.4), які демонстрували високий ступінь відповідності між вхідним і відновленим часовими рядами (рис. 3.12, 3.13).

Це підтверджує здатність запропонованого підходу адекватно відтворювати основні характеристики реальних фінансових часових структур.



3.12 – Порівняння результату процесу відновлення (синій) та оригінального ряду (червоний)

Слід відзначити, що для часових рядів із нестабільними викидами отримані результати виявляються нижчими порівняно з іншими випадками, що відображає підвищену мінливість та складність структури таких рядів (рис. 3.14). Водночас низькі відносні помилки разом із високими коефіцієнтами кореляції свідчать про високу точність відтворення форми та динаміки досліджуваних рядів, демонструючи їхню значну відповідність моделі.

Значення коефіцієнта детермінації R^2 підтверджує достатньо високу пояснювальну здатність моделі та відображає її здатність адекватно передавати загальні закономірності даних (табл. 3.5).

Разом із тим, наявність певних абсолютних помилок вказує на існування потенціалу для подальшого удосконалення, зокрема в частині більш точного відтворення окремих нестандартних значень.

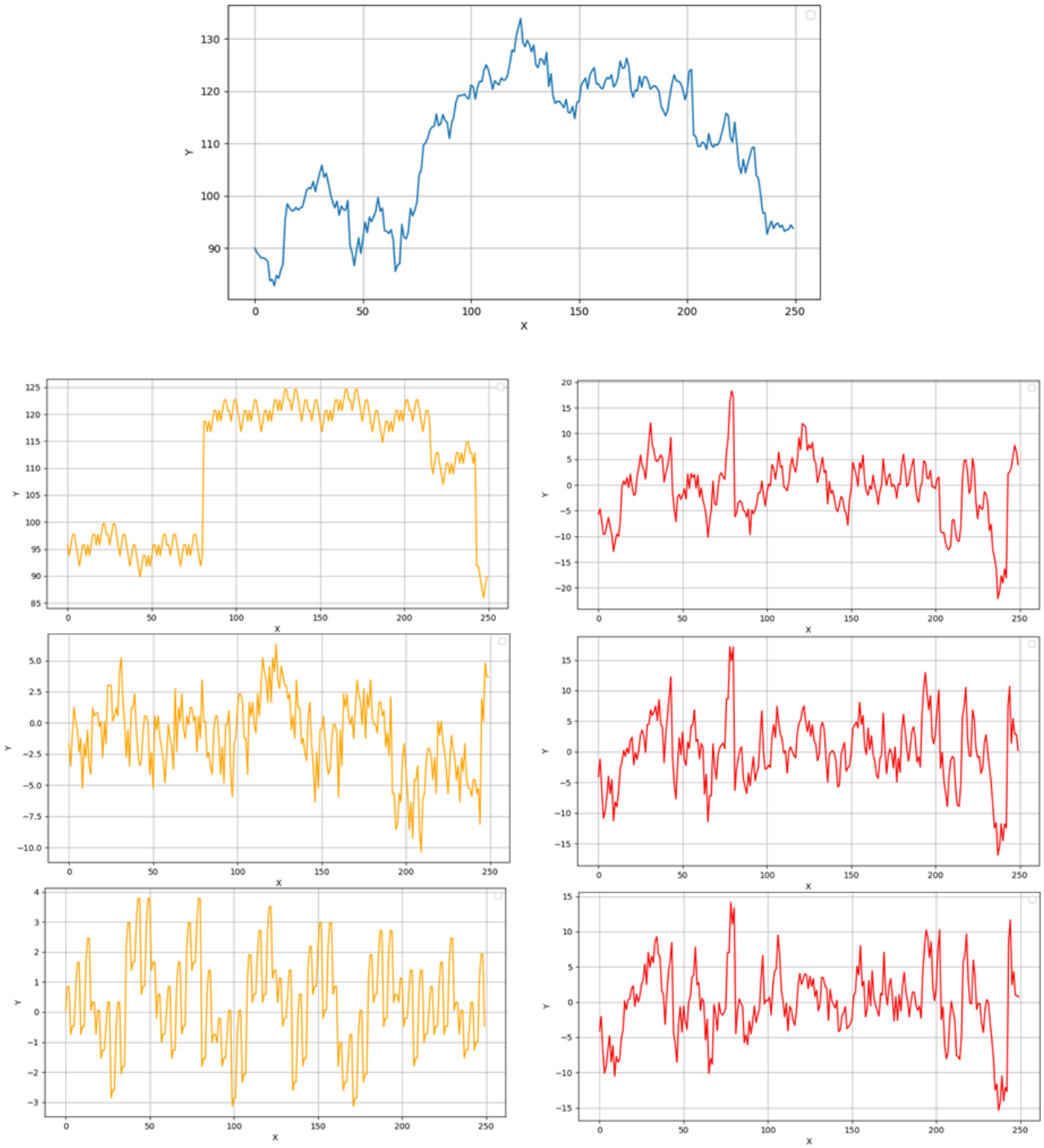


Рисунок 3.13 – Демонстрація ітеративної зміни результату окремої ітерації (помаранчьовий) та її залишку (червоний)

Збільшення кількості ітерацій у процесі відновлення безпосередньо впливає на точність відтворення вихідного ряду. Водночас із поступовим просуванням процесу ефект від кожної наступної ітерації зменшується, що свідчить про поступове насичення процесу та стабілізацію показників точності.

Ця закономірність наочно відображена у табл. 1, де видно, що приріст точності між послідовними ітераціями поступово скорочується.

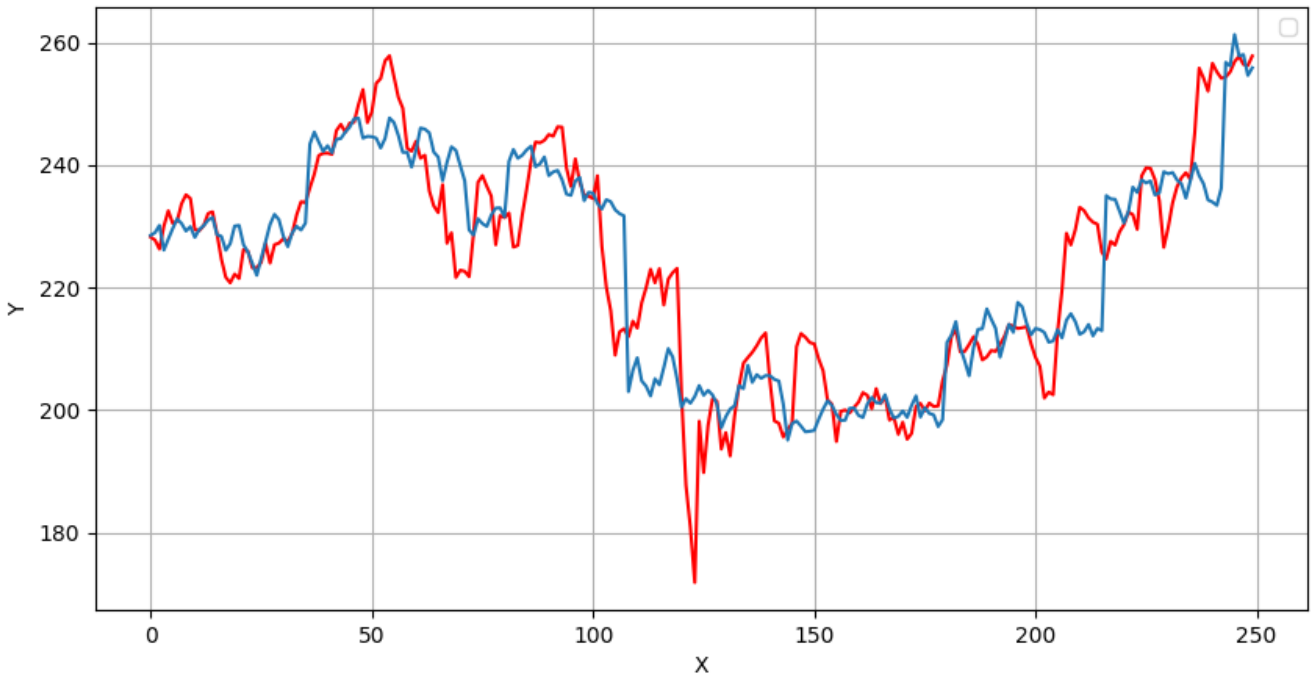


Рисунок 3.14 – Порівняння відновленного (синій) та оригінального (червоний) рядів із аномальними викидами значень

Таблиця 3.5 – Порівняння показників ряду з аномальними значеннями (показники акцій компанії Apple) та без (показники акцій компанії Cisco)

Показник	Значення для ряду акцій Apple	Значення для ряду акцій Cisco
MAE	5.947	1.247
MSE	70.807	2.712
RMSE	8.415	1.647
MAPE	2.67%	1.99%
SMAPE	2.68%	1.99%
R^2	0.780	0.861
Коефіцієнт кореляції Пірсона	0.887	0.928
Коефіцієнт кореляції рангу Спірмена	0.885	0.896

3.4.4 Експеримент 4.2: Дослідження можливостей прогнозування та визначення проблеми її реалізації

Окремим етапом експериментальних досліджень стало впровадження операції прогнозування подальших значень реальних часових рядів на основі віднов-

лених параметрів конструктивних моделей. На цьому етапі було виявлено низку принципових труднощів, пов’язаних із самою природою процесу розгортки L-систем.

Основна проблема полягала у тому, що одна і та сама множина параметрів може породжувати часові ряди різної довжини залежно від кількості ітерацій розгортки. У результаті різні варіанти розгортки формують різні реалізації ряду, що ускладнює безпосереднє використання моделі для прогнозування, оскільки відсутня однозначна відповідність між параметрами та довжиною вихідної послідовності.

Для вирішення цієї проблеми було запропоновано введення додаткового параметра, який визначає довжину розгортки. Згідно з цим підходом, загальна довжина часового ряду формувалася як сума двох складових: довжини навчальної вибірки, на основі якої здійснюється відновлення параметрів, та довжини інтервалу прогнозування. Це дозволяло зафіксувати єдину довжину розгортки та забезпечити формування безперервного ряду, що включає як відомі, так і прогнозовані значення.

Проте результати дослідів показали, що навіть за такої модифікації отримати стабільні та точні прогнози не вдалося. Незважаючи на формальну узгодженість довжини ряду, відхилення між прогнозованими та фактичними значеннями залишалися значними, що свідчить про недостатню адаптованість даного підходу до задачі передбачення.

У ході аналізу було визначено можливі напрями подальших досліджень. Зокрема, встановлено, що точність прогнозування потенційно залежить від довжини інтервалу передбачення: зі збільшенням цього інтервалу може покращуватися узгодженість результатів за рахунок більш повного врахування структури ряду. Аналогічно, збільшення довжини навчальної вибірки сприяє більш точному відновленню параметрів, що також може позитивно впливати на якість прогнозу.

Втім, в межах проведених випробувань зазначені залежності не призвели до отримання стабільно позитивних результатів. Це дозволяє зробити висновок про те, що задача прогнозування на основі відновлених параметрів конструктивних

моделей потребує додаткових досліджень та, ймовірно, розробки спеціалізованих підходів, орієнтованих саме на передбачення, а не лише на відновлення структури ряду.

3.4.5 Експеримент 4.3: Дослідження фрактальних властивостей відновлених стохастичних часових рядів

В межах данного експерименту для низки випробувань, які були виконані у межах Експерименту 4.1, були визначені значення показника Герста та фрактальна розмірність для вхідних та вихідних часових рядів.

Оскільки кожне представлене спостереження є окремим завершеним випробуванням, результати дозволяють оцінити індивідуальну адаптивність моделі до різних структур реальних даних. Основний фокус аналізу зосереджений на здатності алгоритму відтворити фрактальні властивості оригіналу, що виражається через близькість показників H та D.

У випробуваннях, де вхідні часові ряди мали помірну антиперсистентність (H_o у межах 0.42-0.47), модель продемонструвала максимальну точність. Значення розбіжності Δ у таких тестах є мінімальними (зокрема 0.005 та 0.014), що вказує на майже повний збіг структурної складності модельного ряду з оригінальним. Фрактальна розмірність D_m у цих випадках стабільно фіксується на рівні 1.56-1.57, підтверджуючи коректне відтворення щільності коливань.

Проте в окремих випробуваннях зафіксовано зростання розбіжності Δ до рівнів 0.14-0.19. Це свідчить про те, що для певних типів стохастичної динаміки модель схильна змінювати характер ряду. Наприклад, у тестах, де вхідний показник H_o вказував на наявність тренду (понад 0.6), модельний результат H_m зміщувався в зону нижче 0.5. Це призводило до штучного ускладнення траєкторії та зростання модельної фрактальної розмірності D_m порівняно з оригіналом.

У свою чергу, при окремому розгляді показників відновлених рядів у межах ітерацій процесу відновлення не вирізняються сталими межами значень. Для кожного результуючого ряду ітерації значення показника Герста та відповідної фрактальної розмірності суттєво відрізняються від показника попередньої (табл. 3.6).

Таблиця 3.6 – Порівняння зміни показника Герста для локальних результатів

Номер ітерації	Значення показника Герста	Різниця із значенням попередньої ітерації
1	0.363	-
2	0.571	0.208
3	0.493	-0.078
4	0.55	0.057
5	0.505	-0.045
6	0.785	0.28

На відміну від поодиноких результатів у межах окремої ітерації, акумулятивний результуючий часовий ряд, що формується як сума результатів усіх попередніх етапів, має характерну особливість: локалізацію показника Герста. У міру збільшення номера ітерації значення цього показника демонструє дедалі меншу варіативність, поступово стабілізуючись.

Таку динаміку можна кваліфікувати як мінімізацію структурних змін, що накопичуються протягом перебігу процесу відновлення. Це свідчить про те, що кожна наступна ітерація справляє все менший вплив на загальну фрактальну архітектуру результуючого ряду, приводячи його до стійкого фінального стану (табл. 3.7).

Таблиця 3.7 – Порівняння зміни показника Герста для результатів ітерації

Номер ітерації	Значення показника Герста	Різниця із значенням попередньої ітерації
1	0.403	-
2	0.491	0.089
3	0.554	0.063
4	0.595	0.040
5	0.576	-0.018
6	0.580	0.004
7	0.591	0.011
8	0.596	0.005
9	0.599	0.003

Висновки до третього розділу

Узагальнюючи результати проведених досліджень, слід зазначити, що верифікація запропонованого конструктивно-продукційного підходу здійснювалася шляхом поетапного ускладнення об’єктів аналізу. Така організація експериментів дозволила всебічно оцінити ефективність алгоритму, починаючи від ідентифікації

параметрів ідеалізованих детермінованих структур і закінчуючи апроксимацією складних реальних процесів.

На етапі Експерименту 1.1 було проведено 100 досвідів із використанням простих правил підстановки для синтетичних детермінованих рядів. Отримані результати підтвердили фундаментальну спроможність методу до точного відновлення як лінгвістичної структури L-системи, так і її числових характеристик, що стало базовим підтвердженням коректності алгоритму.

У межах Експерименту 1.2 дослідження було розширено до 100 випробувань із застосуванням ускладнених правил, що містили пробільні символи. Це дозволило продемонструвати високу адаптивність методу до розріджених структур даних, де алгоритм успішно ідентифікував параметри моделі в умовах переривчастої генерації символів.

Експеримент 1.3 був присвячений аналізу моделей із множинними правилами та складною аксіомою, для чого було виконано 60 випробувань. У ході випробувань було зафіксовано суттєве зростання обчислювальної складності: максимальна кількість ітерацій для успішного відновлення зросла до 39346, що вказує на пряму залежність між структурною розгалуженістю моделі та часом її ідентифікації.

На другому етапі здійснювалася оцінка прогнозної здатності відновлених моделей на базі 15 часових рядів довжиною 500 точок. Порівняння за метриками MAE та MAPE з класичними методами (ARIMA) та нейромережами (LSTM, CNN-LSTM) довело перевагу запропонованого підходу, оскільки ідентифікація внутрішньої логіки процесу забезпечила точніші результати передбачення, ніж стандартна апроксимація значень.

Експеримент 3 охоплював 50 незалежних випробувань із синтетичними стохастичними рядами для перевірки стійкості методу до шуму. Було встановлено, що похибка відновлення числових показників не перевищує 1.5% від математичного очікування, а алгоритм демонструє характерну властивість мінімізувати дисперсію, що забезпечує формування стабільних структур у динаміці ряду.

На етапі Експерименту 4.1 проводилося відновлення параметрів для реальних фінансових рядів акцій Apple та Cisco (NASDAQ). Застосування методу ітеративної декомпозиції дозволило досягти високої точності відтворення історичної динаміки: показник MAPE для Apple склав 2.67%, а для Cisco – 1.99%. Аналіз за комплексом із 8 метрик (включаючи R^2 та коефіцієнти кореляції Пірсона і Спірмена (понад 0.98) підтвердили, що модель ідентифікує основну структурну компоненту ряду, а процес ітерацій демонструє ефект насичення після досягнення межі структурної ідентифікації.

На етапі Експерименту 4.2 здійснювалася апробація можливостей використання відновлених моделей для прогнозування реальних часових рядів. Результати виявили критичне обмеження, пов'язане з природою фрактальної розгортки: проблема неоднозначності довжини генерації призвела до нестабільності прогнозів на інтервалах за межами навчальної вибірки. Навіть введення додаткового параметра регулювання довжини не дозволило досягти точності, порівнянної з детермінованими рядами, що свідчить про необхідність подальшої розробки механізмів стабілізації розгортки для стохастичних емпіричних даних.

На етапі Експерименту 4.3 були досліджені значення фрактальних показників реальних вхідних та відновлених часових рядів разом з їх поступовими змінами в межах ітерацій процесу відновлення. Метод демонструє високу індивідуальну адаптивність до вхідних параметрів H та D (точність $\Delta = 0.005$), попри чутливість до сильно трендових сигналів. Ключовою особливістю є контраст між локальною та загальною динамікою: якщо в межах окремих ітерацій спостерігається хаотичність показників (різниця до 0.28), то в акумулятивному ряді відбувається прогресуюча стабілізація. Мінімізація структурних змін до рівня 0.003 на фінальних етапах підтверджує, що накопичення ітерацій нівелює похибки окремих випробувань.

Матеріали розділу опубліковані у роботах: [75, 76, 99, 100, 103, 104].

РОЗДІЛ 4
РЕАЛІЗАЦІЯ ПРОГРАМНОГО РІШЕННЯ ДЛЯ АВТОМАТИЗАЦІЇ ВІДНОВЛЕННЯ КОНСТРУКТИВНОЇ МОДЕЛІ ЧАСОВИХ РЯДІВ

4.1 Постановка завдання для автоматизації процесу відновлення

Відповідно до визначених теоретичних положень та практичних підходів, з метою автоматизації процесу проведення експериментальних досліджень була визначена необхідність розробки спеціалізованого програмного забезпечення, яке забезпечувало реалізацію алгоритмів відновлення параметрів конструктивної моделі заданого часового ряду. Розроблене програмне рішення дозволило формалізувати та автоматизувати процес обробки експериментальних даних, а також створило можливість систематичного дослідження ефективності запропонованого підходу.

Крім того, створене програмне забезпечення забезпечило автоматизацію масових експериментальних випробувань, що мало особливе значення на початкових етапах дослідження. Це дозволило виконати первинну верифікацію запропонованого методу, оцінити його працездатність, а також визначити межі коректного застосування розробленого підходу. Проведення серії автоматизованих експериментів сприяло накопиченню статистичних даних щодо результатів відновлення параметрів конструктивної моделі та дозволило більш обґрунтовано оцінити ефективність методу для часових рядів різної природи.

Слід зазначити, що в процесі дослідження часових рядів різного походження підхід до програмної реалізації поступово змінювався та вдосконалювався. Такі зміни були зумовлені необхідністю адаптації розроблених алгоритмів до специфічних властивостей досліджуваних часових рядів, а також прагненням підвищити загальну ефективність програмної системи. Зокрема, значна увага приділялася оптимізації обчислювальних процедур та усуненню вузьких місць у процесі відновлення параметрів конструктивної моделі. Це дозволило підвищити продуктивність програмного рішення, зменшити витрати обчислювальних ресурсів та

забезпечити стабільність функціонування системи при обробці значних обсягів експериментальних даних.

Таким чином, розроблене програмне забезпечення стало важливим інструментом проведення дослідження, оскільки забезпечило автоматизацію експериментальних процедур, підвищило відтворюваність отриманих результатів та створило передумови для подальшого аналізу ефективності методів відновлення параметрів конструктивних моделей часових рядів.

4.2 Визначення загального підходу до проектування

Використання генетичного алгоритму як основної конструкції для реалізації процесу відновлення параметрів дозволило виділити основні логічні сутності та компоненти системи, визначити необхідний набір операцій, а також сформувати загальну структуру процесу відновлення. Такий підхід сприяв формалізації ключових етапів обробки даних та створив основу для подальшого проєктування програмної системи.

Відповідно до цього початковий підхід до проектування програмного додатку базувався на загальних принципах об’єктно-орієнтованого дизайну (Object-Oriented Design, OOD) [38, 95] та доменно-орієнтованого дизайну (Domain-Driven Design, DDD) [24, 46]. Використання зазначених підходів дозволило побудувати логічне відображення процесів предметної області у структурі програмної системи за допомогою відповідних програмних абстракцій.

Застосування принципів OOD та DDD забезпечило можливість формування моделі системи, яка відображає ключові елементи процесу відновлення параметрів конструктивної моделі часового ряду, а також взаємозв’язки між ними. Крім того, використання зазначених підходів дозволило досягти необхідного рівня гнучкості архітектури програмного рішення, що є важливим з огляду на подальший розвиток дослідження. Зокрема, це створило передумови для внесення змін різного масштабу – від модифікації окремих алгоритмічних компонентів до розширення функціональних можливостей системи.

Таким чином, використання принципів об’єктно-орієнтованого та доменно-орієнтованого дизайну стало важливим етапом у процесі проєктування програмної системи, оскільки дозволило забезпечити структурованість програмної архітектури, підвищити її адаптивність та спростити подальшу еволюцію програмного рішення.

4.2.1 Перелік основних логічних сутностей

Відповідно до зазначених принципів у системі були виділені основні доменні сутності, а саме хромосома та популяція.

Хромосома виступає логічною абстракцією, що відповідає визначеній на теоретичному рівні структурі параметрів конструктивної моделі та зберігає повний набір її параметрів.

Популяція є сукупністю хромосом поточної ітерації алгоритму та виступає агрегатом [24] у термінах доменно-орієнтованого проектування. Вона відповідає за збереження та керування поточним набором хромосом, а також надає відповідний програмний інтерфейс (API) для виконання операцій над ними. Впровадження даної сутності дозволило підвищити рівень інкапсуляції, що, у свою чергу, забезпечило незалежність між різними шарами програмного додатка.

Як основні операції системи були визначені класичні оператори генетичного алгоритму, а саме: кросинговер (crossover), мутація (mutation) та селекція (selection). Перші дві операції реалізовувалися на рівні окремих хромосом відповідно до команд, що формувалися популяцією. Операція селекції, у свою чергу, була реалізована на рівні самої популяції та відповідала за формування нового покоління хромосом. Загальний відсоток хромосом, що відсіювалися на кожній ітерації, визначався заздалегідь.

4.3 Монолітна реалізація програмного додатку

На початковому етапі досліджень випробування та експериментальні дослідження проводилися на основі синтетичних детермінованих часових рядів різної складності структури [76]. Основною особливістю таких рядів є їх передбачуваність та відносна простота структурної організації, що робить їх зручними для ви-

користання на початкових етапах перевірки нових методів. Завдяки цьому вони дозволяють із достатнім рівнем достовірності оцінити коректність роботи алгоритмів та загальної логіки побудованого процесу.

Відповідно до цього, окрім основної мети – відновлення параметрів конструктивної моделі, дослідження на даному етапі були спрямовані насамперед на валідацію самого процесу відновлення та перевірку працездатності запропонованої концепції. Таким чином, зазначений етап дослідження фактично відповідав підходу Proof of Concept, метою якого є підтвердження принципової можливості реалізації запропонованого методу.

З урахуванням визначених вимог початкова версія програмного додатка з архітектурної точки зору являла собою простий монолітний застосунок, реалізований у вигляді однопотокової (single-thread) системи (рис. 4.1).

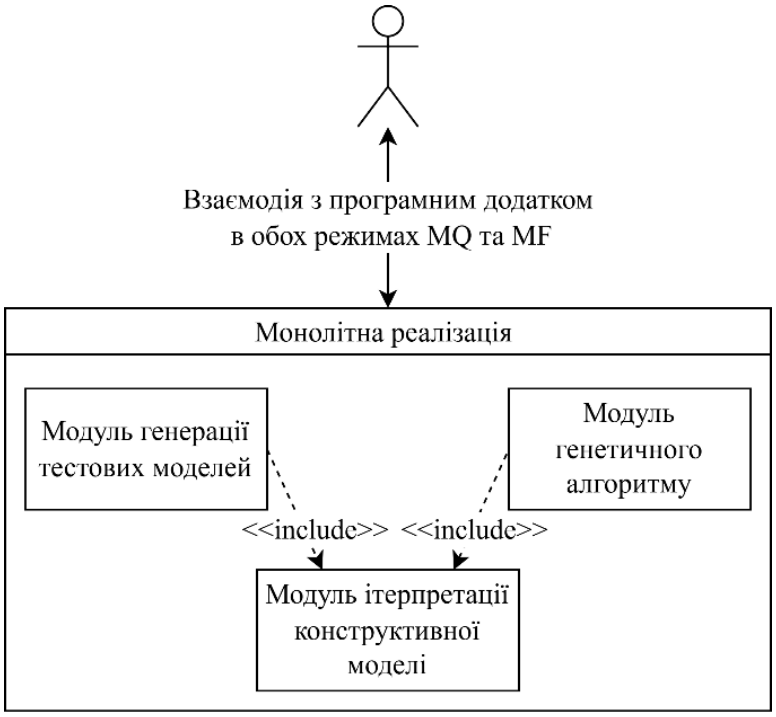


Рисунок 4.1 – Загальна діаграма монолітного підходу до реалізації

Архітектура системи включала набір модулів, які забезпечували виконання основних етапів експериментального процесу (рис. 4.2), зокрема:

- підготовку тестових даних;
- виконання експериментальних обчислень;
- валідацію отриманих результатів;

- діагностику та моніторинг роботи системи.

Кожна з визначених сутностей, відповідно до покладених на неї функцій, керувала окремими етапами загального процесу, передаючи або делегуючи керування потоком виконання іншим компонентам системи (рис. 4.3). Така організація взаємодії між модулями забезпечувала послідовне виконання етапів експерименту та дозволяла контролювати хід обчислювального процесу.

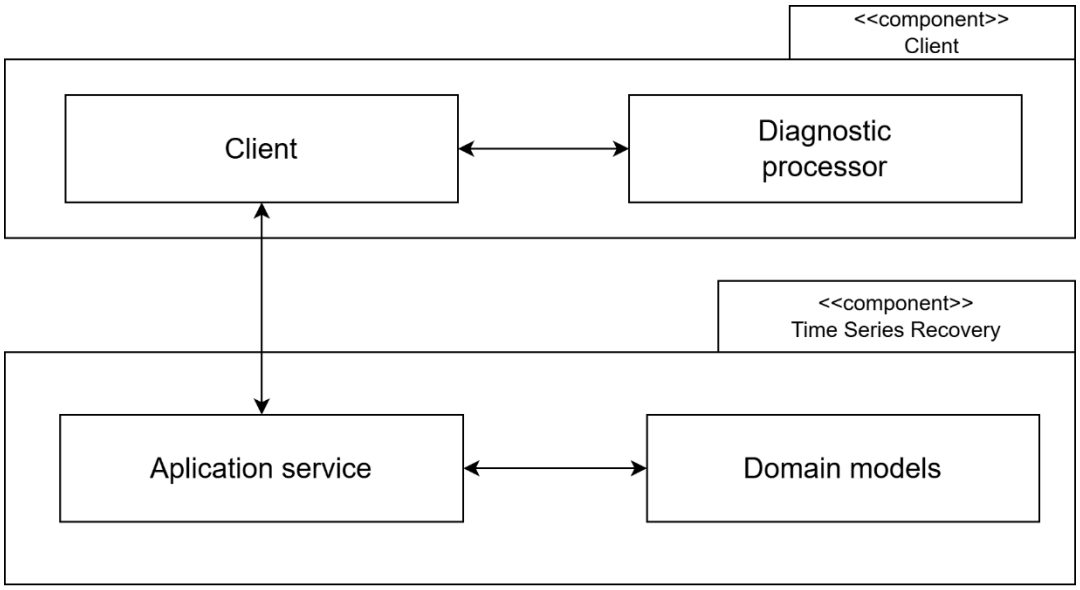


Рисунок 4.2 – Компонентна модель монолітного програмного додатку

На основі даної ітерації реалізації програмного додатка були визначені ключові особливості автоматизації експериментального процесу, а також виявлені основні вузькі місця обчислювальної частини системи. Основними результатами цього етапу стали підтвердження валідності побудованого процесу відновлення та демонстрація можливості його масового застосування при роботі з детермінованими часовими рядами [76].

4.4 Мульти-агентний підхід для обробки стохастичних часових рядів

Відповідно до поставлених цілей, початкова версія реалізації дозволила провести попереднє дослідження особливостей процесу відновлення параметрів та визначити подальші напрями розвитку дослідження. Одним із ключових напрямів було визначено оптимізацію обчислювального процесу та забезпечення можливості його масштабування.

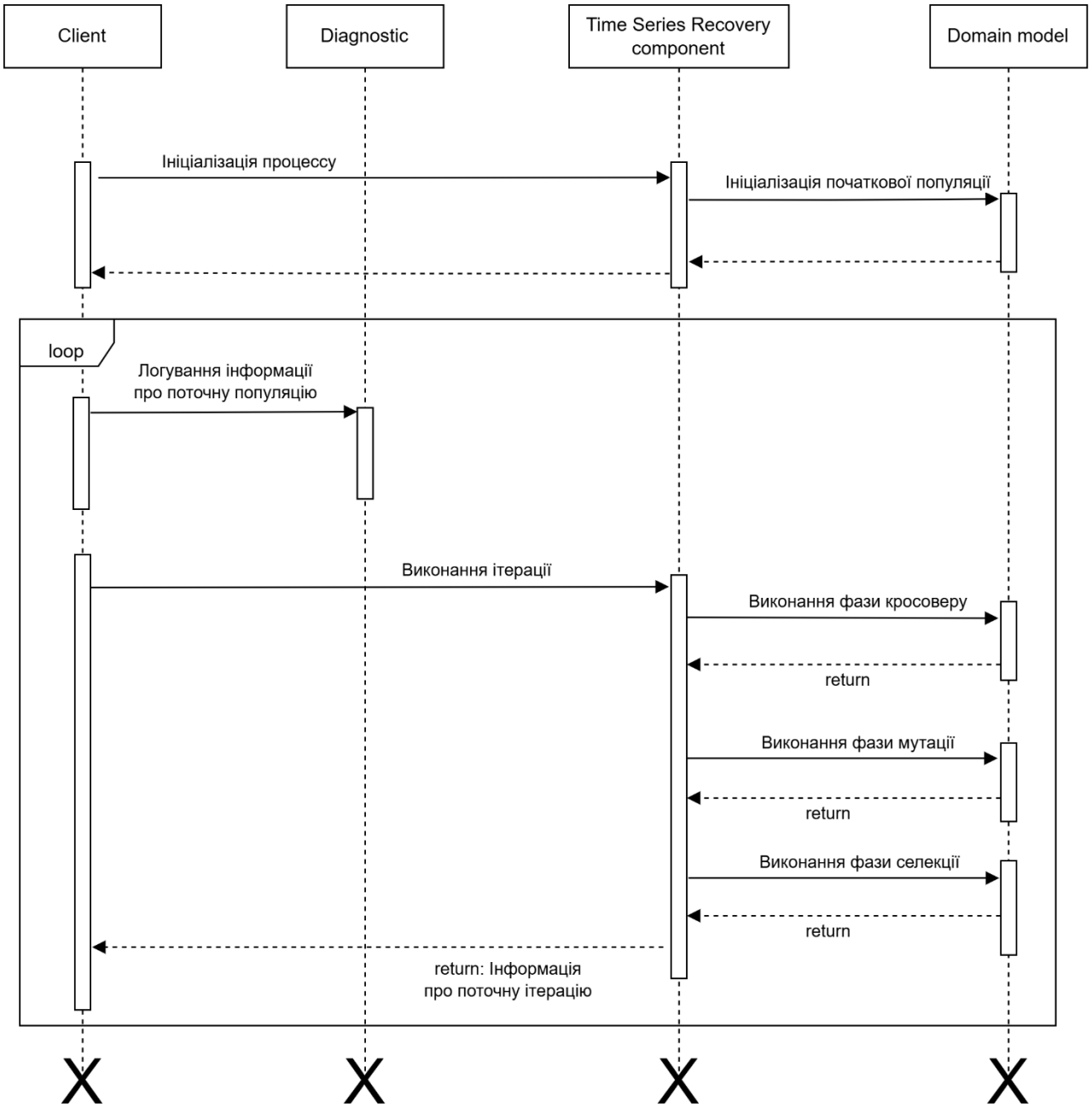


Рисунок 4.3 – UML діаграма послідовності роботи монолітного програмного додатку

Зокрема, для підвищення ефективності роботи генетичного алгоритму необхідним є збільшення розміру популяції, що сприяє покращенню якості пошуку рішень. Водночас це призводить до суттєвого зростання обчислювального навантаження на кожному етапі алгоритму, особливо під час обчислення функції пристосованості (fitness function).

Детальніше опис запропонованого підходу до організації обчислень та результати його порівняння з монолітною реалізацією наведено у [75].

4.4.1 Визначення топології зв'язків обчислювальних сутностей

Використання однопотокової моделі виконання не дозволяє ефективно використовувати обчислювальні ресурси, оскільки обмежується лише вертикальним масштабуванням [4, 11, 38]. Такий підхід ускладнює обробку великих популяцій і знижує загальну продуктивність системи.

У зв’язку з цим, для забезпечення ефективного проведення експериментів і розширення можливостей дослідження різних типів часових рядів, виникла необхідність переходу до горизонтального масштабування [1, 14, 38]. Такий підхід дозволяє розподілити обчислювальне навантаження між кількома обчислювальними вузлами або потоками, що сприяє підвищенню продуктивності системи та забезпечує можливість обробки більш складних і ресурсоємних задач.

Як базовий підхід до реалізації горизонтального масштабування було обрано агентно-орієнтовану модель обчислень, в межах якої процес відновлення параметрів конструктивної моделі інкапсулюється в окремій обчислювальній сутності – агенті, що є складовою обчислювального кластера (рис. 4.4), де кожен з них має відповідну зону відповідальності (табл. 4.1) та відповідні зв’язки (табл. 4.2).

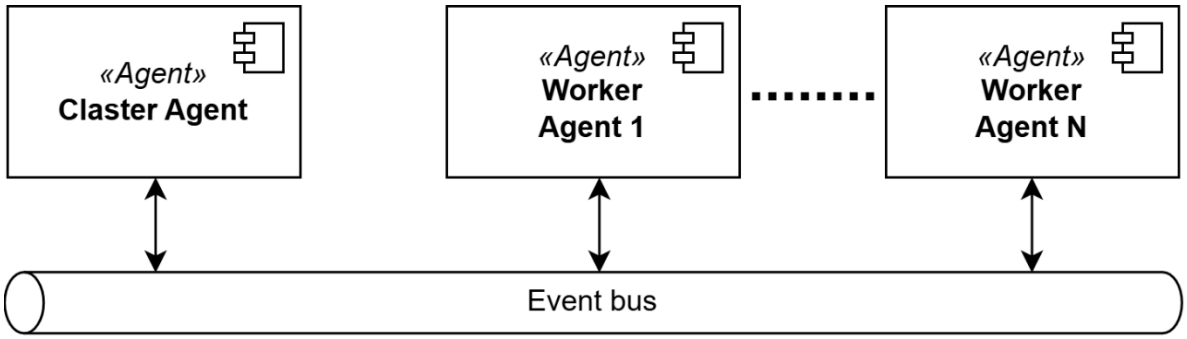


Рисунок 4.4 – Архітектура кластеру агентно-орієнтованого підходу

У процесі розробки архітектури було розглянуто різні підходи до організації топології системи агентів, зокрема “master-slave” та “island” [2]. Модель “master-slave” передбачає делегування окремих етапів обчислень незалежним агентам-виконавцям під керуванням одного або кількох головних агентів. Однак даний підхід виявився недостатньо ефективним для поставленої задачі, оскільки алгоритм не містив достатньо складних підзадач, виділення яких в окремі агенти забезпечило б суттєвий приріст продуктивності.

Таблиця 4.1 – Перелік агентів та їх обов‘язків

Тип агента	Обов’язки	Область знань
Користувацький агент	Приймання та валідація команд користувача з подальшою передачею кластерному агенту	Інформація про користувача, інформація про кластерний агент
Кластерний агент	Веде облік доступних агентів, ініціює обчислювальний процес, керує роботою робочих агентів	Інформація про агентів, конфігурації робочих агентів, специфікація задачі
Робочий агент	Виконує процес відновлення параметрів конструктивної моделі часового ряду	Конфігурація, специфікація задачі
Міграційний агент	Керує процесом міграції хромосом між генетичними пулами відповідно до заданих правил	Буфер генетичного пулу, поточний стан обчислень для кожного агента

Таблиця 4.2 – Відношення між агентами кластеру

Взаємозв’язок	Опис
Користувацький агент – кластерний агент	Отримання команди на запуск обчислень та даних часового ряду, для якого необхідно визначити параметри моделі. Після завершення обчислень кластерний агент передає користувацькому агенту отриманий результат.
Кластерний агент – робочий агент	Локальна реалізація принципу «master–slave». Кластерний агент ініціює роботу всіх робочих агентів, надаючи їм необхідний набір вхідних даних. Після досягнення заданої точності робочий агент надсилає повідомлення кластерному агенту, який, у свою чергу, зупиняє роботу всіх інших робочих агентів.
Кластерний агент – міграційний агент	Кластерний агент передає команди запуску та зупинки обчислювального процесу міграційному агенту.

Натомість підхід “island” передбачає, що кожен агент реалізує повний цикл відновлення конструктивної моделі та функціонує як автономна обчислювальна одиниця. Масштабування в даному випадку досягається шляхом збільшення кіль-

кості агентів та встановлення зв’язків між ними. Додатковою перевагою є можливість використання асинхронної моделі обміну повідомленнями, що підвищує ефективність взаємодії між компонентами системи.

За результатами аналізу було обрано саме підхід “island”. Його реалізація дозволила представити кожного агента як незалежну обчислювальну сутність, що забезпечило паралелізацію процесу відновлення. Крім того, така архітектура дозволила застосувати асинхронний механізм обміну даними на основі шини повідомлень за принципом *“fire-and-forget”* [20, 28].

4.4.2 Визначення підходу до обміну інформацією між агентами

Кожен агент, незалежно від інших, виконував процес відновлення конструктивної моделі на основі отриманих даних із використанням генетичного алгоритму. Завдяки розподілу обчислювального процесу між агентами було впроваджено механізм міграції, тобто обміну найкращими представниками поколінь між агентами в процесі роботи.

Було розглянуто два основні підходи до реалізації міграції: peer-to-peer та міграція через допоміжний банк хромосом. Перший підхід передбачає безпосередній обмін хромосомами між агентами за визначеною схемою або у довільному порядку. Його основним недоліком є прискорення процесу виродження популяції, що негативно впливає на різноманітність рішень і, як наслідок, на ефективність відновлення.

Альтернативний підхід, заснований на використанні допоміжного банку хромосом, передбачає делегування процесу міграції окремому компоненту системи (рис. 4.5). Це дозволяє застосовувати додаткові інтелектуальні механізми, зокрема методи нечіткої логіки або нейронні мережі, для оптимізації процесу обміну. У результаті було обрано саме цей підхід, оскільки він забезпечує кращу декомпозицію функцій між компонентами системи та підвищує ефективність розподіленого генетичного алгоритму [75].

Застосування зазначеного підходу дозволило перебудувати процес обчислень із урахуванням можливості збільшення розміру популяції. У загальному випадку кожен агент виконував обробку окремої популяції, тоді як обмін даними реалізовувався через спеціалізований міграційний компонент.

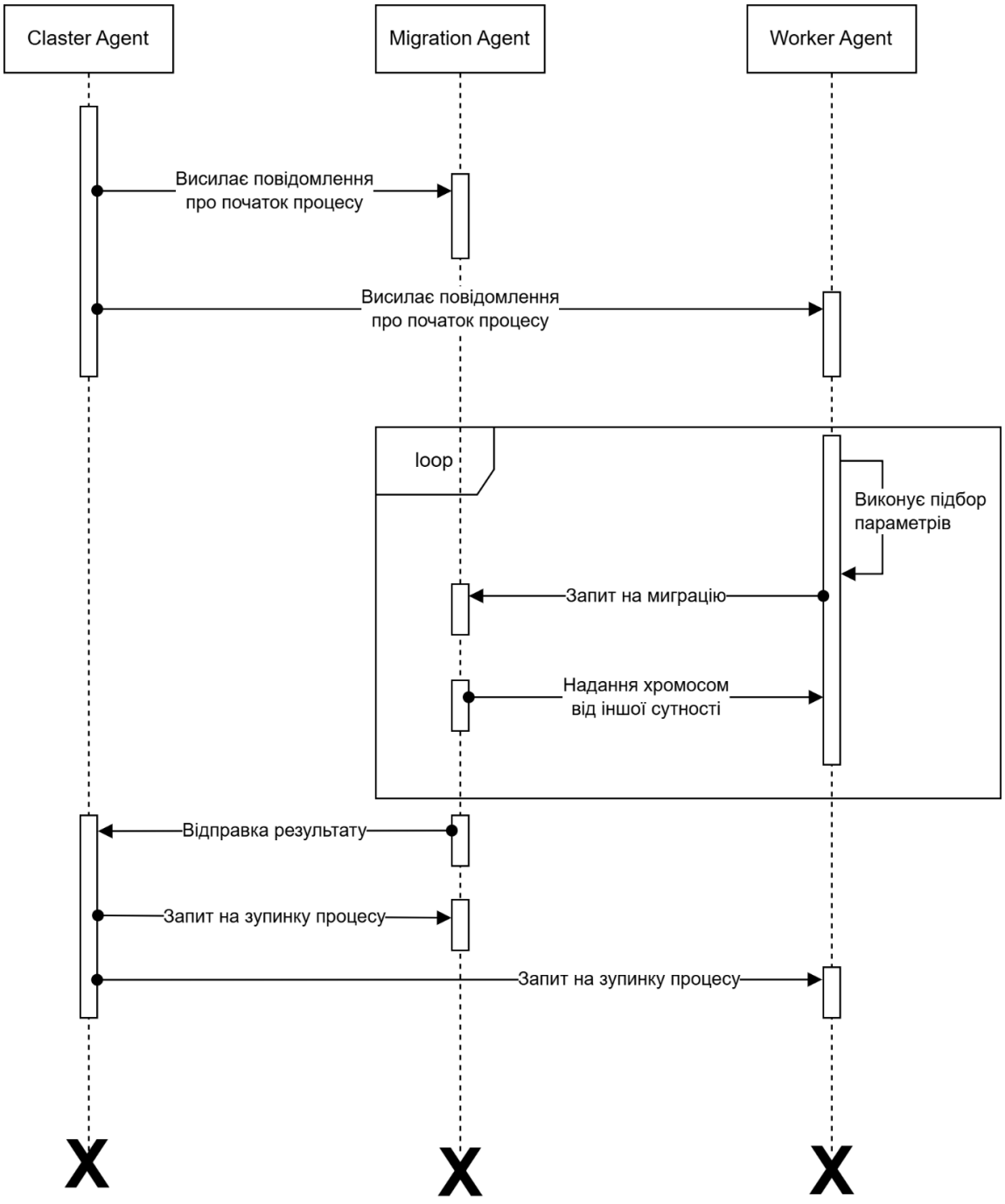


Рисунок 4.5 – UML діаграма взаємодії агентів під час міграції між популяціями

4.5 Вирішення проблеми масштабування окремих етапів ітерації

На основі проведених досліджень подібний підхід показав ефективність роботи з детермінованими часовими рядами, а також забезпечив можливість обробки стохастичних часових рядів для тестування методу у відповідних умовах [75].

Незважаючи на отримані позитивні результати, для роботи зі стохастичними рядами мультиагентна реалізація не є оптимальною. Основною проблемою є нерівноцінність операцій конструктора C_{RS}, а саме складність обчислення показника фітнесу (життєздатності) $q(X_n)$ для стохастичних рядів. Для детермінованих рядів складність операції для всієї популяції описується як $n \cdot k$, де n– кількість хромосом у популяції, а k – довжина ряду.

4.5.1 Стратегія роздільного масштабування та модифікація генетичних операцій

У стохастичному випадку складність критично зростає через варіативність, що призводить до загальної складності $N \cdot M \cdot n \cdot k$, де N– кількість синтетичних рядів, згенерованих однією хромосомою, а M– кількість модельованих рядів для порівняння. Результат обчислення фітнесу базується на сумі найменших відхилень між синтетичними та модельованими рядами.

Відповідно до зазначених особливостей, ця операція є «вузьким місцем» процесу. Попередні підходи страждали від низької часової ефективності через ітеративну природу генетичного алгоритму, що спонукало до пошуку рішень у сфері горизонтального масштабування, як було зазначено при розробці мультиагентного підходу.

Для вирішення визначеної проблеми було прийнято рішення до незалежного масштабування окремих елементів конструктора C_{RS}, а саме алгоритмів $\{B_3, B_4, B_5\}$:

$$\left\langle \begin{matrix} \beta_\theta \rightarrow \langle B_3 \cdot B_4 \cdot B_5 \rangle_{sub_i} \cdot \beta_\theta \\ \beta_\theta \rightarrow \varepsilon \end{matrix} \right\rangle, \tag{4.1}$$

де масштабування реалізується за рахунок виділення підпопуляцій sub_i, для кожної з яких виділяється окремий вузол обчислень, інтегрований в правило підстановки β[76].

В межах мультиагентного середовища логічна структура робочого агента складалася з двох компонентів: операцій генетичного алгоритму (кросинговер, мутація, селекція) та оцінки конструктивної моделі (рис. 4.6).

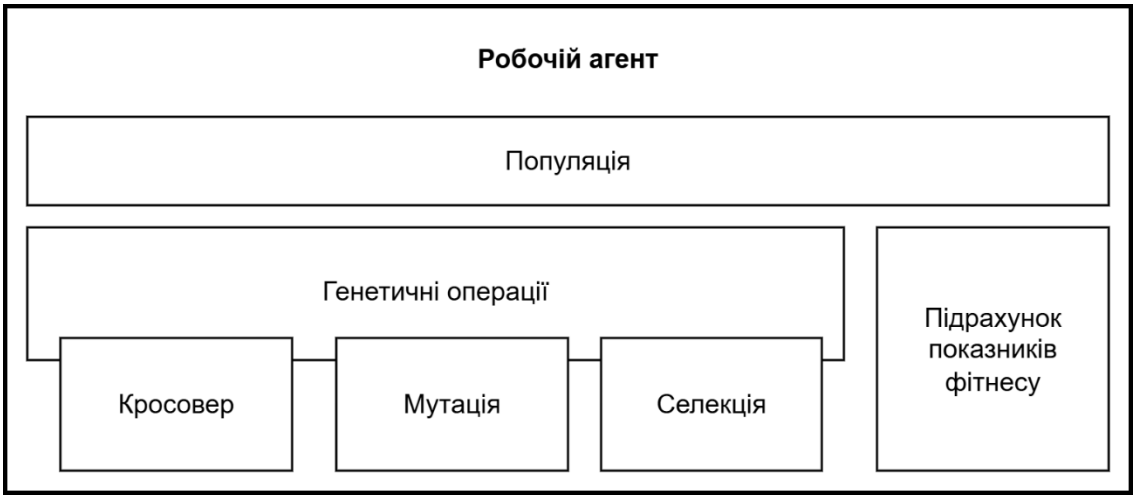


Рисунок 4.6 – Декомпозиція основних елементів обчислювальної сутності

Для роздільного масштабування логіку оцінки фітнесу було виділено як окрему сутність. При цьому кожній сутності для генетичних операцій має відповідати декілька сутностей для оцінювання (рис. 4.7).

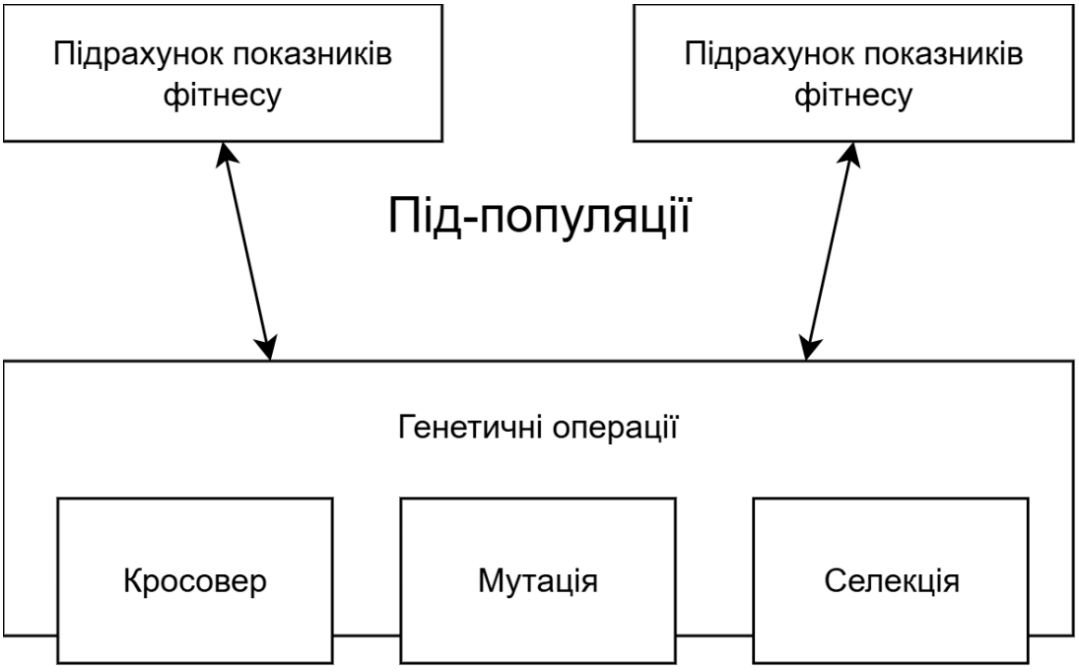


Рисунок 4.7 – Розділення компонентів обчислювальної сутності для окремого маштабування

Необхідність впровадження розбиття та агрегації розширює зону відповідальності алгоритмів $B_7\left|_P^P\right.$, $B_8\left|_P^P\right.$, а саме:

- $B_7\left|_P^{P,\{P_{sub_i}\}}\right.$ – забезпечує розширення популяції новими хромосомами та виділяє під-популяції;
- $B_8\left|_P^{\{P_{sub_n}\}}\right.$ – разом із операцією селекції виконує агрегацію під-популяції.

Такий розподіл дозволяє розпаралелити процес обчислення показників життєздатності. Залежно від розміру підпопуляцій досягається зменшення загальної кількості послідовних операцій і підвищення швидкодії. Для максимальної ефективності кількість сутностей розрахунку має відповідати кількості підпопуляцій, що дає змогу виконувати обчислення за один прохід; інакше хромосоми очікуватимуть своєї черги.

Відповідно до поставленої мети, були визначені наступні сутності-сервіси:

- генетичний сервіс (Genetic service) – відповідає за реалізацію операцій кросинговеру та мутації. Також ініціалізує початкову популяцію. Імплементує алгоритми B_1, B_2;
- сервіс оцінки показників фітнесу (Evaluation service) – відповідає за обчислення показників фітнесу хромосом популяції. Імплементує алгоритми B_3, B_4, B_5;
- сервіс агрегації (Aggregation service) – відповідає за об’єднання підпопуляцій та виконання операції селекції перед передачею даних на нову ітерацію. Імплементує алгоритми B_6, B_7, B_8.

4.5.2 Загальний архітектурний підхід та мультисервісна парадигма

Було розглянуто два підходи до реалізації комунікації: синхронний та асинхронний. Синхронний підхід на базі HTTP/2 [55] або gRPC [30] забезпечував надійний транспорт, проте створював складну топологію (рис. 4.8).

Даний підхід забезпечував надійний та простий механізм пересилки та виключав необхідність впровадження складної мануальної синхронізації. Загальним

чином він представляє собою розширення вже зазначеної мультиагентної реалізації.

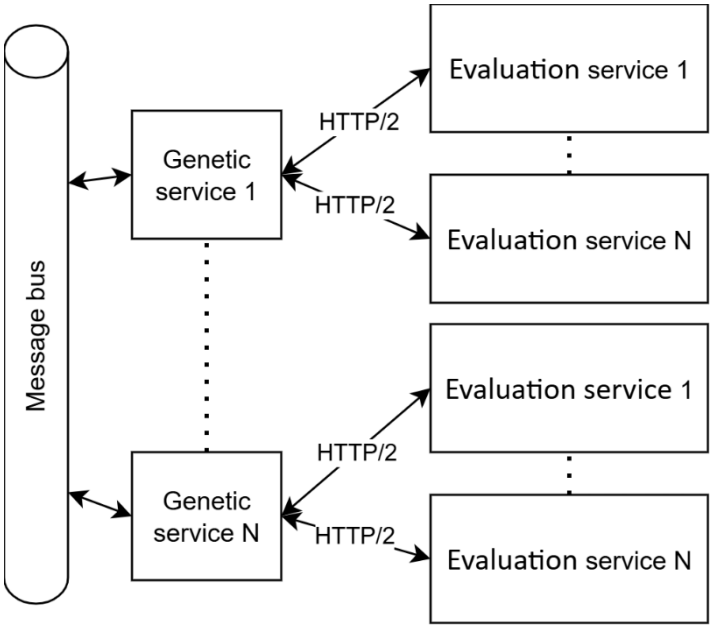


Рисунок 4.8 – Горизональне маштабування із використанням синхронного підходу до коммунікації

Втім, основним його недоліком є складність самої топології, яка б забезпечувала легкість масштабування та гнучкі залежності між обчислювальними сутностями. Подібна топологія при використанні потребує окремих налаштувань для кожного робочого кластера, що нівелює можливості простого горизонтального масштабування навіть у порівнянні з мультиагентним підходом.

Як альтернативу було спроєктовано асинхронну мультисервісну архітектуру, де обмін даними відбувається через шину повідомлень (рис. 4.9). Це дозволяє відокремити процеси один від одного та полегшити масштабування кожного елемента. Для відсутності колізій кожен сервіс підписаний на відповідну чергу.

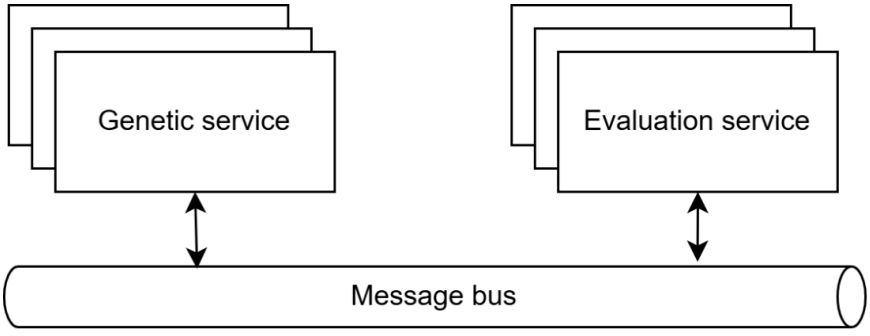


Рисунок 4.9 - Горизональне маштабування із використанням асинхронного підходу до коммунікації

Сам процес комунікації є циклічним, а саме, сервіси обмінюються інформацією, допоки не буде знайдено рішення або не буде здійснена мануальна зупинка процесу.

Архітектура побудована так, щоб кожен генетичний сервіс міг координувати роботу декількох сервісів оцінювання, які обробляють підпопуляції паралельно. Це дозволяє системі динамічно реагувати на зміну навантаження, додаючи обчислювальні вузли для обробки великих популяцій, що особливо критично для стохастичних рядів.

4.5.3 Реалізація програмного додатку на базі провайдеру хмарного середовища AWS

Реалізація прототипу була виконана на платформі Amazon Web Services із використанням сервісів AWS Lambda [60, 65] та Amazon SQS [87]. Архітектура системи побудована таким чином, що основна логіка генетичних операцій і обчислення фітнес-функції розподілена між окремими функціональними компонентами, реалізованими у вигляді незалежних сервісів (рис. 4.10).

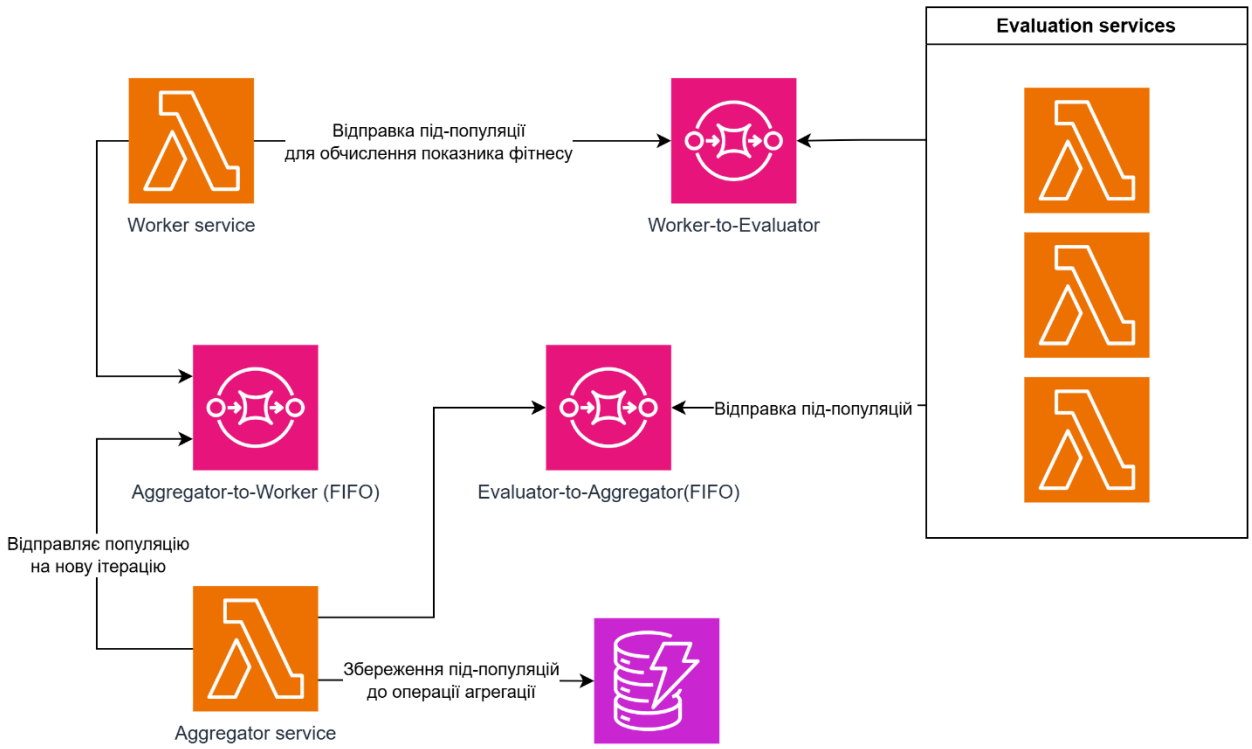


Рисунок 4.10 – Архітектура програмного рішення на базі компонентів AWS

У процесі розробки було обрано serverless-підхід до розгортання та хостингу компонентів системи. Такий вибір зумовлений необхідністю забезпечення високої швидкості внесення змін і спрощення процесу розробки на початкових етапах дослідження. На цьому етапі прикладне рішення орієнтоване насамперед на гнучкість і швидкість ітерацій, без необхідності виконання додаткових налаштувань інфраструктури, які зазвичай є обов'язковими на етапі фінальної інтеграції програмного продукту. Саме тому було прийнято рішення використовувати зазначений підхід, незважаючи на загальні рекомендації щодо обмежень використання сервісу AWS Lambda.

Ще однією перевагою AWS Lambda є вбудований механізм автоматичного масштабування [65], який дозволяє системі динамічно збільшувати або зменшувати кількість одночасно виконуваних функцій залежно від поточного навантаження. Це забезпечує ефективне використання обчислювальних ресурсів, дозволяє обробляти пікові навантаження без попереднього резервування інфраструктури та спрощує підтримку системи в умовах змінної інтенсивності обчислювальних задач.

Окремо слід відзначити економічний аспект обраної архітектури. Відповідно до політики платформи Amazon, оплата використання функцій AWS Lambda здійснюється за принципом «pay-as-you-go» [34], тобто тільки за фактичний час виконання. Така модель є доцільною в умовах дослідницького використання системи, оскільки вона експлуатується обмеженим колом користувачів і не передбачає одночасного виконання великої кількості експериментів. Це дозволяє ефективно оптимізувати витрати на обчислювальні ресурси без втрати функціональності.

Особливу роль в архітектурі системи відіграє сервіс Amazon SQS, який використовується як проста, але надійна шина повідомлень для організації асинхронної взаємодії між компонентами. Такий підхід забезпечує слабку зв'язаність компонентів, підвищує стійкість системи до збоїв і дозволяє гнучко масштабувати окремі етапи обчислювального процесу.

Крім того, використання Amazon SQS дозволяє реалізувати буферизацію навантаження, що є критичним у випадках нерівномірного надходження задач або

обмежень на кількість одночасних викликів функцій AWS Lambda. Повідомлення у черзі можуть оброблятися з урахуванням політик повторних спроб (retry) та механізмів відкладеної обробки, що підвищує надійність виконання [23].

Додатково використовується механізм тригерів SQS → AWS Lambda, який забезпечує автоматичний запуск відповідних функцій при появі нових повідомлень у черзі. Це дозволяє реалізувати подієво-орієнтовану модель обчислень без необхідності централізованого керування процесом, спрощує оркестрацію задач і забезпечує природну масштабованість системи відповідно до кількості вхідних повідомлень.

Для збереження стану експериментів і забезпечення узгодженої роботи розподілених компонентів використовується сервіс Amazon DynamoDB [21]. Основним призначенням бази даних є зберігання метаданих експериментів, включаючи параметри моделей, проміжні результати обчислень, інформацію про поточний стан популяцій та результати виконання окремих етапів алгоритму. Вибір саме Amazon DynamoDB обумовлений його здатністю забезпечувати стабільно низькі затримки операцій читання та запису навіть при високій інтенсивності звернень.

Крім того, використання Amazon DynamoDB є ключовим для вирішення задачі агрегації підпопуляцій у розподіленому середовищі. Оскільки функції AWS Lambda не мають власного стану та не підтримують кешування між викликами, виникає необхідність використання зовнішнього високопродуктивного сховища для збереження проміжних результатів. DynamoDB у цьому випадку виступає як централізований швидкодіючий шар доступу до даних, що дозволяє агрегатору ефективно зчитувати результати обробки підпопуляцій та записувати нові стани популяції з мінімальними затримками. Це забезпечує узгодженість даних, зменшує час ітерацій генетичного алгоритму та підвищує загальну продуктивність системи.

4.5.4 Проблеми синхронізації агрегації

Ключовим викликом при впровадженні асинхронного підходу до реалізації еволюційного алгоритму стала операція агрегації результатів оцінювання підпо-

пуляцій. У межах початкової архітектури для виконання цього завдання застосовувався спеціалізований сервіс-агрегатор, взаємодія з яким відбувалася через механізм обміну повідомленнями з використанням черг типу FIFO (First-In-First-Out). Вибір такого архітектурного патерна був зумовлений необхідністю чіткої координації розподілених обчислювальних вузлів та інтеграції їхніх локальних результатів у єдину глобальну популяцію для наступної ітерації (рис. 4.11).

Черги типу FIFO традиційно розглядаються як оптимальне рішення для розподілених систем, де критичною є вимога збереження строгого хронологічного порядку обробки подій та уникнення станів гонки (race conditions). Завдяки детермінованій послідовності вилучення повідомлень, яка є ідентичною порядку їх надходження, такі черги забезпечують надійну синхронізацію та послідовну зміну стану глобальної системи. Вони гарантують семантику обробки "точно один раз" (exactly-once processing), що є необхідною умовою для коректної агрегації фітнес-оцінок. Для забезпечення строгих гарантій впорядкованості в межах паралельних процесів застосовувався принцип партиціювання черги, де ключем групи повідомлень (Message Group ID) виступав унікальний ідентифікатор популяції або експерименту. Таке рішення дозволяло системі маршрутизувати та лінеаризувати всі результати обчислень еволюційного процесу, формуючи суворо впорядкований потік даних без взаємного блокування незалежних експериментів.

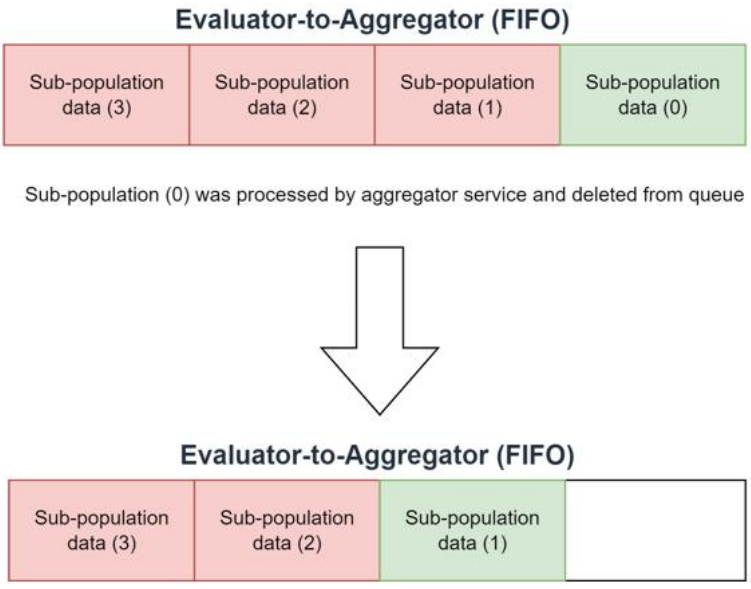


Рисунок 4.11 – Проблема швидкодії FIFO підходу до синхронізації

Однак емпіричне тестування запропонованої архітектури на масштабованих популяціях обсягом у 1000 особин виявило суттєву деградацію продуктивності: загальний час виконання однієї еволюційної ітерації перевищував 10 хвилин. Детальний аналіз системи показав, що вимога суворої впорядкованості, імплементована через FIFO-чергу з використанням ключа експерименту, генерувала критичне «вузьке місце». Оскільки сервіс агрегації функціонував у синхронному ітеративному режимі, він був змушений обробляти кожне повідомлення в межах групи строго послідовно. Ця нерозпаралелювана секція виконання повністю нівелювала виграш у продуктивності, досягнутий завдяки паралельному обчисленню фітнес-функцій на попередніх етапах алгоритму.

Для подолання проблеми латентності, притаманної FIFO-чергам у даному контексті, парадигма агрегації була фундаментально переглянута у бік подієво-орієнтованої архітектури з розподіленою синхронізацією на рівні бази даних. Замість традиційних черг повідомлень для передачі масивів даних нова архітектура використовує базу даних DynamoDB як спільне сховище стану. Процедура ініціюється сервісом-евалюатором, який після завершення оцінювання підпопуляції зберігає результати у таблицю DynamoDB та надсилає асинхронне сповіщення (notification event) до сервісу-агрегатора. Це повідомлення є легковагим та містить лише ключ популяції або експерименту. Отримавши сповіщення, сервіс-агрегатор використовує цей ключ для виконання запиту до бази даних та підрахунку поточної кількості зібраних підпопуляцій. У разі недосягнення цільового розміру популяції сервіс завершує поточний потік обробки, не блокуючи роботу інших вузлів-евалюаторів.

Умова, за якої набирається необхідна кількість хромосом, викликає запуск процедури фінальної агрегації. Оскільки кілька вузлів сервісу-агрегатора можуть одночасно обробляти повідомлення про надходження різних підпопуляцій, виникає ризик колізій при записі результатів агрегації та дублювання даних. Зазначений підхід до синхронізації обумовлений використанням serverless-архітектури, яка характеризується відсутністю постійного стану (stateless) та незалежним життєвим циклом окремих екземплярів функцій [54]. У таких умовах стає неможли-

вим використання класичних низькорівневих примітивів синхронізації, таких як м'ютекси, семафори або блокування в оперативній пам'яті, оскільки обчислювальні процеси ізольовані один від одного та можуть виконуватися на різних фізичних серверах.

Для запобігання ситуаціям, коли два або більше незалежних безсерверних екземплярів одночасно фіксують наявність достатньої кількості підпопуляцій і намагаються виконати агрегацію (рис. 4.12), було застосовано вбудований механізм умовних оновлень (conditional updates), який є частиною функціональності DynamoDB [21].

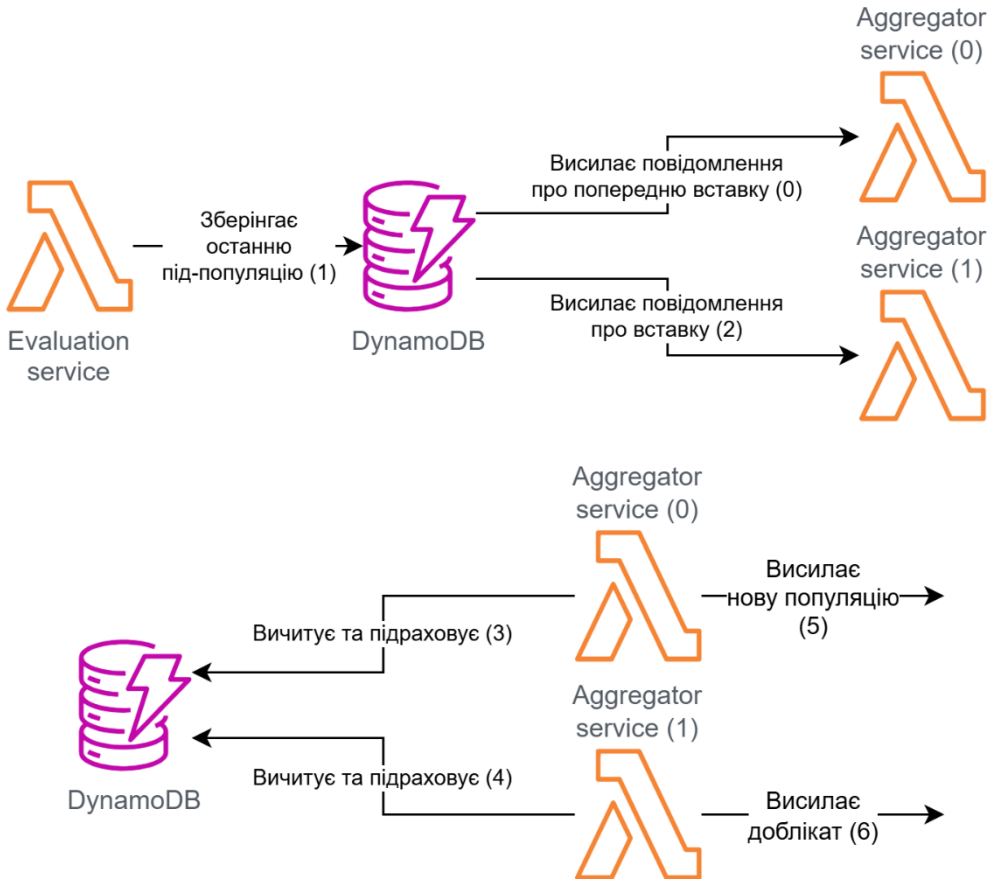


Рисунок 4.12 – Іллюстрація проблеми синхронізації

Ключовим елементом нової синхронізаційної моделі стала оптимістична транзакція, що базується на записі метаданих експерименту. Як референтний об’єкт, що координує синхронізацію в конкурентному середовищі, використовується запис метаданих експерименту, де фіксується поточний номер ітерації. Сервіс-агрегатор, який першим фіксує накопичення достатньої кількості хромосом, ініціює операцію умовного оновлення цього запису. Операція спрямована на ін-

крементацію параметра номера популяції виключно за умови, що його поточне значення в базі даних збігається з очікуваним. Це виконання є атомарним на рівні бази даних, тому будь-які паралельні запити з аналогічною умовою завершуються системною помилкою відхилення транзакції (рис. 4.13).

Такий механізм гарантує атомарність та виключає можливість повторного виконання агрегації в межах однієї ітерації, дозволяючи масштабувати агрегатори горизонтально без ризику колізій.

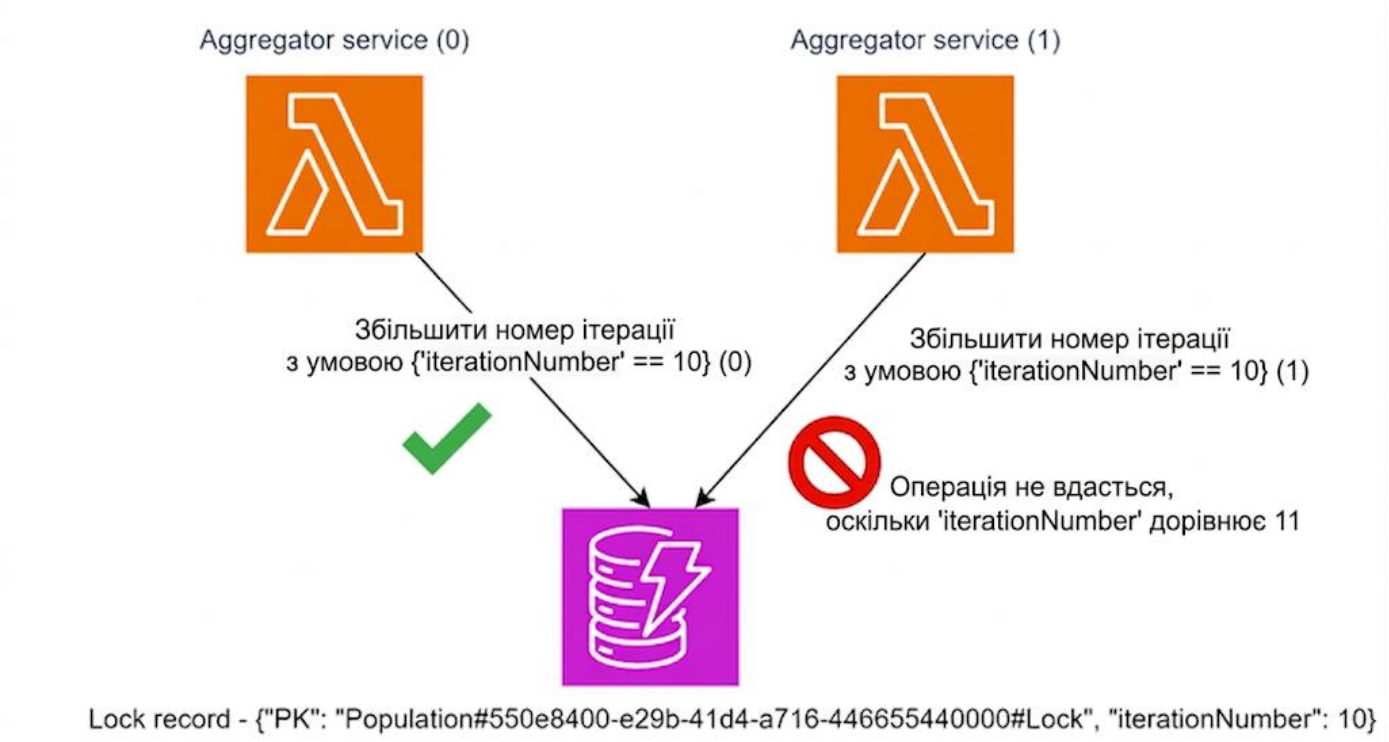


Рисунок 4.13 – Вирішення проблеми синхронізації на основі впровадження умовного оновлення

4.5.5 Модернізація основних сутностей та їх логічна реалізація

Відповідно до вищезазначених особливостей реалізації були сформовані наступні структури відображення сутностей на прикладному рівні реалізації.

Популяція/Експеримент – сутність, яка відображає безпосередньо поточну ітерацію або поточний запуск експерименту процесу відновлення параметрів моделі. Для збереження метаданих експериментів використовується таблиця DynamoDB з унікальним ідентифікатором експерименту (PK) та сортувальним ключем (SK). Така організація дозволяє зберігати різні типи записів для одного

експерименту і забезпечує ефективний доступ до даних через Query за Partition Key. Крім того, таблиця включає основні атрибути сутності (табл. 4.3):

```
{
    "PK": "EXP#8f14e45f-e3d9-4a7b-8a3c-1b2f3d4e5a6b",
    "SK": "META",
    "created_datetime": "2026-04-06 12:34:56",
    "last_updated_datetime": "2026-04-06 12:34:56",
    "iteration_count": 150,
    "fitness_value": "0.98765",
    "finished_datetime": null
}
```

Таблиця 4.3 – Опис параметрів сутності популяції/експеримента

Поле	Тип даних	Опис
PK	String	Partition Key – унікальний ідентифікатор експерименту у форматі EXP#{UUID}
SK	String	Sort Key – визначає тип запису, для метаданих використовується "META"
created_datetime	String	Дата і час створення запису у форматі YYYY-MM-DD HH:MM:SS
last_updated_datetime	String	Дата і час останнього оновлення
iteration_count	Number	Кількість виконаних ітерацій
fitness_value	String	Найкраще значення цільової функції (fitness) для експерименту
finished_datetime	String / Null	Дата і час завершення експерименту; null якщо експеримент ще триває

Підпопуляція представляє собою частину загальної популяції хромосом, виділену для окремої обробки у рамках експерименту (табл. 4.4). Вона дозволяє ефективно керувати великими наборами моделей, забезпечуючи сегментоване збереження та незалежну обробку кожного блоку даних. Підпопуляції забезпечують контрольований ітеративний процес відновлення параметрів, дозволяють оцінювати прогрес fitness окремих хромосом і полегшують масштабування експе-

риментів, зберігаючи послідовність еволюції моделей навіть у випадку стохастичних або складних часових рядів:

```
{
    "PK": "EXP#8f14e45f-e3d9-4a7b-8a3c-1b2f3d4e5a6b",
    "SK": "MOD#3fa85f64-5717-4562-b3fc-2c963f66afa6",
    "constract_models": "789c4b4f4d4a4e2d...",
    "TTL": 1717761296
}
```

У свою чергу, кожен елемент під-популяції являє собою логічне відображення сутності хромосоми (табл. 4.5):

```
{
    "key": "f19c7938-2287-45fa-8b19-8eaebecb44c9",
    "age": 0,
    "fitness": "1.7976931348623157E+308",
    "mathExp": -3.975401,
    "mathExpDiv": 3.5446966,
    "dispersion": 0,
    "axiom": "f+ff-",
    "rules": {"f": ["-f+f+f-f-f+"]
    }
}
```

Таблиця 4.4 – Опис параметрів сутності підпопуляції (з закодованими даними)

Поле	Тип даних	Опис
`PK`	String	Partition Key – ідентифікатор експерименту/популяції
`SK`	String	Sort Key – унікальний ідентифікатор підпопуляції (`MOD#{set_id}`)
`constract_models`	String	Стислі дані моделей підпопуляції у форматі hex
`TTL`	Number	Час життя запису у секундах

Таблиця 4.5 - Опис параметрів сутності елемента популяції

Поле	Тип даних	Опис
`key`	String	Унікальний ідентифікатор елемента популяції
`age`	Number	Вік елемента у кількості ітерацій

fitness	String / Number	Значення цільової функції (fitness) для оцінки якості відновлення
mathExp	Number	Математичне очікування (середнє значення) параметрів елемента
mathExpDiv	Number	Дисперсія математичного очікування або нормалізоване відхилення
dispersion	Number	Дисперсія елемента; для стабільних рядів часто прагне до 0
axiom	String	Початкова аксіома L-системи для генерації часового ряду
rules	Object	Множина правил підстановки L-системи; ключі – символи, значення – масив правил

Для оптимізації процесу агрегації та підрахунку оброблених підпопуляцій окремо також створюється та зберігається запис із кількістю елементів у кожній підпопуляції (табл. 4.6). Такий підхід дозволяє швидко оцінювати розмір підпопуляції без необхідності декомпресувати чи повторно обробляти всі моделі, що значно зменшує витрати ресурсів під час читання даних із бази. Крім того, наявність цього запису оптимізує операції підрахунку та агрегування, що забезпечує більш ефективне виконання ітеративних обчислень:

```
{
    "PK": "EXP#8f14e45f-e3d9-4a7b-8a3c-1b2f3d4e5a6b",
    "SK": "COUNT#3fa85f64-5717-4562-b3fc-2c963f66afa6",
    "count": 150,
    "TTL": 1717761296
}
```

Таблиця 4.6 - Опис параметрів сутності кількості елементів у під-популяції

Поле	Тип даних	Опис
PK	String	Partition Key – ідентифікатор експерименту
SK	String	Sort Key – унікальний ідентифікатор підрахунку підпопуляції (COUNT#{set_id}). Значення set_id є спільним із головним записом підпопуляції.

`count`	Number	Кількість моделей у підпопуляції
`TTL`	Number	Час життя запису у секундах

4.5.6 Реалізація неблокуючих конвеєрів та керування потоком даних

Для забезпечення максимальної швидкості виконання генетичних операцій та мінімізації латентності в межах обчислювального вузла було реалізовано механізм неблокуючого потоку керування. Фундаментальна ідея запропонованого підходу полягала у повній відмові від традиційних ітераційних циклів для кожного етапу генетичного алгоритму на користь потокової конвеєрної структури.

У межах розробленого на мові Golang рушія системи кожен логічний вузол конвеєра (node) представлений окремою функцією, що виконується у власній легковиваженій нитці керування (goroutine). Ці функції відповідають за атомарні операції генетичного пошуку, зокрема оператори схрещування (кросинговеру), мутації, а також фінальну підготовку сформованих особин до відправлення на оцінювання фітнес-функції. Організація такого конвеєра дозволяє досягти високого рівня гранулярності паралелізму, оскільки різні хромосоми можуть одночасно перебувати на різних стадіях генетичної модифікації.

Логічна структура кожного функціонального вузла містить покажчик на наступну функцію в ланцюгу та відповідний канал (channel) для вихідних повідомлень. Архітектура асинхронної взаємодії побудована на базі вбудованих конкурентних примітивів мови Go, що дозволяє організувати передачу даних без явного використання класичних блокувань пам'яті та критичних секцій. Вхідні та вихідні канали кожного з вузлів виконують роль транспортних магістралей для передачі типізованих структур даних, що описують генотипи хромосом. При цьому вихідний канал поточного функціонального вузла є вхідним каналом для наступного вузла в ланцюгу обробки, формуючи спрямований ациклічний граф передачі даних. Після завершення локальної операції над особиною функція-вузол негайно транслює результат у вихідний канал, забезпечуючи безперервний потік даних.

Щоб повністю нівелювати ризик виникнення взаємних блокувань (deadlocks) та вимушених простоїв при високій інтенсивності вхідного потоку даних, у кожен міжузловий канал було введено буфер обміну. Розмір зазначеного буфера детермінується та динамічно адаптується відповідно до максимально очікуваної кількості хромосом у популяції для поточної ітерації. Наявність буферизованих каналів гарантує, що операція запису в канал не буде блокувати горутину-продуцента доти, доки буфер не заповниться повністю. Це забезпечує асинхронне відчеплення (decoupling) суміжних етапів алгоритму, завдяки чому жоден етап генетичного конвеєра не блокується в очікуванні готовності наступного компонента прийняти дані, що призводить до максимальної утилізації доступного процесорного часу (рис. 4.14).

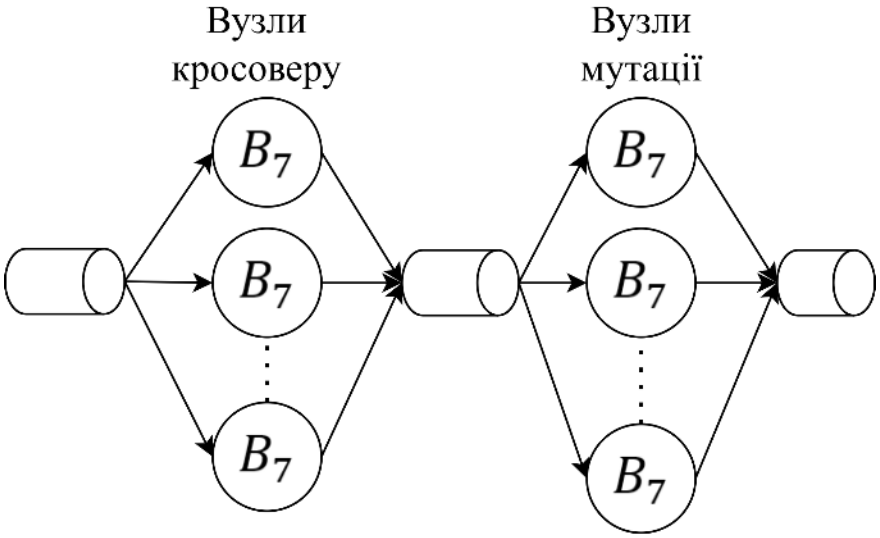


Рисунок 4.14 – Відображення вузлової системи із відповідними алгоритмічними операціями

Виконання кожної функції-вузла є повністю ізольованим, а ініціалізація та запуск усього конвеєра здійснюється через каскад рекурсивних викликів функцій, що генерують зв'язану мережу активних горутин. Для управління життєвим циклом цієї мережі та запобігання витоку ресурсів архітектура рушія передбачає впровадження спеціалізованих глобальних каналів для координації та обробки виняткових ситуацій. Зокрема, реалізовано канал помилок (error channel), призначений для централізованого перехоплення, агрегації та логування помилок, що виникають під час асинхронного виконання функцій. Будь-який горутин, що сти-

кається з некоректним станом даних, транслює дескриптор помилки в цей канал, не перериваючи роботу всього конвеєра без крайньої потреби.

Паралельно функціонує загальний канал завершення (done channel), який є критичним елементом керування в архітектурі майстра пайплайну (orchestrator node). Даний канал використовується майстром для ініціації процедури скасування (canceling) контексту, який передається кожному з функціональних вузлів як аргумент під час етапу первинної ініціалізації. Кожна горутина у вузлах конвеєра містить конструкцію вибору (select), що постійно моніторить стан каналу завершення та системного контексту. При відправці сигналу або закритті каналу завершення майстром ініціюється каскадна відміна контексту, внаслідок чого відповідні вузли синхронно припиняють подальшу обробку та переходять у режим коректного завершення поточного потоку. Такий механізм є необхідним для детермінованого припинення обчислень при досягненні критеріїв зупинки або виникненні критичних збоїв у stateless-середовищі (рис. 4.15).

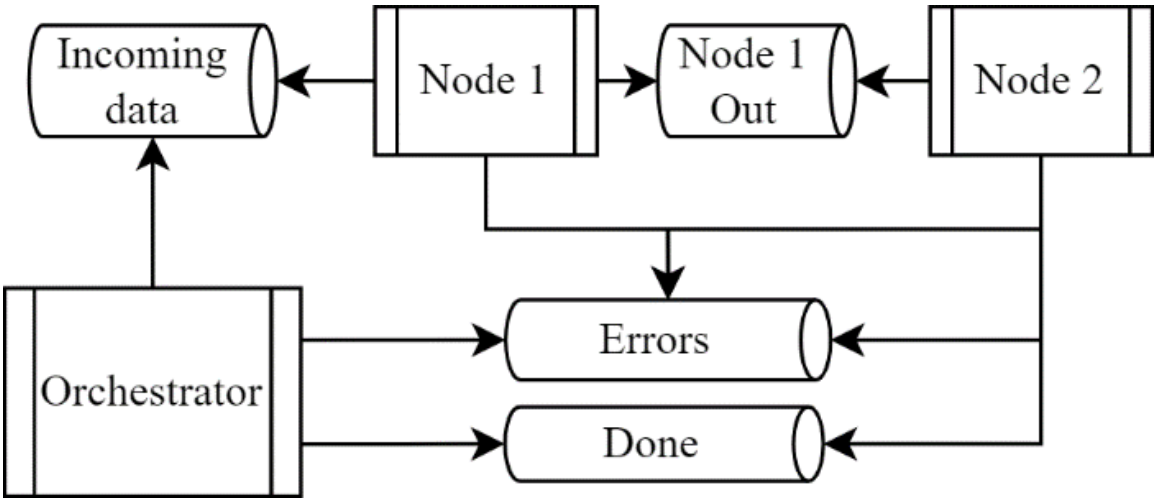


Рисунок 4.15 – Глобальна архітектура фасаду вузлової системи

Такий підхід до роздільного масштабування компонентів на рівні конкурентних потоків мови Go дозволив максимально ефективно розпоряджатися доступними обчислювальними ресурсами. Емпіричні дослідження доводять, що хоча при роботі з малими популяціями накладні витрати на ініціалізацію каналів, планувальник горутин та пересилання даних між розподіленими сутностями є дещо вищими за класичний монолітний підхід, при масштабуванні до великих об'ємів даних час виконання ітерацій виходить на стабільне «плато». Це підтверджує ви-

соку масштабованість, відсутність деградації продуктивності при зростанні навантаження та стійкість обраної архітектурної моделі.

4.6 Модифікація программного додатку для покриття відновлення параметрів реальних рядів

У межах розробленої архітектури генетичний алгоритм з обробкою популяцій був інтегрований як підпорядкований функціональний елемент більш високорівневої структури. У фінальній реалізації ієрархія сутностей була розширена за рахунок впровадження концепту епох, де представлений раніше процес генетичного пошуку виконує роль вкладеної фази відновлення параметрів.

Відповідно до цієї дворівневої моделі керування кожна ітерація верхнього рівня визначається як епоха, під час якої виконується послідовний підбір параметрів спочатку для основного, а потім для залишкового ряду. На кожному з цих двох етапів ініціюється повноцінний автономний запуск генетичного алгоритму. Кількість епох є конфігурованим параметром системи і задається користувачем на етапі ініціалізації експерименту аналогічно до того, як задається гранична кількість внутрішніх ітерацій (поколінь) для самого генетичного алгоритму.

Процедура керування епохами та переходу між ними має тривіальний характер і не потребує впровадження складних елементів розподіленої синхронізації або важковагових механізмів координації. На відміну від процесів агрегації всередині генетичного алгоритму, де виникає висока конкурентність через одночасну роботу багатьох безсерверних функцій, виконання епох є строго послідовним. Нова епоха не може розпочатися доти, доки повністю не завершиться попередня, а всередині самої епохи фаза обробки залишкового ряду строго детермінована завершенням фази основного ряду. Оскільки конкурентний доступ до стану епохи на запис відсутній, для фіксації прогресу достатньо простих лінійних операцій оновлення статусу.

Для збереження цілісності даних та можливості повної ретроспективної реконструкції обчислювального процесу було модифіковано схему бази даних. У фізичній моделі збереження результатів кожен автономний запис про виконання

генетичного алгоритму містить обов'язкове посилальне поле, що вказує на батьківську епоху. Завдяки такій структурі індексування забезпечується детермінована відновлення відношень «один-до-множини» [28] між епохою та відповідними їй ітераційними циклами генетичного пошуку. Записи експерименту зберігають метадані про свій внутрішній стан для легкого агрегування результатів (рис. 4.16).

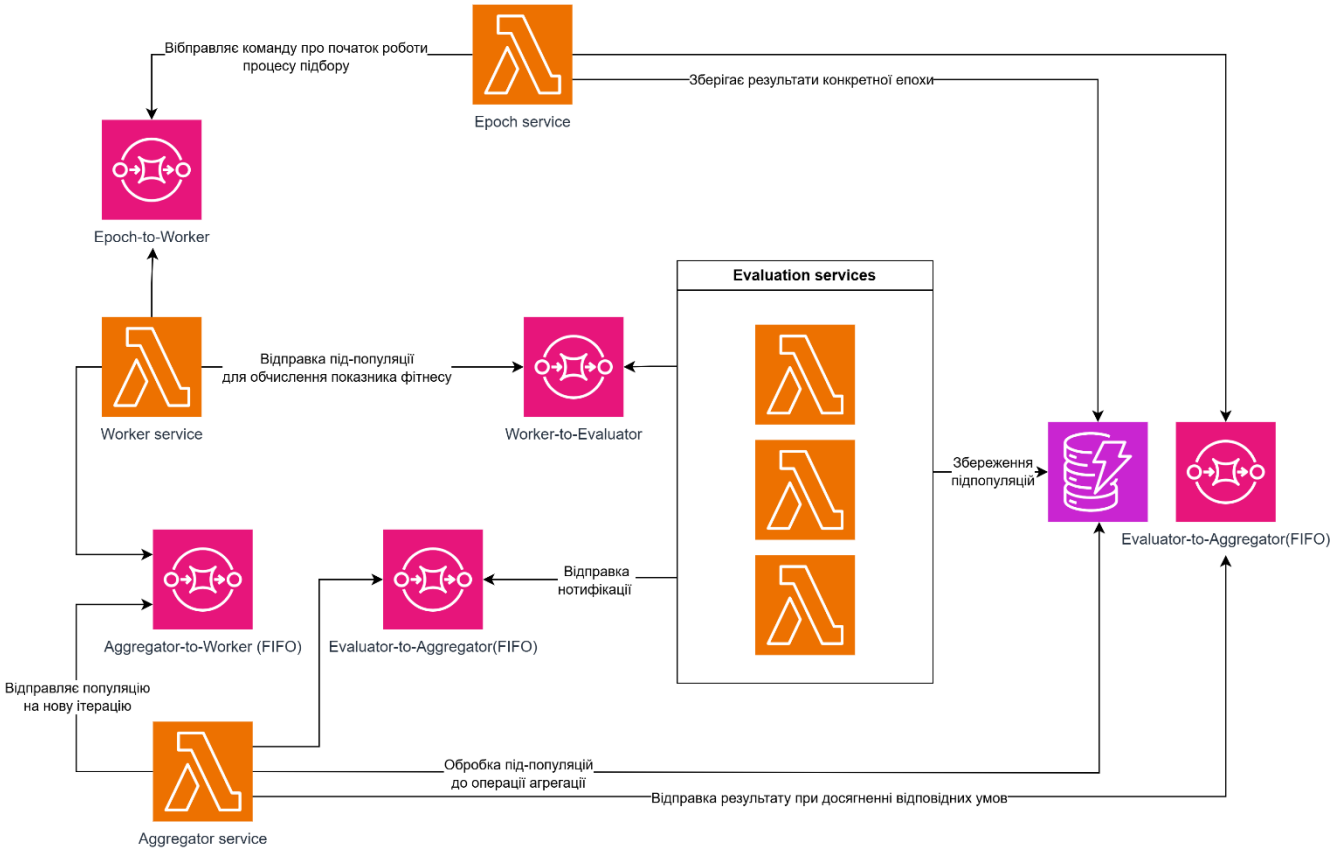


Рисунок 4.16 – Архітектура програмного рішення на базі компонентів AWS відповідно до впровадження епох як елементів функціоналу

Програмний контроль за виконанням епох здійснюється за допомогою базових операцій у DynamoDB. Після фіксації завершення роботи генетичного алгоритму для основного ряду (коли набрано необхідну кількість хромосом та виконано атомарну агрегацію), керуючий процес ініціює наступну фазу в межах тієї ж епохи для залишкового ряду.

Тільки після успішної фіксації та збереження параметрів обох конструктивних моделей (основної та залишкової) статус поточної епохи оновлюється в базі даних як завершений, і система переходить до обчислення нового вхідного сигналу для наступної епохи.

Висновки до четвертого розділу

Реалізація програмного рішення для автоматизації процесу відновлення параметрів конструктивної моделі часового ряду пройшла складний шлях від простого монолітного застосунку до розподіленої хмарної архітектури.

На початковому етапі розробки виникла необхідність формалізувати та автоматизувати обробку експериментальних даних для забезпечення систематичного дослідження запропонованого математичного підходу. З цією метою було спроєктовано монолітну однопотокову систему, що базувалася на принципах об'єктно-орієнтованого (OOD) та доменно-орієнтованого (DDD) дизайну. Виділення ключових логічних сутностей, таких як хромосома та популяція, а також імплементація класичних операторів генетичного алгоритму дозволили побудувати чітке логічне відображення процесів предметної області. Цей етап фактично відповідав підходу "Proof of Concept", який дозволив підтвердити принципову можливість реалізації запропонованого методу та виконати первинну верифікацію на детермінованих часових рядах, які відзначаються відносною простотою структурної організації та передбачуваністю.

Проте подальші дослідження виявили суттєві обмеження монолітної архітектури при переході до роботи зі стохастичними часовими рядами. Однопотокова модель виконання обмежувалася лише вертикальним масштабуванням, що ускладнювало обробку великих популяцій, необхідних для покращення якості пошуку рішень, і призводило до зниження загальної продуктивності системи. Для вирішення цієї проблеми було здійснено перехід до горизонтального масштабування з використанням мультиагентного підходу. За результатами аналізу різних топологій була обрана модель "island", де кожен агент функціонував як автономна обчислювальна одиниця, реалізуючи повний цикл відновлення, а обмін найкращими представниками поколінь відбувався за допомогою спеціалізованого міграційного агента через допоміжний банк хромосом. Це дозволило уникнути швидкого виродження популяції та підвищити ефективність розподіленого генетичного алгоритму.

Незважаючи на успіхи мультиагентної моделі, обробка стохастичних рядів виявила нове критичне «вузьке місце» – операцію обчислення показника фітнесу. Через високу варіативність стохастичних рядів складність цієї операції критично зросла, що змусило дослідників перейти до стратегії роздільного масштабування. Логіку оцінки фітнесу було відокремлено від генетичних операцій (кросинговеру, мутації, селекції) в окрему сутність, що дозволило одному генетичному сервісу координувати роботу декількох паралельних сервісів оцінювання. Для реалізації цієї асинхронної мультисервісної архітектури було обрано платформу Amazon Web Services із використанням serverless-підходу на базі AWS Lambda та шини повідомлень Amazon SQS. Цей вибір забезпечив вбудовані механізми автоматичного динамічного масштабування та економічну ефективність за моделлю «pay-as-you-go», дозволяючи системі миттєво реагувати на пікові навантаження без попереднього резервування інфраструктури.

Впровадження розподіленої обчислювальної мережі виявило складні проблеми синхронізації, зокрема на етапі агрегації результатів оцінювання підпопуляцій. Використання традиційних FIFO-черг для координації розподілених вузлів генерувало критичне «вузьке місце», оскільки сервіс-агрегатор був змушений обробляти повідомлення строго послідовно, що нівелювало виграш від паралельних обчислень і збільшувало час ітерації до понад 10 хвилин для великих популяцій. Для подолання цієї проблеми парадигму було переглянуто на користь подієво-орієнтованої архітектури з використанням високопродуктивного сховища Amazon DynamoDB. Оскільки безсерверна архітектура унеможливлювала використання класичних примітивів блокування в оперативній пам'яті (м'ютексів чи семафорів), синхронізація була реалізована через механізм умовних оновлень (conditional updates) на рівні бази даних. Ця оптимістична транзакція гарантувала атомарність виконання операцій, повністю виключаючи ризик дублювання даних або виникнення станів гонки при одночасній роботі кількох агрегаторів.

Паралельно з глобальною хмарною оптимізацією була проведена глибока модернізація логічного ядра системи на рівні програмного коду. Відбулася повна відмова від традиційних ітераційних циклів на користь потокової неблокуючої

конвеєрної структури, реалізованої за допомогою конкурентних примітивів мови Golang. Кожен логічний етап генетичного пошуку виконувався у власній легковиваженій горутині, а дані передавалися через буферизовані канали, що забезпечувало високий рівень гранулярності паралелізму. Впровадження буферів обміну дозволило асинхронно відчепити суміжні етапи алгоритму, запобігаючи взаємним блокуванням (deadlocks) та максимізуючи утилізацію процесорного часу. Для надійного керування цим комплексом паралельних процесів було розроблено систему скасування контексту та централізовані канали обробки помилок.

Фінальним етапом еволюції програмного рішення стала його адаптація до специфіки реальних часових рядів шляхом впровадження ієрархічної концепції епох. Генетичний алгоритм став підпорядкованим функціональним елементом, який послідовно запускався спочатку для основного, а потім для залишкового ряду в межах однієї епохи. На відміну від висококонкурентних процесів агрегації, управління епохами мало строго послідовний та лінійний характер, що виключало стани гонки на цьому рівні. Модифікація схеми бази даних для підтримки відношень «один до множини» між епохою та ітераціями забезпечила цілісність даних та можливість ретроспективного аналізу всього процесу. Таким чином, послідовна еволюція архітектурних підходів дозволила створити відмовостійкий, економічно обґрунтований та високопродуктивний програмний комплекс, який не лише підтвердив життєздатність теоретичного підходу, але й забезпечив надійний інструментарій для повномасштабного дослідження методів відновлення параметрів складних часових рядів будь-якої природи.

Матеріали розділу опубліковані у роботах: [75, 76, 77, 102, 105, 106]

ЗАГАЛЬНІ ВИСНОВКИ

У дисертаційній роботі наведено теоретичне узагальнення та нове розв’язання науково-прикладної задачі відновлення параметрів конструктивно-продукційних моделей часових рядів різної природи. Запропонований підхід забезпечує перехід від традиційної апроксимації до реконструкції внутрішньої структури та логіки формування складних динамічних процесів.

Основні результати досліджень полягають у наступному:

1. Здійснено комплексний аналіз існуючих підходів, який підтвердив обмеженість класичних статистичних методів та засобів машинного навчання у задачах структурної ідентифікації, що дозволило обґрунтувати доцільність застосування конструктивно-продукційного моделювання.

2. Формалізовано систему спеціалізованих конструкторів та відповідних алгоритмів, в тому числі генетичний, що дозволило представити повноцінний обернений процес моделювання, а саме відтворення математичної моделі на основі аналізу значень часового ряду.

3. Розроблено методи відновлення параметрів для синтетичних детермінованих рядів, які, на відміну від існуючих засобів, дозволяють достовірно ідентифікувати та відновити фрактальну складову існуючого ряду.

4. Запропоновано ітеративний підхід до відновлення параметрів стохастичних та реальних часових рядів шляхом їх декомпозиції, що забезпечує виявлення закономірностей змін складових частин ряду відповідно до їхніх фрактальних властивостей.

5. Удосконалено метод оцінки близькості (фітнес-функції) на основі ансамблевого зіставлення реалізацій, що дозволило коректно врахувати варіативність стохастичних процесів і суттєво підвищити точність їхньої реконструкції.

6. Розширено апарат формальних граматик для опису взаємодії фрактальних компонентів, що забезпечило формалізацію складних структурних зв’язків між елементами динамічного процесу.

7. Отримав подальший розвиток метод кодування гібридних хромосом у генетичному алгоритмі, адаптовані базові генетичні оператори, що дозволило поєд-

нати символьні правила L-систем та числові математичні параметри в межах єдиної структури.

8. Удосконалено архітектурний підхід до програмної реалізації через впровадження механізму ізольованого масштабування окремих операцій відповідно до їхньої обчислювальної складності.

9. Інтегровано засоби генетичного пошуку як базові модулі-конструктори у загальну систему моделювання, що забезпечило універсальність запропонованого методу для роботи з часовими рядами різної природи.

10. Розроблено програмний комплекс, який забезпечує повну автоматизацію процесу експериментальних досліджень та створює умови для практичного впровадження розроблених методів.

11. Експериментально підтверджено ефективність підходу: для детермінованих рядів досягнуто точного відновлення структури, а для стохастичних синтетичних та реальних даних – стабільного відтворення динаміки через ітеративну декомпозицію параметрів та правил підстановки. Так для реального часового ряду з вартості акцій компанії Cisco відновлена послідовність складових конструктивних моделей має поліпшені показники якості порівняно з іншими підходами MAE: 1.247, MSE: 2.712, RMSE: 1.647, MAPE: 1.99%, SMAPE: 1.99%, R^2: 0.861, Pearson Corr: 0.928, Spearman Corr: 0.896.

12. Результати роботи впроваджено у навчальний процес університету.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1.Abbott, M. L., & Fisher, S. (2015). The Art of Scalability: Scalable Web Architecture, Processes, and Organizations for the Modern Enterprise (2nd ed.). Addison-Wesley.

2.Alba, E & Luque, G. (2011). Parallel Genetic Algorithms: Theory and Real World Applications. Springer. DOI: 10.1007/978-3-642-22084-5

3.Albeverio, S., Pratsiovytyi, M., Torbin, G. (2004). Fractal probability distributions and transformations preserving the Hausdorff–Besicovitch dimension. Ergodic Theory and Dynamical Systems, 24(1), 1–16. DOI: 10.1017/S0143385703000397.

4.Allspaw, J., & Robbins, J. (2020). Web Operations: Keeping the Data On Time. O'Reilly Media.

5.Al-Selwi, S.M., Hassan, M.F., Abdulkadir, S.J., Muneer, A., Sumiea, E.H., Alqushaibi, A., Ragab, M.G. (2024). RNN-LSTM: From applications to modeling techniques and beyond – Systematic review. Journal of King Saud University - Computer and Information Sciences, 36(5), 102068. DOI: 10.1016/j.jksuci.2024.102068.

6.Armstrong, J.S., Collopy, F. (1992). Error measures for generalizing about forecasting methods: Empirical comparisons. International Journal of Forecasting, 8(1), 69–80. DOI: 10.1016/0169-2070(92)90008-W.

7.Bansal, I.R., Ashourvan, A., Bertolero, M., Bassett, D.S., Pequito, S. (2022). Model-based stationarity filtering of long-term memory data applied to resting-state blood-oxygen-level-dependent signal. PLOS ONE, 17(7). DOI: 10.1371/journal.pone.0268752.

8.Bashar, M.A., Nayak, R., Astin-Walmsley, K., Heath, K. (2023). Machine learning for predicting propensity-to-pay energy bills. Intelligent Systems with Applications, 17, 200176. DOI: 10.1016/j.iswa.2023.200176.

9.Bhalla, V., Bhalla, A., Jain, R. (2025). A comparison of ARIMA, SARIMA, and statistical models for estimating malaria incidence and death. Sustainable Computing and Intelligent Systems: Proceedings of SCIS 2024, vol. 2, 1296, p. 185. DOI: 10.1007/978-981-96-3314-2_14

10.Belozyorov, V., Zaytsev, V. (2016). Recurrence analysis of time series generated by 3D autonomous quadratic dynamical system depending on parameters. Visnyk of Dnipropetrovsk University. Series: Modeling, 8(8), 56–70. DOI: 10.15421/141604.

11.Bondi, A. B. (2021). Foundations of Software and System Performance Engineering: Process, Performance Modeling, Requirements, Testing, Optimization, and Management. Addison-Wesley.

12.Bulakh, V.; Kirichenko, L.; Radivilova, T. Binary Classification of Fractal Time Series by Machine Learning Methods. In Lecture Notes in Computational Intelligence and Decision Making: ISDMCI 2019. Advances in Intelligent Systems and Computing, vol. 1020; Springer: Cham, 2020; pp. 701–711. DOI: 10.1007/978-3-030-26474-1_49.

13.Bulakh, V.; Kirichenko, L.; Radivilova, T. Time Series Classification Based on Fractal Properties. In Proceedings of the 2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP), Lviv, Ukraine, 21–25 August 2018; pp. 198–201. DOI: 10.1109/DSMP.2018.8478532.

14.Burns, B. (2018). Designing Distributed Systems: Patterns and Paradigms for Modern Microservices. O'Reilly Media.

15.Castro, A.L., Marcato, A.L.M., De Aguiar, E.P. (2023). Multifractal analysis of the Brazilian electricity market. IEEE Access, 11, 98939–98957. DOI: 10.1109/ACCESS.2023.3313099.

16.Chen, J.-F., Wang, W.-M., Huang, C.-M. (1995). Analysis of an adaptive time-series autoregressive moving-average (ARMA) model for short-term load forecasting. Electric Power Systems Research, 34(3), 187–196. DOI: 10.1016/0378-7796(95)00977-1.

17.Chicco, D., Warrens, M.J., Jurman, G. (2021). The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. PeerJ Computer Science, 7, e623. DOI: 10.7717/peerj-cs.623.

18.Contreras, J., Espinola, R., Nogales, F.J., Conejo, A.J. (2003). ARIMA models to predict next-day electricity prices. IEEE Transactions on Power Systems, 18(3), 1014–1020. DOI: 10.1109/TPWRS.2002.804943.

19.Dabral, P.P., Murry, M.Z. (2017). Modelling and forecasting of rainfall time series using SARIMA. Environmental Processes, 4, 399–419. DOI: 10.1007/s40710-017-0226-y.

20.Daigneau, R. (2011). Service Design Patterns: Fundamental Design Solutions for SOAP/WSDL and RESTful Web Services. Addison-Wesley.

21.DeBrie, A. (2020). The DynamoDB Book: A Comprehensive Guide to Data Modeling with DynamoDB. Self-published.

22.Elmaz, F., Eyckerman, R., Casteels, W., Latré, S., Hellinckx, P. (2021). CNN-LSTM architecture for predictive indoor temperature modeling. Building and Environment, 206, 108327. DOI: 10.1016/j.buildenv.2021.108327.

23.Estrin, E. (2021). Cloud Security Handbook: Design and implement cost-effective and secure cloud solutions. Packt Publishing.

24.Evans, E. (2004). Domain-Driven Design: Tackling Complexity in the Heart of Software. Addison-Wesley.

25.Ghaffarpour, R., Zamanian, S. (2022). Fractal-based lightning model for investigation of lightning direct strokes to the communication towers. Electrical Engineering, 104, 2543–2551. DOI: 10.1007/s00202-022-01503-w.

26.Goodwin, P., Lawton, R. (1999). On the asymmetry of the symmetric MAPE. International Journal of Forecasting, 15(4), 405–408. DOI: 10.1016/S0169-2070(99)00007-2.

27.Gottimukkala, S.R. (2025). Applying the Multifractal Model of Asset Returns (MMAR) to financial markets: insights and limitations. In: Proceedings of the 2025 10th International Conference on Signal Processing and Communication (ICSC), Noida, India, pp. 277–282. DOI: 10.1109/ICSC64553.2025.10968868.

28.Hohpe, G., & Woolf, B. (2003). Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions. Addison-Wesley.

29.Hyndman, R.J., Athanasopoulos, G. (2018). Forecasting: principles and practice. 2nd ed. OTexts: Melbourne, Australia. OTexts.com/fpp2.

30.Indrasiri, K., & Kuruppu, D. (2020). gRPC: Up and Running: Building Cloud Native Applications with Go and Java for Docker and Kubernetes. O'Reilly Media.

31.Jin, Y.-C., Cao, Q., Wang, K.-N., Zhou, Y., Cao, Y.-P., Wang, X.-Y. (2023). Prediction of COVID-19 data using improved ARIMA-LSTM hybrid forecast models. IEEE Access, 11, 67956–67967. DOI: 10.1109/ACCESS.2023.3291999.

32.Kaur, J., Parmar, K.S., Singh, S. (2023). Autoregressive models in environmental forecasting time series: a theoretical and application review. Environmental Science and Pollution Research, 30, 19617–19641. DOI: 10.1007/s11356-023-25148-9.

33.Kashif, K., Ślepaczuk, R. (2025). LSTM-ARIMA as a hybrid approach in algorithmic investment strategies. Knowledge-Based Systems, 320, 113563. DOI: 10.1016/j.knosys.2025.113563.

34.Kavis, M. J. (2014). Architecting the Cloud: Design Decisions for Cloud Computing Service Models (SaaS, PaaS, and IaaS). John Wiley & Sons.

35.Kirichenko, L., Radivilova, T., Stepanenko, J. (2021). Applying recurrence plots to classify time series. Computational Linguistics and Intelligent Systems, 16–26.

36.Kirichenko, L.; Radivilova, T.; Stepanenko, J. Machine Learning in Classification Time Series with Fractal Properties. sData 2019, 4, 5. DOI: 10.3390/data4010005.

37.Kirichenko, L.; Bulakh, V.; Radivilova, T. Fractal Time Series Analysis of Social Network Activities. In Proceedings of the 2017 4th International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T), Kharkiv, Ukraine, 2017; pp. 456–459. DOI: 10.1109/INFOCOMMST.2017.8246438.

38.Kleppmann, M. (2017). Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems (Updated ed.). O'Reilly Media.

39.Konar, S., Auluck, N., Ganesan, R., et al. (2022). A non-linear time series based artificial intelligence model to predict outcome in cardiac surgery. Health Technology, 12, 1169–1181. DOI: 10.1007/s12553-022-00706-2.

40.Kontopoulou, V.I., Panagopoulos, A.D., Kakkos, I., Matsopoulos, G.K. (2023). A review of ARIMA vs. machine learning approaches for time series forecasting in data driven networks. Future Internet, 15, 255. DOI: 10.3390/fi15080255.

41.Lee, C.-M., Ko, C.-N. (2011). Short-term load forecasting using lifting scheme and ARIMA models. Expert Systems with Applications, 38(5), 5902–5911. DOI: 10.1016/j.eswa.2010.11.033.

42.Makridakis, S. (1993). Accuracy measures: theoretical and practical concerns. International Journal of Forecasting, 9(4), 527–529. DOI: 10.1016/0169-2070(93)90079-3.

43.Md, A.Q., Kapoor, S., Junni, A.V.C., Sivaraman, A.K., Tee, K.F., Sabireen, H., Janakiraman, N. (2023). Novel optimization approach for stock price forecasting using multi-layered sequential LSTM. Applied Soft Computing, 134, 109830. DOI: 10.1016/j.asoc.2022.109830.

44.Meenakshi, M., Gowrisankar, A. (2025). Multifractal analysis on ozone depletion and climate change: the time series approaches. Contemporary Mathematics, 6(1), 594–616. https://ojs.wiserpub.com/index.php/CM/article/view/6141.

45.Mensi, W., Kumar, A.S., Vo, X.V., et al. (2024). Asymmetric multifractality and dynamic efficiency in DeFi markets. Journal of Economic and Financial Studies, 48, 280–297. DOI: 10.1007/s12197-023-09655-6.

46.Millet, S., & Tune, N. (2015). Patterns, Principles, and Practices of Domain-Driven Design. Wrox.

47.Monge, M., Infante, J. (2023). A fractional ARIMA (ARFIMA) model in the analysis of historical crude oil prices. Vol. 4. Issue 1. DOI: 10.46557/001c.36578.

48.Mukaka, M.M. (2012). A guide to appropriate use of correlation coefficient in medical research. Malawi Medical Journal, 24(3), 69–71.

49.Myers, R.H. (1990). Classical and modern regression with applications. Vol. 2. Belmont, CA: Duxbury Press.

50.Oktaviani, F., Miftahuddin, Setiawan, I. (2021). Forecasting sea surface temperature anomalies using the SARIMA ARCH/GARCH model. Journal of Physics: Conference Series, 1882, 012020. DOI: 10.1088/1742-6596/1882/1/012020.

51.Ospina, R., Gondim, J.A.M., Leiva, V., Castro, C. (2023). An overview of forecast analysis with ARIMA models during the COVID-19 pandemic: Methodology and case study in Brazil. Mathematics, 11, 3069. DOI: 10.3390/math11143069.

52.Ouma, Y.O., Cheruyot, R., Wachera, A.N. (2022). Rainfall and runoff time-series trend analysis using LSTM recurrent neural network and wavelet neural network with

satellite-based meteorological data. Complex & Intelligent Systems, 8, 213–236. DOI: 10.1007/s40747-021-00365-2.

53.Park, M.-J., Yang, H.-S. (2024). Comparative Study of Time Series Analysis Algorithms Suitable for Short-Term Forecasting in Implementing Demand Response Based on AMI. Sensors, 24(22), 7205. DOI: 10.3390/s24227205.

54.Patil, S. (2022). Applying the Twelve-Factor App Methodology to Enterprise Systems. 9. 116-124. 10.5281/zenodo.18184272.

55.Pollard, B. (2019). HTTP/2 in Action. Manning Publications.

56.Pramanik, A., Sultana, S., Rahman, M.S. (2022). Time series analysis and forecasting of Monkeypox disease using ARIMA and SARIMA model. In: 13th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kharagpur, India, pp. 1–7. DOI: 10.1109/ICCCNT54827.2022.9984345.

57.Prusinkiewicz, P., Lindenmayer, A. (2012). The Algorithmic Beauty of Plants. Springer Science & Business Media.

58.Ray, S., Das, S.S., Mishra, P., et al. (2021). Time series SARIMA modelling and forecasting of monthly rainfall and temperature in the South Asian countries. Earth Systems and Environment, 5, 531–546. DOI: 10.1007/s41748-021-00205-w.

59.Ray, S., Lama, A., Mishra, P., Biswas, T., Das, S.S., Gurung, B. (2023). An ARIMA-LSTM model for predicting volatile agricultural price series with random forest technique. Applied Soft Computing, 149, part A, 110939. DOI: 10.1016/j.asoc.2023.110939.

60.Roberts M., Chapin J. (2017). What Is Serverless*?* Sebastopol: O'Reilly Media.

61.Salem, F.M. (2022). Recurrent Neural Networks (RNN). In: Recurrent Neural Networks. Springer, Cham. DOI: 10.1007/978-3-030-89929-5_3.

62.Saraswat, M., Srishti. (2022). Leveraging genre classification with RNN for book recommendation. International Journal of Information Technology, 14, 3751–3756. DOI: 10.1007/s41870-022-00937-6.

63.Shu, Z., Chan, P. W. (2025). Application of Fractal Analysis on Wind Speed Time Series: A Review. Advances in Wind Engineering, 2, 100028. DOI: 10.1016/j.awe.2024.100028.

64.Mielniczuk J., Wojdyłło P. (2007). Estimation of Hurst exponent revisited. Computational Statistics & Data Analysis, 51, 4510–4525. DOI: 10.1016/j.csda.2006.07.033.

65.Sbarcea, B. (2020). Serverless Architectures on AWS: With Examples using AWS Lambda. Packt Publishing.

66.Serletis, A., Rosenberg, A.A. (2007). The Hurst exponent in energy futures prices. Physica A: Statistical Mechanics and its Applications, 380, 325–332.

67.Shabani, M., Magris, M., Tzagkarakis, G., et al. (2023). Predicting the state of synchronization of financial time series using cross recurrence plots. Neural Computing & Applications, 35, 18519–18531. DOI: 10.1007/s00521-023-08674-y.

68.Shakhovska, N., Shymanskyi, V., Prymachenko, M. (2025). FractalNet-LSTM model for time series forecasting. Computers, Materials & Continua, 82(3), 4469–4484. DOI: 10.32604/cmc.2025.062675.

69.Shen, J., Liu, F., Tu, Y., et al. (2021). Finding gene network topologies for given biological function with recurrent neural network. Nature Communications, 12, 3125. DOI: 10.1038/s41467-021-23420-5.

70.Shynkarenko, V., Lytvynenko, M., Chyhir, R., Nikitina, I. (2020). Modeling of lightning flashes in thunderstorm front by constructive production of fractal time series. In: Advances in Intelligent Systems and Computing, vol. 1080, Springer, pp. 173–185. DOI: 10.1007/978-3-030-33695-0_13.

71.Shynkarenko, V.I., Chyhir, R.R. (2025). Constructive-synthesizing modeling of a thunderstorm front using a genetic algorithm. System Technologies, 4(159), 189–199. DOI: 10.34185/1562-9945-4-159-2025-19.

72.Shynkarenko, V.I., Chyhir, R.R. (2024). The software tool of constructive-synthesizing modeling. Problems in Programming, 2-3, 107–115. DOI: 10.15407/pp2024.02-03.107.

73.Shynkarenko, V.I., Ilman, V.M., “Constructive-Synthesizing Structures and Their Grammatical Interpretations. Part I. Generalized Formal Constructive-Synthesizing Structure”. Cybernetics and Systems Analysis, 50(5), 2014, pp. 655–662. DOI: 10.1007/s10559-014-9655-z.

74.Shynkarenko, V.I., Ilman, V.M., “Constructive-Synthesizing Structures and Their Grammatical Interpretations. Part II. Refining Transformations”. Cybernetics and Systems Analysis, 50(6), 2014, pp. 829–841. DOI: 10.1007/s10559-014-9674-9.

75.Shynkarenko, V., Zhadan, A., & Halushka, O. (2024). Multi-Agent System for Reconstruction Constructive Models of Stochastic Fractal Time Series. In Proceedings of the Information Technology and Implementation (IT&I) Workshop: IT Infrastructure and Applied Solutions (IT&I-WS 2024: ITIAS), Kyiv, November 20 – 21. 66-77.

76.Shynkarenko, V., Zhadan, A. (2020). Modeling of the deterministic fractal time series by one rule constructors. In: Proceedings of the 2020 IEEE 15th International Conference on Computer Sciences and Information Technologies (CSIT), Zbarazh, Ukraine, pp. 336–339. DOI: 10.1109/CSIT49958.2020.9321923.

77.Shynkarenko, V., Zhadan, A. (2024). Multiservice architecture of software for stochastic fractal time series forecasting. 2024 IEEE Conference on Computer Sciences and Information Technologies, 1–4. DOI: 10.1109/CSIT65290.2024.10982626.

78.Skalozub, V., Ilman, V., Bilyy, B. (2020). Constructive multiplayer models for ordering a set of sequences, taking into account the complexity operations of formations. Science and Transport Progress. Bulletin of Dnipropetrovsk National University of Railway Transport, 61–76. DOI: 10.15802/stp2020/213232.

79.Sohail, A., Afroz, H., Singh, J.N., Goel, D., Jasim, L.H. (2024). Stock price prediction using LSTM. In: Proceedings of the 2024 1st International Conference on Advances in Computing, Communication and Networking (ICAC2N), Greater Noida, India, pp. 48–53. DOI: 10.1109/ICAC2N63387.2024.10895015.

80.Srihari, G., et al. (2024). Predictive modeling of return volatility in sustainable investments: An in-depth analysis of ARIMA, GARCH, and ARCH techniques. Investment Management & Financial Innovations, 21(1), 213. DOI: 10.21511/imfi.21(1).2024.17.

81.Srivastava, P., Mishra, P.K. (2021). Stock market prediction using RNN LSTM. In: Proceedings of the 2021 2nd Global Conference for Advancement in Technology (GCAT), Bangalore, India, pp. 1–5. DOI: 10.1109/GCAT52182.2021.9587540

82.Thirunavukarasu, R., Kotei, E. (2024). A comprehensive review on transformer network for natural and medical image analysis. Computer Science Review, 53, 100648. DOI: 10.1016/j.cosrev.2024.100648.

83.Tarmanini, C., Sarma, N., Gezegin, C., Ozgonenel, O. (2023). Short term load forecasting based on ARIMA and ANN approaches. Energy Reports, 9, suppl. 3, 550–557. DOI: 10.1016/j.egyr.2023.01.060.

84.Toner, W., Darlow, L. (2024). An analysis of linear time series forecasting models. arXiv preprint arXiv:2403.14587. DOI: 10.48550/arXiv.2403.14587.

85.Tsukamoto, M. (2025). Mean Hausdorff dimension of some infinite-dimensional fractals. Journal d'Analyse Mathématique, 155, 235–286. DOI: 10.1007/s11854-024-0353-0

86.Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I. (2017). Attention is all you need. In: Guyon, I., Von Luxburg, U., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS), Advances in Neural Information Processing Systems, 30. Curran Associates, Inc. DOI: 10.48550/arXiv.1706.03762.

87.Vohra, D. (2020). Serverless Computing: A Modular Approach with AWS Lambda and SQS. Springer.

88.Wang, H., Yao, J. (2025). Forecasting carbon emissions in power systems using Autoformer and LSTM: a case study of the Spanish electricity market. In: Proceedings of the 2025 8th International Conference on Energy, Electrical and Power Engineering (CEEPE), Wuxi, China, pp. 1445–1450. DOI: 10.1109/CEEPE64987.2025.11033988.

89.Wang, J.-D., Susanto, C.O.N. (2023). Traffic flow prediction with heterogenous data using a hybrid CNN-LSTM model. Computers, Materials and Continua, 76(3), 3097–3112. DOI: 10.32604/cmc.2023.040914.

90.Wang, K., Zhang, J., Li, X., Zhang, Y. (2023). Long-term power load forecasting using LSTM-Informer with ensemble learning. Electronics, 12, 2175. DOI: 10.3390/electronics12102175.

91.Weigend, A. (1994). Time Series Prediction: Forecasting the Future and Understanding the Past. London. DOI: 10.4324/9780429492648.

92.Wu, H., Xu, J., Wang, J., Long, M. (2021). Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In: Advances in Neural Information Processing Systems, 34, 22419–22430. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2021/file/bcc0d400288793e8bdcd7c19a8ac0c2b-Paper.pdf.

93.Wu, J.M.T., Li, Z., Herencsar, N., et al. (2023). A graph-based CNN-LSTM stock price prediction algorithm with leading indicators. Multimedia Systems, 29, 1751–1770. DOI: 10.1007/s00530-021-00758-w.

94.Yilmaz, D., Büyüktahtakın, İ.E. (2023). Learning optimal solutions via an LSTM-optimization framework. Operations Research Forum, 4, 48. DOI: 10.1007/s43069-023-00224-5

95.Young, G. (2022). Object-Oriented Design Methodologies. CRC Press.

96.Zha, W., Liu, Y., Wan, Y., Luo, R., Li, D., Yang, S., Xu, Y. (2022). Forecasting monthly gas field production based on the CNN-LSTM model. Energy, 260, 124889. DOI: 10.1016/j.energy.2022.124889.

97.Працьовитий, М. В., Вовк, Ю. Ю., Лисенко, І. М., Ратушняк, П. (2021). Суперпозиції функцій з фрактальними властивостями. Нелінійні коливання, 24(4), 498–517.

98.Турбін, А. Ф., Працьовитий, М. В. (1992). Фрактальные множества, функции, распределения. Наукова думка.

99.Шинкаренко, В. І., Жадан, А. А. (2022). Конструктивно-продукційне моделювання фрактальних складових часових рядів. Сучасні інформаційні та комунікаційні технології на транспорті, в промисловості та освіті. Дніпро. с. 71.

100.Шинкаренко, В. І., Жадан, А. А. (2023). Конструктивно-продукційне моделювання детермінованих часових рядів з фрактальними властивостями. Інформаційні технології в металургії та машинобудуванні. 222–226. DOI: 10.34185/1991-7848.itmm.2023.01.060.

101. Шинкаренко, В. І., Жадан, А. А. (2024). Мультиагентне середовище відновлення фрактальних часових рядів. Інформаційні технології в металургії та машинобудуванні. 232–236. DOI: 10.34185/1991-7848.itmm.2024.01.041.

102.Шинкаренко, В. І., Жадан, А. А. (2024). Проблеми синхронізації розподілених обчислень на прикладі мікросервісної реалізації генетичного алгоритму. Сучасні інформаційні та комунікаційні технології на транспорті, в промисловості та освіті. Дніпро. с. 93.

103.Шинкаренко, В. И., Жадан, А. А. (2025). Використання методів конструктивно-продукційного моделювання для прогнозування детермінованих часових рядів. Системи і технології, 6, 124-137. DOI: 10.34185/1562-9945-5-161-2025-12.

104.Шинкаренко, В. І., Жадан, А. А. (2025). Конструктивно-продукційне моделювання часових рядів на основі поетапного розкладання. Сучасні інформаційні та комунікаційні технології на транспорті, в промисловості та освіті. 98.

105.Шинкаренко, В. И., Жадан, А. А. (2025). Реалізація процесу відновлення конструктивно-продукційних моделей фрактальних часових рядів. Проблеми програмування, 4, 3-11. DOI: 10.15407/pp2025.04.

106.Шинкаренко, В. І., Жадан, А. А. (2025). Реалізація розподілення процесу роботи генетичного алгоритму шляхом впровадження каналів та вузлів обробки повідомлень. Інформаційні технології в металургії та машинобудуванні. 215–219. DOI: 10.34185/1991-7848.itmm.2025.01.037.

107.Шинкаренко В. І., Жадан А. А. (2025). Сучасні методи та засоби роботи з часовими рядами. Наука і прогрес транспорту. № 4(112). 142–154. DOI: 10.15802/stp2025/341773.

108.Шинкаренко, В. І., Жадан, А. А., Галушко, О. В. (2023). Мультиагентне конструктивне моделювання часових рядів. Сучасні інформаційні та комунікаційні технології на транспорті, в промисловості та освіті. Дніпро. с. 64.

109.Шинкаренко, В. И., Литвиненко, К. В., Чигирь, Р. Р., Жадан, А. А. (2018). Констуктивно-продукционное моделирование фракталов. Інтелектуальні системи прийняття рішень і проблеми обчислювального інтелекту. Матеріали міжнародної наукової конференції. Херсон. 289–291.

110. Шинкаренко, В. И., Литвиненко, К. В., Чигирь, Р. Р., Жадан, А. А. (2018). Констуктивно-продукционное моделирование фрактальных временных рядов на основе L-систем. В Проблеми математичного моделювання. Матеріали Всеукраїнської науково-методичної конференції Кам'янське: ДДТУ. 161–163.

111.Шинкаренко, В. І., Литвиненко, К. В., Чигір, Р. Р., Жадан, А. А. (2017). Вариативность уточняющих преобразований конструктивно-продукционного моделирования. Теоретичні та прикладні аспекти побудови програмних систем (TAAPSD'2017). Київ. 225–230.

112.Шинкаренко, В. І., Чигір, Р. Р., Жадан, А. А. (2017). Комп'ютерна програма «Моделювання часових рядів із заданими фрактальними властивостями». Свідоцтво про реєстрацію авторського права на твір № 72579 від 27.06.2017.

113. Шинкаренко, В. І., Чигір, Р. Р., Жадан, А. А. (2019). Комп'ютерна програма «Моделювання взаємопов'язаних часових рядів й геометричних фракталів породжуваних в L-системах». Рішення про реєстрацію договору, який стосується права автора на твір № 4369 від 31.05.2019.

114. Шинкаренко, В. І., Чигір, Р. Р., Жадан, А. А. (2019). Комп'ютерна програма «Рекурентний аналіз часових рядів породжуваних в L-системах». Рішення про реєстрацію договору, який стосується права автора на твір № 4367 від 29.05.2019.

115.Шинкаренко, В. І., Чигир, Р. Р., Жадан, А. А. (2017). Моделювання взаємопов'язаних часових рядів та зображень L-подібних конструктивно-продукційних структур. Інформаційні технології в металургії та машинобудуванні. Дніпро. с. 98.

Додаток А Список публікацій здобувача та відомості про апробацію результатів дисертації

Праці у фахових виданнях МОН України

1.Шинкаренко, В. И.[1], ***Жадан, А. А.[2]*** (2025). Використання методів конструктивно-продукційного моделювання для прогнозування детермінованих часових рядів. Системи і технології, 6, 124-137. DOI: 10.34185/1562-9945-5-161-2025-12 ([1] – керування дослідженням, обговорення результатів, [2] – визначення дослідницьких підходів, розробка програмного рішення).

2.Шинкаренко, В. И.[1], ***Жадан, А. А.[2]*** (2025). Сучасні методи та засоби роботи з часовими рядами. Наука і прогрес транспорту. 4(112), 142–154. DOI: 10.15802/stp2025/341773 ([1] – керування дослідженням, обговорення результатів, [2] – проведення аналітичних досліджень, порівняння існуючих методів, формалізація результатів).

3.Шинкаренко, В. И.[1], ***Жадан, А. А.[2]*** (2025). Реалізація процесу відновлення конструктивно-продукційних моделей фрактальних часових рядів. 4, 3-11. DOI: 10.15407/pp2025.04 ([1] – керування дослідженням, обговорення результатів, [2] – проведення архітектурного аналізу програмного рішення, визначення основних компонентів, реалізація та верифікація програмного додатка).

Праці міжнародних конференцій, включені до міжнародної наукометричної бази Scopus

4.Shynkarenko, V.[1], ***Zhadan, A.[2]*** (2020). Modeling of the deterministic fractal time series by one rule constructors. In: Proceedings of the 2020 IEEE 15th International Conference on Computer Sciences and Information Technologies (CSIT), Zbarazh, Ukraine, pp. 336–339. DOI: 10.1109/CSIT49958.2020.9321923 ([1] – керування дослідженням, обговорення результатів, визначення теоретичних засад, [2] – визначення підходів до реалізації, написання програмного додатка, верифікація результатів).

5.Shinkarenko, V.[1], ***Zhadan, A.[2]*** (2024). Multiservice Architecture of Software for Stochastic Fractal Time Series Forecasting. 2024 IEEE 19th International Conference

on Computer Science and Information Technologies (CSIT), 1-4. DOI: 10.1109/CSIT65290.2024.10982626 ([1] – керування дослідженням, обговорення результатів, [2] – архітектурний аналіз, визначення підходів до реалізації, написання програмного додатка, верифікація результатів).

6.Shynkarenko, V.[1], ***Zhadan, A.,[2]*** & Halushka, O.[3] (2024). Multi-Agent System for Reconstruction Constructive Models of Stochastic Fractal Time Series. Proceedings of the Information Technology and Implementation (IT&I) Workshop: IT Infrastructure and Applied Solutions (IT&I-WS 2024: ITIAS), Kyiv, Ukraine, November 20 – 21, pp. 66-77. ([1] – керування дослідженням, обговорення результатів, [2] – архітектурний аналіз, визначення підходів до реалізації, написання програмного додатка, 3 – написання програмного додатка, аналіз результатів).

Матеріали міжнародних наукових конференцій

7.Шинкаренко, В. І.[1], Литвиненко, К. В.[2], Чигір, Р. Р.[3], ***Жадан, А. А.[4]*** (2017). Вариативность уточняющих преобразований конструктивно-продукционного моделирования. В «Теоретичні та прикладні аспекти побудови програмних систем» (TAAPSD'2017). Київ. с. 225–230. ([1] – теоретичні положення, [2] – підготовка матеріалов до друку, обговорення результатів, [3] – розробка програмного забезпечення, проведення досліджень з лінійних геометричних фракталів, [4] – проведення дослідження з фрактальних часових рядів)

8.Шинкаренко, В. І.[1], Чигир, Р. Р.[2], ***Жадан, А. А.[3]*** (2017). Моделювання взаємопов'язаних часових рядів та зображень L-подібних конструктивно-продукційних структур. В «Інформаційні технології в металургії та машинобудуванні». Дніпро. с. 98. ([1] – теоретичні положення, [2] – розробка програмного забезпечення, [3] – тестування та верифікація результатів)

9.Шинкаренко, В. И.[1], Литвиненко, К. В.[2], Чигирь, Р. Р.[3], ***Жадан, А. А.[4]*** (2018). Констуктивно-продукционное моделирование фракталов. В «Інтелектуальні системи прийняття рішень і проблеми обчислювального інтелекту». Матеріали міжнародної наукової конференції. Херсон: ПП Вишемирський В.С. с. 289–291 ([1] – керування дослідженням, обговорення результатів, [2] – підготовка матеріалов до

друку, обговорення результатів, [3] – розробка програмного забезпечення та конструктивно-продукційних моделей фрактальних структур, проведення досліджень, [4] – розробка частини програми, обговорення результатів)

10.Шинкаренко, В. І.[1], ***Жадан, А. А.[2]*** (2022). Конструктивно-продукційне моделювання фрактальних складових часових рядів. В «Сучасні інформаційні та комунікаційні технології на транспорті, в промисловості та освіті». Дніпро. с. 71 ([1] – керування дослідженням, обговорення результатів, теоретичні засади, [2] – програмна реалізація, аналіз результатів).

11.Шинкаренко, В. І.[1], ***Жадан, А. А.[2]*** (2023). Конструктивно-продукційне моделювання детермінованих часових рядів з фрактальними властивостями. В «Інформаційні технології в металургії та машинобудуванні». Дніпро. с. 222–226. DOI: 10.34185/1991-7848.itmm.2023.01.060 ([1] – керування дослідженням, обговорення результатів, теоретичні засади, [2] – програмна реалізація, аналіз результатів).

12.Шинкаренко, В. І.[1], ***Жадан, А. А.[2]***, Галушко, О. В.[3] (2023). Мультиагентне конструктивне моделювання часових рядів. В «Сучасні інформаційні та комунікаційні технології на транспорті, в промисловості та освіті». Дніпро. с. 64 ([1] – керування дослідженням, обговорення результатів, [2] – програмна реалізація, архітектурний аналіз, [3] – програмна реалізація, аналіз результатів).

13.Шинкаренко, В. І.[1], ***Жадан, А. А.[2]*** (2024). Мультиагентне середовище відновлення фрактальних часових рядів. В «Інформаційні технології в металургії та машинобудуванні». Дніпро. с. 232–236. DOI: 10.34185/1991-7848.itmm.2024.01.041 ([1] – керування дослідженням, обговорення результатів, [2] – програмна реалізація, аналіз результатів).

14.Шинкаренко, В. І.[1], ***Жадан, А. А.[2]*** (2024). Проблеми синхронізації розподілених обчислень на прикладі мікросервісної реалізації генетичного алгоритму. В «Сучасні інформаційні та комунікаційні технології на транспорті, в промисловості та освіті». Дніпро. с. 93 ([1] – керування дослідженням, обговорення результатів, [2] – аналіз підходів до реалізації, імплементація прототипів, верифікація результатів).

15.Шинкаренко, В. І.[1], ***Жадан, А. А.[2]*** (2025). Реалізація розподілення процесу роботи генетичного алгоритму шляхом впровадження каналів та вузлів обробки повідомлень. В «Інформаційні технології в металургії та машинобудуванні». Дніпро. с. 215–219. DOI: 10.34185/1991-7848.itmm.2025.01.037 ([1] – керування дослідженням, обговорення результатів, [2] – аналіз підходів до реалізації, імплементація прототипів, верифікація результатів).

16.Шинкаренко, В. І.[1], ***Жадан, А. А.[2]*** (2025). Проблеми синхронізації розподілених обчислень на прикладі мікросервісної реалізації генетичного алгоритму. В «Сучасні інформаційні та комунікаційні технології на транспорті, в промисловості та освіті». Дніпро. с. 98 ([1] – керування дослідженням, обговорення результатів, [2] – аналіз підходів до реалізації, імплементація прототипів, верифікація результатів).

Матеріали наукових конференцій

17. Шинкаренко, В. И.[1], Литвиненко, К. В.[2], Чигирь, Р. Р.[3], ***Жадан, А. А.[4]*** (2018). Констуктивно-продукционное моделирование фрактальных временных рядов на основе L-систем. В Проблеми математичного моделювання. Матеріали Всеукраїнської науково-методичної конференції. Кам'янське: ДДТУ. 161–163 ([1] – керівництво роботою, [2] – підготовка матеріалов до друку, обговорення результатів, [3] – розробка програмного забезпечення для моделювання часових рядів, [4] – планування та проведення дослідження)

Свідоцтва про реєстрацію авторського права

18. Шинкаренко, В. І.[1], Чигір, Р. Р.[2], ***Жадан, А. А.***[3] (2019). Комп'ютерна програма «Рекурентний аналіз часових рядів породжуваних в L-системах». Рішення про реєстрацію договору, який стосується права автора на твір № 4367 від 29.05.2019. ([1] – керівництво розробкою, [2] – розробка частини програми, тестування, [3] – розробка частини програми)

19. Шинкаренко, В. І.[1], Чигір, Р. Р.[2], ***Жадан, А. А.***[3] (2019). Комп'ютерна програма «Моделювання взаємопов'язаних часових рядів й геометричних фракталів породжуваних в L-системах». Рішення про реєстрацію договору, який стосується пра-

ва автора на твір № 4369 від 31.05.2019. ([1] – керівництво розробкою, [2] – розробка частини програми, [3] – розробка частини програми, тестування)

20. Шинкаренко, В. І.[1], Чигір, Р. Р.[2], ***Жадан, А. А.***[3] (2017). Комп'ютерна програма «Моделювання часових рядів із заданими фрактальними властивостями». Свідоцтво про реєстрацію авторського права на твір № 72579 від 27.06.2017. ([1] – керівництво розробкою, [2] – розробка частини програми, тестування, [3] – розробка частини програми)

Додаток Б Акт впровадження

«ЗАТВЕРДЖУЮ»
Перший проректор
Українського державного університету
науки і технологій
_____ д.т.н., проф. Анатолій РАДКЕВИЧ
_____ 14.04.2026 р.

АКТ

Про використання результатів дисертації
«КОНСТРУКТИВНО-ПРОДУКЦІЙНЕ МОДЕЛЮВАННЯ ФРАКТАЛЬНИХ СКЛАДОВИХ ЧАСОВИХ РЯДІВ»
Жадана Артема Анатолійовича,
представленої на здобуття наукового ступеня доктора філософії
за спеціальністю 122 «Комп'ютерні науки»

Цей акт складений про те, що у навчальному процесі Дніпровському інституту інфраструктури і транспорту Українського державного університету науки і технологій при підготовці аспірантів за спеціальністю 122 «Комп'ютерні науки» використовуються наукові та практичні результати, що отримані в дисертації Жадана А.А., при викладені дисципліни «Ефективність інформаційних систем та комп'ютерних технологій».

Зав. кафедри
Комп'ютерні інформаційні технології
к.т.н., доцент Вадим ГОРЯЧКІН

Декан факультету
Комп'ютерні технології і системи
к.т.н., доцент Володимир МАЛОВІЧКО