

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Міністерство освіти і науки України

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Міністерство освіти і науки України

Кваліфікаційна наукова
праця на правах рукопису

Горбатюк Владислав Сергійович

УДК 004.032.26

ДИСЕРТАЦІЯ

**ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ПРОГНОЗУВАННЯ
НЕСТАЦІОНАРНИХ ЧАСОВИХ РЯДІВ НА ОСНОВІ НЕЙРОННИХ
МЕРЕЖ**

05.13.06 – інформаційні технології

Подається на здобуття наукового ступеня кандидата технічних наук

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



Горбатюк В.С.

Науковий керівник: Чумаченко Олена Іллівна, д.т.н., доцент

Київ – 2021

АНОТАЦІЯ

Горбатюк В.С. Інформаційна технологія прогнозування нестационарних часових рядів на основі нейронних мереж. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня кандидата технічних наук за спеціальністю 05.13.06 – інформаційні технології. – Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського» МОН України, Київ, 2021.

Задача прогнозування часових рядів є актуальною науково–прикладною проблемою, потреба у вирішенні якої виникає в тому числі у системах підтримки прийняття рішень, де на основі результатів прогнозування виконується планування та приймаються ті чи інші рішення, і точність прогнозу має велике значення. При цьому у загальному випадку побудувати адекватну прогнозуючу модель неможливо, і тому усі існуючі методи спираються на додаткові припущення про характер часових рядів, що прогнозуються. Застосування окремого методу для побудови прогнозуючої моделі для часового ряду, який не задовольняє припущення цього методу, призводить до погіршення точності прогнозів отриманої моделі. Оскільки існуючі реальні часові ряди є дуже різноманітними за своїми властивостями, актуальною є задача розробки інформаційної технології, методів та моделей для прогнозування часових рядів, які відповідають певним припущенням, що відрізняються від припущень існуючих методів (за умови реальної наявності таких часових рядів та потреби у їх прогнозуванні), у такий спосіб підвищуючи точність прогнозів отриманих прогнозуючих моделей для цих часових рядів.

Використання технологій на основі штучного інтелекту дозволяє позбавитися окремих припущень та недоліків класичних відомих підходів, що

базуються на застосуванні теорії ймовірності та випадкових процесів. На сьогодні найбільш перспективним напрямком інтелектуальних технологій вважаються штучні нейронні мережі завдяки їх універсальності та вражаючим результатам, отриманим у різноманітних сферах їх використання, таким як: аналіз зображень та відео, аналіз тексту, аналіз мови та інші. Тому наукове завдання дисертації полягає у розробці інформаційної технології прогнозування нестационарних часових рядів на основі штучних нейронних мереж, що задовольняють певні припущення.

У роботі виділено клас нестационарних часових рядів, що відповідають додатковим припущенням, який названо класом нестационарних часових рядів з декількома потенційними умовними розподілами, та наведено приклади реальних часових рядів, що належать до цього класу. Розглянуто основні існуючі методи прогнозування часових рядів, проаналізовано основні переваги та недоліки цих методів при застосуванні для прогнозування рядів з виділеного класу. На основі аналізу недоліків і переваг існуючих методів та властивостей виділеного класу часових рядів обґрунтована потреба в новій інформаційній технології і розроблено новий загальний підхід побудови прогнозуючої моделі для класу нестационарних часових рядів з декількома потенційними умовними розподілами.

Синтезовано нову модель штучного нейрону Sigmoid Piecewise (SP), який є однією з найпростіших прогнозуючих моделей, що складаються з декількох локальних моделей та моделі-компоненту, що визначає, яку з локальних моделей застосовувати до вхідного вектору. Така модель краще підходить для опису часових рядів з декількома потенційними умовними розподілами, і проведені порівняльні тести на реальних вибірках це підтверджують.

Вдосконалено багаторядний алгоритм МГУА шляхом використання

методів зворотного поширення помилки та вибіркового виключення, що дозволило оптимальніше налаштовувати параметри прогнозуючої моделі.

Синтезовано нові методи побудови прогнозуючих моделей для класу нестационарних часових рядів з декількома потенційними умовними розподілами на основі розробленого загального підходу та методів налаштування локальних моделей. Перший розроблений метод базується на застосуванні методу кластеризації та штучних нейронних мереж з SP нейронами в якості локальних моделей, з використанням спеціальної регуляризації в процесі налаштування локальних моделей. Другий метод використовує регуляризований метод комбінації експертів, де в якості моделей–експертів використовуються штучні нейронні мережі з SP нейронами.

Розроблено метод м'якої кластеризації на основі роздільних гіперповерхонь. Метод дозволяє вирішити задачу кластеризації шляхом налаштування параметрів нейронної мережі, у такий спосіб дозволяючи використовувати розроблені моделі. Іншою перевагою методу є можливість знаходити кластери, які розділені в просторі гіперповерхнями, складність яких залежить від розміру мережі, тобто збільшення мережі дозволяє отримати більш складну роздільну гіперповерхню.

Нові методи були порівняні з існуючими методами на реальних вибірках і показали кращі результати на тестових підвибірках. У такий спосіб, отримано нові та вдосконалено існуючі методи побудови прогнозуючих моделей, що враховують властивості часових рядів з декількома потенційними умовними розподілами та дозволяють отримати моделі з більшою точністю прогнозу.

Розглянуто вимоги та сценарії використання інформаційної технології (ІТ) прогнозування нестационарних часових рядів (ПНЧР) на основі штучних нейронних мереж з використанням розроблених методів, моделей та

алгоритмів. На основі наведених вимог та сценаріїв використання розроблено архітектуру інформаційної технології.

Розроблено ІТ для прогнозування часових рядів, що використовує створені методи та моделі прогнозування часових рядів та розглянуто приклади її застосування для побудови прогнозуючих моделей і систем для реальних вибірок. За рахунок використання синтезованих методів та моделей, що враховують властивості нестационарних часових рядів з декількома потенційними умовними розподілами, ця ІТ дозволяє отримувати точніший прогноз для рядів з цього класу.

Інформаційну технологію впроваджено для задачі прогнозування попиту на лінійку пристроїв мікропроцесорного релейного захисту серій на державному підприємстві «Виробниче об'єднання Київприлад», що дозволило точніше визначити потребу в комплектуючих та витратних матеріалах і тим самим скоротити витрати.

Ключові слова: штучні нейронні мережі, прогнозування нестационарних часових рядів, метод групового урахування аргументів, інформаційна технологія, метод комбінації експертів, м'яка кластеризація, нелінійна оптимізація.

ABSTRACT

Horbatiuk V.S. Information technology for non-stationary time series forecasting based on neural networks. – Qualifying scientific work published as a manuscript.

Thesis for a Candidate Degree in Engineering, specialty 05.13.06 – information technologies. – National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute» of Ministry of Education and Science of Ukraine, Kyiv, 2021.

The paper is devoted to the development of new methods for solving the current scientific and applied problem of time series forecasting, namely a certain class of non-stationary time series. This problem is of great scientific and practical importance, and serves as basis for another important problem – planning. Thus, the development of new forecasting methods with better forecasting accuracy increases planning efficiency.

The thesis considers the problem of non-stationary time series forecasting and presents current practical problems that need to be solved, thus confirming the relevance of this problem. A class of non-stationary time series with several potential conditional distributions is selected, for which information technology has been developed, which includes new methods, models, and algorithms. After allocating a specific class of time series, it becomes possible to use the properties of the class in the development of new methods, models, and algorithms.

Existing methods, algorithms and models for non-stationary time series forecasting are considered, which allowed to analyze the main advantages and disadvantages of these methods when being used to forecast time series from a selected class, and, therefore, to take them into account when developing new methods. The need for new information technology is substantiated based on the

analysis of the existing methods' shortcomings.

The properties of non-stationary time series class with several potential conditional distributions are investigated in terms of further development of new methods for constructing forecasting models for series of this class. The investigated properties allowed to synthesize new methods for constructing more accurate forecasting models for time series of the selected class.

A new general method for constructing a forecasting model for non-stationary time series class with several potential conditional distributions has been developed based on the time series properties analysis belonging to this class. The method considers the series properties of the selected class and the shortcomings of the existing forecasting methods when being applied to this class. Thus, it is more suitable for constructing the forecasting time series models with several potential conditional distributions.

A new model of the artificial Sigmoid Piecewise (SP) neuron has been developed, being one of the simplest forecasting models which consists of several local models and a component model that determines which of the local models to apply to the input vector. This model is better suited for describing the time series with several potential conditional distributions, and comparative tests carried out on real samples confirm this.

The multi-row GMDH algorithm has been improved by using the methods of error back propagation and selective exclusion, which allowed to optimally adjust the parameters of the forecasting model.

The problem of constructing the active distribution estimator component when constructing a forecasting model for the non-stationary time series class with several potential conditional distributions is considered and new methods of its solution are developed. A method using a soft clustering algorithm based on separate hypersurfaces has been synthesized. The method allows to solve the clustering

problem by adjusting the parameters of the neural network or any other model to approximate the unknown functions, thus allowing to use the methods and models developed in the second section of this paper. The ability to find clusters that are separated in space by hypersurfaces of any complexity (with sufficient approximating ability of the model) is another advantage of the method. A method which uses a regularized algorithm of expert combination has also been developed. The method is an improvement of the usual mixture of experts algorithm due to the use of special regularization which allows to reduce the impact of the retraining problem and improve the local models forecasting properties.

New methods for constructing forecasting models for non-stationary time series class with several potential conditional distributions have been synthesized on the basis of the developed general method, methods for local models setting up, and methods for constructing the active distribution estimator component. The first developed method grounds on the clustering algorithm and artificial neural networks application with SP neurons as local models, using special regularization in the process of local models setting up. The second method uses a regularized mixture of experts algorithm, where artificial neural networks with SP neurons are used as expert models.

The new methods have been compared with existing ones on the several real samples and they showed better results on test subsamples. Thus, the existing methods have been improved and the new methods for constructing the forecasting models have been obtained that consider the properties of time series with several potential conditional distributions and produce models with higher accuracy of the forecast.

Requirements and scenarios of using information technology for non-stationary time series forecasting based on artificial neural networks using the developed methods, models, and algorithms are considered. Based on the given

requirements and usage scenarios, the structure of the new information technology has been developed consisting of a list of key components, the relationships between them and the architecture of individual components.

Information technology for time series forecasting is developed, which uses the designed methods and models of time series forecasting and examples of its application of forecasting models and systems construction for real samples are considered. Due to the use of synthesized methods and models which consider the properties of non-stationary time series with several potential conditional distributions, this information technology allows to obtain a more accurate forecast for series of this class.

The above-mentioned information technology was implemented to forecast the demand for the line of devices for microprocessor-based relay protection series at the state enterprise «Production Association Kyivprylad», which allowed to determine the demand for components and consumables more precisely and, thereby, to reduce costs.

Keywords: artificial neural networks, non-stationary time series forecasting, group method of data handling, information technology, mixture of experts method, soft clustering, nonlinear optimization.

СПИСОК ОПУБЛІКОВАНИХ ПРАЦЬ ЗА ТЕМОЮ ДИСЕРТАЦІЇ

Монографії:

1. Интеллектуальные методы прогнозирования [Текст]: монография / В. М. Синеглазов, Е. И. Чумаченко, В. С. Горбатюк. – К.: Освіта України, 2013. – 219 с. *(Особистий внесок: аналіз проблеми прогнозування нестационарних часових рядів та існуючих методів її вирішення).*

У виданнях іноземних держав–членів ЄС:

2. An algorithm for solving the problem of forecasting / Victor Sineglazov, Elena Chumachenko, Vladyslav Gorbatyuk // Aviation. – 2013. №1(17), p. 9–13. (Видання іноземної держави–члена ЄС, Литва. Входить до міжнародних наукометричних баз ESCI, DOAJ, EBSCOhost, Elsevier Bibliographic Databases: SCOPUS, El Compendex Gale, Google Scholar, ICONDA, Microsoft Academic, MyScienceWork, OpenAIRE2020, ProQuest). *(Особистий внесок: розробка вдосконаленої моделі прогнозування на основі нейронних мереж та методу групового урахування аргументів).*

3. Using a mixture of experts' approach to solve the forecasting task / Victor Sineglazov, Elena Chumachenko, Vladyslav Gorbatyuk // Aviation. – 2014. №3(18), p. 129–133. (Видання іноземної держави–члена ЄС, Литва. Входить до міжнародних наукометричних баз ESCI, DOAJ, EBSCOhost, Elsevier Bibliographic Databases: SCOPUS, El Compendex Gale, Google Scholar, ICONDA, Microsoft Academic, MyScienceWork, OpenAIRE2020, ProQuest). *(Особистий внесок: розробка спеціального регуляризуючого критерію, що застосовується для побудови локальних моделей).*

4. Forecasting aircraft miles flown time series using deep learning–based hybrid approach / Victor Sineglazov, Elena Chumachenko, Vladyslav Gorbatyuk // Aviation. – 2018. № 1(22), p. 6–12. (Видання іноземної держави–члена ЄС, Литва. Входить до міжнародних наукометричних баз ESCI, DOAJ, EBSCOhost,

Elsevier Bibliographic Databases: SCOPUS, El Compendex Gale, Google Scholar, ICONDA, Microsoft Academic, MyScienceWork, OpenAIRE2020, ProQuest). *(Особистий внесок: розробка вдосконаленого багаторядного алгоритму методу групового урахування аргументів).*

5. An algorithm for the corporate shares price forecasting / V. Sineglazov, O. Chumachenko, V. Gorbatiuk // Information Technology in Selected Areas of Management 2017. – 2018, p. 73–79. (Видання іноземної держави–члена ЄС, Польща. Входить до міжнародної наукометричної бази Scopus, Google Scholar). *(Особистий внесок: розробка загального підходу до побудови моделі).*

У фахових наукових виданнях України:

6. Soft clustering algorithm based on separating hypersurfaces / O. I. Chumachenko, V. S. Gorbatiuk // Electronics and control systems. – 2017. Volume 2, № 52, p. 11–15. (Наукове фахове видання з технічних наук. Входить до міжнародних наукометричних баз РІНЦ, Google Scholar, EBSCO, Index Copernicus, Наукова періодика України). *(Особистий внесок: розробка вдосконаленого методу м'якої кластеризації).*

7. Structural–parametric synthesis of the feedforward neural networks with sigmoid piecewise–type neurons / M. Z. Zgurovsky, O. I. Chumachenko, V. S. Gorbatiuk // Electronics and control systems. – 2018. Volume 4, № 58, p. 42–47. (Наукове фахове видання з технічних наук. Входить до міжнародних наукометричних баз РІНЦ, Google Scholar, EBSCO, Index Copernicus, Наукова періодика України). *(Особистий внесок: розробка нової моделі штучного нейрону sigmoid piecewise).*

8. Комплексування декількох алгоритмів під час розв'язання задачі прогнозування / О. І. Чумаченко, В. С. Горбатюк // Житомир: Вісник ЖДТУ. Серія: Технічні науки. – 2016. №1 (76), С. 101–106. (Наукове фахове видання з технічних наук. Входить до міжнародних наукометричних баз Index

Copernicus, DOAJ, Google Scholar, WorldCat, BASE, eLibrary, Наукова періодика України). *(Особистий внесок: розробка методу прогнозування, що базується на використанні декількох локальних прогнозуючих моделей).*

9. A method for building a forecasting model with dynamic weights / Victor Sineglazov, Elena Chumachenko, Vladyslav Gorbatyuk // Eastern–European Journal of Enterprise Technologies. – 2014. [S.l.], v. 2, n. 4(68), p. 4–8. (Наукове фахове видання з технічних наук. Входить до міжнародних наукометричних баз Scopus, CrossRef, Index Copernicus, DOAJ, Google Scholar, OpenAIRE, WorldCat, CORE, SIS, CAS, BASE, EBSCO, Cite Factor, Наукова періодика України). *(Особистий внесок: розробка алгоритму побудови прогнозуючої моделі, який застосовує динамічні ваги).*

10. Алгоритм решения задачи прогнозирования / Е.И. Чумаченко, В.С. Горбатюк // Штучний інтелект. – 2012. № 2, С. 24–31. (Наукове фахове видання з технічних наук. Входить до міжнародних наукометричних баз Google Scholar, Наукова періодика України). *(Особистий внесок: розробка алгоритму прогнозування на базі штучних нейронних мереж).*

11. Метод решения задачи прогнозирования на основе комплексирования оценок / В. М. Синеглазов, Е. И. Чумаченко, В. С. Горбатюк // Индуктивное моделирование сложных систем. – 2012. Выпуск 4, С. 214–223. (Наукове фахове видання з технічних наук. Входить до міжнародних наукометричних баз РІНЦ, Google Scholar, Наукова періодика України). *(Особистий внесок: дослідження існуючих методів прогнозування, що використовують декілька локальних моделей–нейронних мереж).*

12. Использование искусственных нейронных сетей для задачи прогнозирования / Е. И. Чумаченко, В. С. Горбатюк // Электроника и системы управления. – 2012. № 1, С. 113–119. (Наукове фахове видання з технічних наук. Входить до міжнародних наукометричних баз РІНЦ, Google Scholar,

EBSCO, Index Copernicus, Наукова періодика України). *(Особистий внесок: аналіз переваг та недоліків застосування штучних нейронних мереж в якості моделей прогнозування).*

Матеріали конференцій:

13. One approach for the forecasting task solution / Victor Sineglazov, Elena Chumachenko, Vladyslav Gorbatyuk // Безпека в авіації та космічні технології: тези доповідей V Всесвітнього конгресу «Авіація у XXI столітті». – 2012, р. 3.5.49–3.5.53.

14. Method for predicting failure risk of UAV navigation systems / Elena Chumachenko, Vladyslav Gorbatyuk // Methods and systems of navigation and motion control (MSNMC – 2012). Book of abstracts. – 2012, р. 63–65.

15. Алгоритм решения задачи прогнозирования / Е.И. Чумаченко, В.С. Горбатюк // Интеллектуальные системы принятия решений и проблемы вычислительного интеллекта (ISDMCI – 2012): тезисы докладов. – 2012, С. 428–430.

16. Интеллектуальная система прогнозирования рисков послеоперационных осложнений / В. М. Синеглазов, Е. И. Чумаченко, В. С. Горбатюк // Интеллектуальные системы принятия решений и проблемы вычислительного интеллекта (ISDMCI – 2013): тезисы докладов. – 2013, С. 289–291.

17. Applying Different Neural Network's Topologies to the Forecasting Task / Victor Sineglazov, Elena Chumachenko, Vladyslav Gorbatyuk // International Conference in Inductive Modelling (ICIM – 2013). Book of abstracts. – 2013, р. 217–220.

18. Forecasting the Demand for UAV Using Different Neural Networks Topology / E. I. Chumachenko, V. S. Gorbatiuk // The 2–nd International

Conference, Actual Problems of Unmanned Air Vehicles Development Proceedings. – 2013, p. 62–64.

19. Комплексування декількох алгоритмів при вирішенні задачі прогнозування / О. І. Чумаченко, В. С. Горбатюк // Інформаційно–комп’ютерні технології – 2016: тези доповідей VIII Міжнародної науково–технічної конференції. – Житомир: ЖДТУ. – 2016, С. 95–96.

20. Метод побудови прогнозуючої моделі з динамічними параметрами / В. М. Синєглазов, О. І. Чумаченко, В. С. Горбатюк // Інформаційні технології та комп’ютерне моделювання: тези доповідей Міжнародної науково–практичної конференції. – Івано–Франківськ: Супрун В. П. – 2016, С. 23–26.

21. Object's movement prediction in 3D space using neural networks / O. Chumachenko, V. Gorbatiuk // 4th International Conference Methods and systems of navigation and motion control (MSNMC–2016). Book of abstracts. – 2016, p. 255–258.

22. A New Approach in Cluster Analysis / Victor Sineglazov, Elena Chumachenko, Vladyslav Gorbatiuk // 4th International Conference on Actual Problems of Unmanned Aerial Vehicles Developments (APUAVD–2017). Book of abstracts. – 2017, p. 223–227.

23. Нова модель штучного нейрону для побудови нейронних мереж у задачі апроксимації / О.І. Чумаченко, В.С. Горбатюк // Інформаційні технології та комп’ютерне моделювання: тези доповідей Міжнародної науково–практичної конференції. – Івано–Франківськ. – 2018, С. 296–299.

24. Алгоритм м’якої кластеризації на основі розділяючих гіперповерхонь / В.С. Горбатюк // Електроніка – 2018: тези доповідей XI Міжнародної науково–технічної конференції молодих вчених. – 2018, С. 425–430.

25. An algorithm for the corporate shares price forecasting / V. Sineglazov, O. Chumachenko, V. Gorbatiuk // XVIII International Scientific Conference «Corporate Governance – Theory and practice». Book of abstracts. – 2018, p. 101–102.

26. Прогнозування часових рядів з використанням регуляризованої суміші експертів / В. М. Синєглазов, О. І. Чумаченко, В. С. Горбатюк // Інформаційні системи та технології ICT–2018: тези доповідей VII Міжнародної науково–технічної конференції. – 2018, р. 457–459.

ЗМІСТ

ВСТУП	18
РОЗДІЛ 1. ПРОБЛЕМА ПРОГНОЗУВАННЯ НЕСТАЦІОНАРНИХ ЧАСОВИХ РЯДІВ ТА ІСНУЮЧИХ МЕТОДІВ ЇЇ ВИРІШЕННЯ	26
1.1. Постановка задачі прогнозування на основі наявних спостережень та огляд проблеми нестационарних часових рядів	26
1.2. Опис класу нестационарних часових рядів з декількома потенційними умовними розподілами	31
1.3. Огляд існуючих методів та алгоритмів побудови прогнозуючих моделей	33
Висновки	40
РОЗДІЛ 2. ПОБУДОВА ПРОГНОЗУЮЧОЇ МОДЕЛІ ДЛЯ КЛАСУ НЕСТАЦІОНАРНИХ ЧАСОВИХ РЯДІВ З ДЕКІЛЬКОМА ПОТЕНЦІЙНИМИ УМОВНИМИ РОЗПОДІЛАМИ	42
2.1. Розробка загального підходу побудови прогнозуючої моделі для нестационарних часових рядів з декількома потенційними умовними розподілами	42
2.2. Вдосконалення алгоритму МГУА шляхом використання алгоритмів зворотного поширення помилки та вибіркового виключення	44
2.3. Синтез нової моделі нейрону типу Sigmoid Piecewise	54
2.3.1. Налаштування параметрів нейрону Sigmoid Piecewise	60
2.3.2. Порівняння з ReLU-подібними моделями нейронів	64
2.3.3. Дослідження ефективності нейронів типу Sigmoid Piecewise на реальних вибірках	65
2.4. Використання SP нейронів у складі штучних нейронних мереж	69
Висновки	82
РОЗДІЛ 3. МЕТОДИ ПОБУДОВИ ПРОГНОЗУЮЧИХ МОДЕЛЕЙ ДЛЯ КЛАСУ НЕСТАЦІОНАРНИХ ЧАСОВИХ РЯДІВ З ДЕКІЛЬКОМА ПОТЕНЦІЙНИМИ УМОВНИМИ РОЗПОДІЛАМИ	83
3.1. Розробка методів розв'язання задачі побудови компоненту оцінювача активного розподілу	83
3.1.1. Вдосконалення методу м'якої кластеризації шляхом застосування моделі роздільних гіперповерхонь	84
3.1.2. Використання методу комбінації експертів (mixture of experts) для налаштування компоненту оцінювача активного розподілу	90
3.2. Реалізація методу побудови прогнозуючої моделі для нестационарних часових рядів з декількома потенційними умовними розподілами на основі використання методу кластеризації та ШНМ	90
3.3. Реалізація методу побудови прогнозуючої моделі для нестационарних часових рядів з декількома потенційними умовними розподілами на основі використання методу комбінації експертів	92
3.4. Тестування на штучних та реальних даних	94
Висновки	98

РОЗДІЛ 4. ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ПРОГНОЗУВАННЯ НЕСТАЦІОНАРНИХ ЧАСОВИХ РЯДІВ НА ОСНОВІ ШТУЧНИХ НЕЙРОННИХ МЕРЕЖ	100
4.1. Основні сценарії використання інформаційної технології	100
4.2. Технологічні аспекти програмної реалізації інформаційної технології прогнозування нестационарних часових рядів.....	103
4.3. Аналіз процесів обробки даних в інформаційній технології прогнозування нестационарних часових рядів.....	109
4.4. Інструкція із застосування інформаційної технології для виконання основних поставлених задач	114
4.5. Впровадження інформаційної технології для прогнозування попиту на лінійку пристроїв мікропроцесорного релейного захисту серій на державному підприємстві «Виробниче об'єднання Київприлад».....	121
Висновки	129
ВИСНОВКИ.....	131
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	133
ДОДАТОК А.....	145
ДОДАТОК Б	151
ДОДАТОК В.....	153

ВСТУП

Актуальність теми. У сучасному світі інформаційні технології застосовуються для вирішення широкого спектру завдань, у тому числі у системах підтримки прийняття рішень, які в більшості випадків потребують прогнозування часових рядів різного походження: економічних, технічних, природних та інших. На основі прогнозів ці системи будують плани та/або пропонують відповідні рішення, у такий спосіб, неправильний прогноз може призвести до неправильних планів та рішень. Усе це вказує на важливість задачі прогнозування. Нажаль, є багато чинників, що значно ускладнюють вирішення цієї задачі: непостійний характер прогнозованого об'єкту/процесу, великий рівень шуму у наявних даних, ненормальний розподіл шуму, мала кількість наявних даних та інші. Питанням прогнозування часових рядів займалися вітчизняні та іноземні вчені: Т.В. Алесинська, Є.В. Бодянський, А.Г. Івахненко, Ю.П. Зайченко, О.Г. Руденко, В.С. Степашко, П.І. Бідюк, F.A. Amir, J.L. Elman, R.A. Fisher, T. Hanazawa, G. Hinton, W.S. McCulloch, A. Waibel та інші.

Класичні підходи до розв'язання задачі прогнозування в основному ґрунтуються на застосуванні математичної статистики та теорії ймовірності і зазвичай припускають, що модель прогнозованого об'єкту відома з точністю до параметрів (лінійна, поліноміальна, експоненціальна та інші), та припускають стаціонарність прогнозованого об'єкту або деякий конкретний вигляд нестаціонарності.

Використання технологій на основі штучного інтелекту дозволяє позбавитися окремих припущень та недоліків класичних підходів. На сьогодні найбільш перспективним напрямком інтелектуальних технологій вважаються штучні нейронні мережі, у тому числі завдяки їх універсальності та

вражаючим результатам, отриманим у різноманітних сферах їх використання, таких як: аналіз зображень, відео, тексту, мови та інші. Однак, більшість методів прогнозування на основі штучного інтелекту припускають, що прогнозований процес можна описати використовуючи єдину модель, і таке припущення не завжди виконується. Крім того, якість прогнозуючої моделі, отриманої за допомогою цих методів, сильно залежить від кількості наявних даних, тому при невеликому розмірі навчальної вибірки зазвичай виникає проблема перенавчання, коли прогнозуюча модель фактично починає відтворювати шум, присутній у вибірці.

Доцільною і актуальною є задача розробки нових методів, моделей та інформаційної технології прогнозування нестационарних часових рядів з використанням нейронних мереж, що дозволить підвищити точність прогнозу за рахунок використання більш загальних припущень щодо прогнозованого процесу та підвищення стійкості моделі до проблеми перенавчання при малій кількості наявних даних.

Зв'язок роботи з науковими програмами, планами, темами. Дисертаційне дослідження виконувалось відповідно до планів НДР, програм і договорів, що виконувались в НТУУ «КПІ імені Ігоря Сікорського»:

— НДР «Методологія побудови інформаційних систем з інтелектуальною підтримкою прийняття рішень» (номер державної реєстрації 0111U002510);

— НДР «Розробка методів, алгоритмів структурно–параметричного синтезу гібридних нейронних мереж та створення на її основі інтелектуальних систем розв'язання прикладних задач» (номер державної реєстрації 0117U002595).

Мета і задачі дослідження. Метою роботи є підвищення точності прогнозу для нестационарних часових рядів шляхом створення нової

інформаційної технології прогнозування нестационарних часових рядів з використанням методів та моделей на основі нейронних мереж.

Об'єктом дослідження є процес прогнозування нестационарних часових рядів з використанням методів на основі нейронних мереж.

Предмет дослідження – моделі, методи та інформаційна технологія, на основі яких здійснюється прогнозування нестационарних часових рядів.

Основні завдання роботи, які потрібно розв'язати для досягнення мети:

- аналіз проблеми прогнозування нестационарних часових рядів та існуючих методів її вирішення;
- обґрунтування необхідності розроблення нової інформаційної технології;
- визначення класу нестационарних часових рядів, для якого буде розроблятися інформаційна технологія;
- розробка загального підходу до побудови прогнозуючої моделі для визначеного класу нестационарних часових рядів на основі аналізу властивостей часових рядів, що належать до цього класу, та переваг і недоліків існуючих методів;
- розробка нових та вдосконалення існуючих моделей для апроксимації функцій на базі штучних нейронних мереж, враховуючи специфіку визначеного класу нестационарних часових рядів;
- розробка нових методів побудови прогнозуючих моделей для визначеного класу нестационарних часових рядів на основі розробленого загального підходу та моделей;
- розробка інформаційної технології прогнозування нестационарних часових рядів на основі запропонованих методів та моделей;
- експериментальна перевірка розробленої інформаційної технології.

Методи дослідження базуються на положеннях:

- теорії штучних нейронних мереж та скінченних алгебраїчних структур, що дозволило синтезувати нові типи нейронів і архітектури штучних нейронних мереж;
- теорії оптимізації, дискретної математики та лінійної алгебри, що дозволило удосконалити методи навчання штучних нейронних мереж;
- математичного аналізу, що дозволило виявити властивості аналізованих функцій;
- імітаційного та комп'ютерного моделювання, що дозволило підтвердити достовірність отриманих теоретичних результатів;
- математичної статистики, що дозволило дослідити результати експериментів;
- методів програмування та теорії обчислювальної складності алгоритмів, що дозволило реалізувати спеціалізовану програмну систему для вирішення задачі прогнозування.

Як засіб розв'язання поставлених завдань використовувалося математичне та комп'ютерне моделювання. У дослідженнях використовувались програмні пакети Tensorflow, Keras та Matlab а також мова програмування C#.

Наукова новизна одержаних результатів.

1. Уперше запропоновано модель штучного нейрону, яка відрізняється від існуючих тим, що складається з трьох зважених суматорів та нової функції активації, яка залежить від трьох змінних, що дозволяє отримати модель, яка апроксимує певну просту кусково–лінійну функцію і при цьому дозволяє незалежно налаштовувати параметри гіперплощини, яка задає півпростори з різними лінійними функціями. За рахунок цих відмінностей нова модель має покращені апроксимуючі властивості.

2. Уперше розроблено метод розв'язання задачі прогнозування для нестационарних часових рядів, який базується на вирішенні задачі кластеризації та побудові декількох локальних прогнозуючих моделей, що відрізняється використанням вдосконаленого методу кластеризації та спеціального регуляризуючого критерію. Завдяки використанню цього критерію зменшується вплив розмірності вибірки на якість вихідної прогнозуючої моделі та підвищується стійкість локальних моделей до перенавчання.

3. Вдосконалено метод м'якої кластеризації шляхом використання моделі роздільних гіперповерхонь, що дозволяє вирішувати задачу кластеризації для певного класу критеріїв як задачу оптимізації диференційованої функції від фіксованої кількості аргументів. Таким чином кількість параметрів, які необхідно налаштовувати, не залежить від кількості прикладів, що дає можливість обмежити кількість параметрів за необхідності.

4. Вдосконалено багаторядний метод групового урахування аргументів для вирішення задачі прогнозування шляхом застосування алгоритмів зворотного поширення помилки та вибіркового виключення, що дозволило покращити точність прогнозу.

Наукові результати і висновки, отримані у дисертації, підкріплено порівняльними тестами результатів вирішення прикладних задач, отриманих шляхом використання розробленої інформаційної технології та шляхом використання відомих методів та моделей прогнозування нестационарних часових рядів.

Практичне значення одержаних результатів. На основі розроблених методів та моделей розроблено спеціалізовану інформаційну технологію для вирішення задач прогнозування нестационарних часових рядів. Інформаційна технологія була апробована шляхом:

1. Використання для прогнозування попиту на лінійку пристроїв мікропроцесорного релейного захисту серій на державному підприємстві «Виробниче об'єднання Київприлад», що дозволило визначити попит на комплектуючі та витратні матеріали і тим самим скоротити витрати (акт впровадження від 12.06.2018).

2. Використання в навчальному процесі кафедри авіаційних комп'ютерно–інтегрованих комплексів факультету аеронавігації, електроніки та телекомунікацій Національного авіаційного університету, зокрема у навчальному курсі «Нейротехнології в комп'ютерно–інтегрованих комплексах» (акт впровадження від 25.09.2020).

Особистий внесок здобувача. Основні положення і результати дисертаційної роботи одержані автором самостійно. У друкованих працях, написаних у співавторстві, авторові належить: [1] – аналіз проблеми прогнозування нестационарних часових рядів та існуючих методів її вирішення; [2] – розробка вдосконаленої моделі прогнозування на основі нейронних мереж та застосування методу групового урахування аргументів; [3] – розробка спеціального регуляризуючого критерію, що застосовується для побудови локальних моделей; [4] – розробка вдосконаленого багаторядного алгоритму методу групового урахування аргументів; [5] – розробка загального підходу до побудови прогнозуючої моделі; [6] – розробка вдосконаленого методу м'якої кластеризації; [7] – розробка нової моделі штучного нейрону sigmoid piecewise; [8] – розробка методу прогнозування, що базується на використанні декількох локальних прогнозуючих моделей; [9] – розробка алгоритму побудови прогнозуючої моделі, який застосовує динамічні ваги, що є варіантом загального підходу використання декількох локальних моделей; [10] – розробка алгоритму прогнозування на базі штучних нейронних мереж; [11] – дослідження існуючих методів прогнозування, що використовують

декілька локальних моделей–нейронних мереж; [12] – аналіз переваг та недоліків застосування штучних нейронних мереж у якості моделей прогнозування.

Апробація результатів дисертації. Основні результати дисертації доповідались та обговорювались на наступних 13-ти міжнародних і вітчизняних конференціях: V Всесвітній конгрес «Авіація у XXI столітті» – «Безпека в авіації та космічні технології». – Київ: НАУ, 25–27 вересня, 2012 р.; Methods and systems of navigation and motion control (MSNMC – 2012). – Kyiv: NAU, October 9–12, 2012; Интеллектуальные системы принятия решений и проблемы вычислительного интеллекта (ISDMCI – 2012). – м. Євпаторія, 27–31 травня, 2012 р.; Интеллектуальные системы принятия решений и проблемы вычислительного интеллекта (ISDMCI – 2013). – м. Євпаторія, 20–24 травня, 2013 р.; International Conference in Inductive Modelling (ICIM – 2013). – Kyiv: IRTC, September 15–21, 2013; VIII Міжнародна науково–технічна конференція «Інформаційно–комп’ютерні технології – 2016». – Житомир: ЖДТУ, 22–23 квітня, 2016 р.; Міжнародна науково–практична конференція «Інформаційні технології та комп’ютерне моделювання». – м. Івано–Франківськ: Прикарпатський національний університет імені Василя Стефаника, 23–28 травня, 2016 р.; 4th International Conference «Methods and systems of navigation and motion control (MSNMC–2016)». – Kyiv: NAU, October 18–20, 2016; 4th International Conference on Actual Problems of Unmanned Aerial Vehicles Developments (APUAVD–2017). – Kyiv: NAU, October 17 – 19, 2017; Міжнародна науково–практична конференція «Інформаційні технології та комп’ютерне моделювання». – м. Івано–Франківськ: Прикарпатський національний університет імені Василя Стефаника, 14–19 травня, 2018 р.; Міжнародна науково–технічна XI конференція молодих вчених «Електроніка – 2018». – м. Київ: НТУУ «КПІ імені Ігоря Сікорського», 3–5 квітня, 2018 р.;

XVIII International Scientific Conference «Corporate Governance – Theory and practice». – Kraków: AGH University of Science and Technology, June 21–22, 2018; 7–ма Міжнародна науково–технічна конференція «Інформаційні системи та технології ICT–2018». – м. Харків – Коблеве: Харківський національний університет радіоелектроніки, 10–15 вересня, 2018 р.

Публікації. За матеріалами дисертації опубліковано 26 друкованих праць, серед яких 1 монографія, 11 статей (4 з них у закордонних наукових виданнях країн, що входять до ЄС, і є у наукометричній базі даних Scopus, 5 – у фахових наукових виданнях України, що входять до бібліографічної бази даних CiteFactor, Index Copernicus та інших, 2 – у фахових виданнях України з технічних наук, які представлені у базах даних: DOAJ; Ulrich's Periodicals Directory; eLibrary; Google Scholar, ResearchBib, UIF), 14 тез доповідей на наукових конференціях.

Структура та обсяг дисертації. Робота складається зі вступу, чотирьох розділів та висновків до них, загальних висновків, що містять основні результати, списку використаних джерел (117 найменувань) та 3 додатків. Загальний обсяг роботи складає 160 сторінок. Основний зміст дисертаційної роботи викладений на 115 сторінках і містить 79 рисунків, 6 таблиць.

РОЗДІЛ 1. ПРОБЛЕМА ПРОГНОЗУВАННЯ НЕСТАЦІОНАРНИХ ЧАСОВИХ РЯДІВ ТА ІСНУЮЧИХ МЕТОДІВ ЇЇ ВИРІШЕННЯ

Задача прогнозування часових рядів виникає тоді, коли з'являється потреба оцінювати майбутню поведінку деякого процесу/об'єкту. Вирішення цієї задачі є необхідним базисом для виконання майже будь-якого планування. У загальному вигляді задачу прогнозування часових рядів вирішити з ідеальною точністю неможливо, оскільки поведінка багатьох процесів/об'єктів не є детермінованою. Тому багато існуючих методів для вирішення такої задачі базуються на використанні математичних теорій ймовірності та статистики. У будь-якому випадку, для розробки та аналізу нових методів спочатку потрібно обрати формальну математичну постановку задачі прогнозування часових рядів, що буде розглянута далі.

1.1. Постановка задачі прогнозування на основі наявних спостережень та огляд проблеми нестационарних часових рядів

Часовим рядом [9, 10] називають послідовність числових спостережень деякого процесу/об'єкту/явища, взятих у певні моменти часу, зазвичай з однаковим проміжком часу між спостереженнями. Математично часовий ряд позначають як $\{y_t\}, t=1, \dots, N$, де y_t – значення часового ряду в момент часу t . У роботі розглядається авторегресійна постановка задачі прогнозування, згідно з якою робиться припущення, що майбутні спостереження певним чином залежать від поточних (наявних) спостережень, у такий спосіб, на основі наявних спостережень можна (з деякою точністю) побудувати прогноз майбутніх. Отже, авторегресійна постановка задачі прогнозування виглядає наступним чином:

Дано:

- Навчальний часовий ряд $\{y_t\}, t = 1, \dots, N$.
- Горизонт прогнозування $h = 1, 2, \dots$ – на скільки кроків вперед потрібно прогнозувати.
- Функціонал $F: \Omega \rightarrow R$, що оцінює якість прогнозуючої моделі (тут Ω – простір усіх можливих авторегресійних прогнозуючих моделей).

Потрібно:

Знайти авторегресійну прогнозуючу модель вигляду $\hat{y}_{t+h} = f^*(y_t, \dots, y_{t-k+1})$, що максимізує функціонал якості моделі: $f^* = \operatorname{argmax}_{f \in \Omega} \{F(f)\}$.

Варто зауважити, що зазвичай неможливо обчислити функціонал F напряму, оскільки він може включати в себе невідомі складові та/або нескінченну кількість складових, у цьому випадку використовують певне наближення $\hat{F}: \Omega \rightarrow R$ до функціоналу F , наприклад, принцип мінімізації емпіричного ризику [11, 12].

Найбільш популярними методами дослідження авторегресійної постановки задачі прогнозування є теорія ймовірності та математична статистика [13, 14]. Згідно з цими методами на основі наявних даних оцінюється умовний розподіл ймовірності майбутнього спостереження при заданих значеннях наявних спостережень $\hat{p}(y_{t+h} | y_t, \dots, y_{t-k+1})$ і далі будується відповідна авторегресійна прогнозуюча модель, що використовує оцінений розподіл, наприклад:

$$\hat{y}_{t+h} = f(y_t, \dots, y_{t-k+1}) = \operatorname{argmax}_{y_{t+h}} \{ \hat{p}(y_{t+h} | y_t, \dots, y_{t-k+1}) \},$$

або

$$\hat{y}_{t+h} = f(y_t, \dots, y_{t-k+1}) = E[\hat{p}(y_{t+h} | y_t, \dots, y_{t-k+1})].$$

З точки зору постійності справжнього умовного розподілу

$p(y_{t+h} | y_t, \dots, y_{t-k+1})$ для різних значень t усі випадкові процеси можна поділити на 2 класи:

1. Стаціонарні у вузькому сенсі процеси [15], для яких виконується умова $F(y_{t+h}, y_t, \dots, y_{t-k+1}) = F(y_{k+h}, y_k, \dots, y_1), \forall t = 1, 2, \dots$, тобто функція розподілу ймовірностей для сусідніх точок не змінюється з плином часу t , у такому випадку і умовний розподіл $p(y_{t+h} | y_t, \dots, y_{t-k+1})$ не буде змінюватися з плином часу t . У такий спосіб, якщо часовий ряд спостережень $\{y_t\}, t = 1, \dots, N$ згенеровано стаціонарним у вузькому сенсі процесом, то потрібно лише оцінити один єдиний умовний розподіл $p(y_{t+h} | y_t, \dots, y_{t-k+1})$, що значно полегшує задачу оцінки розподілу.

2. Нестационарні процеси [16], сумісний розподіл $F(y_{t+h}, y_t, \dots, y_{t-k+1})$ яких може змінюватися з плином часу t , що ускладнює задачу оцінки умовного розподілу $p(y_{t+h} | y_t, \dots, y_{t-k+1})$, оскільки на основі наявного навчального часового ряду $\{y_t\}, t = 1, \dots, N$ тепер потрібно оцінити потенційно більш ніж один умовний розподіл, крім того, умовний розподіл для майбутніх спостережень також може змінитися.

Отже, побудова прогнозуючих моделей для нестационарних часових рядів є складною задачею, і у загальному вигляді вона досі не є вирішеною. Крім того, вважається, що більшість реальних часових рядів є нестационарними [17]. Розглянемо приклади різноманітних сфер, де виникає потреба прогнозувати нестационарні часові ряди.

- Економіка – багато економічних показників залежать від стану макроекономіки та інших показників. Наприклад, графік щоденних цін акцій ІВМ з 1978 року [18] має наступний вигляд:



Рисунок 1.1. Ціна акцій IBM, 1978–2018 роки

Тут помітні декілька періодів з різним характером зміни ціни акції. Різкий спад ціни акції з 2007 року та повернення до попереднього рівня у 2010 році пояснюється світовою кризою, яка відбувалася приблизно у цей період, що можна підтвердити графіком індексу Доу–Джонса [19], який є одним з основних макроекономічних показників:



Рисунок 1.2. Індекс Доу–Джонса, 1978–2018 роки

- Різноманітним кліматичним показникам притаманна сезонність – простий випадок неоднорідності, коли стани з різною поведінкою показника відомі наперед. Для деяких показників стани не є такими очевидними. Графік з відхиленнями середньорічної температури повітря [20] виглядає наступним чином:

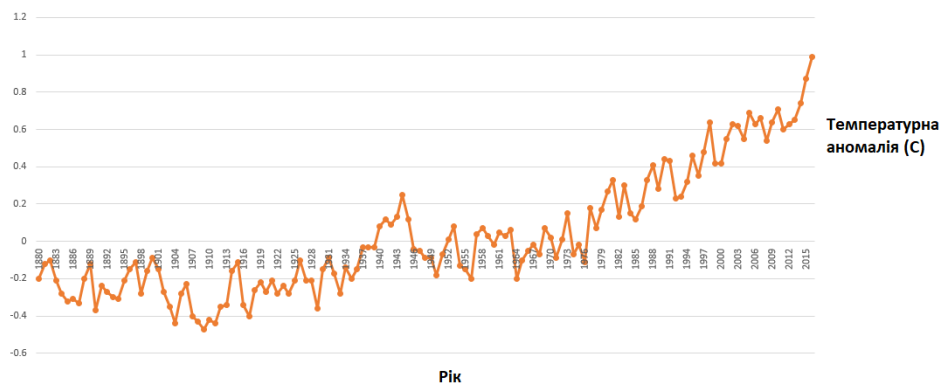


Рисунок 1.3. Відхилення середньорічної температури повітря

Візуально помітно, що приблизно з 1970 року наявний позитивний тренд, відсутній у попередніх роках.

- Прогнозування потреби/рівня споживання певного ресурсу. Наприклад, достатньо розповсюдженою та досліджуваною проблемою є прогнозування навантаження на електромережу. У роботі [21] проведено дослідження навантаження на електромережу у різні дні тижня та виділено п'ять станів з різною поведінкою процесу. Кожний стан відповідав за один або декілька днів. Для прикладу наведемо графіки для понеділку та неділі:

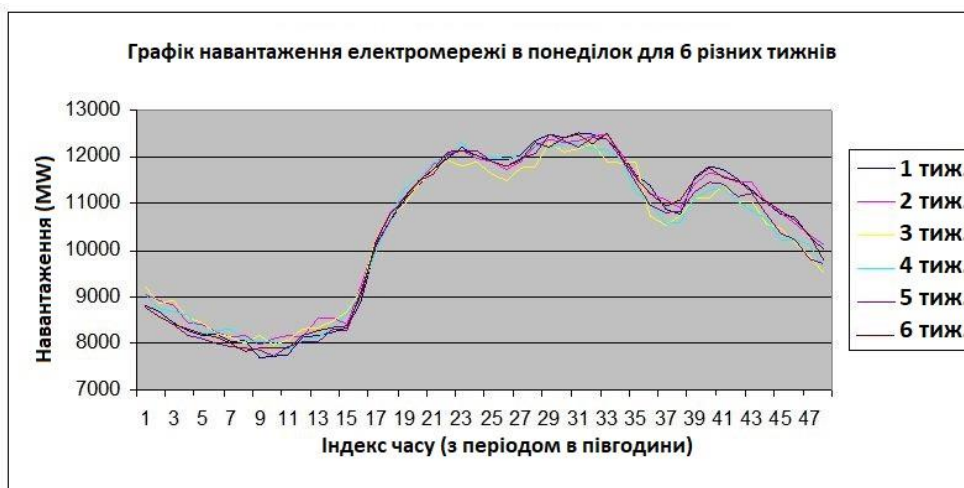


Рисунок 1.4. Електромережеве навантаження в понеділок



Рисунок 1.5. Електромережеве навантаження в неділю

Як добре видно з графіків, поведінка процесу навантаження електромережі дійсно суттєво відрізняється у цих двох станах.

Розглянуті приклади показують, що нестационарні часові ряди дійсно зустрічаються у багатьох сферах, де виникає потреба у вирішенні задачі прогнозування, а отже, розробка нових методів та алгоритмів побудови прогножуючих моделей для нестационарних часових рядів є актуальною.

1.2. Опис класу нестационарних часових рядів з декількома потенційними умовними розподілами

У загальному випадку нестационарного часового ряду умовний розподіл $p(y_{t+h} | y_t, \dots, y_{t-k+1})$ може бути різним для кожного моменту часу t , що робить задачу побудови адекватної прогножуючої моделі практично неможливою, оскільки за такої умови:

- наявний навчальний ряд не несе ніякої інформації про майбутню поведінку ряду, адже умовний розподіл $p(y_{t'+h} | y_t, \dots, y_{t-k+1})$ для майбутніх моментів часу $t' > N$ може суттєво відрізнятися від умовних розподілів, що згенерували значення з навчального ряду;
- оцінка розподілу $p(y_{t+h} | y_t, \dots, y_{t-k+1})$ по лише одній його реалізації

$y_{t+h} \sim p(y_{t+h} | y_t, \dots, y_{t-k+1})$ буде дуже неточною.

Через це більшість методів побудови прогнозуючих моделей для нестационарних часових рядів розглядають певні класи цих рядів, що володіють деякими додатковими ознаками. Розглянемо клас нестационарних часових рядів, який назвемо часовими рядами з декількома потенційними умовними розподілами. Математично ряди цього класу можна описати наступним чином:

1) Існує фіксована кількість потенційних умовних розподілів $p_i(y_{t+h} | y_t, \dots, y_1), i = 1, \dots, K$.

2) Кожне значення y_{t+h} часового ряду генерується згідно одного з потенційних умовних розподілів, який назвемо «активним розподілом в момент часу $t+h$ »:

$$p(y_{t+h} | y_t, \dots, y_{t-k+1}) = p_{a(t+h)}(y_{t+h} | y_t, \dots, y_{t-k+1}), a(t+h) \in \{1, \dots, K\},$$

де $a(t+h)$ є функцією, що повертає номер активного розподілу в момент часу $t+h$.

3) З плином часу зміна активного розподілу виникає рідко, і після виникнення зміни новий активний розподіл не буде змінюватися протягом певного проміжку часу.

Розглянуті у попередньому пункті приклади реальних часових рядів з різноманітних сфер можна віднести до рядів, що належать до описаного класу. Наприклад, візуально проаналізувавши ряд ціни акції компанії ІВМ можна помітити, що він складається з декількох сегментів умовної стаціонарності таких, як сегмент початкового повільного росту, сегменти бурхливого росту/падіння, сегменти з волатильною поведінкою та інші, що відповідають різним станам макроекономіки/інших зовнішніх чинників, які впливали на поведінку часового ряду. У якості іншого прикладу часового ряду, що належить

до наведеного класу, можна розглянути показники деякого сенсору, наприклад датчику серцевої активності, що спостерігає за об'єктом, який може знаходитися в одному з кінцевої множини станів, і в залежності від активного стану генерує той чи інший сигнал.

У такий спосіб, даний клас нестационарних часових рядів зустрічається у різних прикладних задачах прогнозування і при цьому має певні властивості, що спрощують побудову прогнозуючих моделей.

Розглянемо основні існуючі методи та алгоритми прогнозування, які можна застосовувати для побудови прогнозуючих моделей для класу часових рядів з декількома потенційними умовними розподілами та проаналізуємо їх недоліки та переваги.

1.3. Огляд існуючих методів та алгоритмів побудови прогнозуючих моделей

Кореляційно–регресійний аналіз

Цей метод може застосовуватися як для вирішення задач прогнозування значення залежної змінної від значень набору предикторів, так і для вирішення задач прогнозування часових рядів, де в якості предикторів виступають попередні значення часового ряду та/або час.

Соціально–економічні явища є результатом одночасного впливу великого числа причин. При вивченні конкретних залежностей одні ознаки є факторами, що обумовлюють зміну інших ознак. Ознаки цієї групи називаються *факторними*, а ознаки, які є результатом впливу цих факторів, називаються *результативними*.

Оцінка параметрів рівнянь регресії здійснюється методом найменших квадратів (МНК) [22, 23]. Сутність МНК полягає в знаходженні параметрів моделі, при яких мінімізується сума квадратів відхилень справжніх значень від

прогнозних, отриманих за обраним рівнянням регресії [24], наприклад, для рівняння прямої:

$$\sum (y_i - \hat{y})^2 = \sum (y_i - a_0 - a_1 \cdot x_i)^2 \rightarrow \min.$$

Основні переваги методу: оптимальність вихідної прогнозуючої моделі при виконанні певних передумов, а також її простота та робастність. Крім того, побудова і застосування прогнозуючої моделі потребує дуже малої кількості обчислень (у порівнянні з більшістю інших методів).

Основні недоліки методу: оптимальність моделі при виконанні передумов та її простота одночасно є основними недоліками, оскільки якщо передумови не виконуються або якщо потрібна більш складна модель, якість прогнозуючої моделі, отриманої згідно кореляційно-регресійного аналізу, може бути доволі низькою. У контексті нестационарних часових рядів з декількома потенційними умовними розподілами застосування такого методу призведе до отримання єдиної моделі, яка буде відповідати певному усередненню усіх потенційних умовних розподілів, що знову-таки призведе до поганої якості прогнозу.

Інтегрована модель авторегресії-рухомого середнього

Загальний вигляд інтегрованої моделі авторегресії-рухомого середнього – IAPC (k, m, d) (англ. autoregressive integrated moving average, ARIMA) [25, 26] визначається наступним рівнянням:

$$\Delta^d y_t = \alpha_0 + \alpha_1 \Delta^d y_{t-1} + \dots + \alpha_k \Delta^d y_{t-k} + \varepsilon_t + \beta_1 \varepsilon_{t-1} + \beta_2 \varepsilon_{t-2} + \dots + \beta_m \varepsilon_{t-m},$$

де $\alpha_1, \dots, \alpha_k, \beta_1, \dots, \beta_m$ – коефіцієнти моделі; k – порядок авторегресії; m – порядок рухомого середнього; d – порядок інтегрованості часового ряду.

Зауважимо, що APC модель може бути перетворена в модель авторегресії (AP) (k):

$$y_t = \alpha_1 y_{t-1} + \dots + \alpha_k y_{t-k} + \xi_t,$$

де помилка ξ_t задовольняє властивості процесу рухомого середнього порядку m – $\xi_t = \varepsilon_t - \beta_1 \varepsilon_{t-1} - \beta_2 \varepsilon_{t-2} - \dots - \beta_m \varepsilon_{t-m}$, або в модель рухомого середнього – РС (m) шляхом заміни змінних y_{t-i} через лінійні комбінації помилок і подальшого приведення подібних членів після розкриття дужок, що дозволяє спростити математичний аналіз ІАРС моделей:

$$\begin{aligned} y_t = & \alpha_1 (\varepsilon_{t-1} - \beta_1 \varepsilon_{t-2} - \dots - \beta_m \varepsilon_{t-m-1}) + \\ & + \alpha_2 (\varepsilon_{t-2} - \beta_1 \varepsilon_{t-3} - \dots - \beta_m \varepsilon_{t-m-2}) + \\ & + \dots + \alpha_k (\varepsilon_{t-k} - \beta_1 \varepsilon_{t-k-1} - \dots - \beta_m \varepsilon_{t-k-m}) + \\ & + \varepsilon_t - \beta_1 \varepsilon_{t-1} - \dots - \beta_m \varepsilon_{t-m}, \end{aligned}$$

Інтегрована складова моделі полягає в тому, що початковий часовий ряд може бути перетворений шляхом застосування оператора кінцевої різниці деякого порядку, у такий спосіб отримуючи різницево–стаціонарний або інтегрований часовий ряд.

Згідно з методологією Бокса–Дженкінса [26], побудова ІАРС моделі для часового ряду полягає в визначенні змінних порядків авторегресії k , рухомого середнього m та інтегрованості d шляхом використання різноманітних тестів та критеріїв з подальшим визначенням відповідних коефіцієнтів шляхом використання алгоритму МНК [22].

Основні переваги та недоліки методу: модель ІАРС дещо схожа з моделлю, що розглядається в методі кореляційно–регресійного аналізу і через те має схожі переваги та недоліки. У порівнянні з моделлю кореляційно–регресійного аналізу, модель ІАРС має менше передумов та дозволяє описувати більш складні процеси, однак при цьому процес налаштування моделі потребує більше обчислень і є більш складним.

Штучні нейронні мережі

Штучні нейронні мережі (ШНМ) [27, 28] являють собою систему з'єднаних і взаємодіючих між собою штучних нейронів [29]. Параметри ШНМ зазвичай налаштовуються згідно з алгоритмом зворотного поширення помилки [30, 31]. ШНМ можуть виявляти складні залежності між вхідними і вихідними даними та виконувати узагальнення. Здатність нейронної мережі до прогнозування впливає з того, що вона має здібності виділення прихованих залежностей між вхідними та вихідними даними та узагальнення [32, 33]. На сьогоднішній день найбільш популярною областю теорії ШНМ є глибинне навчання [34, 35], що досліджує застосування нейронних мереж з великою кількістю шарів для широкого спектру завдань. Одним з перспективних напрямків глибинного навчання в контексті задачі прогнозування часових рядів є застосування рекурентних нейронних мереж [36, 37].

Однією з переваг ШНМ перед іншими методами прогнозування є відсутність потреби у явному виборі моделі прогнозованого процесу, що спрощує прогнозування та прибирає необхідність залучення людини–експерта для вибору моделі. Другою перевагою є їх універсальність: ШНМ успішно застосовуються як для прогнозування рядів [38], так і для розпізнавання зображень [39, 40, 41], кластеризації вибірки даних [42], машинного перекладу [43], розпізнавання мови [44], розробки інтелектуальних агентів [45] та ін..

Основні переваги та недоліки методу: на відміну від методів кореляційно–регресійного аналізу та моделі ІАРС, для штучних нейронних мереж в більшості випадків не існує деяких передумов, які гарантують оптимальність отриманої моделі, що є одним із недоліків цього методу. Іншою проблемою є нестабільність та відсутність гарантій знаходження оптимальних параметрів в процесі навчання мережі, оскільки на даний момент не існує методів навчання нейронних мереж (окрім тривіальних випадків), які

гарантовано знаходять глобальний мінімум функції помилки. Крім того, як налаштування так і застосування нейронної мережі зазвичай потребує значно більшої кількості обчислень, ніж більшість інших методів, розглянутих у цьому розділі. Однак, даному методу притаманна одна важлива перевага, яка часто переважає над недоліками: можливість описувати складні нелінійні залежності, причому в залежності від структури та параметрів мережі вона може описувати дуже широкий та різноманітний спектр функцій без потреби втручання експерту та мануального перебору різних потенційних функцій для знаходження такої, що найкраще відповідає прогнозованому процесу.

У контексті нестационарних часових рядів з декількома потенційними умовними розподілами нейронні мережі здатні за рахунок нелінійної функції активації в нейронах певним чином розділяти простір вхідних векторів на регіони, на яких локальна поведінка мережі буде різною, у такий спосіб реалізуючи щось подібне до комбінації простих локальних моделей, де кожна модель є активною тільки на своїй ділянці простору вхідних векторів. Але при використанні існуючих активаційних функцій локальні функції мережі на різних частинах простору завжди пов'язані з самим способом розбиття вхідного простору на частини, тобто зміна розбиття простору на частини завжди призводить до зміни параметрів локальних моделей для цих ділянок, що є небажаним обмеженням.

Метод групового урахування аргументів

Метод групового урахування аргументів (МГУА) [46] – сімейство індуктивних алгоритмів для математичного моделювання об'єктів, що описуються великою кількістю параметрів. Метод базується на рекурсивному відборі моделей, які комбінуються для отримання складніших моделей. У більшості відомих алгоритмів МГУА для побудови складних моделей застосовуються комбінації простих моделей – так званих базисних функцій. У

якості базисних функцій найчастіше використовують наступні функції: поліноми Колмогорова–Габора [47], кінцеві ряди Фур’є [48], вейвлети [49] та інші.

Основні переваги та недоліки методу: основною перевагою МГУА є так звана властивість самоорганізації прогнозуючої моделі, тобто автоматичний синтез оптимальної в деякому сенсі структури моделі – це робить даний метод дуже привабливим для застосування у комбінації з штучними нейронними мережами, вибір оптимальної структури яких є важливим і складним завданням. Різні алгоритми МГУА мають різні недоліки: наприклад, комбінаторний алгоритм потребує повного перебору великої кількості потенційних структур, що потребує експоненціально великої кількості обчислень в загальному випадку, і робить даний алгоритм фактично непридатним для задач з великої розмірністю вхідного простору; багаторядний алгоритм є значно ефективнішим з точки зору кількості обчислень, але при цьому втрачається оптимальність параметрів фінальної моделі. У контексті нестационарних часових рядів з декількома потенційними умовними розподілами МГУА притаманні ті ж самі властивості, переваги та недоліки, що й ШНМ.

Приховані марківські моделі

При використанні прихованих марківських моделей [50] вважається, що часовий ряд $\vec{y} = [y_1, \dots, y_n]^T$ генерується об’єктом з прихованим станом, множина станів скінченна та фіксована, і поточний стан об’єкту може змінюватися протягом часу. Тоді кожне значення y_t генерується за розподілом $y_t \sim P_{z_t}(y)$, що задається поточним станом z_t . Для побудови прогнозуючої моделі необхідно оцінити приховані стани $z_{1,\dots,n}$ та параметри розподілів $P_1(y), \dots, P_m(y)$. При виконанні марківської властивості [51], згідно якої стан z_t

залежить лише від стану z_{t-1} , тобто $P(z_t | z_{t-1}, \dots, z_1) = P(z_t | z_{t-1})$, та властивості $P(z_t | z_{t-1}) = P(z_{t'} | z_{t'-1}), \forall t, t'$ можна використати алгоритм Баума–Велша [52] для обчислення оптимального вектору станів \vec{z} та параметрів розподілів $P_1(y), \dots, P_m(y)$.

Основні недоліки методу:

- якщо потрібно враховувати потенційну залежність поточного стану t не лише від попереднього $t-1$, а й від станів $t-2, t-3, \dots, t-N$, то кількість параметрів, яку потрібно оцінювати, зростає експоненційно від N ;
- враховується лише потенційна залежність поточного стану від попереднього стану, не враховуються залежність поточного спостереження від попереднього стану та поточного стану від попереднього спостереження.

Основні переваги методу: перевагою методу є теоретична обґрунтованість його застосування для побудови прогнозуючої моделі для нестационарних часових рядів з декількома потенційними умовними розподілами, оскільки сама постановка задачі, рішенням якої є метод, враховує декілька прихованих станів об'єкту, що прогнозується.

У такий спосіб, усі розглянуті існуючі методи побудови прогнозуючих моделей для нестационарних часових рядів при їх застосуванні до рядів з декількома потенційними умовними розподілами мають такі основні недоліки:

- Використання єдиної моделі незалежно від активного умовного розподілу замість використання оптимальної моделі для кожного умовного розподілу негативно впливає на точність прогнозу такої єдиної моделі – цей недолік притаманний таким методам як кореляційно–регресійний аналіз, ІАРС, ШНМ та МГУА.

- Наявність передумов, при невиконанні яких застосування методу не рекомендується – притаманно таким методам як кореляційно–регресійний аналіз, ІАРС, приховані марківські моделі.
- Занадто спрощена модель для опису нестационарних часових рядів з декількома потенційними умовними розподілами, яка не враховує певні потенційні залежності – притаманно прихованим марківським моделям.
- Використання нелінійних моделей в деяких існуючих методах можна розглядати як застосування декількох простіших моделей для різних частин вхідного простору, що дозволяє отримувати точніші моделі для виділеного класу часових рядів. Однак, у моделях, що використовуються в існуючих методах, функція розбиття вхідного простору на частини пов'язана з відповідними локальними моделями – тобто зміна розбиття простору буде призводити до зміни локальних моделей і навпаки, що є небажаним. Даний недолік притаманний таким методам як ШНМ, МГУА.

Цільовою аудиторією більшості відомих існуючих програмних пакетів для побудови прогнозуючих моделей – Matlab, R, SAS та інші – є інженери з досвідом написання програмного коду. Це є певним недоліком, оскільки спеціаліст в області прогнозування часових рядів не обов'язково буде мати досвід написання програмного коду і через це не буде здатним використовувати наведені програмні пакети.

Описані недоліки основних існуючих методів та програмних пакетів для побудови прогнозуючих моделей обґрунтовують необхідність розроблення інформаційної технології для прогнозування часових рядів з декількома потенційними умовними розподілами.

Висновки

- На основі аналізу проблеми прогнозування нестационарних

часових рядів і актуальних практичних задач, що потребують її вирішення, показано актуальність даної проблеми.

- За рахунок виділення класу нестационарних часових рядів з декількома потенційними умовними розподілами, для якого буде розроблятися інформаційна технологія, властивості цього класу можна використовувати при розробці нових методів, моделей та алгоритмів, що дозволить отримати прогноуючі моделі з підвищеною точністю прогнозів.
- Завдяки аналізу існуючих методів, алгоритмів та моделей прогнозування нестационарних часових рядів виділено їх основні недоліки та переваги при застосуванні для прогнозування рядів з обраного класу, для їх подальшого врахування при розробці інформаційної технології.
- Виділення основних недоліків існуючих методів обґрунтовує потребу в розробці нової інформаційної технології.

РОЗДІЛ 2. ПОБУДОВА ПРОГНОЗУЮЧОЇ МОДЕЛІ ДЛЯ КЛАСУ НЕСТАЦІОНАРНИХ ЧАСОВИХ РЯДІВ З ДЕКІЛЬКОМА ПОТЕНЦІЙНИМИ УМОВНИМИ РОЗПОДІЛАМИ

Аналіз переваг та недоліків існуючих методів прогнозування нестационарних часових рядів при їх застосуванні до рядів з класу нестационарних часових рядів з декількома потенційними умовними розподілами разом з аналізом властивостей часових рядів даного класу дозволяє синтезувати загальний підхід побудови прогнозуючої моделі для часових рядів, що належать до виділеного класу, який буде позбавлений основних недоліків та враховуватиме головні властивості відповідних часових рядів.

2.1. Розробка загального підходу побудови прогнозуючої моделі для нестационарних часових рядів з декількома потенційними умовними розподілами

Проаналізувавши властивості класу нестационарних часових рядів з декількома потенційними розподілами можна зробити висновок, що для побудови прогнозуючої моделі потрібно:

- 1) оцінити усі потенційні умовні розподіли $p_i(y_{k+h} | y_k, \dots, y_1), i = 1, \dots, K$;
- 2) маючи на вході вектор послідовних значень часового ряду вигляду $[y_t, \dots, y_{t-k+1}]$ вміти визначати справжній активний розподіл $p_{a(t+h)}(y_{t+h} | y_t, \dots, y_{t-k+1})$, згідно з яким буде згенероване майбутнє значення y_{t+h} .

Володіючи даною інформацією, ідеальна прогнозуюча модель могла б завжди визначати справжній активний розподіл і на його основі обчислювати оптимальне значення прогнозу \hat{y}_{t+h} . Отже, доволі логічною виглядає наступна загальна схема побудови прогнозуючої моделі, що певним чином апроксимує

ідеальну:

1) Налаштування компоненту, що для заданого вектору $[y_t, \dots, y_{t-k+1}]$ визначає номер $a(t+h) \in \{1, \dots, K\}$ відповідного активного розподілу. Фактично, даний компонент повинен певним чином аналізувати отриманий вектор $[y_t, \dots, y_{t-k+1}]$, і, спираючись на властивість нечастої зміни активного розподілу даного класу нестационарних часових рядів, оцінити номер найбільш ймовірного активного розподілу $a(t) = a(t-1) = \dots = a(t-k+1)$, який згенерував цей вектор спостережень $[y_t, \dots, y_{t-k+1}]$. Отриманий номер розподілу повертається як результат $a(t+h)$, знову-таки спираючись на властивість нечастої зміни активного розподілу – тобто припускаючи, що $a(t+h) = a(t) = a(t-1) = \dots = a(t-k+1)$. Назвемо цей компонент компонентом «оцінки активного розподілу». Один із варіантів налаштування компоненту оцінки активного розподілу є вирішення задачі кластеризації, оскільки різні підпоследовності часового ряду, які згенеровані одним і тим же розподілом, повинні бути певним чином схожими один на одного, а підпоследовності, що згенеровані різними розподілами, мають достатньо відрізнятися, тому задачу пошуку найбільш ймовірного активного розподілу можна розглядати як задачу знаходження кластеру векторів $\{[y_{t_i}, \dots, y_{t_i-k+1}]\}$, що є найбільш схожими на даний вектор $[y_t, \dots, y_{t-k+1}]$.

2) Застосувавши компонент оцінки активного розподілу до усіх векторів $\{[y_{t_i}, \dots, y_{t_i-k+1}]\}$ з навчальної вибірки ми можемо доповнити кожен навчальний приклад вигляду $\langle [y_{t_i}, \dots, y_{t_i-k+1}], y_{t_i+h} \rangle$ інформацією про відповідний номер оціненого розподілу $a(t_i+h)$: $\langle [y_{t_i}, \dots, y_{t_i-k+1}], y_{t_i+h}, a(t_i+h) \rangle$. Використовуючи цю додаткову інформацію стає можливо відібрати усі навчальні приклади, що відповідають певному номеру потенційного умовного

розподілу $p_j(y_{k+h} | y_k, \dots, y_1)$: $\{<[y_{t_i}, \dots, y_{t_i-k+1}], y_{t_i+h}, a(t_i + h) >: a(t_i + h) = j\}$, та оцінити прогножуючу модель для цього розподілу, використовуючи тільки ці приклади. Прогножуючу модель, що відповідає певному розподілу $p_j(y_{k+h} | y_k, \dots, y_1)$ назовемо локальною.

У цьому розділі буде розглянуто другий пункт загальної схеми – налаштування локальних моделей, та запропоновано декілька нових методів структурного та параметричного синтезу нейронних мереж, які можна використовувати як локальні прогножуючі моделі. Крім того, буде проаналізовано певні проблеми нейронних мереж з багатьма прихованими шарами та запропоновано новий штучний нейрон, який є придатним для використання в мережах з невеликою кількістю прихованих шарів. Крім того, певні властивості нового нейрону роблять його добрим кандидатом для використання у складі компоненту оцінки активного розподілу або у складі єдиної нейронної мережі, що буде реалізовувати одночасно як компонент оцінки активного розподілу, так і локальні прогножуючі моделі.

2.2. Вдосконалення алгоритму МГУА шляхом використання алгоритмів зворотного поширення помилки та вибіркового виключення

У роботі [53], присвяченій історії розвитку методів глибинного навчання, багаторядний алгоритм МГУА був названий одним із перших відомих методів побудови глибоких нейронних мереж – тобто задовго до появи терміну «глибинне навчання» цей алгоритм дозволяв ефективно навчати глибокі поліноміальні мережі. Дійсно, модель, що будується у ході найбільш типової реалізації багаторядного алгоритму МГУА [54] – коли часткові моделі будуються для усіх можливих пар вхідних змінних, можна представити у вигляді наступної мережі з поліноміальними нейронами [55, 56]:

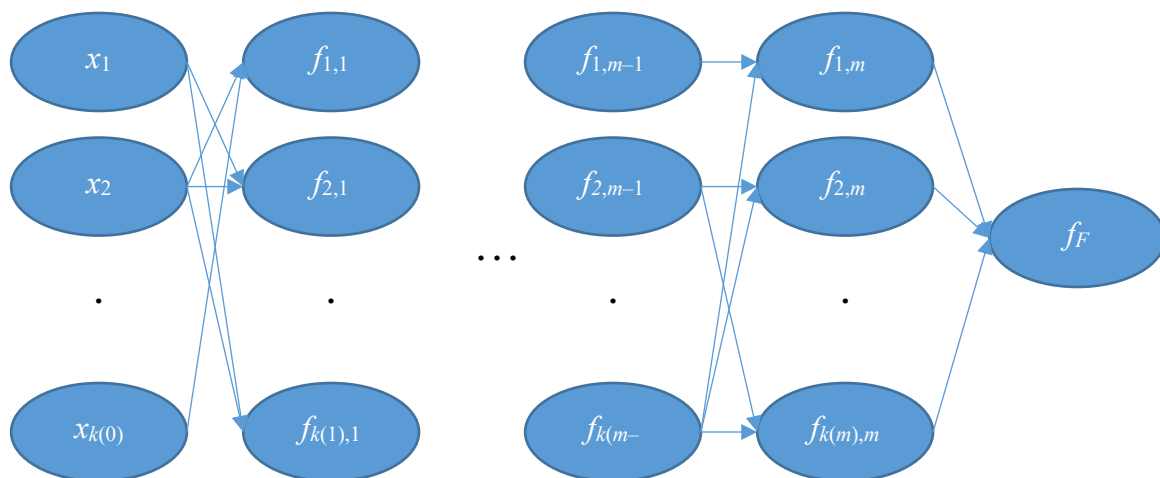


Рисунок 2.1. Поліноміальна мережа

Тут:

- m – кількість прихованих шарів мережі;
- $k(0)$ – кількість вхідних змінних, $k(1), \dots, k(m)$ – кількість поліноміальних нейронів у прихованому шарі з відповідним номером;

- $x_1, \dots, x_{k(0)}$ – вхідні змінні;
- $f_{1,i}, \dots, f_{k(i),i}; i=1, \dots, m$ – поліноміальні нейрони, що реалізують

функцію вигляду $f(x, y) = \sum_{s=0}^2 \sum_{i=0}^s a_{i,s-i} x^i y^{s-i}$, де i – номер шару, у якому

знаходиться відповідний нейрон;

- f_F – вихідний поліноміальний нейрон.

Отже, в процесі виконання багаторядного алгоритму МГУА фактично виконується налаштування структури та параметрів багатошарової поліноміальної нейронної мережі. Крім того, якщо проаналізувати цей алгоритм, то можна помітити певну схожість з етапом жадібного переднавчання [57, 58] в певних алгоритмах навчання глибоких нейронних мереж [59, 60], оскільки параметри усіх нейронів у прихованому шарі з номером i налаштовуються незалежно від параметрів нейронів в інших шарах,

тобто навчання виконується пошарово. Через такий пошаровий характер налаштування фінальне значення усіх параметрів мережі не буде гарантовано найкращим з точки зору критерію, що оптимізується, оскільки для цього потрібно виконати одночасне налаштування усіх параметрів.

Щоб довести неоптимальність параметрів, отриманих при застосуванні багаторядного алгоритму МГУА, візьмемо просту поліноміальну мережу:

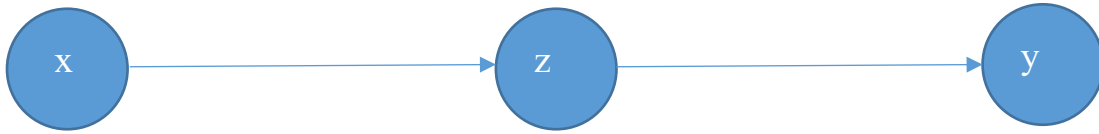


Рисунок 2.2. Двошарова поліноміальна мережа

Тобто маємо 2 часткові моделі: $z = \sum_{i=0}^2 a_i x^i$ та $y = \sum_{i=0}^2 b_i z^i$. Нехай задано навчальну вибірку вигляду:

$$\langle X, y \rangle, X = \begin{bmatrix} x_1 \\ \dots \\ x_n \end{bmatrix}, y = \begin{bmatrix} y_1 \\ \dots \\ y_n \end{bmatrix}.$$

Згідно з багаторівневим алгоритмом МГУА, вектор параметрів $\vec{a} = [a_0, a_1, a_2]^T$ буде знайдений наступним чином, відповідно до методу найменших квадратів [61]: $\vec{a} = (T^T T)^{-1} T^T y$, $T = [t_{i,j}]$, $t_{i,j} = x_i^{j-1}$, $i \in \{1, \dots, n\}$, $j \in \{1, 2, 3\}$. Далі виконується перехід до наступного рівня, де розраховується вектор параметрів \vec{b} . Для цього розраховується вектор $\vec{z} = T\vec{a} = T(T^T T)^{-1} T^T y$, матриця прикладів наступного рівня $Z = [m_{i,j}]$, $m_{i,j} = z_i^{j-1}$, $i \in \{1, \dots, n\}$, $j \in \{1, 2, 3\}$, і нарешті вектор параметрів $\vec{b} = (Z^T Z)^{-1} Z^T y$. У результаті розрахунку векторів \vec{a} та \vec{b} отримуємо фінальний вираз функції, що описується відповідною поліноміальною мережею: $y(x) = \sum_{i=0}^2 b_i (\sum_{j=0}^2 a_j x^j)^i$. Але очевидно, що функція $y(x)$

є поліномом вигляду $y^*(x) = \sum_{i=0}^4 c_i x^i$, оптимальний вектор параметрів якого

можна розрахувати як:

$$\vec{c} = (Q^T Q)^{-1} Q^T y, Q = [q_{i,j}], q_{i,j} = x_i^{j-1}, i \in \{1, \dots, n\}, j \in \{1, 2, 3, 4, 5\}.$$

Порівнявши рівняння для розрахунку векторів \vec{a} , \vec{b} , \vec{c} та виразів для утворення поліномів $y(x)$ і $y^*(x)$ можна зробити висновок, що в загальному випадку поліном $y(x)$, отриманий у результаті застосування багаторядного алгоритму МГУА, не буде рівний оптимальному поліному $y^*(x)$.

Для експериментальної перевірки використаємо наступний поліном:

$$y(x) = \sum_i^4 a_i x^i, \vec{a} = [2, 0.2, -0.8, 0.1, -0.2]^T,$$

і згенеруємо навчальну вибірку. Для цього візьмемо вектор вхідних значень $\vec{x} = [-5, -4.9, -4.8, \dots, 4.9, 5]^T$, і отримаємо вихідні значення шляхом додавання випадкового шуму до відповідного значення поліному:

$$y_i = y(x_i) + N(0, 1).$$

У результаті маємо наступний часовий ряд:

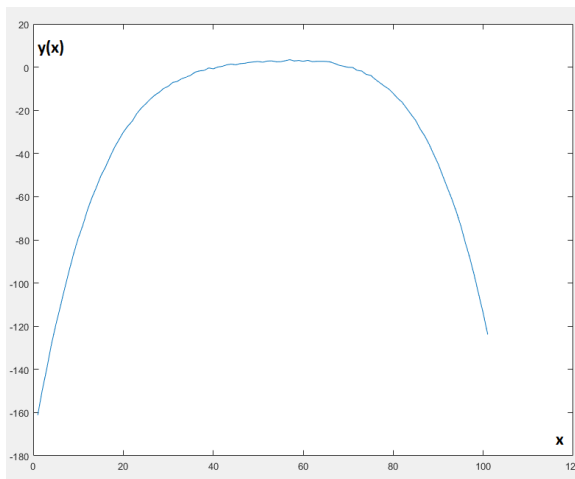


Рисунок 2.3. Отримана навчальна вибірка

Якщо оцінити справжній поліном $\hat{y}(x) = \sum_{i=0}^4 \hat{a}_i x^i$ використовуючи МНК для згенерованої навчальної вибірки, то отримаємо наступний вектор оцінок коефіцієнтів:

$$\hat{a} = [2.0486, 0.1565, -0.8121, 0.1043, -0.1990]^T.$$

Нормована середня квадратична помилка (СКП) моделі з таким вектором параметрів дорівнює $4.86 \cdot 10^{-4}$, сумарна абсолютна відстань оцінок параметрів від справжніх значень дорівнює 0.11.

Тепер навчимо двошарову поліноміальну мережу з квадратичними нейронами згідно багаторядного алгоритму МГУА:



Рисунок 2.4. Двошарова мережа з квадратичними нейронами

Відповідно до алгоритму, спочатку знайдемо оптимальний вектор параметрів моделі $z(x) = \sum_{i=0}^2 b_i x^i$, після чого розрахуємо оптимальний вектор параметрів моделі $y^{(g)}(x) = \sum_{i=0}^2 c_i [z(x)]^i$ (при фіксованих параметрах b_0, b_1, b_2).

Усі параметри розраховуються згідно МНК. У результаті отримаємо наступні вектори параметрів:

$$\vec{b} = [13.13, 1.75, -5.16]^T, \vec{c} = [-1.36, 0.37, -0.007]^T.$$

Нормована СКП моделі $y^{(g)}(x)$ дорівнює 0.0016 – що гірше в 3.5 рази за помилку моделі $\hat{y}(x)$. На основі векторів \vec{a} і \vec{b} моделі $y^{(g)}(x)$ можна розрахувати відповідний вектор параметрів $\vec{a}^{(g)}$ поліному 4 ступеня, який задає ця поліноміальна мережа:

$$\alpha^{(g)} = [2.2743, 0.3201, -0.9467, 0.1296, -0.1908]^T.$$

Отже, сумарна абсолютна відстань оцінок параметрів $\alpha^{(g)}$ від справжніх значень дорівнює 0.598 – що в 5 разів гірше за відповідну відстань оцінок $\hat{\alpha}$.

Нижче наведені графіки прогнозу обох моделей у порівнянні з виходом справжньої моделі:

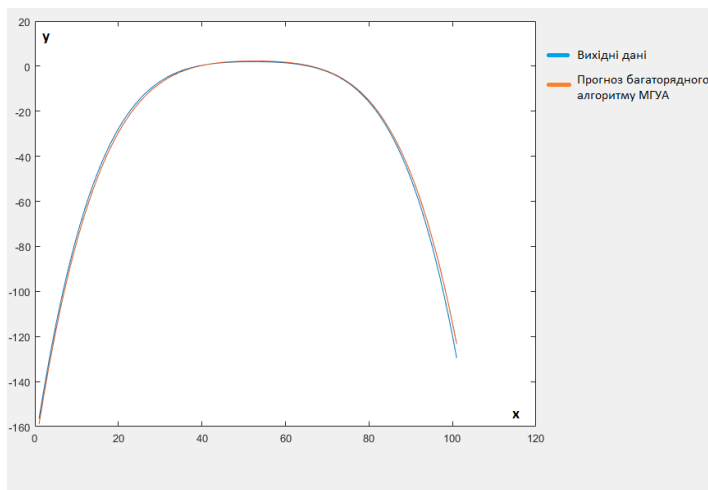


Рисунок 2.5. Графіки справжньої моделі та моделі, отриманої згідно алгоритму МГУА

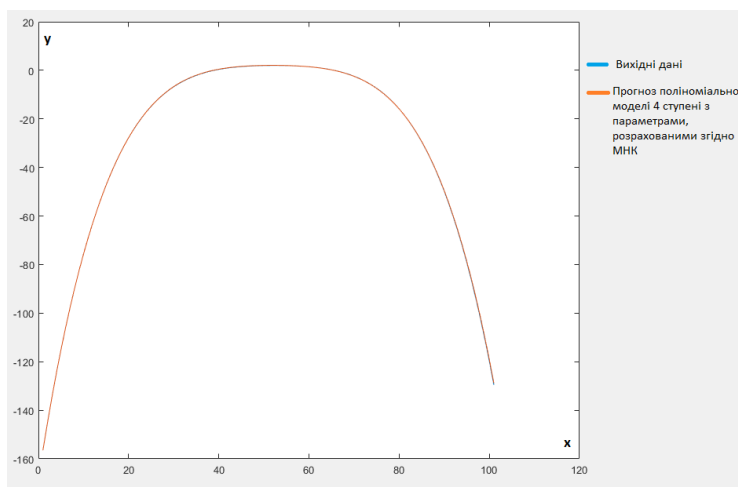


Рисунок 2.6. Графік справжньої моделі та моделі, отриманої у результаті налаштування параметрів поліному 4 ступеня напряму

Отже, на цьому прикладі ми підтвердили, що параметри поліноміальної

мережі, отримані при застосуванні багаторядного алгоритму МГУА, не є оптимальними з точки зору критерію мінімуму середньої квадратичної помилки моделі.

Приведений далі метод намагається певним чином виправити описаний недолік багаторядного алгоритму МГУА. Основна ідея запропонованого методу – додання фази глобального доналаштування [62] параметрів поліноміальної мережі, отриманої у результаті виконання багаторядного алгоритму МГУА – аналогічно з методами глибинного навчання, що складаються з фаз перед– та до–налаштування параметрів мережі. У такий спосіб, цей метод можна розглядати як конкретну реалізацію загального підходу розбиття процесу синтезу прогнозуючої моделі на 2 стадії перед– та до–налаштування, де для перед–налаштування використовується багаторядний алгоритм МГУА, а для до–налаштування – метод зворотного поширення помилки та метод вибіркового виключення [63, 64].

Перевагою використання багаторядного алгоритму МГУА для стадії перед–налаштування є автоматичний синтез структури поліноміальної мережі [65, 66], включаючи: знаходження зайвих вхідних змінних та зв'язків між нейронами, визначення кількості нейронів у прихованих шарах і так далі.

Метод, що пропонується, складається з наступних кроків:

1. Попередня обробка вхідних даних: перехід до різницевого часового ряду, його нормалізація та вкладення для отримання матриці вхідних векторів та вектору вихідних значень, з використанням певної розмірності вкладення k .

2. Стадія перед–налаштування:

- 2.1. Розбиття усієї вибірки на 2 підвибірки: навчальну та валідаційну випадковим чином, де $ts \in \{1, 2, \dots, 99\}$ відсотків усіх прикладів відбираються до навчальної вибірки, а $100 - ts$ відсотків – до валідаційної.

2.2. Будується $C(k,2)$ часткових моделей вигляду:

$$f(x,y) = \sum_{s=0}^2 \sum_{i=0}^s a_{i,s-i} x^i y^{s-i},$$

де як входи x, y кожної часткової моделі використовується унікальна пара поточних вхідних змінних (на першій ітерації це оригінальні вхідні змінні). Наприклад, для першої часткової моделі $x = x_1, y = x_2$, для другої – $x = x_1, y = x_3$ і так далі. Параметри кожної часткової моделі налаштовуються окремо, використовуючи лінійну регресію на навчальній вибірці. Обчислювальна складність побудови кожної часткової моделі – $O(N)$ (при фіксованій кількості змінних), де N – кількість навчальних прикладів. Отже загальна обчислювальна складність цього кроку – $O(C(k,2) * N) = O(k^2 * N)$.

2.3. Для кожної налаштованої часткової моделі розраховується її помилка на валідаційній вибірці:

$$E(f) = \sum_{\langle \vec{x}, y \rangle \in V} (f(\vec{x}) - y)^2.$$

2.4. Відбираються S_l моделей з найкращим значенням помилки на валідаційній вибірці, де l – номер поточного рівня (часто для простоти імплементації обирають $S_l = k, \forall l$). Якщо значення помилки на валідаційній вибірці найкращої моделі поточного рівня l більше за відповідне значення помилки найкращої моделі попереднього рівня $l-1$, то усі моделі поточного рівня відкидаються, і найкраща модель попереднього рівня обирається як фінальна, тобто її вихід буде вважатися виходом повної моделі. Після цього виконання ітерацій припиняється.

2.5. У іншому випадку виходи обраних моделей для кожного вектору із поточної матриці вхідних векторів $X^{(l)}$ використовуються для формування нової матриці вхідних векторів $X^{(l+1)}$:

$$X^{(l+1)} = [m_{i,j}], m_{i,j} = f_i^{(l)}(\bar{x}_j^{(l)}), i \in \{1, \dots, S_l\}, j \in \{1, \dots, n\},$$

де $\bar{x}_i^{(l)}, i = 1, \dots, n$ – стовпець під номером i матриці $X^{(l)}$, $f_j^{(l)}(\bar{x}_i^{(l)}), j = 1, \dots, S_l, i = 1, \dots, n$ – застосування часткової моделі $f_j^{(l)}$ до вектору $\bar{x}_i^{(l)}$, при цьому обираються лише ті входи з $\bar{x}_i^{(l)}$, які потрібні частковій моделі $f_j^{(l)}$. Після формування нової матриці вхідних векторів виконується перехід до наступної ітерації $l + 1$, починаючи з кроку 2.2.

2.6. У такий спосіб, критерій зупинки ітерацій стадії переднавчання у запропонованому методі можна сформулювати наступним чином:

$$\begin{cases} l \geq 2, \\ \min_{f \in F_{l+1}} \{E(f)\} \geq \min_{f \in F_l} \{E(f)\} \end{cases},$$

де l – номер поточної ітерації, F_l – множина усіх часткових моделей поточної ітерації.

2.7. У результаті виконання стадії перед-налаштування отримуємо поліноміальну нейронну мережу, схематично аналогічну мережі, що зображена на рис. 2.1.

3. Стадія до-налаштування, у ході якої параметри отриманої мережі налаштовуються з використанням алгоритму еластичного зворотного поширення помилки `gror` [67] та методу вибіркового виключення:

3.1. Для коректного застосування методу вибіркового виключення потрібно попередньо помножити ваги усіх нейронів на коефіцієнт $1/p$, де $p = 0.2$ для вагів між нейронами першого прихованого шару та шаром вхідних змінних і $p = 0.5$ для усіх інших вагів.

3.2. Виконуються 2 «проходи» для усіх пар вигляду <вхідний вектор, вихідне значення> з вибірки:

- прямий прохід – вектор вхідних значень подається на входи мережі

і послідовно обчислюються виходи кожного нейрону, поки не буде обчислений вихід останнього, вихідного нейрону;

- зворотній прохід – обчислюються похідні функції помилки по усіх параметрах усіх нейронів.

3.3. При цьому згідно з методом вибіркового виключення, при виконанні проходів для кожної пари $\langle \vec{x}, y \rangle$ з навчальної вибірки, де \vec{x} – вхідний вектор, y – відповідне вихідне значення, з ймовірністю p ($p = 0.2$ для фіктивних нейронів, що описують вхідні змінні і $p = 0.5$ для усіх нейронів) виключається кожен нейрон мережі, тобто вважається, що усі вхідні та вихідні ваги такого нейрону рівні 0, а отже, і похідні функції помилки по цим вагам також будуть рівні 0. Таке випадкове виключення нейронів в процесі навчання мережі на практиці дозволяє вирішувати проблему перенавчання і призводить до покращення узагальнюючих властивостей мережі.

3.4. Усі обчислені похідні сумуються по усім прикладам із навчальної вибірки.

3.5. Значення параметрів усіх нейронів моделі оновлюється за наступним правилом:

$$\Delta_{ij}^{(t)} = \begin{cases} \alpha^+ \times \Delta_{ij}^{(t-1)}, & \frac{\partial E^{(t)}}{\partial w_{ij}} \times \frac{\partial E^{(t-1)}}{\partial w_{ij}} \geq 0, \\ \alpha^- \times \Delta_{ij}^{(t-1)}, & \frac{\partial E^{(t)}}{\partial w_{ij}} \times \frac{\partial E^{(t-1)}}{\partial w_{ij}} < 0, \end{cases}$$

$$w_{ij}^{(t)} = w_{ij}^{(t-1)} - \text{sign}\left(\frac{\partial E^{(t)}}{\partial w_{ij}}\right) * \Delta_{ij}^{(t)}.$$

Якщо знак похідної функції помилки по деякому параметру співпадає зі знаком аналогічної похідної на попередній ітерації, то значення корекції для цього параметру буде помножене на деякий фактор $\alpha^+ > 1$; у іншому випадку значення корекції буде помножене на фактор $\alpha^- < 1$. На найпершій ітерації

використовують деяке константне значення корекції для усіх параметрів: $\Delta_{ij}^0 = c, \forall j, \forall i$. Зазвичай, використовують наступні значення гіперпараметрів: $\alpha^+ = 1.2, \alpha^- = 0.5, c = 0.1$.

3.6. Після оновлення усіх параметрів обчислюється помилка мережі на навчальній вибірці: якщо значення помилки менше, ніж на попередній ітерації, то навчання продовжується і виконується перехід до наступної ітерації, починаючи з кроку 3.2, у іншому випадку навчання зупиняється, і усі параметри мережі повертаються до значень, які були на попередній ітерації.

3.7. Після зупинки навчання відповідно до методу вибіркового виключення усі ваги мережі помножуються на відповідне значення коефіцієнту ρ .

4. У результаті отримано звичайну поліноміальну нейронну мережу, а отже, її використання для прогнозування на нових даних виконується як звичайно: вектор нових входних даних \vec{x} подається на вхідний шар мережі, після чого поступово обчислюються виходи усіх нейронів шар за шаром, поки не буде обчислений вихід єдиного нейрону в останньому шарі, який і буде значенням прогнозу мережі.

2.3. Синтез нової моделі нейрону типу Sigmoid Piecewise

Розглянемо наступну мережу, що використовує квадратичні нейрони:

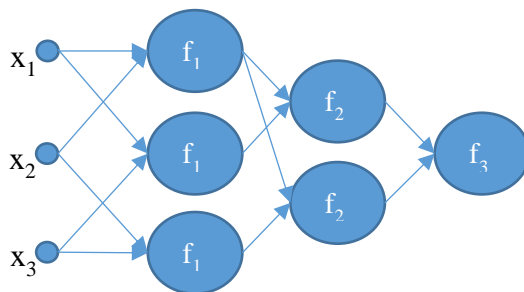


Рисунок 2.7. Поліноміальна мережа з квадратичними нейронами

Така мережа може бути представлена відповідним поліномом 8 ступеня, для повного опису якого потрібно 165 вагових коефіцієнтів, при цьому мережа має 36 вагових коефіцієнтів – власне ця економія у кількості вагових коефіцієнтів, потрібних для опису функції, і вважається однією з переваг багатошарових архітектур.

Однак, як мінімум у випадку поліноміальних мереж, ми можемо записати повну систему рівнянь, що описує як розрахувати кожен із 165 коефіцієнтів повного поліному маючи 36 коефіцієнтів мережі. І тоді стає очевидно, що неможливо описати будь-який вектор із простору \mathbb{R}^{165} маючи лише 36 параметрів – це означає, що ця мережа може описати у кращому випадку лише поліноми, що належать до деякого 36-вимірного підпростору у просторі усіх можливих поліномів 8 ступеня з 3 змінними, тобто реальної економії не досягається.

Насправді, ситуація навіть гірша. Якщо проаналізувати найпростіший випадок багатошарової поліноміальної мережі з квадратичними нейронами вигляду:



Рисунок 2.8. Найпростіший випадок багатошарової поліноміальної мережі з квадратичними нейронам

і скласти систему рівнянь для отримання коефіцієнтів поліному $f(x) = c_0 + c_1x + c_2x^2 + c_3x^3 + c_4x^4$ із коефіцієнтів квадратичних нейронів $f_{11}(x) = a_0 + a_1x + a_2x^2$ та $f_{21}(x) = b_0 + b_1x + b_2x^2$, то отримаємо наступну систему:

$$\begin{cases} c_0 = b_0 + b_1 a_0 + b_2 a_0^2 \\ c_1 = b_1 a_1 + 2a_0 a_1 b_2 \\ c_2 = b_1 a_2 + 2b_2 a_0 a_2 + b_2 a_1^2 \\ c_3 = 2a_1 a_2 b_2 \\ c_4 = b_2 a_2^2 \end{cases}$$

Якщо виконати декілька елементарних перетворень цієї системи, отримаємо наступну рівність: $a_2(4c_4c_2c_3 - c_3^3 - 8c_1c_4^2) = 0$, із якої випливає, що якщо $a_2 \neq 0$, то на коефіцієнти c_1, c_2, c_3, c_4 накладається обмеження, у такий спосіб, ця поліноміальна мережа, що має 6 параметрів, здатна описати лише такі поліноми 4 ступеня, що задовольняють цьому обмеженню, тобто насправді ця мережа здатна описувати лише поліноми що належать до певного 4-вимірного підпростору у 5-вимірному просторі усіх можливих поліномів 4 ступеня від однієї змінної. Отже, насправді багаточарові поліноміальні мережі із n параметрами здатні описувати лише поліноми з деякого підпростору з розмірністю менше за n із усього простору поліномів ступеня, який досягається відповідною мережею. Отже, замість економії ми отримуємо зайві параметри, а знайшовши відповідні обмеження ми б могли налаштовувати меншу кількість параметрів.

Інша проблема багаторівневих архітектур полягає в зайвому ускладненні моделі мережі. Порівняємо моделі поліному 4 ступеня від однієї змінної та мережі з 2 шарами по 1 квадратичному нейрону у кожному:

$$\begin{cases} f_{poly}(x) = c_0 + c_1x + c_2x^2 + c_3x^3 + c_4x^4 \\ f_{net}(x) = b_0 + b_1(a_0 + a_1x + a_2x^2) + b_2(a_0 + a_1x + a_2x^2)^2 \end{cases}$$

Обидві моделі описують поліном 4 ступеня (причому як було показано раніше, модель мережі не здатна описувати усі можливі поліноми 4 ступеня), але модель мережі значно складніша – нелінійна по параметрам, похідні по параметрам будуть залежати від значення інших параметрів, що призводить до ускладнення процесу навчання моделі. Також, при налаштуванні параметрів

мережі і використанні середньої квадратичної помилки як критерію навчання, функція, яку потрібно мінімізувати, очевидно не буде опуклою функцією від параметрів мережі, а мінімізація неопуклих функції в загальному випадку є невіршеною задачею [68].

Отже, багат шарові поліноміальні нейронні мережі мають певні недоліки: вони описують певний поліном деякого ступеня від своїх вхідних змінних, але роблять це неефективним чином – використовуючи надмірну кількість параметрів і занадто ускладнюючи функціональний вигляд поліному, що описується. Однак, одношарові поліноміальні нейронні мережі, хоча й позбавлені цих недоліків, не здатні описувати складні нелінійні процеси – сума будь-якої кількості поліномів деякого ступеня залишається поліномом того ж ступеня. Навіть використовуючи нейрони, для яких доведено теоретично, що при достатній кількості таких нейронів у прихованому шарі мережа зможе описати будь-яку гладку функцію, на практиці потрібно надто велика кількість цих нейронів для апроксимації складних залежностей, і використовуючи декілька шарів нейронів зазвичай вдається досягнути потрібної точності апроксимації використовуючи значно менше нейронів [69, 70, 71]. Пропонується новий штучний нейрон Sigmoid Piecewise (SP), що має складнішу ніж у існуючих нейронів математичну модель, що дозволяє в багатьох обмежитись використанням одношарових мереж:

$$SP(\vec{x}; \vec{w}^{(+)}, \vec{w}^{(-)}, \vec{h}) = \frac{\langle \vec{w}^{(+)}, \vec{x} \rangle}{1 + e^{-k \langle \vec{h}, \vec{x} \rangle}} + \frac{\langle \vec{w}^{(-)}, \vec{x} \rangle}{1 + e^{k \langle \vec{h}, \vec{x} \rangle}}; k > 0.$$

За рахунок використання цієї моделі SP нейрон реалізовує просту схему, що складається з трьох зважених суматорів, два з яких виконують роль локальних моделей, де кожна модель є активною на окремому регіоні вхідного простору, а третій суматор задає гіперплощину, яка розділяє вхідний простір на два регіони. У такій моделі параметри гіперплощини, що розділяє простір

на регіони та параметри локальних моделей, які є активними на відповідних регіонах, є незалежними, що є важливою перевагою над існуючими моделями штучних нейронів та робить SP нейрон більш придатним для побудови прогнозуючих моделей для нестационарних часових рядів з декількома потенційними умовними розподілами. Фактично, ця модель є апроксимацією кусково–лінійної функції, що складається з роздільної гіперплощини \vec{h} , та двох лінійних функцій: $\vec{w}^{(+)} \cdot \vec{x}, \vec{h} \cdot \vec{x} \geq 0$ та $\vec{w}^{(-)} \cdot \vec{x}, \vec{h} \cdot \vec{x} < 0$. При переході між двома регіонами, які задаються роздільною гіперплощиною, замість жорсткого «переключення» між локальними моделями застосовується логістична сигмоїда $\text{sigm}(x; k) = (1 + e^{-kx})^{-1}, k > 0$, що є його гладкою та диференційовною апроксимацією:

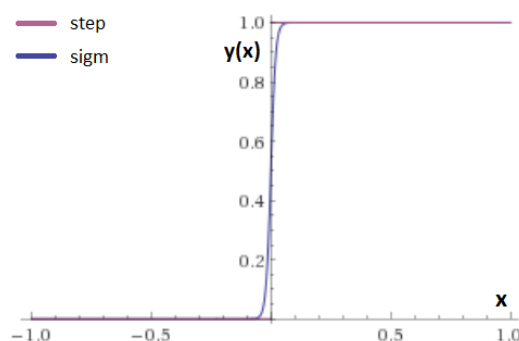


Рисунок 2.9. Порівняння функцій $\text{step}(x)$ та $\text{sigm}(x; k)$ при $k = 100$

Структурно, SP нейрон можна представити так (F – функція активації):

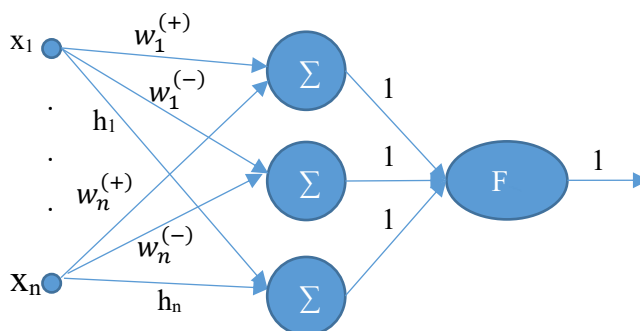


Рисунок 2.10. Структура SP нейрону

Як відомо, класичний математичний нейрон складається з 2 частин: блоку–зваженого суматору $S(\vec{x};\vec{w}) = \langle \vec{w}, \vec{x} \rangle$ та функції активації $F(x; \vec{\theta})$, що описують модель $f(\vec{x}; \vec{w}, \vec{\theta}) = F(S(\vec{x}; \vec{w}); \vec{\theta})$. У випадку SP нейрону структура дещо складніша: маємо 3 блоки суматори – $S^{(h)}(\vec{x}; \vec{h}), S^{(+)}(\vec{x}; \vec{w}^{(+)})$, $S^{(-)}(\vec{x}; \vec{w}^{(-)})$ та функцію активації, що залежить від трьох змінних –

$$F(S^{(h)}, S^{(+)}, S^{(-)}) = \frac{S^{(+)}}{1 + e^{-kS^{(h)}}} + \frac{S^{(-)}}{1 + e^{kS^{(h)}}}, \text{ які разом задають наступну функцію:}$$

$$SP(\vec{x}; \vec{h}, \vec{w}^{(+)}, \vec{w}^{(-)}) = F(S^{(h)}(\vec{x}; \vec{h}), S^{(+)}(\vec{x}; \vec{w}^{(+)}), S^{(-)}(\vec{x}; \vec{w}^{(-)})).$$

Оскільки функція активації $F(S^{(h)}, S^{(+)}, S^{(-)})$ є функцією від 3 змінних (при деякому фіксованому значенні k), у загальному випадку її неможливо представити дво– або тривимірним графіком. Проте, це стає можливим для часткових випадків: наприклад, нехай $S^{(-)} = c \times S^{(+)}$, де c – деяка константа; при $c = 5$ та $k = 2$ отримаємо наступний графік функції активації:

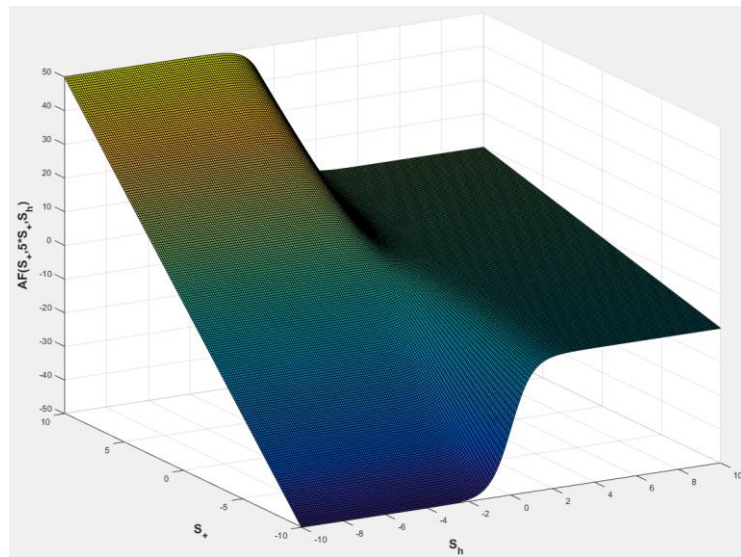


Рисунок 2.11. Графік функції активації при $c = 5$ та $k = 2$

У контексті побудови прогнозуючої моделі для нестационарних часових рядів з декількома потенційними умовними розподілами SP нейрон можна

розглядати як одну з найпростіших прогнозуючих моделей, що складаються з декількох локальних моделей, та моделі-компоненту, що визначає, яку з локальних моделей застосовувати до даного вхідного вектору. Тобто вектор параметрів $\vec{w}^{(+)}$ задає першу локальну модель, що відповідає одному з двох потенційних розподілів, вектор $\vec{w}^{(-)}$ – другу локальну модель, що відповідає другому потенційному розподілу, а вектор \vec{h} – параметри моделі-компоненту, що згідно значень вхідного вектору визначає, який потенційний розподіл є активним, а отже, яку локальну модель використовувати для отримання прогнозу. Комбінуючи декілька таких нейронів вже можна імітувати більше ніж 2 потенційні умовні розподіли.

2.3.1. Налаштування параметрів нейрону Sigmoid Piecewise

Налаштування параметрів SP нейрону можна виконувати застосовуючи певну модифікацію алгоритму градієнтного спуску, використовуючи наступні формули для розрахунку перших похідних:

$$\begin{aligned}
 \frac{\partial f}{\partial w_q^{(+)}} &= \frac{x_q}{1 + e^{-k\langle \vec{h}, \vec{x} \rangle}}, \\
 \frac{\partial f}{\partial w_q^{(-)}} &= \frac{x_q}{1 + e^{k\langle \vec{h}, \vec{x} \rangle}}, \\
 \frac{\partial f}{\partial h_q} &= -\frac{\langle \vec{w}^{(+)}, \vec{x} \rangle}{(1 + e^{-k\langle \vec{h}, \vec{x} \rangle})^2} e^{-k\langle \vec{h}, \vec{x} \rangle} \times (-kx_q) - \frac{\langle \vec{w}^{(-)}, \vec{x} \rangle}{(1 + e^{k\langle \vec{h}, \vec{x} \rangle})^2} e^{k\langle \vec{h}, \vec{x} \rangle} \times (kx_q) = \\
 &= \frac{\langle \vec{w}^{(+)}, \vec{x} \rangle \times (kx_q)}{(1 + e^{-k\langle \vec{h}, \vec{x} \rangle}) \times (1 + e^{-k\langle \vec{h}, \vec{x} \rangle}) \times e^{k\langle \vec{h}, \vec{x} \rangle}} - \frac{\langle \vec{w}^{(-)}, \vec{x} \rangle \times (kx_q)}{(1 + e^{k\langle \vec{h}, \vec{x} \rangle}) \times (1 + e^{k\langle \vec{h}, \vec{x} \rangle}) \times e^{-k\langle \vec{h}, \vec{x} \rangle}} = \\
 &= \frac{\langle \vec{w}^{(+)}, \vec{x} \rangle \times (kx_q)}{(1 + e^{-k\langle \vec{h}, \vec{x} \rangle}) \times (1 + e^{k\langle \vec{h}, \vec{x} \rangle})} - \frac{\langle \vec{w}^{(-)}, \vec{x} \rangle \times (kx_q)}{(1 + e^{k\langle \vec{h}, \vec{x} \rangle}) \times (1 + e^{-k\langle \vec{h}, \vec{x} \rangle})} = \\
 &= \frac{(\vec{w}^{(+)}, \vec{x}) \times (kx_q) - \langle \vec{w}^{(-)}, \vec{x} \rangle \times (kx_q)}{(1 + e^{-k\langle \vec{h}, \vec{x} \rangle}) \times (1 + e^{k\langle \vec{h}, \vec{x} \rangle})} = \frac{(kx_q) \times (\langle \vec{w}^{(+)}, \vec{x} \rangle - \langle \vec{w}^{(-)}, \vec{x} \rangle)}{2 + e^{-k\langle \vec{h}, \vec{x} \rangle} + e^{k\langle \vec{h}, \vec{x} \rangle}}.
 \end{aligned}$$

Графіки залежності $\frac{\partial f}{\partial w_q^{(+)}}$, $\frac{\partial f}{\partial w_q^{(-)}}$ від x_q та $S^{(h)} = \langle \vec{h}, \vec{x} \rangle$ при $k=1$:

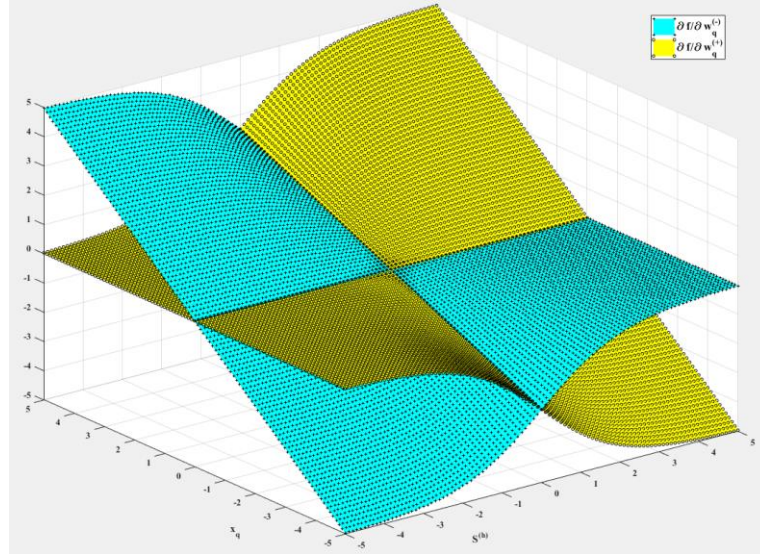


Рисунок 2.12. Залежність $\frac{\partial f}{\partial w_q^{(+)}}$ та $\frac{\partial f}{\partial w_q^{(-)}}$ від x_q та $S^{(h)} = \langle \vec{h}, \vec{x} \rangle$

Графік залежності $\frac{\partial f}{\partial h_q}$ від $t = x_q \times (\langle \vec{w}^{(+)}, \vec{x} \rangle - \langle \vec{w}^{(-)}, \vec{x} \rangle)$ та $S^{(h)}$, при $k=1$:

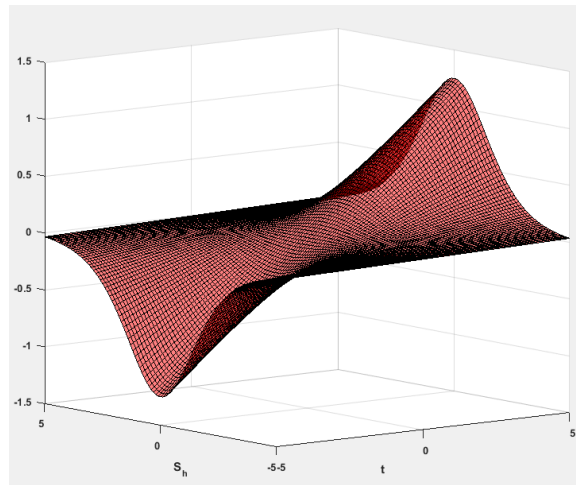


Рисунок 2.13. Залежність $\frac{\partial f}{\partial h_q}$ від $t = x_q \times (\langle \vec{w}^{(+)}, \vec{x} \rangle - \langle \vec{w}^{(-)}, \vec{x} \rangle)$ та $S^{(h)}$

Сума $\frac{\partial f}{\partial w_q^{(+)}} + \frac{\partial f}{\partial w_q^{(-)}}$ може бути спрощена:

$$\begin{aligned} \frac{\partial f}{\partial w_q^{(+)}} + \frac{\partial f}{\partial w_q^{(-)}} &= \frac{x_q}{1 + e^{-k\langle \vec{h}, \vec{x} \rangle}} + \frac{x_q}{1 + e^{k\langle \vec{h}, \vec{x} \rangle}} = \\ &= \frac{x_q \times (1 + e^{k\langle \vec{h}, \vec{x} \rangle}) + x_q \times (1 + e^{-k\langle \vec{h}, \vec{x} \rangle})}{(1 + e^{-k\langle \vec{h}, \vec{x} \rangle}) \times (1 + e^{k\langle \vec{h}, \vec{x} \rangle})} = \frac{x_q \times (2 + e^{k\langle \vec{h}, \vec{x} \rangle} + e^{-k\langle \vec{h}, \vec{x} \rangle})}{2 + e^{k\langle \vec{h}, \vec{x} \rangle} + e^{-k\langle \vec{h}, \vec{x} \rangle}} = x_q. \end{aligned}$$

У результаті аналізу наведених рівнянь та графіків маємо такі висновки:

1. Вихід суматору $S^{(h)} = \langle \vec{h}, \vec{x} \rangle$ має значний вплив на усі похідні. Значення $S^{(h)}$ задає як далеко та по яку сторону від роздільної площини (з вектором нормалі \vec{h}) знаходиться вектор \vec{x} .

2. Сума $\frac{\partial f}{\partial w_q^{(+)}} + \frac{\partial f}{\partial w_q^{(-)}}$ дорівнює x_q при усіх значеннях параметрів.

Однак, відносний вклад кожної з похідних залежить від $S^{(h)}$: при $S^{(h)} \ll 0$ похідна $\frac{\partial f}{\partial w_q^{(-)}}$ буде близька до x_q , а похідна $\frac{\partial f}{\partial w_q^{(+)}}$ до 0, і навпаки, при

$S^{(h)} \gg 0$. Відносний вклад $\frac{\partial f}{\partial w_q^{(-)}}$ та $\frac{\partial f}{\partial w_q^{(+)}}$ буде приблизно рівний при $S^{(h)} \approx 0$.

3. $\frac{\partial f}{\partial h_q} \approx (kx_q) \times (\langle \vec{w}^{(+)}, \vec{x} \rangle - \langle \vec{w}^{(-)}, \vec{x} \rangle)$ при $|S^{(h)}| \approx 0$, та ≈ 0 при

$|S^{(h)}| \gg 0$. Тобто вхідні вектори, які знаходяться далеко від роздільної гіперплощини, будуть в загальному випадку мати менший вплив при налаштуванні вектору \vec{h} в порівнянні з вхідними векторами, що знаходяться ближче до роздільної гіперплощини. Для векторів, що знаходяться близько з роздільною гіперплощиною, значення $\frac{\partial f}{\partial h_q}$ пропорційне різниці

$\langle \vec{w}^{(+)} , \vec{x} \rangle - \langle \vec{w}^{(-)} , \vec{x} \rangle$. У такий спосіб, вектори \vec{x} , для яких $S^{(h)} \approx 0$ та $|\langle \vec{w}^{(+)} , \vec{x} \rangle - \langle \vec{w}^{(-)} , \vec{x} \rangle| \gg 0$ мають найбільший вплив при налаштуванні вектору \vec{h} , і якщо в процесі градієнтного спуску усі вектори із вибірки будуть мати значення $|S^{(h)}| \gg 0$, то усі похідні $\frac{\partial f}{\partial h_q}$ будуть ≈ 0 , і налаштування параметрів

\vec{h} фактично припиниться. Така ситуація може виникнути в двох випадках:

1) усі вектори із вибірки знаходяться по один бік від роздільної гіперплощини – у цьому випадку SP нейрон практично реалізує лінійний нейрон, і навчатися буде або тільки вектор $\vec{w}^{(+)}$ або тільки вектор $\vec{w}^{(-)}$;

2) вектори з вибірки наявні з обох боків роздільної гіперплощини з достатньо великою мінімальною відстанню від векторів з кожного боку до гіперплощини: тобто у вибірці існує 2 чи більше лінійно роздільні кластери і поточна гіперплощина \vec{h} розділяє їх.

4. Якщо на певному кроці алгоритму градієнтного спуску отримано однакові вектори $\vec{w}^{(+)}$ та $\vec{w}^{(-)}$ – оновлення вектору \vec{h} на наступному кроці буде рівним 0. Але ймовірність виконання цієї умови, тим більше протягом декількох ітерацій, є доволі малою, оскільки в більшості випадків похідні $\frac{\partial f}{\partial w^{(+)}}$ та $\frac{\partial f}{\partial w^{(-)}}$ будуть різними на наступному кроці, що призведе до порушення умови $\vec{w}^{(+)} = \vec{w}^{(-)}$.

При налаштуванні багатошарових ШНМ, що застосовують SP нейрони у прихованих шарах, згідно з алгоритмом зворотного поширення помилки, потрібно розраховувати перші похідні моделі нейрону по змінним x_q :

$$\frac{\partial f}{\partial x_q} = \frac{kh_q(\langle \vec{w}^{(+)} , \vec{x} \rangle - \langle \vec{w}^{(-)} , \vec{x} \rangle)}{2 + e^{k\langle \vec{h} , \vec{x} \rangle} + e^{-k\langle \vec{h} , \vec{x} \rangle}} + \frac{w_q^{(+)}}{1 + e^{-k\langle \vec{h} , \vec{x} \rangle}} + \frac{w_q^{(-)}}{1 + e^{k\langle \vec{h} , \vec{x} \rangle}} =$$

$$= \frac{\partial f}{\partial h_q} \frac{h_q}{x_q} + \frac{w_q^{(+)}}{1 + e^{-k \langle \vec{h}, \vec{x} \rangle}} + \frac{w_q^{(-)}}{1 + e^{k \langle \vec{h}, \vec{x} \rangle}}.$$

З останнього рівняння можна зробити висновок, що ця похідна буде ненульовою завжди, коли виконуються умови $w_q^{(+)} \neq 0$ та $w_q^{(-)} \neq 0$, що є корисною властивістю при налаштуванні глибоких ШНМ, оскільки вона зменшує ймовірність виникнення проблеми згасаючих градієнтів.

2.3.2. Порівняння з ReLU-подібними моделями нейронів

У роботі [72] розглянуто нейрон з моделлю вигляду $\text{ReLU}(\vec{x}; \vec{w}) = \max(0, \vec{w}^T \vec{x})$, функція активації якого виглядає наступним чином:

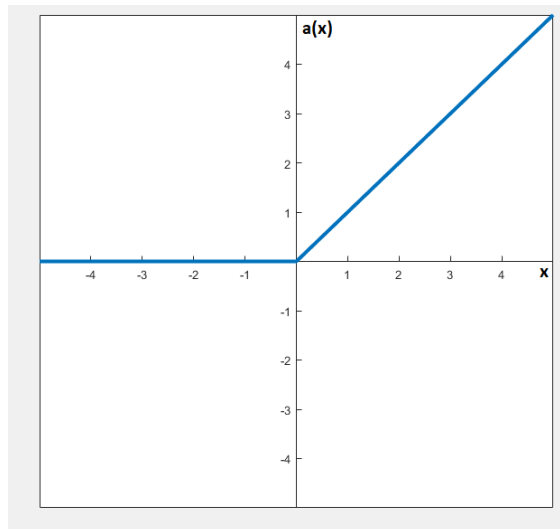


Рисунок 2.14. Графік функції активації ReLU

До родини ReLU-подібних нейронів можна також віднести наступні моделі нейронів:

- Leaky ReLU [73]:

$$\text{LReLU}(\vec{x}; \vec{w}, c) = \begin{cases} \vec{w}^T \vec{x}, & \vec{w}^T \vec{x} > 0 \\ c * \vec{w}^T \vec{x}, & \vec{w}^T \vec{x} \leq 0 \end{cases},$$

де c — константа, зазвичай у діапазоні $(0, 0.5)$;

- Parametric ReLU (PReLU) [39] – має ту ж саму модель, що і Leaky ReLU, але параметр c налаштовується в процесі навчання;
- Maxout [74]:

$$\text{mout}(\vec{x}; \vec{w}^{(1)}, \vec{w}^{(2)}) = \max(\langle \vec{w}^{(1)}, \vec{x} \rangle, \langle \vec{w}^{(2)}, \vec{x} \rangle).$$

Усі ці моделі є частковими випадками більш загальної PL моделі:

$$\text{PL}(\vec{x}; \vec{w}^{(+)}, \vec{w}^{(-)}, \vec{h}) = \begin{cases} \langle \vec{w}^{(+)}, \vec{x} \rangle, \langle \vec{h}, \vec{x} \rangle \geq 0 \\ \langle \vec{w}^{(-)}, \vec{x} \rangle, \langle \vec{h}, \vec{x} \rangle < 0 \end{cases}, \vec{w}^{(+)}, \vec{w}^{(-)}, \vec{h} \in R^n,$$

оскільки їх можна отримати з PL моделі наступним чином:

- $\text{ReLU}(\vec{x}; \vec{w}) = \text{PL}(\vec{x}; \vec{w}^{(+)} = \vec{w}, \vec{w}^{(-)} = 0, \vec{h} = \vec{w})$;
- $\text{LReLU}(\vec{x}; \vec{w}, c) = \text{PReLU}(\vec{x}; \vec{w}, c) = \text{PL}(\vec{x}; \vec{w}^{(+)} = \vec{w}, \vec{w}^{(-)} = c \times \vec{w}, \vec{h} = \vec{w})$;
- $\text{mout}(\vec{x}; \vec{w}^{(1)}, \vec{w}^{(2)}) = \text{PL}(\vec{x}; \vec{w}^{(+)} = \vec{w}^{(1)}, \vec{w}^{(-)} = \vec{w}^{(2)}, \vec{h} = \vec{w}^{(1)} - \vec{w}^{(2)})$.

Оскільки PL модель можна розглядати як частковий випадок моделі SP нейрону (при $k \rightarrow \infty$ SP та PL моделі стають рівними), то моделі ReLU, PReLU, LReLU та mout також є частковими випадками моделі SP нейрону.

2.3.3. Дослідження ефективності нейронів типу Sigmoid Piecewise на реальних вибірках

Порівняємо SP та ReLU нейрони на реальних вибірках згідно наступної процедури:

1. Задається певний рівень помилки $MSE^{(target)} > 0$, якого потрібно досягнути.
2. На першій ітерації тесту навчається два персептрони з одним прихованим шаром з одним нейроном: мережа з ReLU нейроном та мережа з SP нейроном.
3. Якщо помилка обох мереж $< MSE^{(target)}$, то виконання тесту закінчується.

4. На ітерації i маємо i нейронів у прихованому шарі мережі.
5. На ітерації $i+1$ додається один нейрон відповідного типу до прихованого шару кожного персептрону, потім проводиться налаштування обох мереж з фіксацією усіх вхідних та вихідних параметрів попередньо доданих нейронів.
6. Для кожної мережі розраховується і запам'ятовується поточна помилка $MSE^{(i+1)}$ – тобто помилка відповідної мережі, досягнута при використанні $i+1$ нейронів.
7. Якщо на ітерації $i+1$ помилка обох мереж стала $< MSE^{(target)}$ – виконання тесту зупиняється.

Перший тест виконано на вибірці денних значень курсу євро до долару:

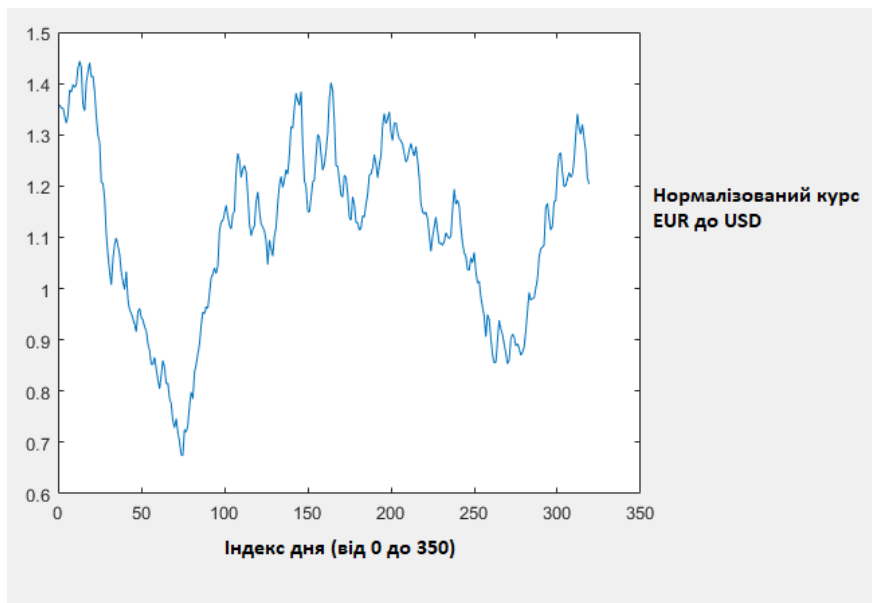


Рисунок 2.15. Вибірка нормалізованих денних значень курсу валют

Навчальна вибірка $\langle X, y : X \rightarrow R \rangle$ була отримана з часового ряду шляхом застосування методу вкладення часових рядів з наступними параметрами: розмірність вкладення $m=5$; горизонт прогнозування $k=4$.

Тобто на основі послідовних значень часового ряду $g_{i-4}, g_{i-3}, \dots, g_i$

потрібно отримати прогноз значення g_{i+4} . СКП наївної моделі $\hat{g}_{i+4} = g_i$ для цієї вибірки дорівнює $MSE^{(naive)}=0.0716$. Задамо цільову СКП $MSE^{(target)}=0.05$ та виконаємо попередньо описану процедуру порівняльного тесту, після чого побудуємо графіки залежності кількості нейронів від досягнутої СКП мережі при використанні цієї кількості нейронів. Як видно з рис. 2.16., для досягнення СКП, меншої за $MSE^{(threshold)} \approx 0.0065$, було необхідно використати в 3 рази більше ReLU нейронів ніж SP нейронів, причому чим більше різниця $MSE^{(threshold)} - MSE, MSE < MSE^{(threshold)}$, тим більше відношення необхідної кількості ReLU нейронів до необхідної кількості SP нейронів, що свідчить про більшу ефективність SP нейрону з точки зору критерію «кількості параметрів, потрібних для досягнення заданого рівня помилки», якщо рівень помилки достатньо малий.

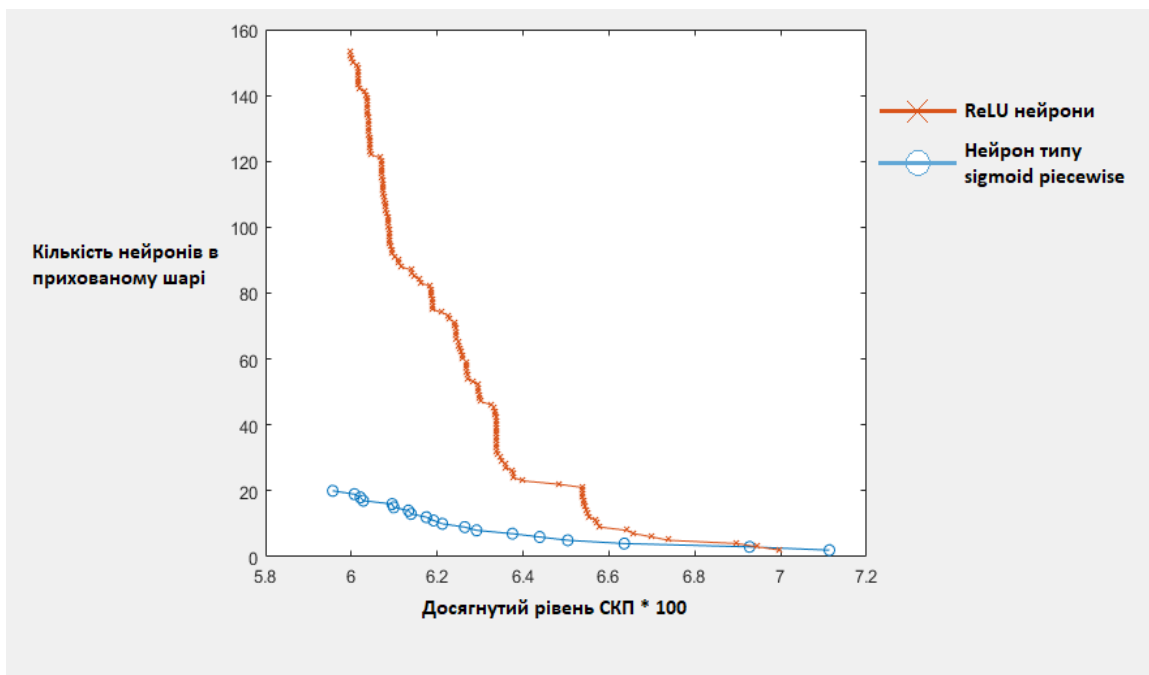


Рисунок 2.16. Залежність кількості нейронів від СКП мережі

Другий тест проводився на часовому ряді процентних ставок за державними облігаціями в Австралії, 1965–1994 роки:

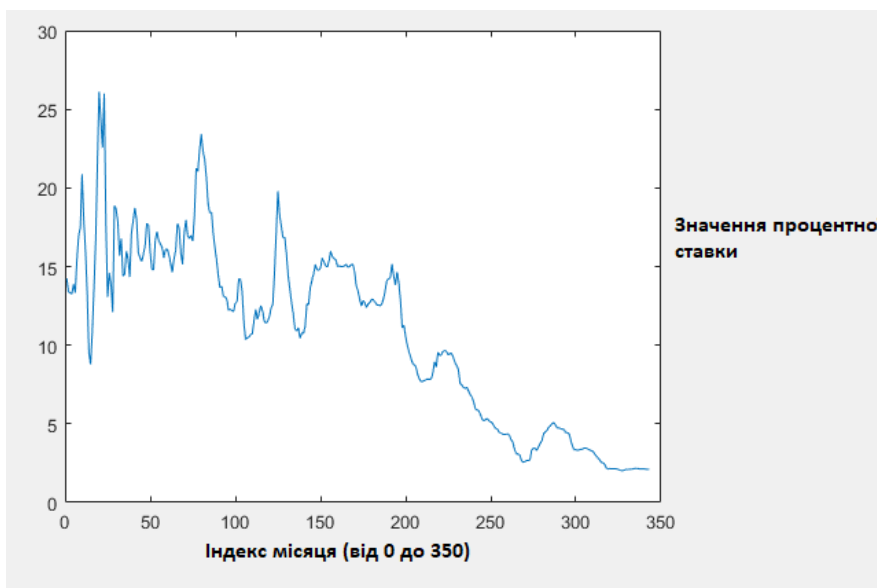


Рисунок 2.17. Графік вибірки, що використовувалися для другого тесту

Після виконання аналогічної процедури отримано СКП наївної моделі $MSE^{(naïve)}=2.915$; при значенні цільової СКП $MSE^{(target)}=1.5$ отримано наступні графіки:

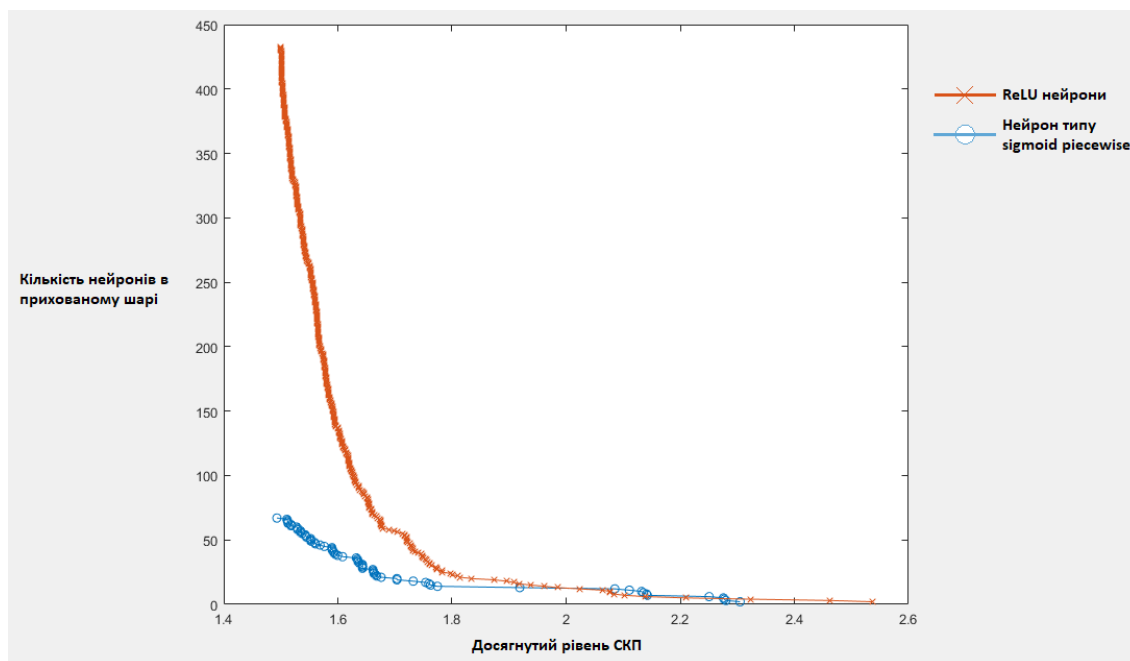


Рисунок 2.18. Залежність кількості нейронів від досягнутої СКП у другому тесті

Для досягнення помилки, меншої за $MSE^{(threshold)} \approx 1.6$, було необхідно використати в 3 рази більше ReLU нейронів ніж SP нейронів, після чого відношення необхідної кількості ReLU нейронів до необхідної кількості SP нейронів продовжувало збільшуватися, як і в першому тесті.

Також було виконано масштабне тестування мереж з SP нейронами на 405 різних часових рядах, взятих зі змагання МЗ. У порівнянні брали участь мережа з ReLU нейронами з 12 нейронами в єдиному прихованому шарі та мережа з SP нейронами з 4 нейронами в єдиному прихованому шарі (тобто кількість параметрів для налаштування в обох мережах була однаковою). У якості вхідних даних для отримання прогнозу використовувалися 4 останні значення часового ряду. Результати тестування наведені у наступних таблицях:

Таблиця 2.1.

**Результати тестування мереж на 405 часових рядах зі змагання МЗ,
прогнозування на 5 кроків вперед**

Тип мережі	Мережа з ReLU нейронами	Мережа з SP нейронами
Сумарна нормалізована СКП на усіх тестових вибірках	121.0895	108.2995

Таблиця 2.2.

**Результати тестування мереж на 405 часових рядах зі змагання МЗ,
прогнозування на 4 кроки вперед**

Тип мережі	Мережа з ReLU нейронами	Мережа з SP нейронами
Сумарна нормалізована СКП на усіх тестових вибірках	110.2880	99.0999

2.4. Використання SP нейронів у складі штучних нейронних мереж

Проаналізуємо поведінку нейронної мережі з одним прихованим шаром, що складається з кусково–лінійних PL нейронів. Оскільки SP нейрон є гладкою

апроксимацією нейрону *piecewise linear*, то результати цього аналізу будуть в деякій мірі поширюватися і на відповідні нейронні мережі з SP нейронами.

Отже, нехай маємо мережу з n входами, L_1 нейронами *piecewise linear* у прихованому шарі, та одним лінійним вихідним нейроном:

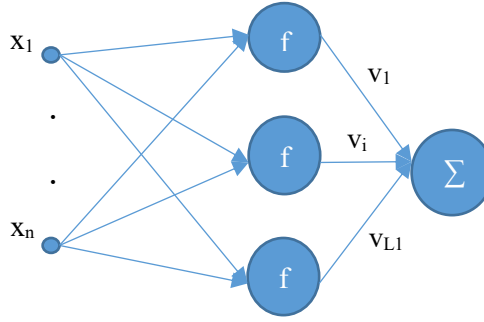


Рисунок 2.19. Вихідна мережа

Розглянемо вектори \vec{h}_i кожного нейрону у прихованому шарі, і введемо

матрицю $H = \begin{bmatrix} \vec{h}_1^T \\ \vdots \\ \vec{h}_{L_1}^T \end{bmatrix}$. Фактично, кожен такий вектор задає гіперплощину у

просторі R^n , що розділяє увесь простір на 2 регіони, і на кожному регіоні відповідний нейрон може застосувати окрему лінійну трансформацію до векторів з цього регіону. Для спрощення подальшого аналізу введемо функцію

$$b(\vec{x}, H) = \text{step}(H\vec{x}), \text{ де } \text{step}(\vec{v}) = \begin{bmatrix} \text{step}(v_1) \\ \vdots \\ \text{step}(v_d) \end{bmatrix}, \text{ step}(x) = \begin{cases} 1, x \geq 0 \\ 0, x < 0 \end{cases}, \text{ яка будь-якому}$$

вектору входів \vec{x} ставить у відповідність двійкове число з L_1 розрядів, де розряд i дорівнює 1, якщо $\vec{h}_i^T \vec{x} \geq 0$ та 0 навпаки. У такий спосіб, стає очевидним, що в найкращому випадку шар з L_1 *piecewise linear* нейронів може розділити увесь вхідний простір на 2^{L_1} регіонів, і в залежності від регіону, до

якого належить \vec{x} , до нього буде застосовуватися різна лінійна трансформація (за умови $\vec{w}_i^{(+)} \neq \vec{w}_i^{(-)}, \forall i$). При цьому кожному регіону можна однозначно поставити у відповідність двійкове число \vec{b} .

Однак, якщо деякий вектор \vec{h}_i можна виразити як лінійну комбінацію інших векторів $\vec{h}_i = \sum_{i \neq i'} c_i \vec{h}_{i'}$, то виникають регіони, на яких значення $\vec{h}_i^T \vec{x}$ буде завжди ≥ 0 або < 0 . У нотації двійкових чисел це означає, що якщо серед L_1 нейронів є нейрони з лінійно залежними векторами \vec{h}_i , то з'являється множина двійкових чисел, що відповідають певним регіонам, які неможливо отримати як результат функції $b(\vec{x}, H), \forall \vec{x}$, а отже, увесь вхідний простір буде розбито на $N < 2^{L_1}$ регіонів. Наприклад, якщо кожен вектор \vec{h}_i є множитком деякого одного вектору – $\vec{h}_i = c_i \vec{v}$, то замість 2^{L_1} регіонів з різними лінійними трансформаціями буде отримано лише 2 регіони.

Отже, з точки зору складності кінцевої функції, що задається такою мережею, найкращим варіантом буде мати прихований шар з L_1 нейронами *piecewise linear*, і при цьому вектори $\vec{h}_1, \dots, \vec{h}_{L_1}$ повинні бути лінійно незалежними – у такому випадку кількість регіонів з різними лінійними трансформаціями буде дійсно рівна 2^{L_1} . Однак, згідно з лемою Стейніца [75] у просторі R^n неможливо знайти більше, ніж n лінійно незалежних векторів – тобто якщо кількість нейронів у прихованому шарі $L_1 > n$ – то незалежно від значень векторів, деякі з них завжди будуть лінійно залежними від інших. Наприклад, у просторі R^3 вектори $\vec{h}_1 = [1, 0, 0]^T, \vec{h}_2 = [0, 1, 0]^T, \vec{h}_3 = [0, 0, 1]^T$ задають базис цього простору – тобто будь-який вектор з R^3 може бути виражений як комбінація цих 3 векторів; і якщо обрати 4 вектор $\vec{h}_4 = [1, 1, 1]$, то він буде

лінійно залежним від цих трьох через рівність $\vec{h}_4 = \vec{h}_1 + \vec{h}_2 + \vec{h}_3$, і замість $2^4 = 16$ регіонів буде отримано 14 регіонів, оскільки $\vec{h}_1 \geq 0 \wedge \vec{h}_2 \geq 0 \wedge \vec{h}_3 \geq 0 \rightarrow \vec{h}_4 \geq 0$ та $\vec{h}_1 < 0 \wedge \vec{h}_2 < 0 \wedge \vec{h}_3 < 0 \rightarrow \vec{h}_4 < 0$:

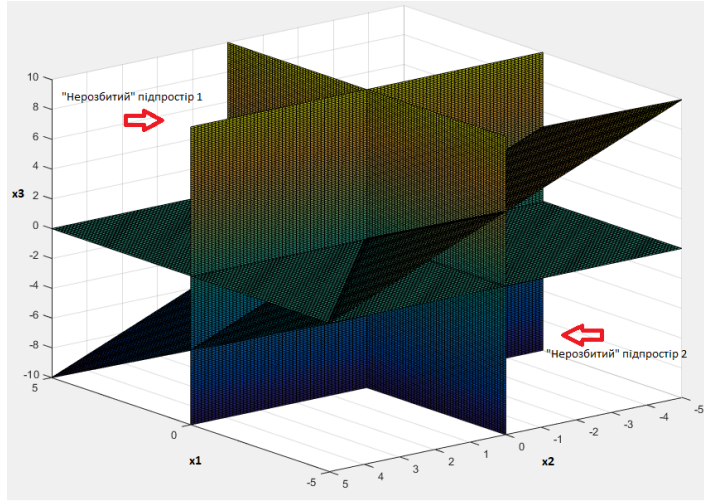


Рисунок 2.20. Візуалізація проблеми втрати регіонів

Як видно з рис. 2.20, при додаванні 4 нейрону з вектором \vec{h}_4 не усі попередньо утворені регіони було розбито на 2 нових регіони, і тому було утворено 14, а не 16 регіонів. Чим більше різниця $L_1 - n$ (за умови $L_1 > n$), тим більше регіонів втрачається по відношенню до найкращого можливого результату у 2^{L_1} підпросторі, і згідно з теоремами Заславського [76] максимально можлива кількість регіонів, отриманих у результаті використання L_1 гіперплощин у n -вимірному просторі дорівнює $O(C_n^{L_1}) = O\left(\frac{L_1!}{(L_1 - n)!n!}\right) \approx O(L_1^n)$ – тобто при $L_1 \gg n$ замість експоненціальної залежності кількості регіонів від кількості нейронів буде отримано поліноміальну залежність.

У такий спосіб, остання кількість нейронів у прихованому шарі, при якій не відбувається втрат з точки зору складності функції, що задається мережею, дорівнює n , і відповідна їй кількість підпросторів – 2^n . Очевидно, що при

малих значеннях n відповідна функція, що задається 2^n регіонами з різними лінійними трансформаціями не буде дуже складною, і в залежності від задачі може знадобитися більш складна функція.

Розглянемо другу складову PL нейрону, а саме вектори $\vec{w}^{(+)}, \vec{w}^{(-)}$. Якщо ми зафіксуємо певний підпростір, якому відповідає двійкове число \vec{b} , то до усіх векторів \vec{x} з цього регіону буде застосовуватися лінійна трансформація

$$W_{\vec{b}} = \begin{bmatrix} \vec{w}_{\vec{b},1} \\ \vdots \\ \vec{w}_{\vec{b},L_1} \end{bmatrix}, \text{ де } \vec{w}_{\vec{b},i} = \begin{cases} \vec{w}_i^{(+)}, b_i = 1 \\ \vec{w}_i^{(-)}, b_i = 0 \end{cases}. \text{ Отже, до певного опуклого конусу, що}$$

$$\text{задається системою нерівностей } H_{\vec{b}} \vec{x} > 0, H_{\vec{b}} = \begin{bmatrix} \vec{h}_{\vec{b},1} \\ \vdots \\ \vec{h}_{\vec{b},L_1} \end{bmatrix}, \vec{h}_{\vec{b},i} = \begin{cases} \vec{h}_i, b_i = 1 \\ -\vec{h}_i, b_i = 0 \end{cases}, \text{ буде}$$

застосовуватися лінійна трансформація $W_{\vec{b}}$. Як відомо, лінійна трансформація від опуклого конусу залишається опуклим конусом у новому просторі [77], тобто кожний опуклий конус, що задається системою нерівностей $H_{\vec{b}} \vec{x} > 0$, буде відображено у деякий опуклий конус $W_{\vec{b}}(H_{\vec{b}}) \vec{y} > 0$ у просторі векторів, утворених з виходів нейронів першого прихованого шару. При цьому відповідні трансформовані конуси у новому просторі можуть перетинатися (і навіть містити один одного), хоча усі початкові конуси не перетиналися. Наприклад, нехай маємо мережу з кількістю входів $n = 2$, і 2 нейрони *piecewise linear* з наступними параметрами:

$$\vec{h}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \vec{h}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \vec{w}_1^{(+)} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \vec{w}_1^{(-)} = \begin{bmatrix} -1 \\ 0 \end{bmatrix}, \vec{w}_2^{(+)} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \vec{w}_2^{(-)} = \begin{bmatrix} 0 \\ -1 \end{bmatrix}.$$

У такому випадку маємо наступні опуклі конуси у просторі входів:

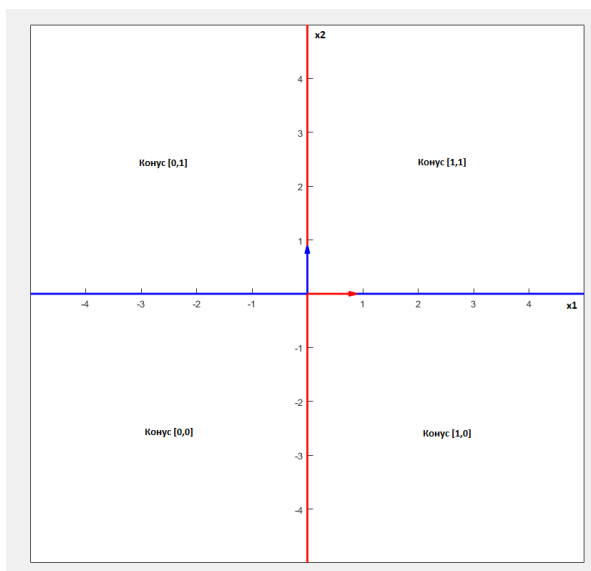


Рисунок 2.21. Візуалізація утворених опуклих конусів з різними лінійними трансформаціями

При цьому згідно з векторами $\vec{w}_{1,+}$, $\vec{w}_{1,-}$, $\vec{w}_{2,+}$, $\vec{w}_{2,-}$ усі 4 опуклі конуси у просторі входів буде перетворено в один і той же конус у просторі виходів першого прихованого шару, що задається системою нерівностей

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} > 0:$$

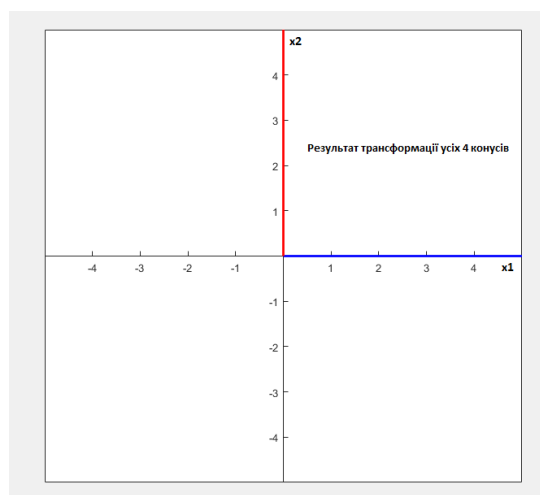


Рисунок 2.22. Візуалізація результату трансформації усіх 4 опуклих конусів у просторі входів

Додамо ще один прихований шар з одним нейроном piecewise linear:

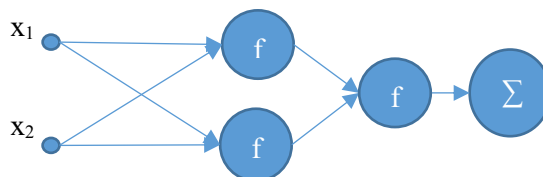


Рисунок 2.23. Структура мережі з прикладу з 2 прихованими шарами

Задамо наступний вектор–нормаль до роздільної площини для цього

нейрону: $\vec{h}_{2,1} = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$. Тоді він буде задавати наступну роздільну гіперповерхню:

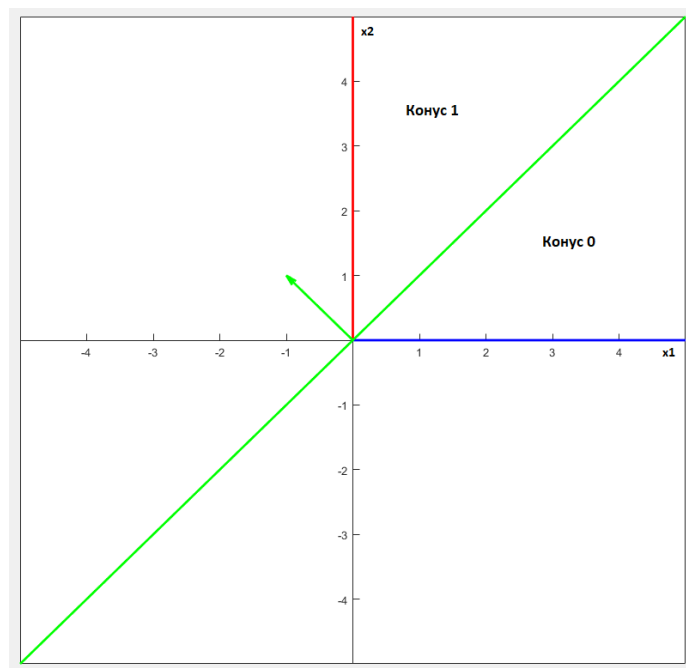


Рисунок 2.24. Конуси у просторі виходів нейронів 1 прихованого шару, утворені роздільною поверхнею нейрону 2 прихованого шару та відображеннями конусів у просторі вхідних векторів, що задаються нейронами 1 прихованого шару у простір виходів нейронів 1 прихованого шару

Оскільки через конфігурацію нейронів першого прихованого шару весь простір R^2 вхідних векторів відображається в перший квадрант простору

виходів нейронів першого прихованого шару, то нейрон у другому прихованому шарі з таким вектором $\vec{h}_{2,1}$ буде розділяти усі конуси простору вхідних векторів, і така мережа буде задавати функцію з $2^3=8$ конусів з різними лінійними трансформаціями, які будуть застосовуватися до вхідних векторів \vec{x} , що належать до цих конусів. Тобто хоча розмірність простору вхідів $n=2$, і при додаванні 3-го нейрону у 1 прихованому шарі було б отримано 6 конусів – додавання ще одного прихованого шару, а переміщення 3-го нейрону до цього шару дозволило отримати 8 конусів, тобто зберегти експоненціальну залежність кількості конусів від кількості нейронів. Якщо позначити вектори цього нейрону як $\vec{w}^{(+)}, \vec{w}^{(-)}$, то ця проста мережа буде задавати наступні 8 конусів, що задаються системою нерівностей вигляду $H\vec{x} > 0$, з відповідними їх лінійними трансформаціями, що задаються матрицями $W \in R^{1 \times 2}$:

$$H_1 = \begin{bmatrix} -1 & 1 \\ 1 & 0 \end{bmatrix}, W_1 = \vec{w}_{2,1,+}^T \times \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \vec{w}_{2,1,+}^T,$$

$$H_2 = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}, W_2 = \vec{w}_{2,1,-}^T \times \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \vec{w}_{2,1,-}^T,$$

$$H_3 = \begin{bmatrix} 0 & -1 \\ 1 & 1 \end{bmatrix}, W_3 = \vec{w}_{2,1,-}^T \times \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} = [w_{2,1,-,1}, -w_{2,1,-,2}]^T,$$

$$H_4 = \begin{bmatrix} 1 & 0 \\ -1 & -1 \end{bmatrix}, W_3 = \vec{w}_{2,1,+}^T \times \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} = [w_{2,1,+,1}, -w_{2,1,+,2}]^T,$$

$$H_5 = \begin{bmatrix} -1 & 0 \\ 1 & -1 \end{bmatrix}, W_5 = \vec{w}_{2,1,+}^T \times \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} = -\vec{w}_{2,1,+}^T,$$

$$H_6 = \begin{bmatrix} 0 & -1 \\ -1 & 1 \end{bmatrix}, W_6 = \vec{w}_{2,1,-}^T \times \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} = -\vec{w}_{2,1,-}^T,$$

$$H_7 = \begin{bmatrix} 0 & 1 \\ -1 & -1 \end{bmatrix}, W_7 = \vec{w}_{2,1,-}^T \times \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} = [-w_{2,1,-,1}, w_{2,1,-,2}]^T,$$

$$H_8 = \begin{bmatrix} -1 & 0 \\ 1 & 1 \end{bmatrix}, W_8 = \vec{w}_{2,1,+}^T \times \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} = [-w_{2,1,+,1}, w_{2,1,+,2}]^T.$$

Або у вигляді графіку:

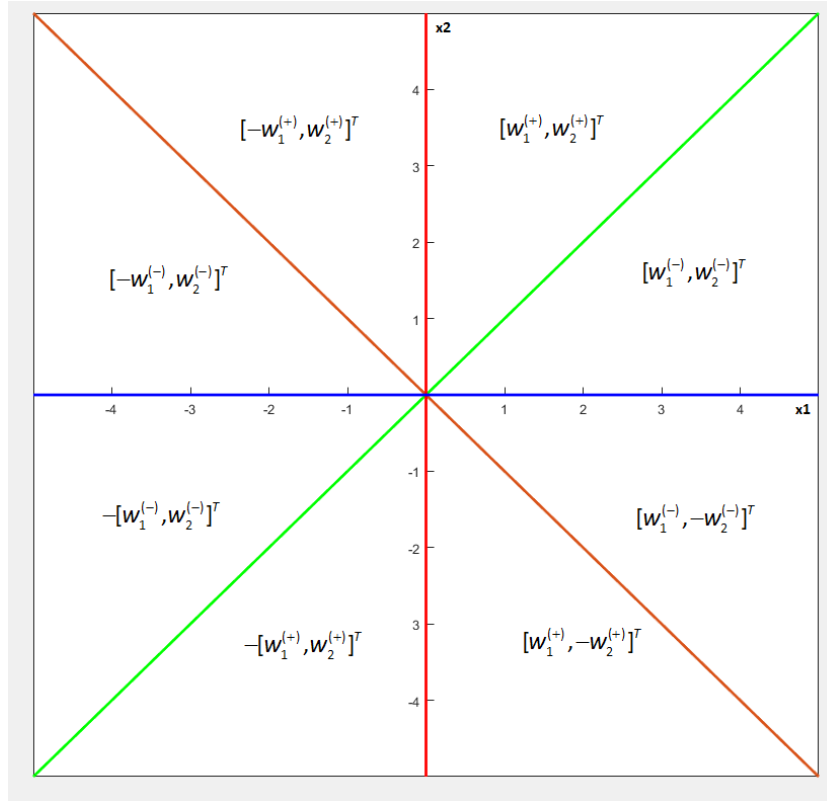


Рисунок 2.25. Отримані конуси з різними лінійними трансформаціями

У такий спосіб, використання у нейронній мережі декількох прихованих шарів замість одного дозволяє отримати в певному сенсі складнішу функцію за деяких умов. Однак, при цьому:

а) для отримання складнішої функції потрібно утримувати параметри мережі у певних регіонах, і якщо в процесі налаштування мережі параметри вийдуть за межі цих регіонів – складність функції, яку описує мережа, почне зменшуватися;

б) навіть при утриманні параметрів мережі в потрібних регіонах, функція мережі є складнішою по відношенню до мережі з одним прихованим шаром з такою ж кількістю параметрів лише візуально, оскільки кількість параметрів однакова, а це означає, що параметри лінійних трансформацій для різних опуклих конусів мають бути пов'язані певними рівняннями між собою, а також з параметрами роздільних гіперплощин, що утворюють ці опуклі конуси. Отже, така нейронна мережа з декількома шарами не буде здатна описати усі можливі лінійні трансформації та усі можливі розбиття на опуклі конуси;

с) при збільшенні шарів значно ускладнюється характер взаємодії усіх параметрів мережі, що призводить до ускладнення процесу налаштування мережі.

Враховуючи ці обмеження, а також проблеми багаторівневих архітектур поліноміальних нейронних мереж, розглянуті у пункті 2.2 разом зі складністю моделі самого SP нейрону, доцільним буде його використання у складі повнозв'язних мереж прямого розповсюдження з одним прихованим шаром. Оскільки SP нейрон є апроксимацією кусково-лінійного PL нейрону, то зважена сума SP нейронів буде апроксимувати відповідну зважену суму PL нейронів. Зважена сума PL нейронів може задавати функцію, що розділяє простір на велику кількість регіонів і на кожному регіоні застосовує локальну лінійну трансформацію – тобто при достатній кількості складових ця зважена сума здатна описати як завгодно складну кусково-лінійну функцію. Отже, навіть використання мережі прямого розповсюдження лише з одним прихованим шаром достатньо для апроксимації функцій з будь-якою точністю при застосуванні достатньої кількості нейронів.

Експериментальний тест ефективності мереж з SP нейронами

Перевіримо ефективність мереж описаної архітектури з SP нейронами у задачах прогнозування на реальних вибірках.

Для цього візьмемо вибірку максимальної за тиждень ціни акції компанії Apple за період 2012–2016 рр.:



Рисунок 2.26. Вибірка максимальної за тиждень ціни акції компанії Apple за період 2012–2016 рр.

З графіку вибірки можна доволі впевнено зробити висновок, що маємо справу з нестационарним процесом, причому доволі явно виділяються декілька різних станів процесу з різною поведінкою: повільне зменшення, повільне зростання, швидке зменшення і так далі.

Проаналізуємо відповідний різницевий ряд: $d_i = x_i - x_{i-1}, i = 2, \dots, N$:

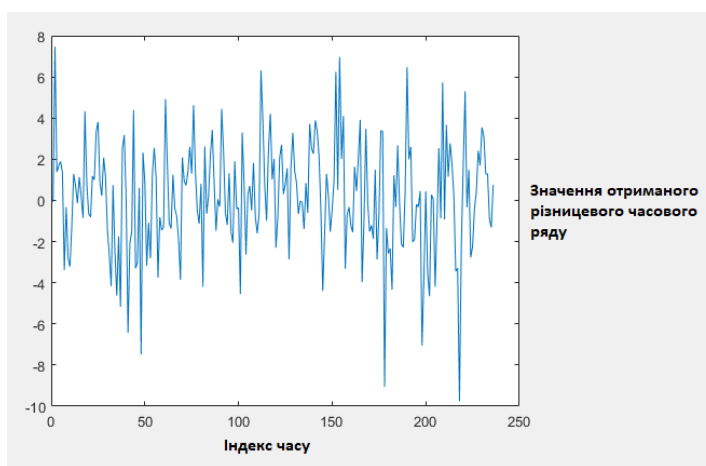


Рисунок 2.27. Різницевий ряд для вибірки максимальної за тиждень ціни акції компанії Apple за період 2012–2016 рр.

Доповнений тест Діккі–Фуллера вказує на стаціонарність процесу, що згенерував дану вибірку, отже немає необхідності у переході до наступних різницевих рядів.

Порівняємо ефективність використання наступних моделей для прогнозування цієї вибірки:

- наївна модель, яка припускає, що майбутнє значення буде рівне поточному: $\hat{x}_{i+1} = x_i$;

- модель ARIMA з використанням методу Бокса–Дженкінса для її побудови та навчання;

- поліноміальна мережа з використанням багаторядного алгоритму МГУА для її побудови та навчання;

- одношаровий персептрон з нейронами типу ReLU у прихованому шарі;

- одношаровий персептрон з SP нейронами у прихованому шарі.

Методика порівняння наступна:

1. Перехід до різницевого ряду $d_i = x_i - x_{i-1}, i = 2, \dots, N$.
2. Для навчання усіх моделей, окрім ARIMA, навчальна вибірка $\langle X, y: X \rightarrow R \rangle$ отримується з різницевого ряду за допомогою застосування методу вкладення часових рядів з розмірністю вкладення рівною 5 та горизонтом прогнозування 1 – тобто для прогнозу значення g_{i+1} використовуються значення $g_{i-4}, g_{i-3}, g_{i-2}, g_{i-1}, g_i$.

3. Перші 70% значень часового ряду відбираються у навчальну вибірку, решта – у тестову.

4. Середня квадратична помилка на навчальній вибірці використовується як критерій навчання. Середня абсолютна помилка (САП) моделі на тестовій вибірці використовується як критерій для порівняння прогнозуючих моделей.

Після налаштування структури та параметрів усіх обраних моделей отримано значення САП на тестовій вибірці, наведені в табл. 2.3.:

Таблиця 2.3.

САП прогнозуючих моделей на тестовій вибірці

Назва моделі/методу	САП на тестовій вибірці
Наївна модель	0.2550
ARIMA	0.2333
Поліноміальна мережа (МГУА)	0.2652
ReLU мережа	0.2365
SP мережа	0.2160

Отже, серед усіх методів та моделей, що порівнювалися, мережа з SP нейронами мала найменшу САП на тестовій вибірці, а САП другого за точністю методу – ARIMA моделі – приблизно на 8% більша.

Порівняймо гістограми помилок моделі ARIMA та мережі з SP нейронами на тестовій вибірці:

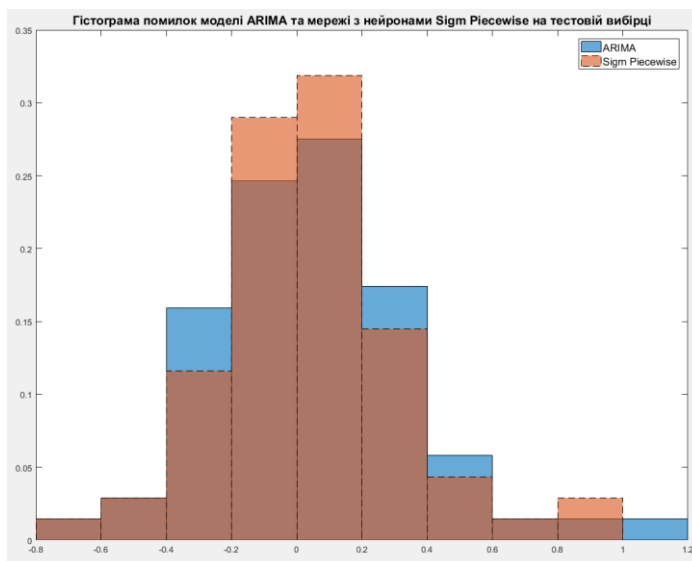


Рисунок 2.28. Гістограма помилок SP мережі та моделі ARIMA

Як видно з графіку з гістограмами, помітна більша кількість помилок на тестовій вибірці мережі з нейронами SP знаходиться в діапазоні $[-0.2; 0.2]$ у

порівнянні з помилками моделі ARIMA.

Висновки

- Розроблено новий загальний підхід побудови прогнозуючої моделі для цього класу часових рядів, що враховує досліджені властивості та недоліки існуючих методів прогнозування при їх застосуванні до рядів з класу. У такий спосіб, новий підхід є більш придатним для побудови прогнозуючих моделей часових рядів з декількома потенційними умовними розподілами.

- Розроблено нову модель штучного нейрону Sigmoid Piecewise (SP), яка складається з декількох локальних моделей та моделі-компоненту, що визначає, яку з локальних моделей застосовувати до вхідного вектору. Така модель краще підходить для опису часових рядів з декількома потенційними умовними розподілами, і проведені порівняльні тести на реальних вибірках це підтверджують: у всіх проведених тестах для досягнення помилки, меншої за певне граничне значення, потрібно було використати приблизно в 2.33 разів менше параметрів в SP мережі ніж в ReLU мережі, і при подальшому зменшенні цільової помилки відношення мінімально необхідної кількості параметрів в ReLU мережі до мінімально необхідної кількості параметрів в SP мережі продовжувало збільшуватись.

- Вдосконалено багаторядний алгоритм МТУА шляхом використання методів зворотного поширення помилки та вибіркового виключення, що дозволило оптимальніше налаштовувати параметри прогнозуючої моделі.

- Проведено експериментальний тест для порівняння ефективності нейронних мереж з SP нейронами: середня абсолютна помилка мережі з SP нейронами на тестовій вибірці була найменшою серед усіх методів, що порівнювалися, та на 9% меншою за помилку другого по точності методу ARIMA.

РОЗДІЛ 3. МЕТОДИ ПОБУДОВИ ПРОГНОЗУЮЧИХ МОДЕЛЕЙ ДЛЯ КЛАСУ НЕСТАЦІОНАРНИХ ЧАСОВИХ РЯДІВ З ДЕКІЛЬКОМА ПОТЕНЦІЙНИМИ УМОВНИМИ РОЗПОДІЛАМИ

У попередньому розділі було розроблено загальний підхід побудови прогнозуючих моделей для класу нестационарних часових рядів з декількома потенційними умовними розподілами, що базується на використанні компоненту оцінювача активного умовного розподілу та декількох локальних прогнозуючих моделей для різних потенційних умовних розподілів, разом з специфічними методами, моделями та алгоритмами для побудови локальних прогнозуючих моделей. Цей розділ присвячений розробці методів побудови компоненти оцінювача активного умовного розподілу та конкретних реалізацій загального підходу побудови прогнозуючих моделей для класу нестационарних часових рядів з декількома потенційними умовними розподілами.

3.1. Розробка методів розв’язання задачі побудови компоненту оцінювача активного розподілу

При побудові прогнозуючої моделі згідно розробленої загальної схеми для нестационарних часових рядів з декількома потенційними умовними розподілами потрібно певним чином оцінювати активний умовний розподіл для заданого вектору послідовних значень часового ряду $[y_t, \dots, y_{t-k+1}]$. Одним з типових підходів до вирішення цієї задачі є використання методів та алгоритмів кластеризації [78, 79, 80] для розбиття усієї навчальної вибірки вигляду $\{<[y_{t_i}, \dots, y_{t_i-k+1}], y_{t_i+h}>, i = 1, \dots, N\}$ на декілька кластерів і надалі вважати, що номер активного розподілу для певного вектору дорівнює номеру його кластеру.

3.1.1. Вдосконалення методу м'якої кластеризації шляхом застосування моделі роздільних гіперповерхонь

Можна виділити наступні основні існуючі методи та алгоритми, які застосовують для кластеризації підпоследовностей часового ряду або сегментації часового ряду – тобто пошуку так званих точок зміни поведінки ряду [81, 82]:

- методи та алгоритми для пошуку точок зміни поведінки ряду, що базуються на застосуванні динамічного програмування [83] або рекурсивних алгоритмів [84, 85, 86];
- класичні методи кластеризації – k-середніх, Гаусівські змішані моделі та інші [87, 88, 89, 90];
- методи, що базуються на теорії штучного інтелекту: нейромережеві автоенкодері [91], нечітка кластеризація [92] та інші.

Задачу кластеризації зазвичай розв'язують як наступну задачу дискретної оптимізації [93, 94, 95]: нехай задано множину векторів $X = (\vec{x}_1, \dots, \vec{x}_n)$, $\vec{x}_i \in R^d, i = 1, \dots, n$, тоді для кожного вектору потрібно визначити номер його кластеру $k \leq n$ у такий спосіб, щоб отриманий вектор номерів кластерів $\vec{k} = [k_1, \dots, k_n]^T$ мінімізував деякий критерій $C(\vec{k}, X)$:

$$\vec{k}^* = \underset{\vec{k}}{\operatorname{argmin}} C(\vec{k}, X).$$

Одним з типових критеріїв є критерій сумарної середньої відстані між векторами кожного кластеру [96]:

$$CR(\vec{k}, X) = \sum_i^K \sum_{\vec{x}_j: k_j=i} [\vec{x}_j - \vec{\mu}_i]^2$$

котрий можна переписати як:

$$CR(\vec{k}, X) = \sum_i^K \frac{1}{|C_i|} \sum_{\vec{x}_j, \vec{x}_l: k_j=i, k_l=i} [\vec{x}_j - \vec{x}_l]^2$$

Ввівши функцію $k(\vec{x})$, яка для кожного вектору задає його номер кластеру, цей критерій можна представити наступним чином:

$$CR(k, X) = \sum_i^K \frac{1}{\sum_{\vec{x} \in X} 1(k(\vec{x})=i)} \sum_{\vec{x}_j, \vec{x}_l} 1(k(\vec{x}_j)=i) \times 1(k(\vec{x}_l)=i) \times [\vec{x}_j - \vec{x}_l]^2,$$

де

$$1(\text{умова}) = \begin{cases} 1, & \text{якщо умова виконується} \\ 0, & \text{якщо умова не виконується} \end{cases}.$$

Нехай маємо 2 кластери, тобто $k(\vec{x}) \in \{0, 1\}$ і пом'якшимо вихідної задачі – дозволимо функції $k(\vec{x})$ видавати значення з діапазону $[0; 1]$, тобто для вектору \vec{x} значення функції $k(\vec{x})$ задає ймовірність приналежності цього вектору до першого кластеру, а значення $1 - k(\vec{x})$ задає ймовірність приналежності до другого кластеру. Тоді пом'якшений критерій можна записати як:

$$CR(k, X) = \left[\sum_{\vec{x} \in X} k(\vec{x}) \right]^{-1} \times \sum_{\vec{x}_j, \vec{x}_l} \left\{ k(\vec{x}_j) \times k(\vec{x}_l) \times [\vec{x}_j - \vec{x}_l]^2 \right\} + \\ \left\{ \sum_{\vec{x} \in X} [1 - k(\vec{x})] \right\}^{-1} \times \sum_{\vec{x}_j, \vec{x}_l} \left\{ [1 - k(\vec{x}_j)] \times [1 - k(\vec{x}_l)] \times [\vec{x}_j - \vec{x}_l]^2 \right\},$$

де дві складові суми відповідають за перший та другий кластер.

Введемо модель поверхні, що розділяє кластери, у вигляді функції $f(\vec{x}; \vec{w}) \in [0, 1]$, яка залежить від певного вектору параметрів \vec{w} і є диференційовною за цими параметрами. Тоді критерій $CR(k, X)$ також буде диференційовною функцією від вектору \vec{w} , і для його оптимізації можна використовувати методи та алгоритми оптимізації нелінійних диференційованих функцій. У такий спосіб, задачу м'якої кластеризації на 2

кластери з використанням критерії сумарної середньої відстані між векторами кожного кластеру можна звести до задачі оптимізації нелінійних диференційованих функцій, яку можна розв'язувати за допомогою алгоритму градієнтного спуску [97]. Для розв'язання задачі м'якої кластеризації на K кластерів можна використовувати модель, виходом якої є дискретний розподіл ймовірності приналежності вектору до кожного з кластерів. Один з варіантів побудови такої моделі є використання softmax функції, що приймає на вхід вектор $\vec{s} \in R^K$ та видає потрібний дискретний розподіл:

$$\text{softmax}(\vec{s})_i = \frac{e^{\vec{s}_i}}{\sum_{j=1}^K e^{\vec{s}_j}}, i = 1, \dots, K.$$

Беручи до уваги переваги мереж з одним прихованим шаром SP нейронів, що наведені у другому розділі, пропонується застосовувати саме ці мережі як моделі роздільних гіперповерхонь. Маємо наступний опис методу м'якої кластеризації на основі використання моделі роздільних гіперповерхонь:

1. Початковий вектор параметрів моделі \vec{w}_0 ініціалізується випадковим чином.
2. Виконується мінімізація критерію $CR(\vec{w}, X)$ шляхом застосування алгоритму градієнтного спуску та отримуємо налаштований вектор параметрів моделі $\vec{w}^{(*)}$.
3. Номер кластеру для кожного вектору з навчальної вибірки отримується шляхом знаходження максимального значення моделі:

$$k(\vec{x}) = \text{argmax}\{f(\vec{x}; \vec{w}^{(*)})\}, \forall \vec{x} \in X.$$

Розглянемо приклад застосування запропонованого методу на штучному прикладі, для чого створимо вибірку з трьома кластерами, де вектори в

кожному кластері згенеровані згідно нормального розподілу:

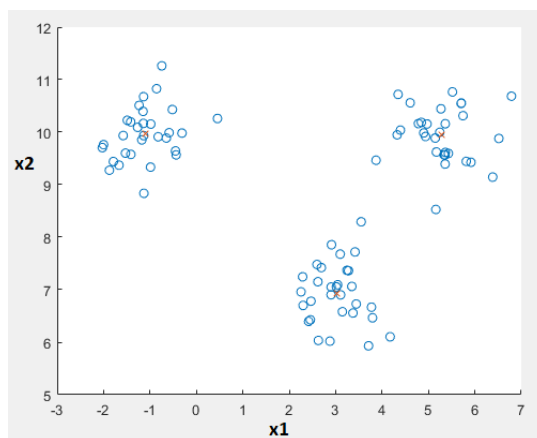


Рисунок 3.1. Штучна вибірка для перевірки роботи вдосконаленого методу м'якої кластеризації

Для простоти, замість використання єдиної моделі, що задає ймовірнісний розподіл приналежності до кожного з 3 кластерів, використаємо підхід «один проти всіх», тобто спочатку налаштуємо першу модель, що буде розділяти усі вектори на 2 кластери, після чого оберемо кластер з більшою дисперсією та налаштуємо другу модель, що буде розділяти його на 2 менші кластери. У якості моделі роздільної гіперповерхні (прямої у даному випадку) оберемо логістичну сигмоїду. У результаті застосування методу для налаштування параметрів моделі отримаємо наступну роздільну пряму:

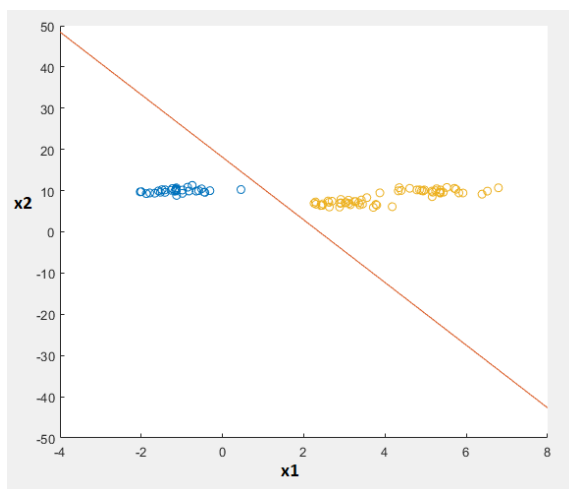


Рисунок 3.2. Отримана роздільна пряма

Отримана роздільна пряма дійсно відокремлює один з кластерів. Далі налаштуємо другу модель, що буде розділяти вектори у кластері з більшою дисперсією. У результаті маємо наступну роздільну пряму:

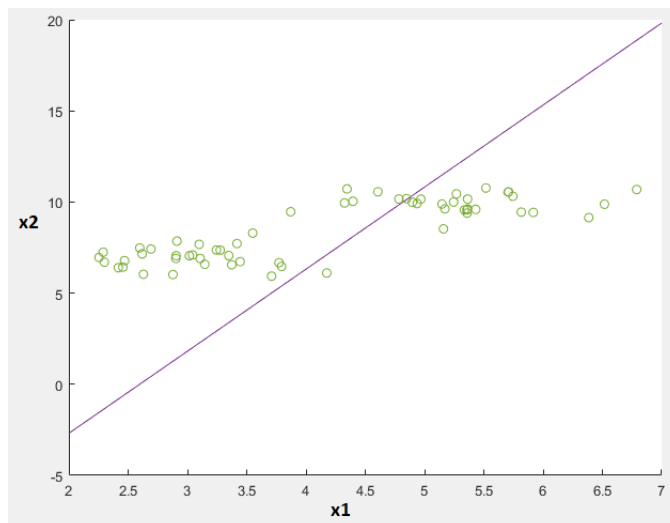


Рисунок 3.3. Друга отримана роздільна пряма

Для отримання складніших роздільних гіперповерхонь потрібно застосувати складнішу мережу. Наприклад, якщо використати мережу, що складається з 2 прихованих шарів з сигмоїдальною функцією активації, де кожен шар складається з 3 нейронів, то можна отримати наступну роздільну гіперповерхню (при навчанні на відповідній вибірці прикладів):

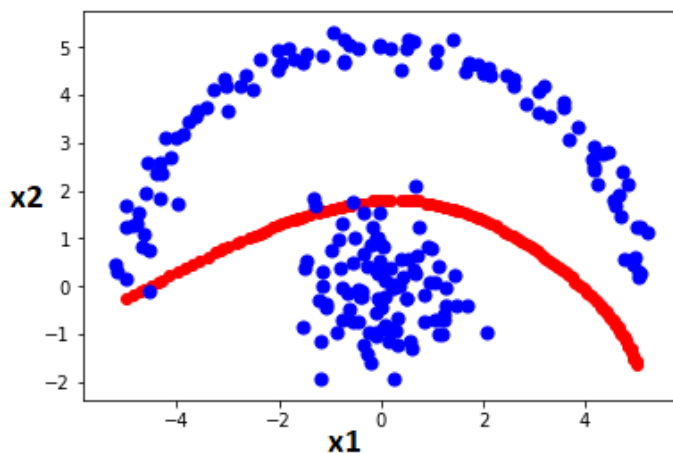


Рисунок 3.4. Приклад складнішої роздільної гіперповерхні, отриманої при навчанні складнішої мережі

Змінюючи параметри цієї мережі, можна отримати різні роздільні гіперповерхні, у тому числі комбінацію декількох гіперповерхонь:

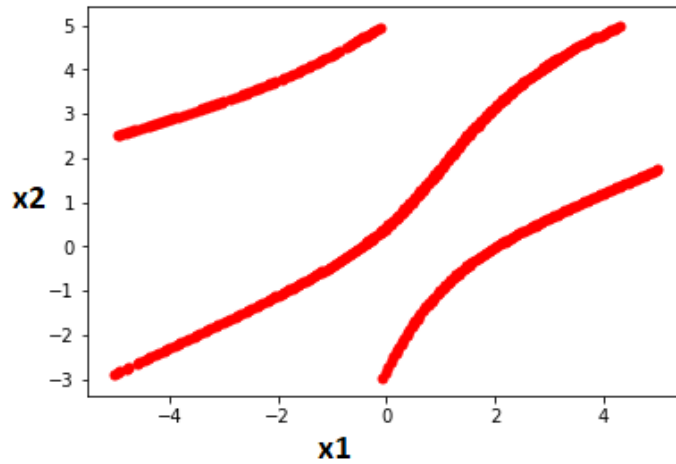


Рисунок 3.5. Роздільні гіперповерхні, отримані при використанні мережі з тією ж структурою, але іншими значеннями параметрів

Запропонований метод має певну схожість з методами нечіткої кластеризації [92, 98, 99], оскільки подібно до цих методів використовується нечіткий розподіл векторів по кластерам, де для вектора $\vec{x}^{(i)}$ задається розподіл належностей до кожного кластеру $\vec{\mu}^{(i)}$ (згідно з теорією нечіткої логіки [100, 101, 102]):

$$\vec{\mu}^{(i)} \in R^K, \sum_j \vec{\mu}_j^{(i)} = 1, \vec{\mu}_j^{(i)} \geq 0.$$

Однак, методи нечіткої кластеризації розраховують розподіли $\vec{\mu}^{(i)}$ для кожного вектору $\vec{x}^{(i)}$, у той час як цей метод розраховує параметри моделі роздільних гіперповерхонь $f(\vec{x}; \vec{w})$, використовуючи яку можна отримати розподіли для кожного вектору як $\mu^{(i)} = f(\vec{x}^{(i)}; \vec{w})$. Тобто кількість параметрів, які необхідно налаштовувати при застосуванні методу, що пропонується, не залежить від кількості векторів у вибірці, що дозволяє значно зменшити

необхідну кількість параметрів для великих вибірок. Також метод дозволяє отримати кластери, що не є розподіленими навколо певного центру, оскільки він базується на гіперповерхнях, які розділяють кластери, а не на їх центрах.

3.1.2. Використання методу комбінації експертів (mixture of experts) для налаштування компоненту оцінювача активного розподілу

Інший підхід, який можна використовувати для побудови компоненту оцінювання активного умовного розподілу, базується на відомому методі комбінації експертів [103]. Згідно з цим методом, налаштування локальних моделей та компоненту оцінювання активного умовного розподілу («стробуюча» модель в термінах оригінальної роботи) виконується одночасно, причому для налаштування параметрів компоненту оцінювання використовується спеціальна функція помилки. Згідно з цією функцією, компонент оцінювання визначає ймовірності того, що кожний потенційний умовний розподіл є активним для даного вхідного вектору, і фінальне значення прогнозу отримується як відповідним чином зважене усереднення прогнозів кожної локальної моделі (згідно з отриманими ймовірностями), що відповідає за окремий потенційний умовний розподіл. У такий спосіб, якщо компонент оцінювання впевнений, що для даного вхідного вектору активним є один з потенційних розподілів, то налаштовуватися будуть лише параметри відповідної локальної моделі, а параметри інших моделей залишаться майже незмінними.

3.2. Реалізація методу побудови прогнозуючої моделі для нестационарних часових рядів з декількома потенційними умовними розподілами на основі використання методу кластеризації та ШНМ

Входи методу:

- вектор спостережень часового ряду $\vec{s} \in R^n$;

- кількість станів–кластерів K ;
- розмірність вкладення ряду d ;
- горизонт прогнозування h .

Етапи методу:

1. Часовий ряд \vec{s} перетворюється на матрицю вхідних векторів $X \in R^{m \times d}$ та цільовий вектор $\vec{y} \in R^m$ шляхом застосування алгоритму вкладення часового ряду з горизонтом прогнозування h та розмірністю вкладення d , де $m = n + 1 - h - d$.

2. Навчається одношаровий персептрон з SP нейронами в прихованому шарі з використанням всієї множини наявних прикладів $S = \{(\vec{x}_i, y_i) : i = 1 \dots m\}$. Для визначення кількості нейронів у прихованому шарі застосовується описаний раніше метод простого структурного синтезу на основі багаторядного алгоритму МГУА. Функція помилки мережі задається стандартною СКП:

$$E(\vec{\theta}) = \sum_{\vec{x}_i, y_i} \left[f(\vec{x}_i; \vec{\theta}) - y_i \right]^2 .$$

Вектор параметрів $\vec{\theta}^{(g)}$ цього персептрону містить інформацією про глобальний характер ряду, що прогнозується.

3. Виконується кластеризація вибірки S використовуючи метод м'якої кластеризації на основі роздільних гіперповерхонь, з використанням одношарового персептрону з SP нейронами в прихованому шарі в якості моделі роздільної гіперповерхні. Для визначення кількості нейронів у прихованому шарі застосовується описаний раніше метод простого структурного синтезу на основі багаторядного алгоритму МГУА. Налаштована модель кластеризації у такий спосіб виконує функцію компоненту оцінювача активного розподілу у загальному підході побудови

прогнозуючих моделей для класу нестационарних часових рядів з декількома потенційними умовними розподілами.

4. Виконується налаштування K локальних мереж:

- вектор параметрів кожної мережі ініціалізуються як $\vec{\theta}^g$;
- при налаштуванні j -ої мережі f_j використовуються лише вектори

з кластеру C_j і наступна функція помилки:

$$E_j(\vec{\theta}) = \sum_{\vec{x}_i \in C_j} (f_j(\vec{x}; \vec{\theta}) - y_i)^2 + \gamma \sum_{\theta_l \in \vec{\theta}} (\theta_l - \theta_l^{(g)})^2.$$

Через використання цієї функції помилки в процесі налаштування, вектор параметрів кожної локальної мережі не зможе занадто «віддалятися» від вектору параметрів $\vec{\theta}^{(g)}$, що зменшує вплив проблеми перенавчання. Параметр $\gamma \in [0, \infty)$ є гіперпараметром, який контролює вплив регуляризуючої складової. Існують різноманітні підходи для вибору значення регуляризуючих гіперпараметрів [104, 105, 106] та інші. На практиці, доволі часто застосовують дуже прості алгоритми: перебір усіх можливих значень із заданої наперед кінцевої множини значень, випадковий перебір.

5. Прогнозування для нових векторів виконується шляхом знаходження відповідного кластеру C_h та застосування локальної мережі, яку було налаштовано на векторах з цього кластеру.

3.3. Реалізація методу побудови прогнозуючої моделі для нестационарних часових рядів з декількома потенційними умовними розподілами на основі використання методу комбінації експертів

Проблемою запропонованого методу на основі використання кластеризації та побудови локальних моделей для кожного кластеру є залежність від якості кластеризації – погана кластеризація може призвести до

значного зменшення точності прогнозуючої моделі. Для зменшення впливу цієї проблеми пропонується вдосконалення методу комбінації експертів шляхом застосування спеціальної регуляризації, що запобігає проблемі перенавчання [107, 108]. Модифікований метод комбінації експертів має наступний вигляд:

1) Навчання єдиної глобальної моделі $f^{(g)}$ використовуючи звичайну середню квадратичну функцію помилки. У результаті цього кроку буде отримано вектор параметрів глобальної моделі $\vec{w}^{(g)}$.

2) Ініціалізуються K моделей експертів, шляхом копіювання глобальної моделі $f^{(g)}$ – це означає, що у даній модифікації усі експерти повинні мати однакові математичні моделі.

3) Ініціалізується стробуюча модель g . Оскільки ця модель повинна оцінювати важливість усіх моделей експертів для заданого вектору, де вектор важливостей є дискретним ймовірнісним розподілом, то можна виконати початкове налаштування параметрів стробуючої моделі для виконання м'якої кластеризації навчальної вибірки із застосуванням вдосконаленого методу м'якої кластеризації на основі роздільних гіперповерхонь.

4) Виконується одночасне налаштування параметрів моделей експертів та стробуючої моделі з використанням наступної функції помилки:

$$\begin{aligned} RE(\vec{x}, \vec{w}_1^{(e)}, \dots, \vec{w}_K^{(e)}, \vec{w}^{(g)}) = \\ = E(\vec{x}, \vec{w}_1^{(e)}, \dots, \vec{w}_K^{(e)}, \vec{w}^{(g)}) + \gamma * \sum_{j=1}^K \sum_{w_i \in \vec{w}_j^{(g)}} (w_i - w_i^{(g)})^2, \\ \gamma \in R^+. \end{aligned}$$

Друга складова цієї функції помилки намагається залишити параметри кожного локального експерту близькими до параметрів глобальної моделі $f^{(g)}$, у такий спосіб, навіть при відносно малій кількості наявних спостережень локальні експерти не зможуть перенавчитися. Відносний вплив другої

складової контролюється гіперпараметром γ .

Після зупинки стадії налаштування параметрів та отримання фінальної прогнозуючої моделі, прогнозування для нових входних векторів виконується наступним чином:

$$f^{(F)}(\vec{x}; \vec{w}_1^{(e)}, \dots, \vec{w}_K^{(e)}, \vec{w}^{(g)}) = \sum_{j=1}^K g(\vec{x}, j; \vec{w}^{(g)}) f_j(\vec{x}; \vec{w}_j^{(e)})',$$

тобто виконується усереднення виходів кожного локального експерту f_j з вагами, що задаються стробуючою моделлю g .

3.4. Тестування на штучних та реальних даних

Перевіримо запропоновані методи на штучно згенерованих даних та реальних вибірках. Для перевірки на штучних згенеруємо вибірку вигляду:

$$y_t = \sin\left(\frac{w_t}{2\pi} \times t\right) + N(0, 0.01), t = 1, \dots, 300, w_{1:100} = 3, w_{101:200} = 6, w_{201:300} = 1,$$

що має наступний графік:

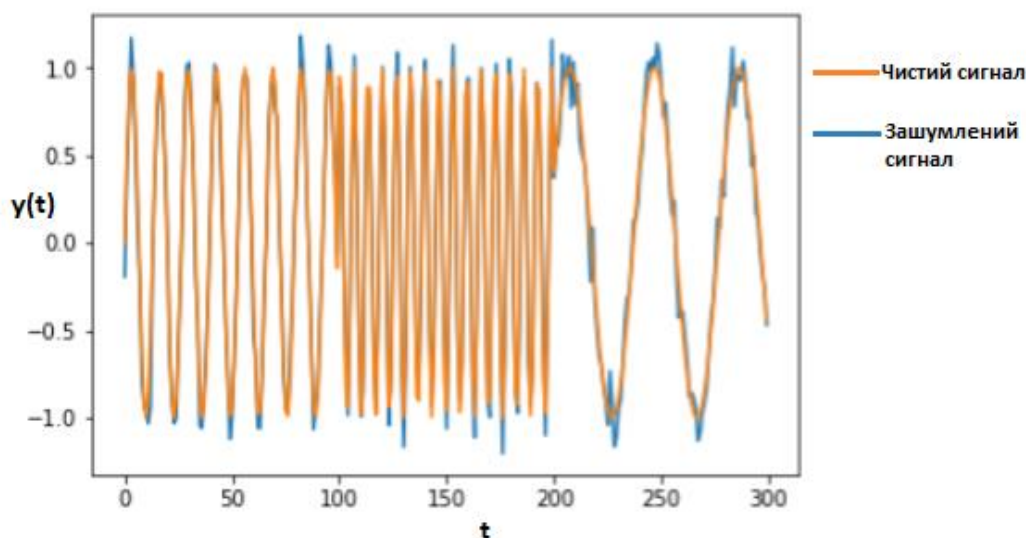


Рисунок 3.6. Згенеровані штучні дані для перевірки методів

Очевидно, що часовий ряд належить до класу нестационарних часових рядів з декількома потенційними умовними розподілами. Застосуємо метод вкладення часових рядів з горизонтом прогнозування $h=2$ та розмірністю вкладення $d=5$ для отримання вибірки $\langle X, y \rangle$, яку випадковим чином розіб'ємо на навчальну – $\langle X^{(train)}, \vec{y}^{(train)} \rangle$ та тестову – $\langle X^{(test)}, \vec{y}^{(test)} \rangle$, де навчальна вибірка буде використовуватися для налаштування моделей, а тестова – для перевірки точності налаштованих моделей. Помилки налаштованого персептрону з 2 прихованими шарами та 12 ReLU нейронами в кожному прихованому шарі склали: СКП=0.098, САП=0.2:

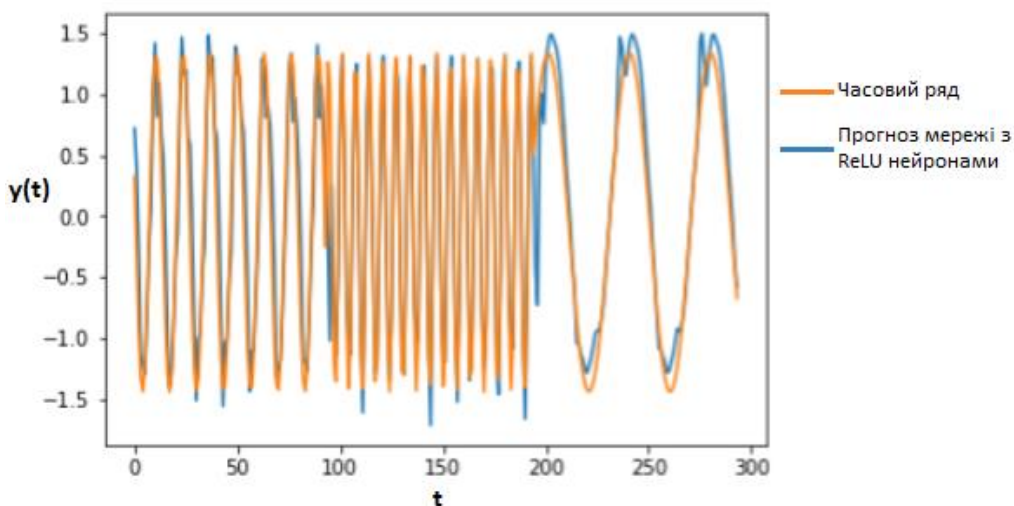


Рисунок 3.7. Часовий ряд та прогноз мережі з ReLU нейронами

Далі було навчено прогножуючу модель згідно з методом комбінації експертів з трьома експертами, де для моделей експертів та стробуючої моделі застосовувалися персептрони з одним прихованим шаром з 12 ReLU нейронів.

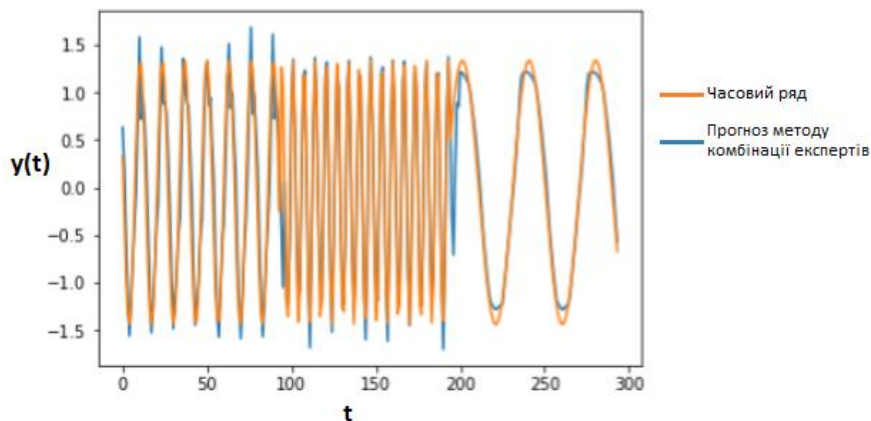


Рисунок 3.8. Часовий ряд та прогноз моделі, побудованої згідно методу комбінації експертів

На графіку видно, що помилки цієї моделі склали: СКП=0.083, САП=0.175, а модель, побудована згідно методу комбінації експертів, помітно краще прогнозує 3 стан часового ряду. Нарешті було налаштовано прогножуючу модель згідно вдосконаленого методу комбінації експертів з використанням додаткової регуляризації, використовуючи перцептрон з одним прихованим шаром з 12 ReLU нейронів для моделей експертів та стробуючої моделі. Помилки цієї моделі склали: СКП=0.079, САП= 0.163:

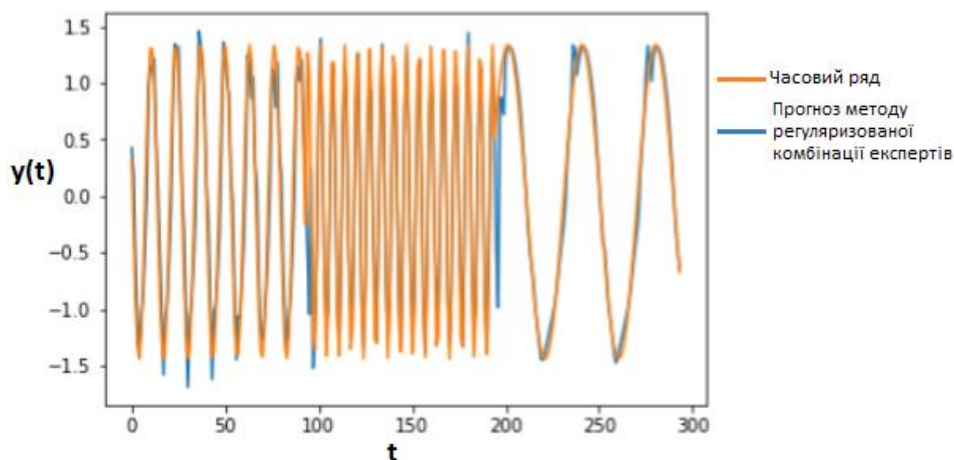


Рисунок 3.9. Часовий ряд та прогноз моделі, побудованої згідно методу комбінації експертів з додатковою регуляризацією

З графіку видно, що додаткова регуляризація покращила точність прогнозу для усіх станів ряду. Помилки усіх налаштованих моделей наведені в табл. 3.1.:

Таблиця 3.1.

Помилки моделей на тестовій вибірці на штучних даних

Модель	СКП	САП
Єдина мережа	0.0984	0.2
Метод комбінації експертів	0.082	0.175
Метод регуляризованої комбінації експертів	0.079	0.163

Для тестування на справжніх даних було використано вибірку з індексом T10Y2Y з масиву даних, який доступний онлайн на сайті Федерального Резервного банку Сент-Луїса [109]:

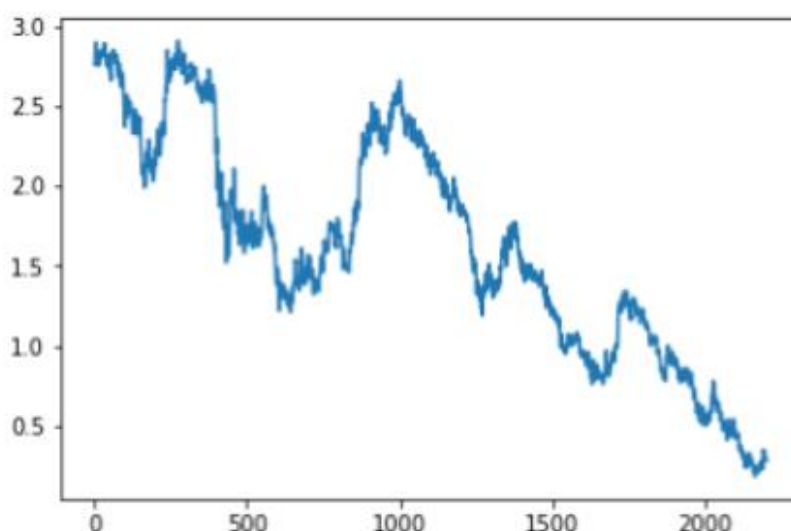


Рисунок 3.10. Тестова вибірка

Для отримання навчальної та тестових вибірок було застосовано вкладення даного часового ряду з горизонтом прогнозування $h=2$ та розмірністю вкладення $d=10$, після чого налаштовано аналогічні моделі, як і

для штучних даних. Помилки налаштованих моделей на тестовій вибірці наведені у табл. 3.2.:

Таблиця 3.2.

Помилки моделей на тестовій вибірці на реальній вибірці

Модель	СКП	САП
Єдина нейронна мережа	0.0057	0.057
Метод комбінації експертів	0.0064	0.063
Метод регуляризованої комбінації експертів	0.0052	0.055

Як видно з таблиці, помилки на тестовій вибірці моделі, налаштованої згідно методу комбінації експертів, більші за помилки єдиної нейронної мережі – таке зниження точності прогнозу може бути наслідком перенавчання моделей експертів. При цьому застосування вдосконаленого методу комбінації експертів з додатковою регуляризацією дозволило отримати модель з найменшими помилками на тестовій вибірці, і можлива причина полягає в тому, що додаткова регуляризація дозволяє зменшити вплив перенавчання при налаштуванні моделей експертів.

Висновки

- Вдосконалено метод м'якої кластеризації шляхом використання моделі роздільних гіперповерхонь, що дає можливість розв'язувати задачу м'якої кластеризації шляхом налаштування параметрів штучної нейронної мережі та знаходити кластери, які розділені в просторі гіперповерхнями, складність яких залежить від розміру мережі – тобто збільшення кількості параметрів мережі дозволяє отримати більш складну роздільну гіперповерхню.

- Розроблено метод побудови прогнозуючих моделей для нестационарних рядів з декількома потенційними умовними розподілами, що

базується на застосуванні методу кластеризації на основі роздільних гіперповерхонь та штучних нейронних мереж з SP нейронами в якості локальних моделей, з використанням спеціальної регуляризації в процесі налаштування локальних моделей.

- Вдосконалено метод комбінації експертів шляхом застосування спеціальної регуляризації та використання штучних нейронних мереж з SP нейронами в якості моделей–експертів, що підвищує стійкість експертів до проблеми перенавчання.

- Нові та вдосконалені методи були порівняні з існуючими методами та алгоритмами на штучних та реальних вибірках, і дозволили досягти зменшення середньої квадратичної помилки прогнозу на тестовій вибірці на 19% на штучних даних та на 9% на реальних даних.

РОЗДІЛ 4. ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ПРОГНОЗУВАННЯ НЕСТАЦІОНАРНИХ ЧАСОВИХ РЯДІВ НА ОСНОВІ ШТУЧНИХ НЕЙРОННИХ МЕРЕЖ

4.1. Основні сценарії використання інформаційної технології

Головна ціль інформаційної технології, що розроблюється – спростити побудову прогнозуючих моделей для нестационарних часових рядів з використанням розроблених методів, алгоритмів та моделей. Виходячи з цієї цілі можна виділити наступні основні сценарії використання застосунку:

Сценарій 1 – побудова нової прогнозуючої моделі використовуючи певну вибірку даних:

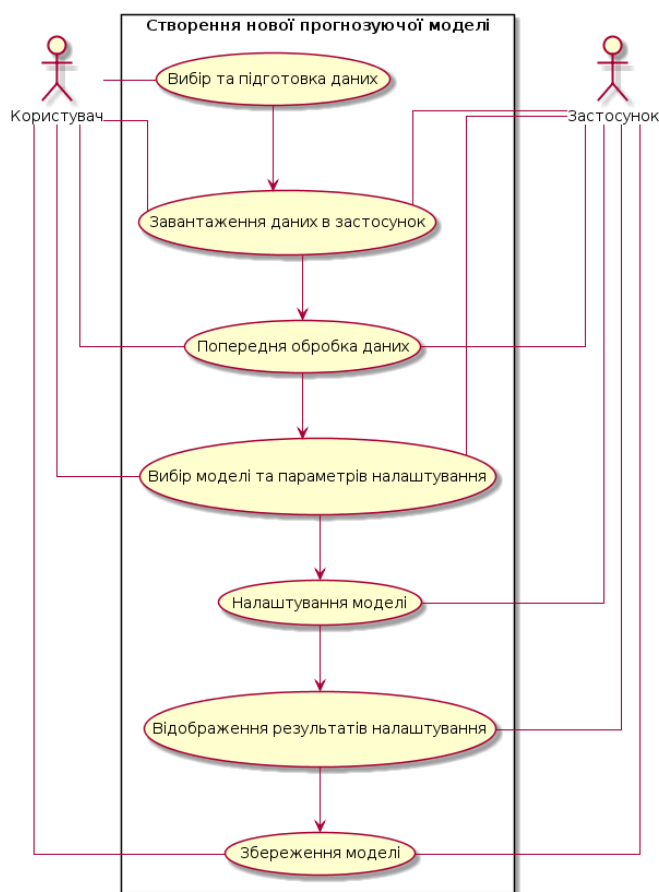


Рисунок 4.1. Діаграма сценарію «побудова нової моделі»

Сценарій 2 – використання збереженої моделі для прогнозування нових даних:

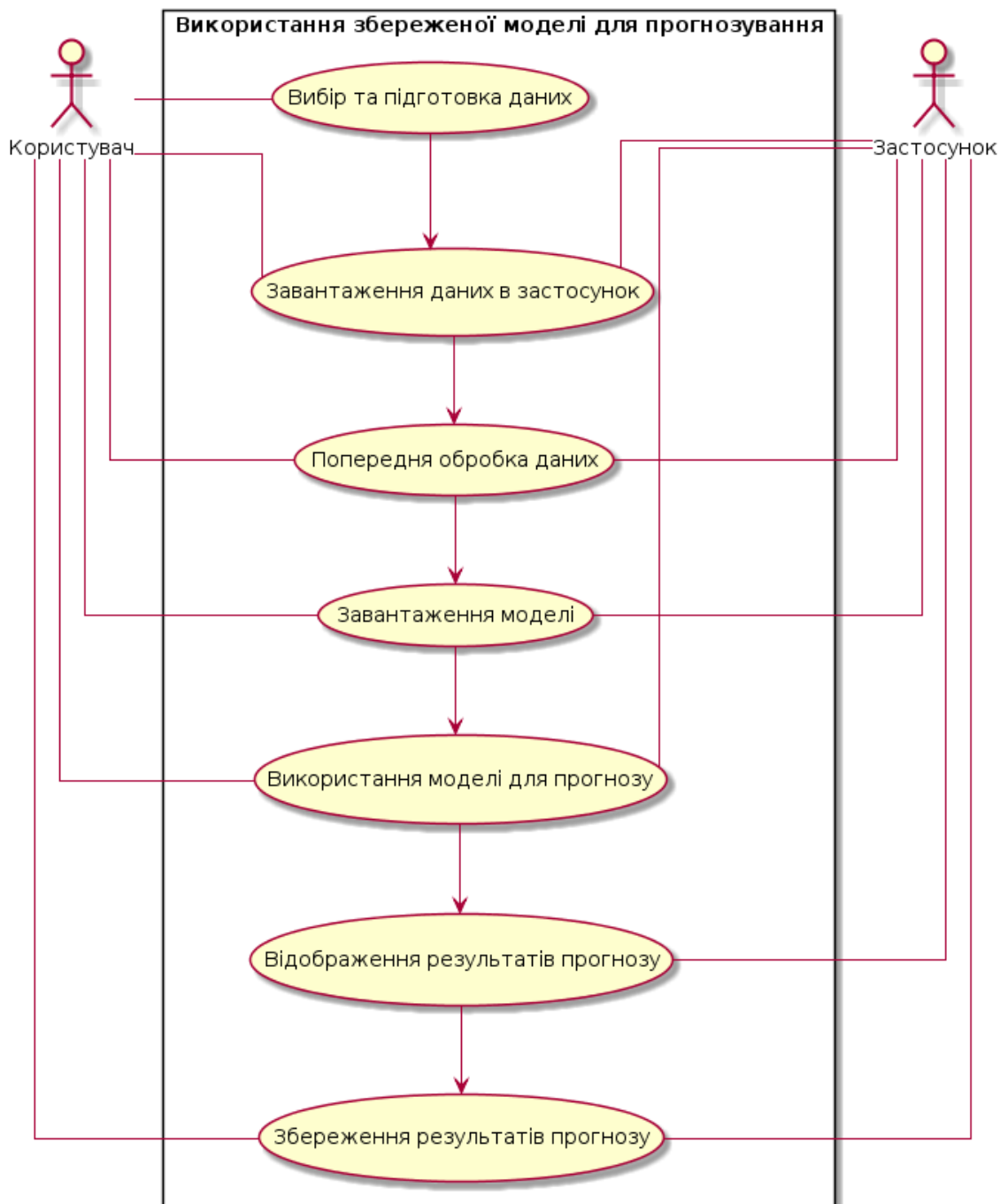


Рисунок 4.2. Діаграма сценарію «використання збереженої моделі для прогнозування нових даних»

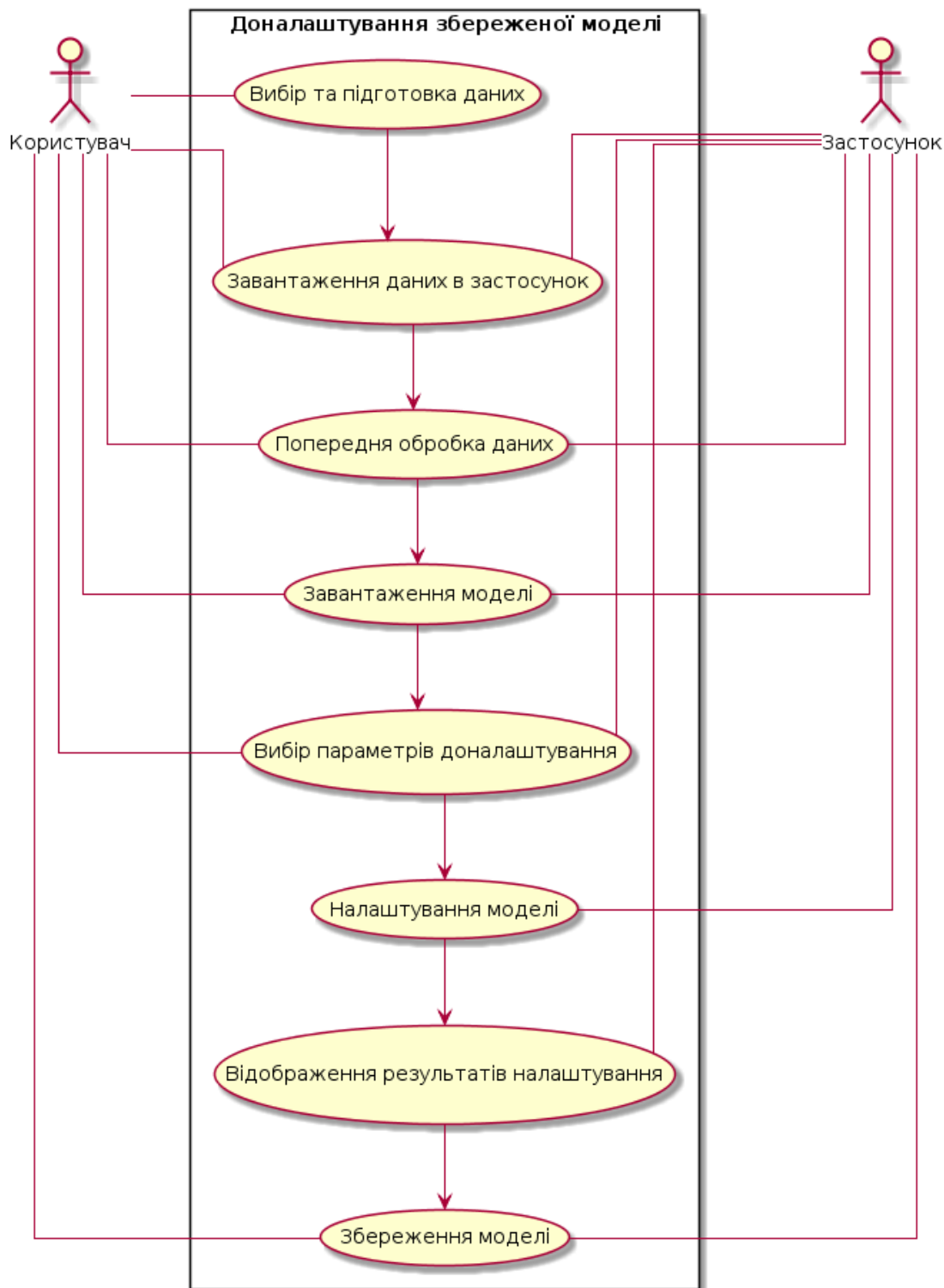
Сценарій 3 – налаштування збереженої моделі:

Рисунок 4.3. Діаграма сценарію «налаштування збереженої моделі»

Сценарій 4 – візуалізація та обробка даних:

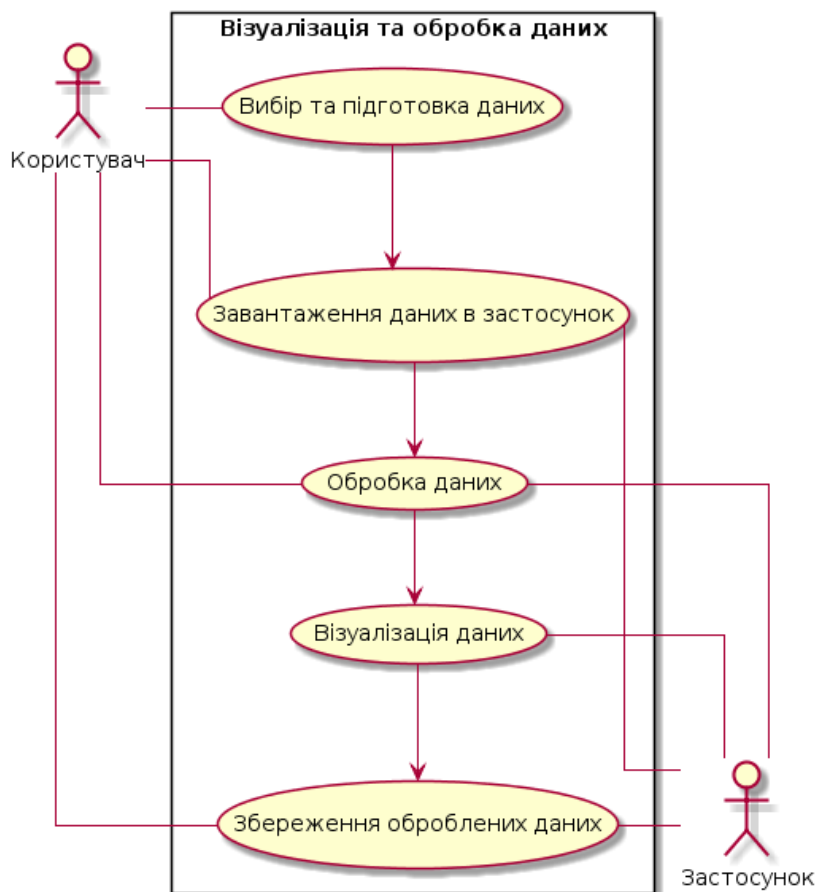


Рисунок 4.4. Діаграма сценарію «візуалізація та обробка даних»

4.2. Технологічні аспекти програмної реалізації інформаційної технології прогнозування нестационарних часових рядів

Головними параметрами інформаційної технології є вхідні дані, операції, які над ними виконуються, та вихідні дані – результати операцій над вхідними даними. Інформаційна технологія прогнозування нестационарних часових рядів (ІТ ПНЧР), що розроблюється, має наступний опис згідно з цими параметрами:

- вхідні дані – часові ряди;

- обробка часових рядів виконується згідно розроблених нових та вдосконалених існуючих методів та моделей;
- вихідними даними є побудовані прогнозуючі моделі, та прогнози, отримані шляхом їх використання;
- вихідні дані ІТ ПНЧР можна в подальшому застосовувати в системах підтримки прийняття рішень.

З точки зору способу взаємодії з користувачем виділяють наступні типи програмної реалізації інформаційних технологій:

- 1) застосунки командного рядку, де взаємодія з користувачем відбувається через вбудований в операційну систему термінал;
- 2) веб-застосунки – взаємодія відбувається через графічний інтерфейс, що надається веб-браузером, який зазвичай іде в комплекті з операційною системою, але не є частиною самої системи;
- 3) застосунки з графічним інтерфейсом, що не потребують браузера для запуску – тобто використовують засоби відображення графічного інтерфейсу, які надаються самою операційною системою;
- 4) забезпечення, яке надає лише програмний інтерфейс (API) як засіб взаємодії – тобто звичайний користувач не має можливості використовувати дане забезпечення.

Для програмної реалізації інформаційної технології ПНЧР на основі використання розроблених методів, алгоритмів та моделей, було обрано тип взаємодії «застосунок з графічним інтерфейсом», через наступні переваги:

- зручність використання для недосвідченого користувача;
- можливість використання складних візуалізацій, які на відміну від веб-застосунків не обмежені можливостями, що надаються веб-браузерами.

Ще одним аспектом програмної реалізації ІТ є її розподіленість – тобто, чи програмна реалізація потребує або підтримує розподілений режим роботи на декількох обчислювальних пристроях (ОП), поєднаних мережею. За цим аспектом можна виділити наступні класи програмних систем (ПС), впорядковані за зростанням складності відповідного класу:

- 1) повністю локальні ПС, які працюють на єдиному ОП та не підтримують роботу на декількох ОП;
- 2) слабо–розподілені ПС, які використовують відносно невелику кількість ОП, поєднаних мережею; зазвичай це пристрої, що належать до єдиної локальної мережі підприємства/установи;
- 3) сильно–розподілені ПС [110], які підтримують/потребують режим роботи на великій кількості ОП, поєднаних мережею; це можуть бути пристрої з багатьох різних локальних мереж, поєднаних в мережі Інтернет.

Найпростішим класом є повністю локальні ПС, і слабо/сильно–розподілені ПС використовуються, коли в цьому є потреба, наприклад, коли обчислювальних можливостей єдиного ОП не вистачає, щоб виконувати поставлені завдання у потрібний проміжок часу. У випадку завдань, що ставляться перед розроблюваною ІТ, значної потреби в розподіленій роботі не виникає, тому було обрано саме повністю локальний варіант реалізації.

Оскільки як тип програмної реалізації ІТ було обрано застосунок із графічним інтерфейсом, то доцільним буде використати класичну трирівневу архітектуру [111, 112] побудови таких застосунків: «Рівень подання» → «Рівень бізнес–логіки» → «Рівень доступу до даних». Згідно з цією архітектурою, застосунок розділяється на 3 впорядковані рівні, де кожен рівень взаємодіє лише з нижчим рівнем шляхом використання інтерфейсів рівнів – тобто не кожен програмний компонент може безпосередньо спілкуватися з будь–яким компонентом нижчого рівня; замість цього, лише одному

спеціально призначеному компоненту дозволено спілкуватися з інтерфейсним компонентом нижчого рівня.

Розроблена ІТ придатна до використання в інформаційних системах (ІС) з типовою структурою «Користувач» → «Робоча станція» → «Мережа» → «Сервер БД» → «БД». Також можливо, що БД або сховище даних знаходяться на робочій станції, і доступ до них виконується засобами локальної ОС, де сама ІТ застосовується на робочій станції, згідно схеми, наведеної на рис. 4.5.

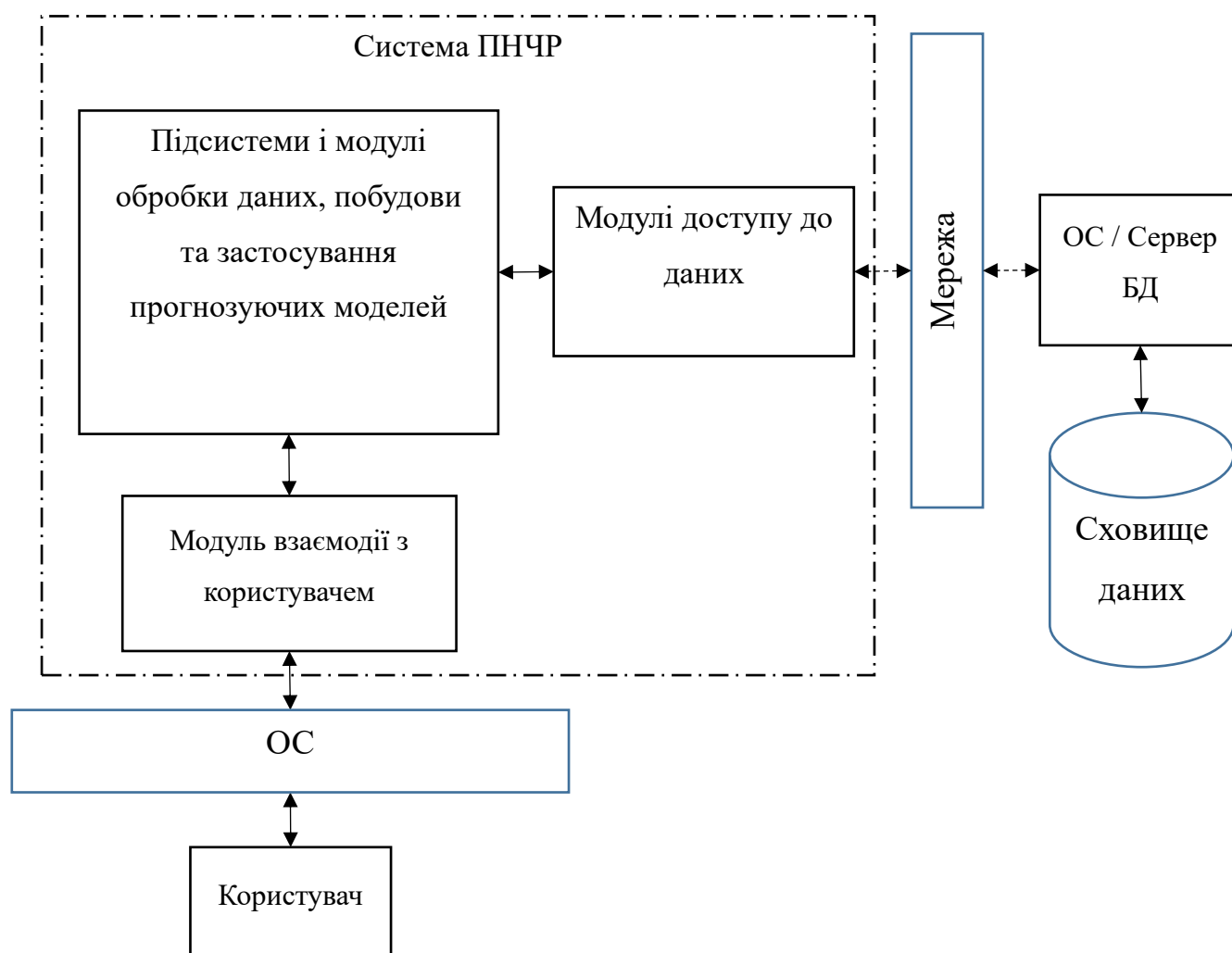


Рисунок 4.5. Інтеграція ІТ ПНЧР в інформаційній системі організації

До складу системи ПНЧР входять наступні модулі:

- *Модуль завантаження даних з зовнішніх джерел:* відповідає за відкриття даних із зовнішніх джерел у різноманітних форматах та їх перетворення до внутрішнього формату ІТ.
- *Модуль трансформації даних:* відповідає за різноманітні засоби обробки та перетворення даних у внутрішньому форматі ІТ.
- *Модуль побудови прогнозуючої моделі:* відповідає за реалізацію розроблених методів та моделей побудови прогнозуючих моделей для нестационарних часових рядів.
- *Модуль застосування прогнозуючої моделі:* застосовує побудовану прогнозуючу модель до заданого набору даних.
- *Модуль взаємодії з користувачем:* реагує на дії користувача та координує роботу інших модулів відповідно до цих дій.
- *Модуль візуалізації даних:* приймає на вхід певний набір даних та параметри візуалізації та відповідним чином відображає ці дані.
- *Модуль збереження даних:* перетворює дані з внутрішнього формату до обраного зовнішнього формату та зберігає їх на зовнішньому сховищі.

Одним з найскладніших модулів системи ПНЧР є модуль взаємодії з користувачем. У зв'язку з тим, що методи побудови прогнозуючих моделей дуже різноманітні по своїй структурі, і до складу різних методів входять різні етапи, даний модуль повинен підтримувати можливість побудови складних і різноманітних обчислювальних схем. Для цього модуль взаємодії з користувачем реалізований згідно парадигми «менеджер–виконавці» [113, 114, 115]: у відповідь на дії користувача даний модуль створює та налаштовує усі інші модулі і зв'язки між ними; після того, як користувач запускає виконання

отриманої схеми, модуль взаємодії з користувачем керує процесом виконання усіх модулів шляхом відправки та отримання повідомлень до/від модулів обчислювальної схеми. Усі модулі системи ПНЧР взаємодіють один з одним згідно наступної схеми:



Рисунок 4.6. Схема взаємодії модулів системи ПНЧР та користувача системи

Цільовою операційною системою для застосунку обрано ОС Windows XP та новіше, для розробки та розгортання застосунку використовується фреймворк Microsoft .NET 4.5 [116]. Для підтримки класичних методів побудови прогнозуючих моделей використовуються пакети Cronos та ML.NET, для роботи з нейронними мережами та реалізації розроблених та вдосконалених у цій роботі методів та моделей – пакет Aforge.NET. Зовнішні

дані завантажуються із файлів ОС або з бази даних використовуючи технологію ODBC. У такий спосіб, схема розгортання системи ПНЧР в інформаційній системі користувача виглядає наступним чином:

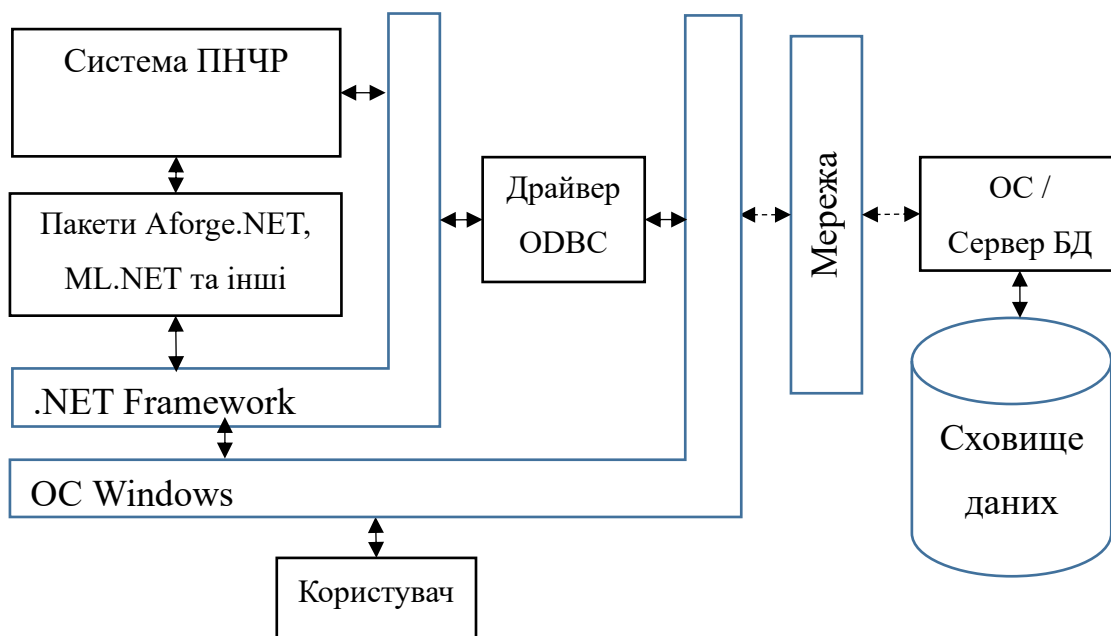


Рисунок 4.7. Схема розгортання системи ПНЧР в ІС організації

4.3. Аналіз процесів обробки даних в інформаційній технології прогнозування нестационарних часових рядів

Розглянемо процеси обробки даних при виконанні основних задач інформаційної технології. Одною з головних задач ІТ ПНЧР є побудова прогнозуючої моделі використовуючи наявні значення певного часового ряду. Для вирішення цієї задачі потрібні наступні етапи обробки даних: завантаження даних, обробка даних, побудова прогнозуючої моделі та збереження даних. Схема процесу обробки даних при виконанні цієї задачі виглядає наступним чином:

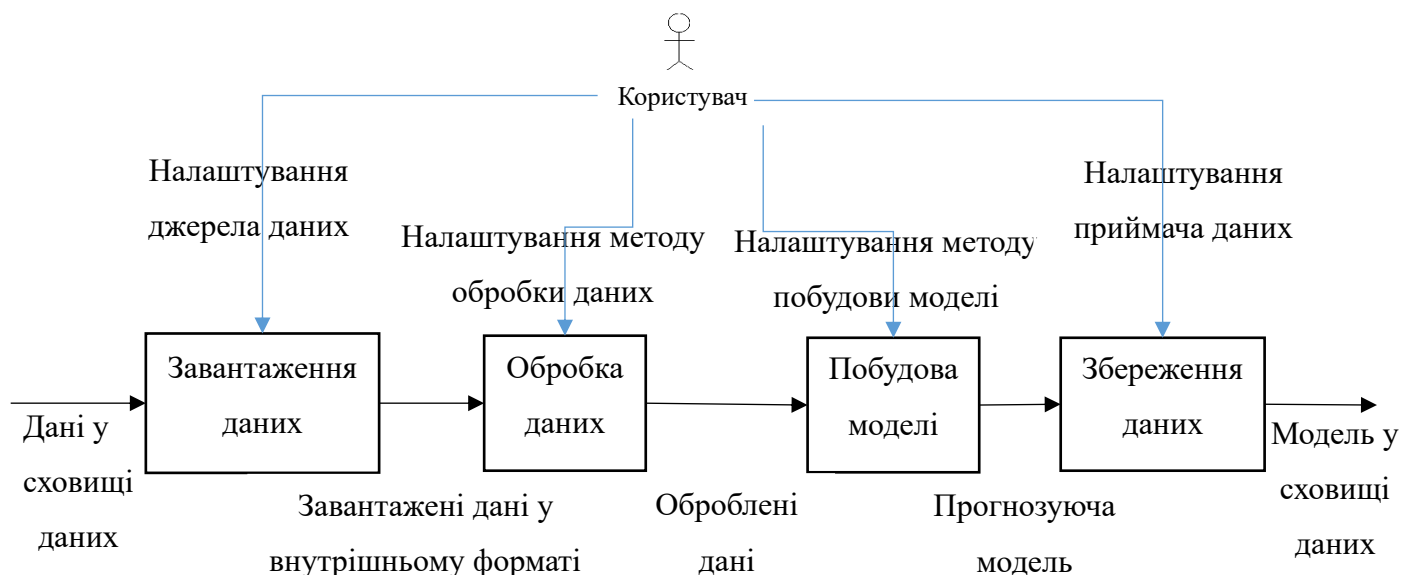


Рисунок 4.8. Схема процесу обробки даних при виконанні задачі побудови та збереження прогнозуючої моделі використовуючи наявні дані

У наступних схемах опущено управляючі сигнали налаштування від користувача до етапів обробки даних, але поведінка кожного етапу залежить від цих сигналів. Другою за важливістю є задача застосування побудованої прогнозуючої моделі для отримання прогнозу на нових даних:

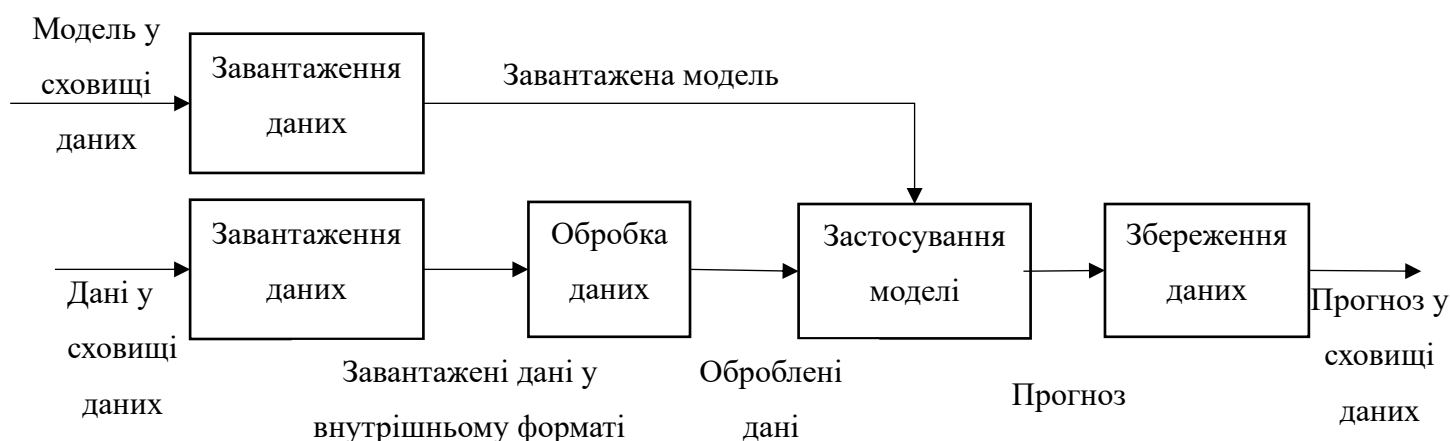


Рисунок 4.9. Схема процесу обробки даних при виконанні задачі застосування збереженої моделі для побудови прогнозу на нових даних

Розроблені та вдосконалені методи та моделі для прогнозування нестационарних часових рядів використовуються в інформаційній технології

на етапі побудові прогнозуючої моделі. Процес обробки даних при застосуванні методу побудови моделі на основі кластеризації та локальних моделей має наступну схему:

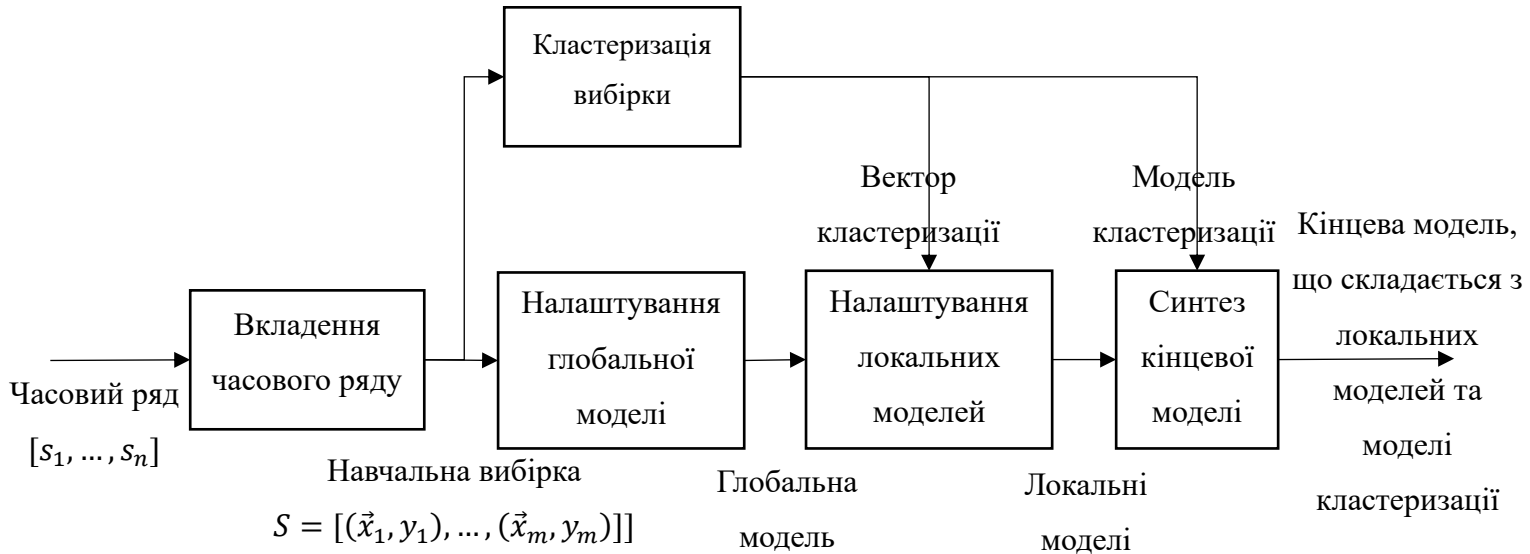


Рисунок 4.10. Схема процесу обробки даних при побудові моделі згідно методу на основі кластеризації та локальних моделей

Схожу схему має процес обробки даних при застосуванні вдосконаленого методу комбінації експертів:



Рисунок 4.11. Схема процесу обробки даних при побудові моделі згідно вдосконаленого методу комбінації експертів

Різниця в тому, що замість моделі кластеризації використовується модель комбінації експертів, і налаштування локальних моделей виконується одночасно з налаштуванням моделі комбінації експертів, відкидаючи потребу в етапі синтезу кінцевої моделі.

Розглянемо основні сценарії взаємодії компонент системи ПНЧР та відповідні діаграми послідовностей.

1. Взаємодія користувача і компонент протягом життєвого циклу:

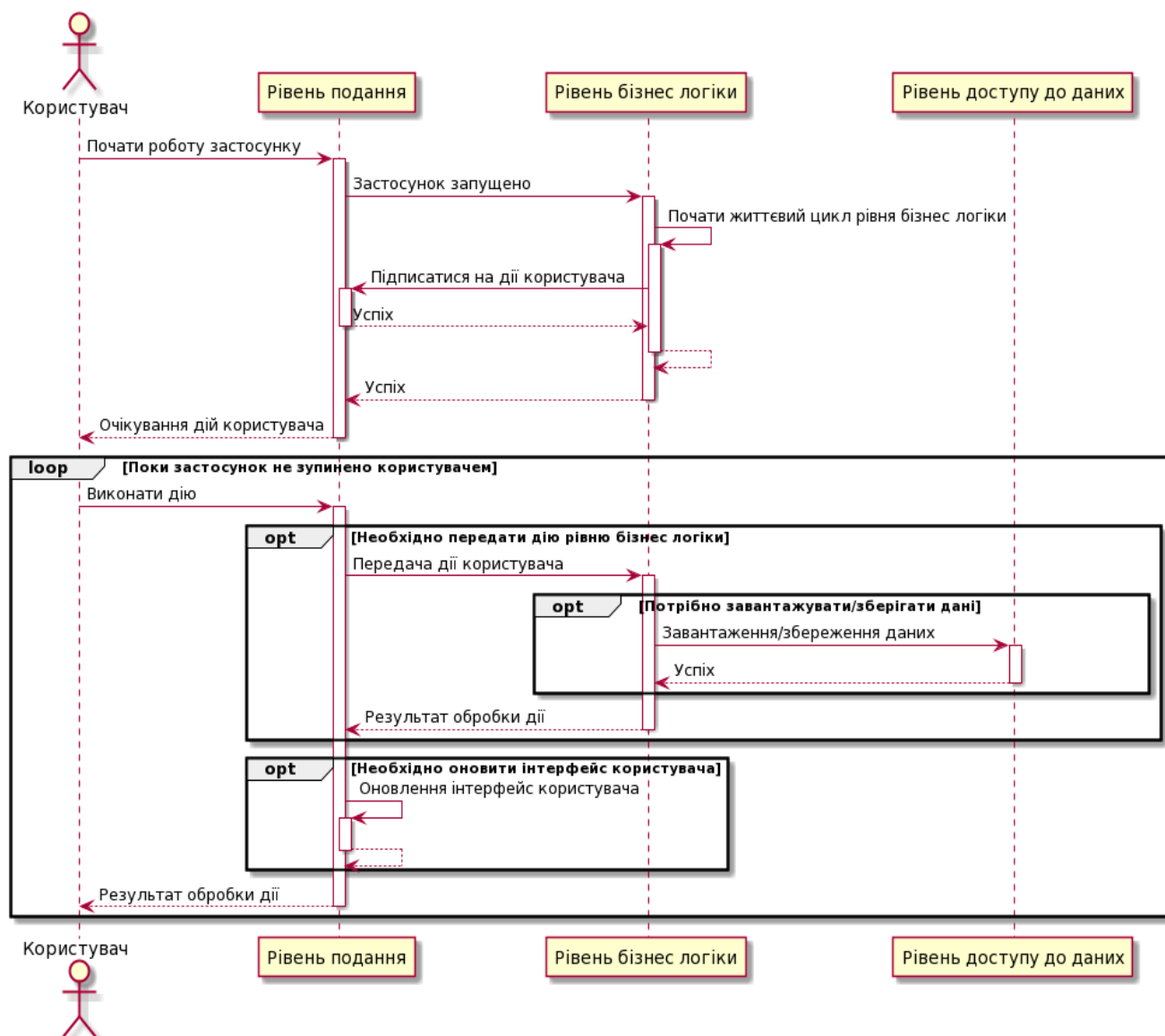


Рисунок 4.12. Діаграма послідовності взаємодії користувача та компонент застосунку протягом життєвого циклу

2. Налаштування параметрів певного компоненту, що входить до складу прогнозуючої системи:

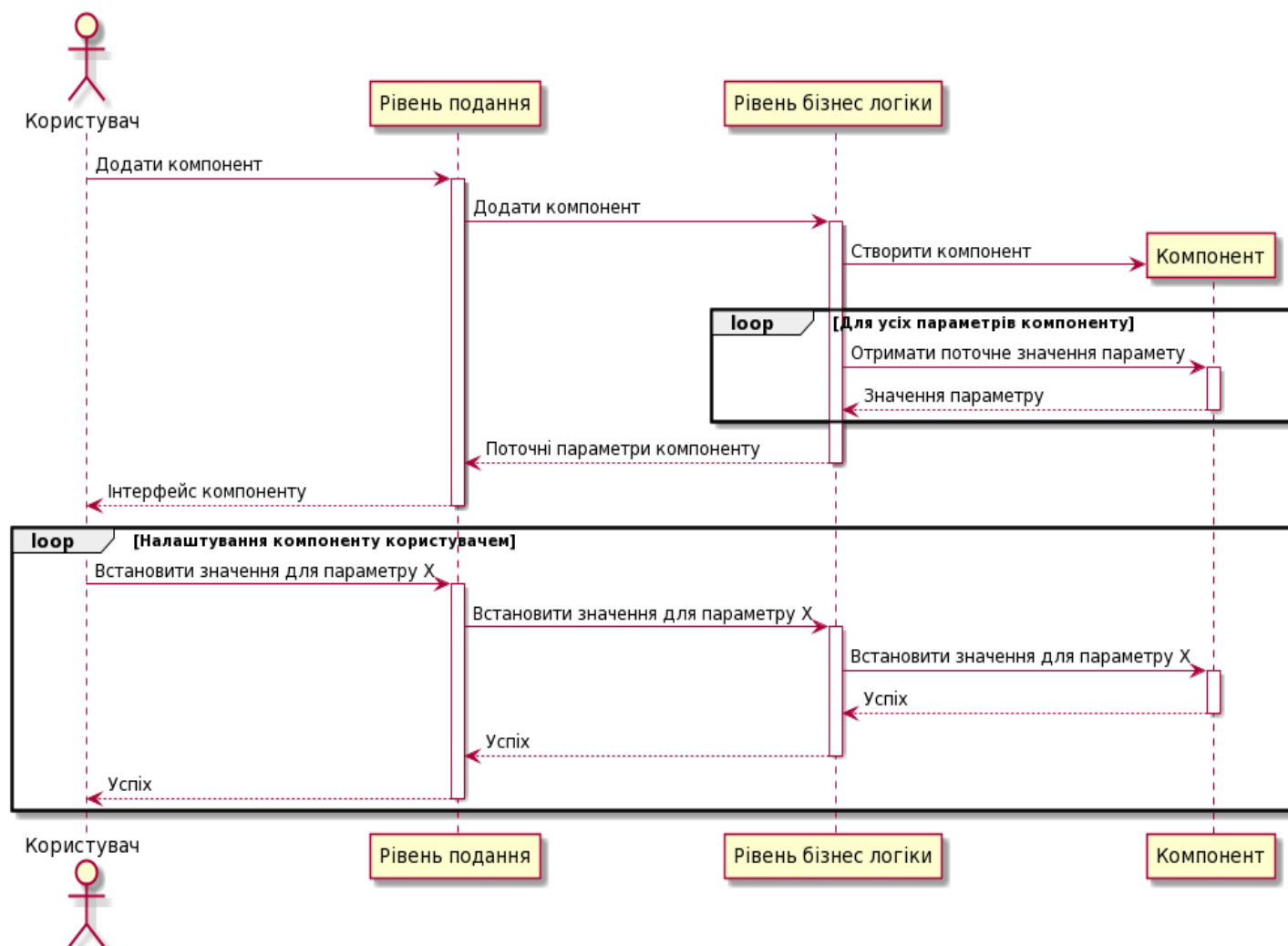


Рисунок 4.13. Діаграма послідовності для сценарію налаштування компоненту прогнозуючої системи

3. Створення графу компонент та його виконання:

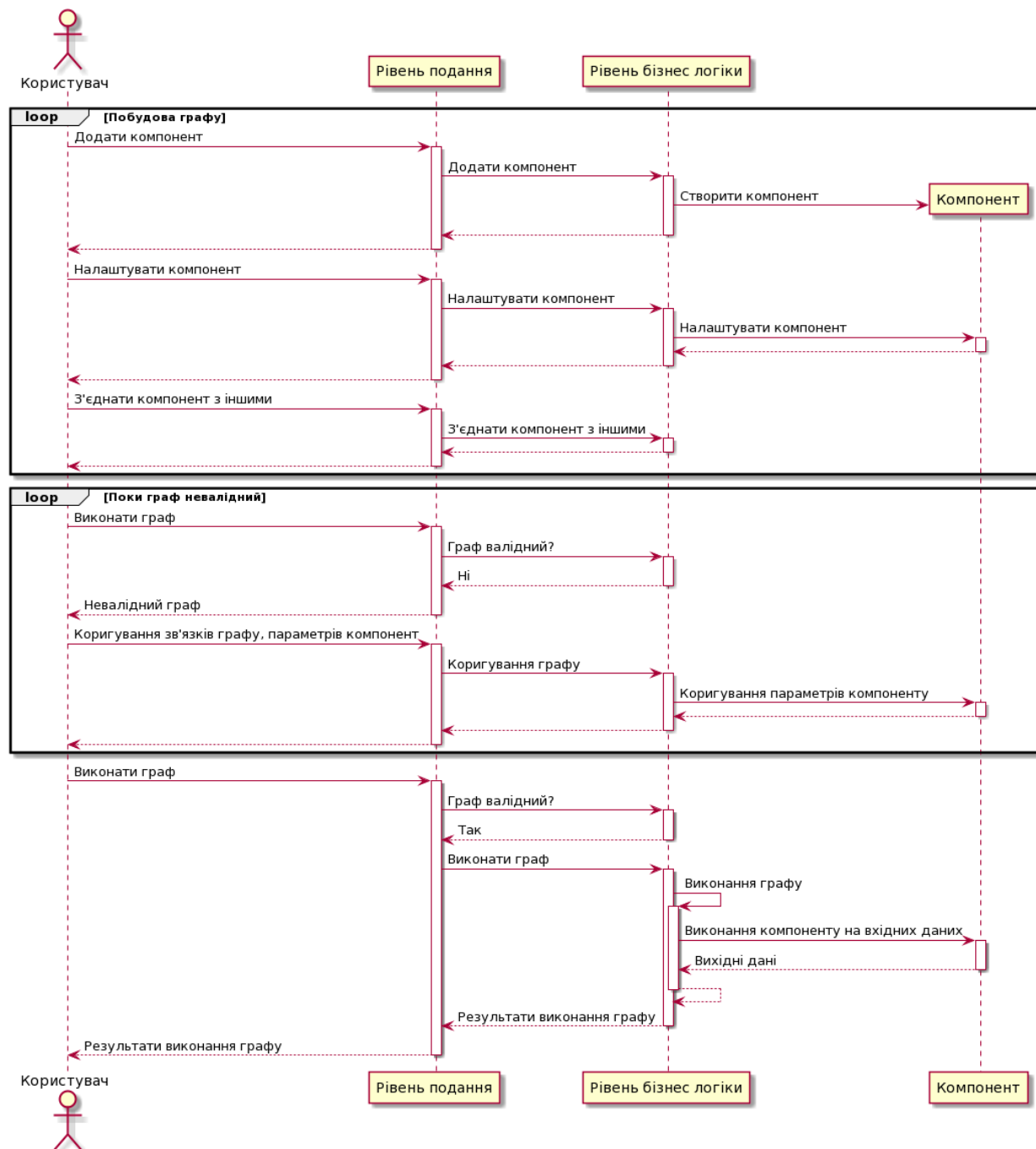


Рисунок 4.14. Діаграма послідовності для сценарію створення та виконання графу компонент

4.4. Інструкція із застосування інформаційної технології для виконання основних поставлених задач

Для більш зручного використання застосунку користувачем в

програмному комплексі ІТ використовується drag-and-drop інтерфейс [117], який дозволяє просто будувати і візуалізувати складні прогнозуючі системи. Завдяки застосуванню drag-and-drop інтерфейсу виконання основних поставлених перед ІТ задач зводиться до побудови певного направленого графу компонент і налаштування їх параметрів.

Задача побудови та налаштування нової прогнозуючої моделі використовуючи певну вибірку даних

Для виконання цієї задачі потрібно спершу завантажити вибірку та візуалізувати її. Для цього необхідно виконати наступні кроки:

- 1) додати до робочої зони компоненти «Завантаження даних/моделі» та «Візуалізація даних»:

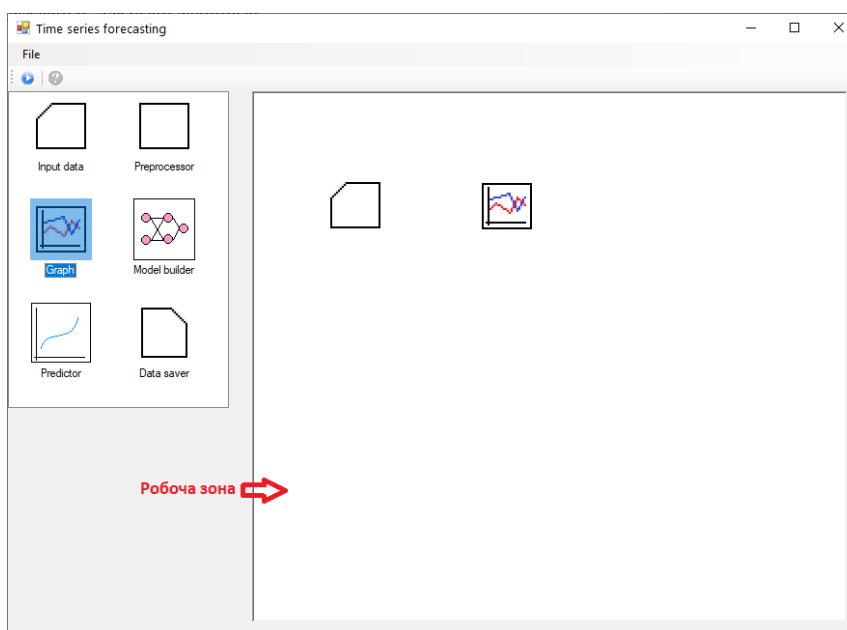


Рисунок 4.15. Робоча зона з доданими компонентами «Завантаження даних/моделі» та «Візуалізація даних»

- 2) налаштувати компонент «Завантаження даних/моделі» вказавши потрібне джерело вхідних даних:

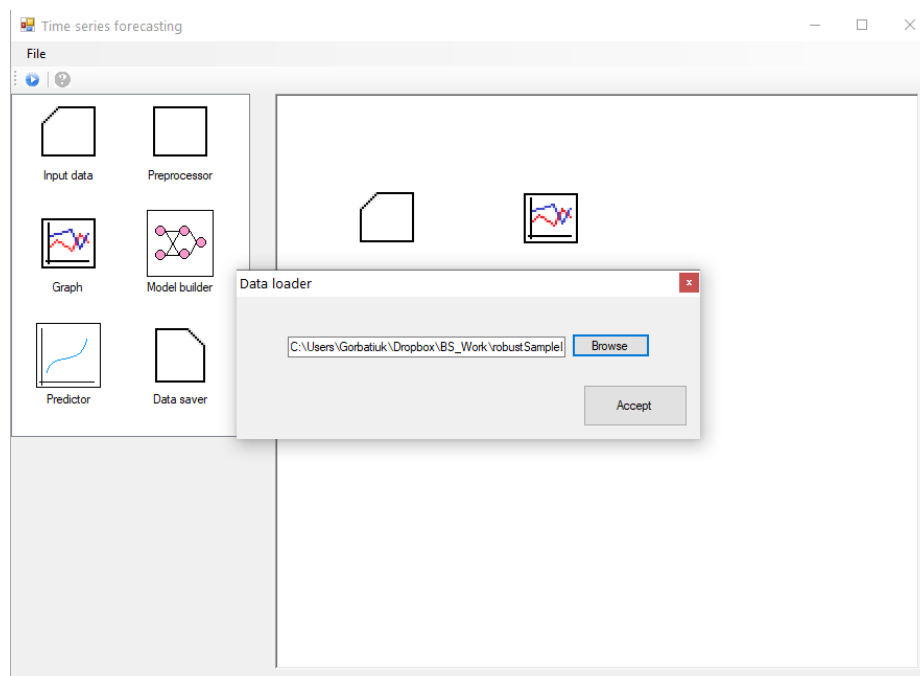


Рисунок 4.16. Налаштування компоненту «Завантаження даних/моделі»

3) поєднати компоненти та запустити поточний граф на виконання:

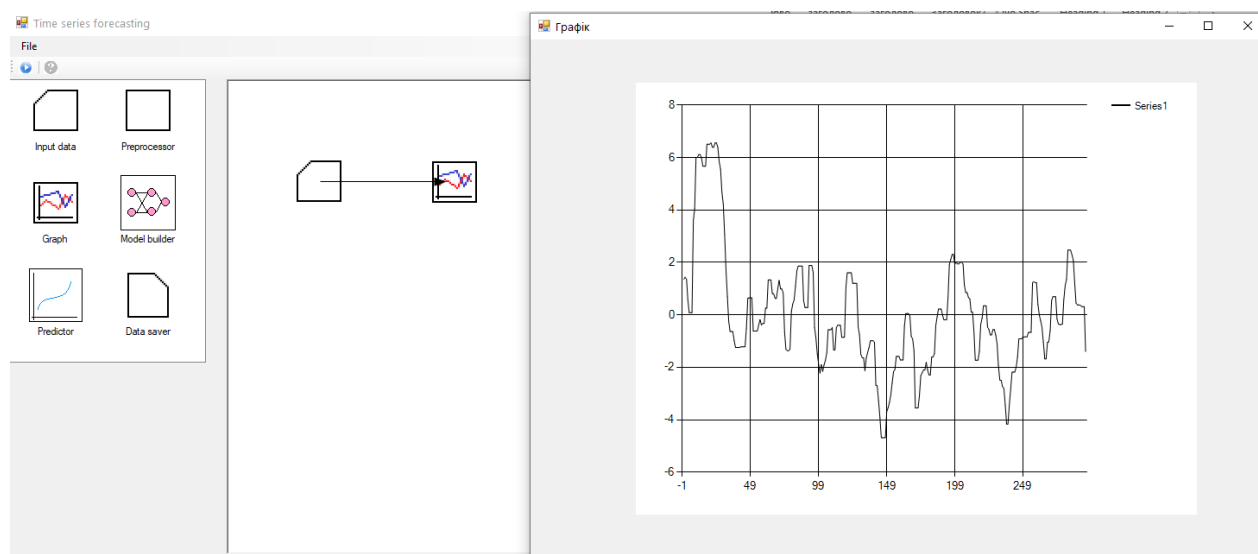


Рисунок 4.17. Результат виконання графу «Завантаження даних/моделі»

—> «Візуалізація даних»

4) за потреби – до графу можна додати компонент «Попередня обробка даних», який може виконувати фільтрування, згладжування, перетворення даних і так далі, після чого запустити граф на виконання

натиснувши на кнопку «Виконати»:

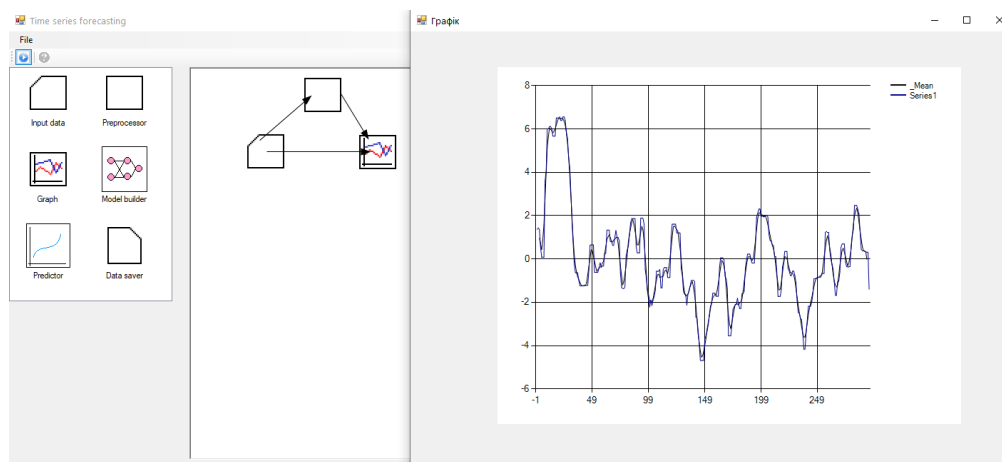


Рисунок 4.18. Використання компоненту «Попередня обробка даних»

Після того, як вхідні дані завантажені та оброблені, їх можна використовувати для налаштування параметрів прогнозуючої моделі. Для цього потрібно:

- 1) додати та налаштувати компонент «Побудова моделі»:

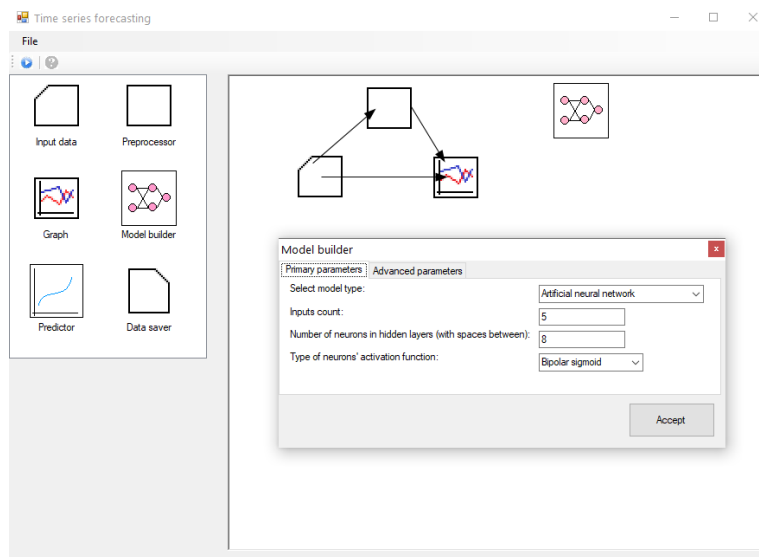


Рисунок 4.19. Налаштування компоненту «Побудова моделі»

- 2) додати компоненти «Використання моделі» і «Візуалізація даних» та поєднати їх з рештою компонент — компонент «Використання моделі» приймає на вхід прогнозуючу модель з компоненту «Побудова моделі» та

вхідні дані для прогнозу з компоненту «Вхідні дані»:

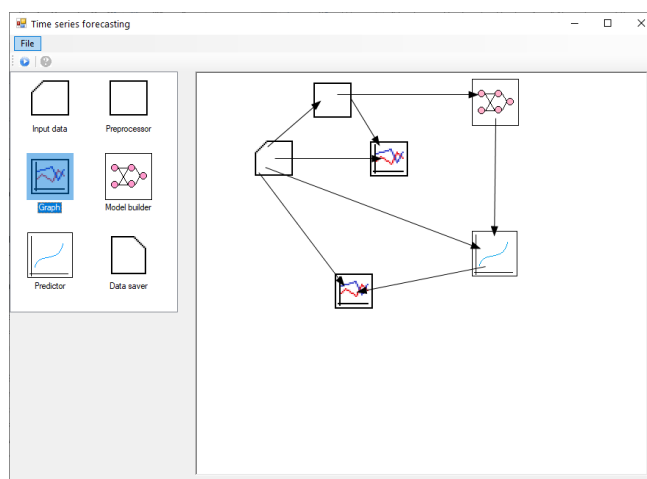


Рисунок 4.20. Фінальний граф разом з доданими компонентами
«Використання моделі» та «Візуалізація даних»

3) запустити граф на виконання та отримати результати використання побудованої прогнозуючої моделі:

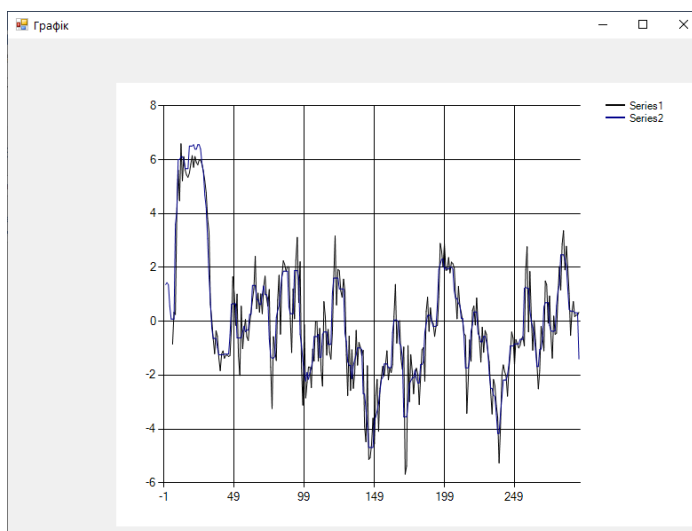


Рисунок 4.21. Результат використання побудованої прогнозуючої моделі

4) Для збереження моделі потрібно додати компонент «Збереження даних/моделі» та вказати шлях у файловій системі ОС, де буде збережена модель, поєднати його з компонентом «Побудова моделі» та знову запустити граф на виконання.

У результаті виконання описаних кроків буде отримана налаштована прогноуюча модель, яку можна використовувати для отримання прогнозів на нових даних.

Задача використання збереженої моделі для прогнозування нових даних

Для використання збереженої моделі для отримання прогнозу на нових даних достатньо побудувати простий граф: один компонент «Завантаження даних/моделі» для завантаження нових даних, один компонент «Завантаження даних/моделі» для завантаження збереженої моделі, компонент «Використання моделі» для отримання прогнозу моделі на нових даних та компонент «Збереження даних/моделі» для збереження результатів прогнозу; опціонально можна додати компоненти «Попередня обробка даних» для виконання необхідної обробки нових даних та «Візуалізація даних» для побудови графіків результату прогнозу:

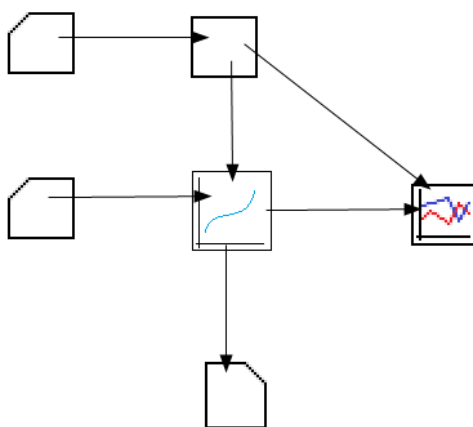


Рисунок 4.22. Приклад графу компонент, який виконує задачу використання збереженої моделі для прогнозування на нових даних

Після належного налаштування компонент наведеного графу та його виконання, будуть завантажені нові дані для прогнозу та збережена прогноуюча модель, нові дані будуть потрібним чином оброблені, а результат

Задача візуалізації та обробки даних

Вибір правильного методу побудови прогнозуючої моделі та їх параметрів потребує попереднього аналізу процесу, що прогнозується, та правильної попередньої обробки даних. Для цього використовують візуалізацію та різні способи трансформації даних. Компоненти «Візуалізація даних» та «Попередня обробка даних» виконують ці функції, і простий граф побудований з їх використанням виглядає наступним чином:

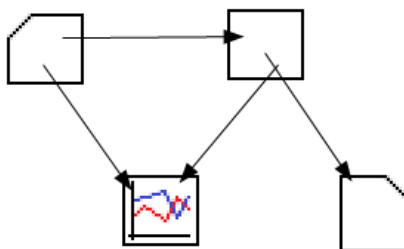


Рисунок 4.24. Приклад графу компонент, який виконує задачу візуалізації та обробки даних

4.5. Впровадження інформаційної технології для прогнозування попиту на лінійку пристроїв мікропроцесорного релейного захисту серій на державному підприємстві «Виробниче об'єднання Київприлад»

Розроблену інформаційну технологію було застосовано для прогнозування попиту на лінійку пристроїв мікропроцесорного релейного захисту серій на державному підприємстві «Виробниче об'єднання Київприлад», що дозволило визначити потребу в комплектуючих та витратних матеріалах і тим самим скоротити витрати (акт впровадження від 12.06.2018). Інформаційна система підприємства, у яку було впроваджено ІТ, відповідає наступним специфікаціям:

1) Система ПНЧР має запускатися на одному ПК з наступними характеристиками:

a) Процесор – Intel Celeron N3060, оперативна пам’ять – 4 GB DDR4 SDRAM, жорсткий диск розміру 128 GB, GPU – відсутній;

b) Операційна система – Windows XP, встановлено Microsoft .NET Framework 4.5, який необхідно для розгортання IT.

2) Дані для побудови прогнозуючих моделей знаходяться на окремому ПК під управлінням ОС Windows XP, доступ до якого виконується з використанням протоколу NFS.

3) Побудовані прогнозуючі моделі, результати прогнозів та різноманітні візуалізації повинні зберігатися на ПК, де знаходяться дані для побудови прогнозуючих моделей.

Згідно з цими специфікаціями інформаційну технологію було впроваджено з використанням наступної схеми розгортання:

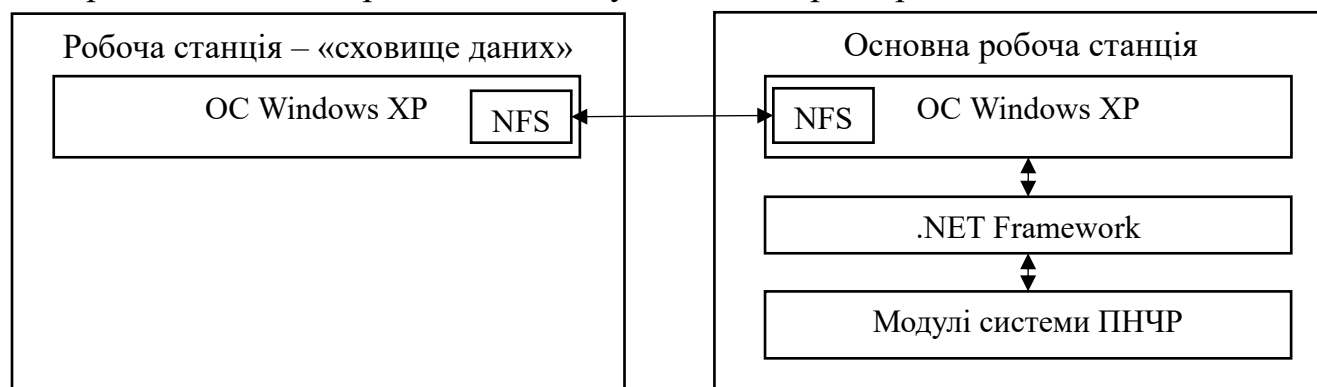


Рисунок 4.25. Схема розгортання IT в інформаційній системі підприємства

Після розгортання, інформаційну технологію було впроваджено для отримання прогнозуючої моделі попиту на лінійку пристроїв мікропроцесорного релейного захисту серій. Для побудови моделі було використано наявні дані у вигляді часового ряду щотижневої кількості проданих пристроїв мікропроцесорного релейного захисту серій за період з 2013 по 2018 рік. Цей ряд з великою ймовірністю є нестационарним, як видно на рис. 4.26.

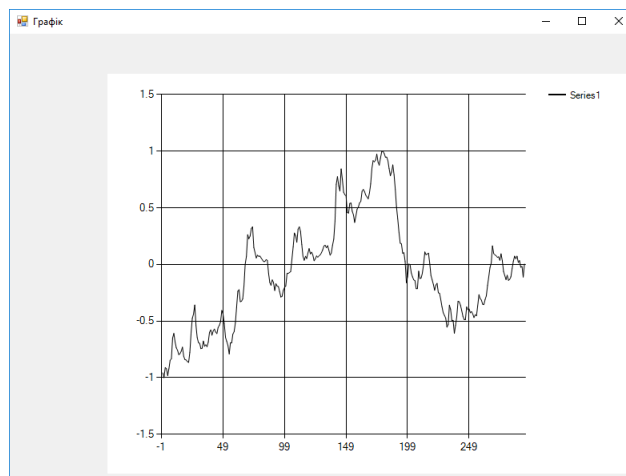


Рисунок 4.26. Графік нормалізованого до діапазону $[-1,1]$ часового ряду кількості проданих пристроїв мікропроцесорного релейного захисту серій за період 01.05.2013–31.12.2018 (перший тиждень відповідає $t = 1$, останній – $t = 295$)

Ряд виглядає як нестационарний з декількома потенційними умовними розподілами, тому доцільно застосувати розроблені методи та алгоритми інформаційної технології. Для попередньої обробки даних використаємо медіанний фільтр, побудувавши наступну схему у програмній системі:

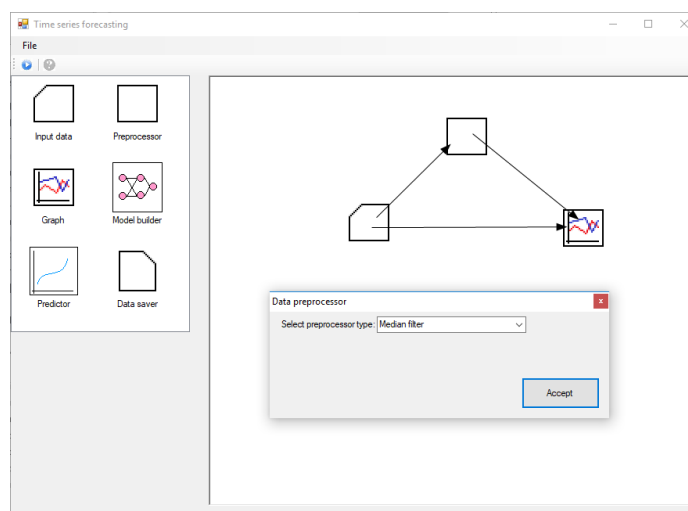


Рисунок 4.27. Побудова схеми для попередньої обробки даних у програмній системі

Виконавши схему, отримаємо наступний результат:

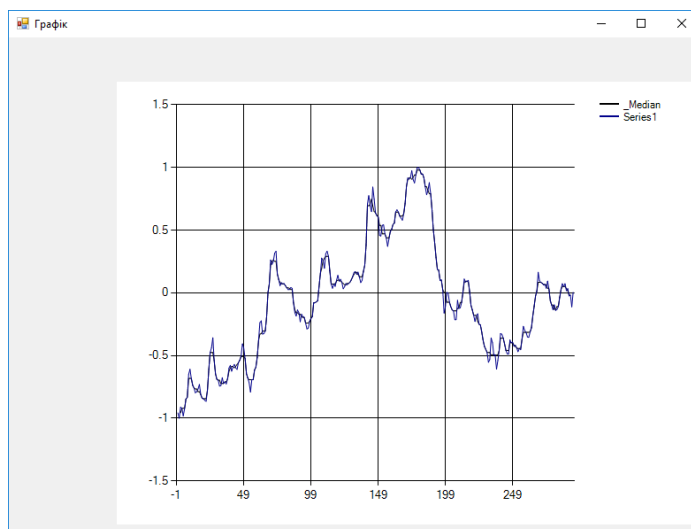


Рисунок 4.28. Результат виконання схеми з попередньою обробкою даних в програмній системі

Розіб'ємо усі наявні дані на навчальну та тестову вибірки та побудуємо декілька різних прогнозуючих моделей у розробленій програмній системі:

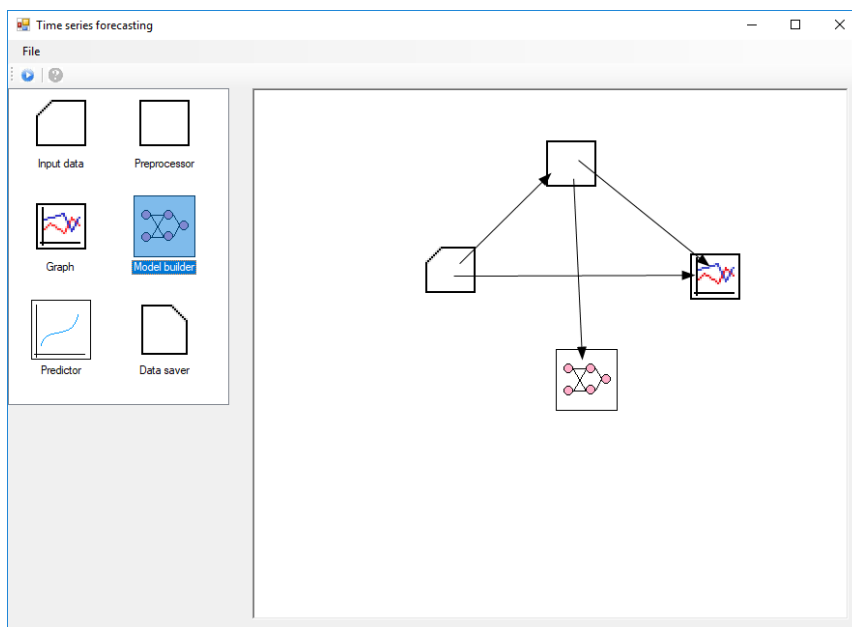


Рисунок 4.29. Схема для побудови різних прогнозуючих моделей з використанням різних методів та алгоритмів в розробленій програмній системі

Побудуємо декілька прогнозуючих моделей використовуючи наступні методи та алгоритми:

- ARIMA;
- багаторядний алгоритм МГУА;
- нейронна мережа з одним прихованим шаром з нейронів ReLU;
- нейронна мережа з одним прихованим шаром з SP нейронів;
- кластеризація навчального часового ряду та побудова нейронної мережі з одним прихованим шаром з SP нейронів для кожного кластеру;
- метод регуляризованої комбінації експертів з використанням нейронних мереж з одним прихованим шаром з SP нейронів для локальних експертів та моделі, що комбінує експертів.

Різноманітні гіперпараметри усіх методів та алгоритмів були обрані згідно процедур, описаних у методах/алгоритмах, та перебору серед декількох рекомендованих значень. Після налаштування відповідних прогнозуючих моделей, отримано значення їх помилок на наступній тестовій вибірці:

Таблиця 4.1.

Помилки моделей на тестовій вибірці

Модель	Середня квадратична помилка на нормованих даних	Середня абсолютна помилка на нормованих даних
ARIMA	0.0107	0.1043
Багаторядний алгоритм МГУА	0.0133	0.1157
ReLU нейронна мережа	0.0172	0.1315
SP нейронна мережа	0.0099	0.1011
Кластеризація з побудовою локальних SP нейронних мереж	0.0095	0.0992
Регуляризована комбінація експертів з використанням SP нейронних мереж	0.0087	0.0925

Для візуалізації прогнозів різних моделей на тестовій вибірці складемо наступну схему в програмній системі:

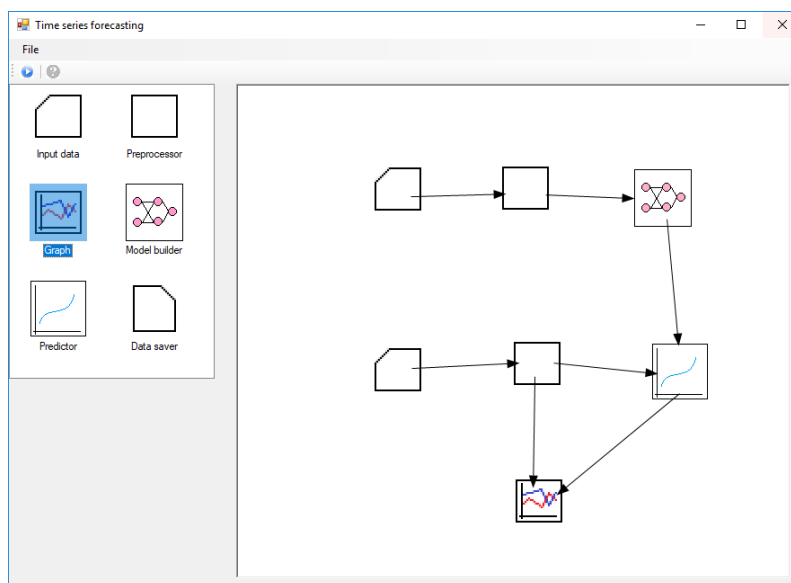


Рисунок 4.30. Схема для візуалізації прогнозів різних моделей на тестовій вибірці

Отримаємо наступні графіки:

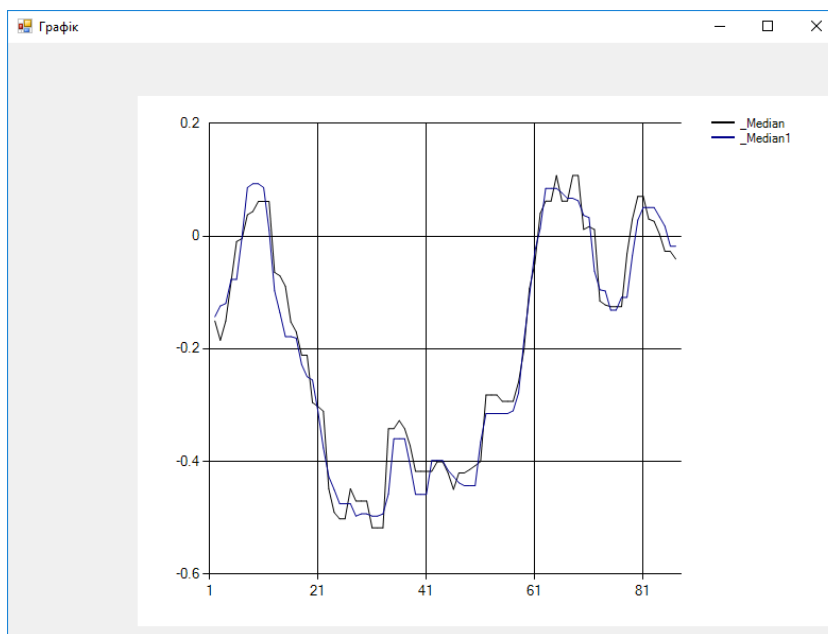


Рисунок 4.31. Тестові дані (синій графік) та прогноз ReLU мережі (чорний графік)

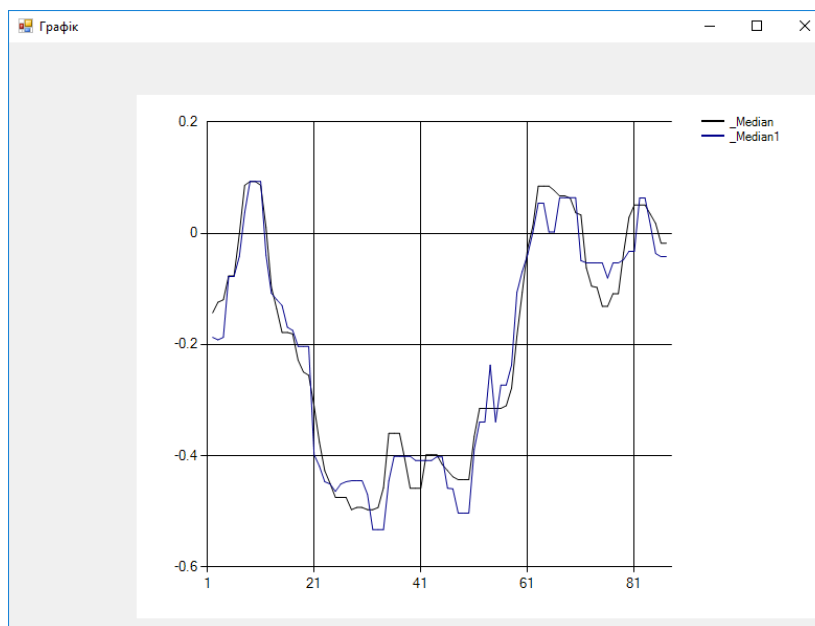


Рисунок 4.32. Тестові дані (чорний графік) та прогноз багаторядного алгоритму МГУА (синій графік)

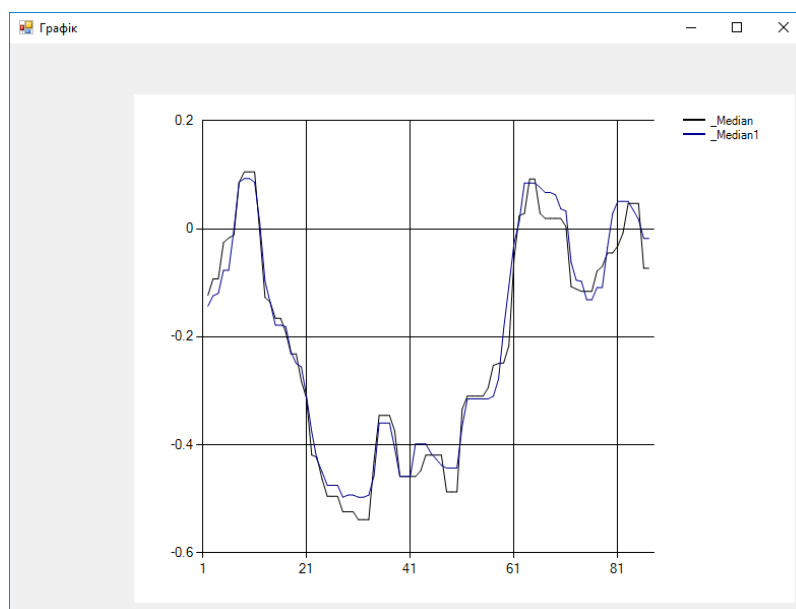


Рисунок 4.33. Тестові дані (синій графік) та прогноз ARIMA (чорний графік)

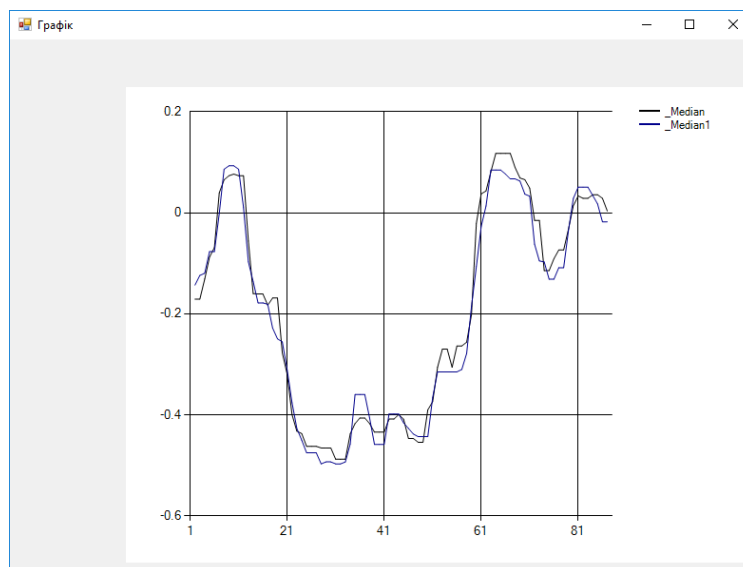


Рисунок 4.34. Тестові дані (синій графік) та прогноз SP мережі (чорний графік)

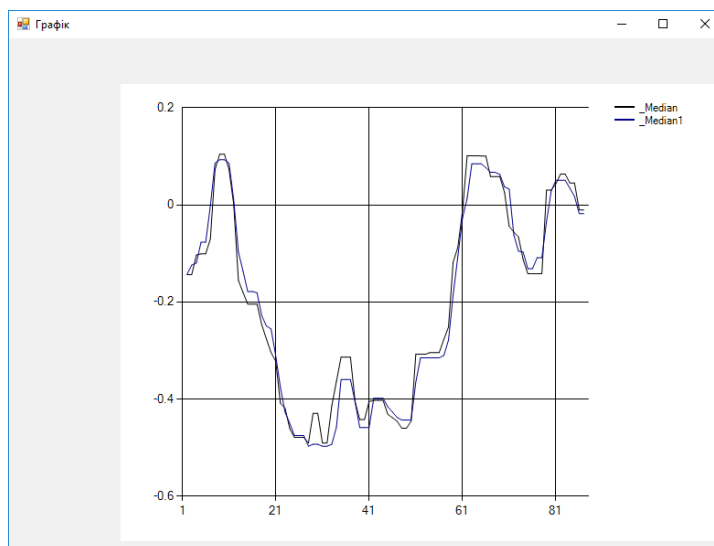


Рисунок 4.35. Тестові дані (синій графік) та прогноз моделі, отриманої шляхом застосування кластеризації та побудови локальних SP мереж (чорний графік)

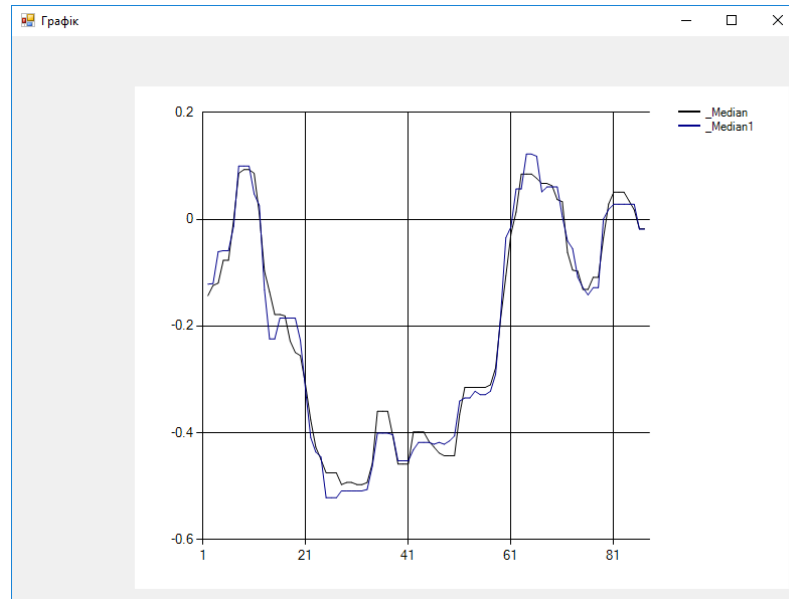


Рисунок 4.36. Тестові дані (чорний графік) та прогноз ReLU мережі (синій графік)

Отже, застосування розробленої інформаційної технології дозволило побудувати прогнозуючу модель, яка на тестовій вибірці мала середню абсолютну помилку менше на 11%, ніж у найближчого методу – ARIMA.

Висновки

- На основі розглянутих вимог та сценаріїв використання розроблено архітектуру інформаційної системи нової інформаційної технології: структурну схему, схеми реалізації окремих компонент. Архітектура інформаційної системи ІТ базується на класичній трирівневій архітектурі програмних застосунків «Рівень подання» → «Рівень бізнес-логіки» → «Рівень доступу до даних», що дозволяє значно спростити кінцевий застосунок та забезпечити вимоги надійності та гнучкості.
- Розроблено програмну систему прогнозування нестационарних часових рядів, що використовує створені методи та моделі прогнозування часових рядів та розглянуто приклади його застосування для побудови

прогнозуючих моделей і систем для реальних вибірок. За рахунок використання синтезованих методів та моделей, що враховують властивості нестационарних часових рядів з декількома потенційними умовними розподілами, програмна реалізація ІТ дозволяє отримувати точніший прогноз для рядів з цього класу.

- Розроблену інформаційну технологію було впроваджено для задачі прогнозування попиту на лінійку пристроїв мікропроцесорного релейного захисту серій на державному підприємстві «Виробниче об'єднання Київприлад», що дозволило побудувати прогнозуючу модель з середньою абсолютною похибкою прогнозу на 11% менше, ніж у найкращого існуючого методу, що брав участь у порівнянні, у такий спосіб дозволивши точніше визначити потребу в комплектуючих та витратних матеріалах і тим самим скоротити витрати.

ВИСНОВКИ

Дисертаційна робота становить собою закінчене наукове дослідження, що вирішує актуальну науково–технічну задачу розробки нових методів, моделей та інформаційної технології прогнозування нестационарних часових рядів з використанням нейронних мереж. Основні досягнуті результати:

1. Визначено клас нестационарних часових рядів з декількома потенційними умовними розподілами, досліджено його властивості, а також переваги і недоліки існуючих методів прогнозування при їх застосуванні до рядів з визначеного класу.

2. Вдосконалено багаторядний алгоритм МГУА шляхом використання методів зворотного поширення помилки та вибіркового виключення, що дає змогу оптимальніше налаштовувати параметри прогнозуючої моделі.

3. Розроблено нову модель штучного нейрону Sigmoid Piecewise (SP), що складається з трьох зважених суматорів та нової функції активації, яка залежить від трьох змінних, що дозволяє отримати модель, яка апроксимує певну просту кусково–лінійну функцію і при цьому дозволяє незалежно налаштовувати параметри гіперплощини, яка задає півпростори з різними лінійними функціями. За рахунок цих відмінностей модель краще підходить для прогнозування рядів з визначеного класу.

4. Вдосконалено метод м'якої кластеризації шляхом використання моделі роздільних гіперповерхонь, що дозволяє вирішувати задачу кластеризації для певного класу критеріїв як задачу оптимізації диференційованої функції. При цьому кількість налаштовуваних параметрів не залежить від кількості прикладів, що дає можливість обмежити кількість параметрів за необхідності.

5. Розроблено новий метод прогнозування рядів з визначеного класу, що використовує вдосконалений метод м'якої кластеризації, SP мережі та спеціальну регуляризацию; вдосконалено метод комбінації експертів шляхом застосування SP мереж та аналогічної регуляризаци, що підвищує стійкість експертів до перенавчання.

6. Проведено експериментальні тести розроблених та вдосконалених моделей та методів. Перший тест досліджував ефективність використання SP нейронів для прогнозування реальних вибірок, і показав, що для досягнення певного значення помилки достатньо було використати в 2 рази менше параметрів в SP мережі, ніж в ReLU мережі. У другому тесті досліджувалися помилки прогнозу мереж з SP та ReLU нейронами, і сумарна помилка мереж з SP нейронами на 405 тестових вибірках була на 10% меншою за відповідну сумарну помилку мереж з ReLU нейронами. У третьому тесті порівнювалися новий метод прогнозування на основі кластеризації і побудови локальних моделей, вдосконалений метод комбінації експертів та існуючі методи на штучних та реальних даних. Згідно з результатами тесту, новий та вдосконалений метод досягли зменшення помилки прогнозу на 19% на штучних і на 9% на реальних даних по відношенню до існуючих методів.

7. Створено інформаційну технологію для прогнозування рядів з визначеного класу на основі розроблених і вдосконалених методів та моделей. IT було впроваджено для прогнозування попиту на лінійку пристроїв мікропроцесорного релейного захисту серій на державному підприємстві «Виробниче об'єднання Київприлад», що дозволило побудувати прогнозуючу модель з похибкою прогнозу на тестовій вибірці на 11% менше, ніж у найкращого існуючого методу, за рахунок чого точніше оцінено потребу в комплектуючих та витратних матеріалах і скорочено витрати.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Sineglazov, Victor, Elena Chumachenko, and Vladyslav Gorbatiuk. "An algorithm for solving the problem of forecasting." *Aviation* 17, no. 1 (2013): 9–13.
2. Sineglazov, V., Olena Chumachenko, and Vladyslav Gorbatiuk. "A method for building a forecasting model with dynamic weights." *Восточно–Европейский журнал передовых технологий* 2 (4) (2014): 4–8.
3. Sineglazov, Victor, Elena Chumachenko, and Vladyslav Gorbatiuk. "Using a mixture of experts' approach to solve the forecasting task." *Aviation* 18, no. 3 (2014): 129–133.
4. Chumachenko, O. I., and V. S. Gorbatiuk. "Soft clustering algorithm based on separating hypersurfaces." *Electronics & Control Systems* 2, no. 52 (2017): 11–15.
5. Чумаченко, Е. И., и В. С. Горбатюк. "Алгоритм решения задачи прогнозирования." *Штучний інтелект* (2012): 24–31.
6. Синеглазов, В. М., Е. И. Чумаченко, и В. С. Горбатюк. "Метод решения задачи прогнозирования на основе комплексирования оценок." *Індуктивне моделювання складних систем* 4 (2012): 214–223.
7. Чумаченко, Е. И., и В. С. Горбатюк. "Использование искусственных нейронных сетей для задачи прогнозирования." *Електроніка та системи управління* 31 (2012): 113–119.
8. Чумаченко, Олена Іллівна, та Владислав Сергійович Горбатюк. "Комплексування декількох алгоритмів під час розв'язання задачі прогнозування." *Вісник Житомирського державного технологічного університету. Серія: Технічні науки* 1 (76) (2016): 101–107.
9. Doob, Joseph L., and Joseph L. Doob. *Stochastic processes*. Vol. 7, no.

2. New York: Wiley, 1953.

10. Parzen, Emanuel. *Stochastic processes*. Society for Industrial and Applied Mathematics, 1999.

11. Vapnik, Vladimir. "Principles of risk minimization for learning theory." *Advances in neural information processing systems*. 1992.

12. Drucker, H., Burges, J. C., Kaufman, L., Smola, A. J., Vapnik, V. "Support vector regression machines." *Advances in neural information processing systems*, (1997): 155–161.

13. Chatfield, Chris. *Time-series forecasting*. CRC press, 2000.

14. Уидроу, Бернард. *Адаптивная обработка сигналов*. Радио и связь, 1989.

15. Ibragimov, Ildar A. "Some limit theorems for stationary processes." *Theory of Probability & Its Applications* 7, no. 4 (1962): 349–382.

16. Dahlhaus, Rainer. "Fitting time series models to nonstationary processes." *The annals of Statistics* 25, no. 1 (1997): 1–37.

17. Stein, Michael L. *Interpolation of spatial data: some theory for kriging*. Springer Science & Business Media, 2012.

18. Yahoo Finance 2018, IBM shares historical data, accessed 01.06.2018. Available from:

<https://query1.finance.yahoo.com/v7/finance/download/IBM?period1=252385200&period2=1567026000&interval=1d&events=history&crumb=Qz1I1HUgxrA>.

19. Yahoo Finance 2018, Dow Jones Industrial Average, accessed 01.06.2018. Available online:

<https://query1.finance.yahoo.com/v7/finance/download/%5EDJI?period1=475794000&period2=1567026000&interval=1d&events=history&crumb=Qz1I1HUgxrA>.

20. National Centers For Enviromental Information, Global Climate Report

– Annual 2015, accessed 01.06.2016. Available from: <https://www.ncdc.noaa.gov/sotc/global/201513>.

21. bin Majid, Shah, Mohd Shahrieel bin Mohd Aras, and Arfah binti Ahmad. "Electricity Load Forecasting Using Data Mining Technique." In *Advances in Data Mining Knowledge Discovery and Applications*. InTech, 2012.

22. Legendre, Adrien Marie. *Nouvelles méthodes pour la détermination des orbites des comètes*. F. Didot, 1805.

23. Fox, John. *Applied regression analysis, linear models, and related methods*. Sage Publications, Inc, 1997.

24. Savitzky, Abraham, and Marcel JE Golay. "Smoothing and differentiation of data by simplified least squares procedures." *Analytical chemistry* 36, no. 8 (1964): 1627–1639.

25. Gurland, John. "Hypothesis Testing in Time Series Analysis." (1954): 197–200.

26. Box, George EP, and Gwilym M. Jenkins. *Time series analysis: forecasting and control, revised ed*. Holden–Day, 1976.

27. Руденко, О. Г., and Є. В. Бодянський. "Штучні нейронні мережі." *Харків: Компанія CMIT* (2006).

28. Bebis, George, and Michael Georgiopoulos. "Feed–forward neural networks." *IEEE Potentials* 13, no. 4 (1994): 27–31.

29. Rosenblatt, Frank. "The perceptron: a probabilistic model for information storage and organization in the brain." *Psychological review* 65, no. 6 (1958): 386.

30. Rumelhart, David E., Geoffrey E. Hinton, and Ronald J. Williams. "Learning representations by back–propagating errors." *nature* 323, no. 6088 (1986): 533.

31. Wan, Eric A. "Temporal backpropagation for FIR neural networks." In

Neural Networks, 1990., 1990 IJCNN International Joint Conference on, pp. 575–580. IEEE, 1990.

32. Wan, Eric A. "Time series prediction by using a connectionist network with internal delay lines." In *SANTA FE INSTITUTE STUDIES IN THE SCIENCES OF COMPLEXITY–PROCEEDINGS VOLUME–*, vol. 15, pp. 195–195. Addison–Wesley publishing co, 1993.

33. Zhang, G. P., Min, Q. "Neural network forecasting for seasonal and trend time series." *European journal of operational research* 160, no. 2 (2005): 501–514.

34. LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. "Deep learning." *nature* 521, no. 7553 (2015): 436.

35. Hinton, Geoffrey E. "Deep belief networks." *Scholarpedia* 4, no. 5 (2009): 5947.

36. Medsker, L. R., and L. C. Jain. "Recurrent neural networks." *Design and Applications* 5 (2001).

37. Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." *Neural computation* 9, no. 8 (1997): 1735–1780.

38. Connor, Jerome T., R. Douglas Martin, and Les E. Atlas. "Recurrent neural networks and robust time series prediction." *IEEE transactions on neural networks* 5, no. 2 (1994): 240–254.

39. He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification." In *Proceedings of the IEEE international conference on computer vision* (2016): 1026–1034.

40. He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, (2016): 770–778.

41. Girshick, Ross, Jeff Donahue, Trevor Darrell, and Jitendra Malik. "Rich feature hierarchies for accurate object detection and semantic segmentation." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587. 2014.
42. Shaham, Uri, Kelly Stanton, Henry Li, Boaz Nadler, Ronen Basri, and Yuval Kluger. "Spectralnet: Spectral clustering using deep neural networks." :1801.01587 (2018).
43. Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." :1409.0473 (2014).
44. Bahdanau, Dzmitry, Jan Chorowski, Dmitriy Serdyuk, Philemon Brakel, and Yoshua Bengio. "End-to-end attention-based large vocabulary speech recognition." In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pp. 4945–4949. IEEE, 2016.
45. Mnih, Volodymyr, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. "Playing atari with deep reinforcement learning." :1312.5602 (2013).
46. Ивахненко, Алексей Григорьевич. *Долгосрочное прогнозирование и управление сложными системами*. Техніка, 1975.
47. Ivakhnenko, Alexey. "Polynomial theory of complex systems". *IEEE Transactions on Systems, Man and Cybernetics*. SMC–1 4 (1971): 364–378.
48. Fourier, J., 1808. Mémoire sur la propagation de la chaleur dans les corps solides. *Nouveau Bulletin des Sciences de la Société Philomathique de Paris*, 6, pp.112–116.
49. Haar, Alfred. "Zur theorie der orthogonalen funktionensysteme." *Mathematische Annalen* 69, no. 3 (1910): 331–371.
50. Schuster-Böckler, Benjamin, and Alex Bateman. "An introduction to hidden Markov models." *Current protocols in bioinformatics* 18, no. 1 (2007): A–

3A.

51. Gardiner, Crispin. *Stochastic methods*. Vol. 4. Berlin: springer, 2009.
52. Welch, L.R., 2003. Hidden Markov models and the Baum–Welch algorithm. *IEEE Information Theory Society Newsletter*, 53(4), pp.10–13.
53. Schmidhuber, Jürgen. "Deep learning in neural networks: An overview." *Neural networks* 61 (2015): 85–117.
54. Юрачковский, Ю. П. "Сходимость многорядных алгоритмов МГУА." *Автоматика* 14, no. 3 (1981): 32–36.
55. Specht, Donald F. "A general regression neural network." *IEEE transactions on neural networks* 2, no. 6 (1991): 568–576.
56. Oh, Sung–Kwun, and Witold Pedrycz. "The design of self–organizing polynomial neural networks." *Information Sciences* 141, no. 3–4 (2002): 237–258.
57. Bengio, Yoshua, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. "Greedy layer–wise training of deep networks." In *Advances in neural information processing systems*, pp. 153–160. 2007.
58. Larochelle, Hugo, Yoshua Bengio, Jérôme Louradour, and Pascal Lamblin. "Exploring strategies for training deep neural networks." *Journal of machine learning research* 10, no. Jan (2009): 1–40.
59. Erhan, Dumitru, Yoshua Bengio, Aaron Courville, Pierre–Antoine Manzagol, Pascal Vincent, and Samy Bengio. "Why does unsupervised pre–training help deep learning?." *Journal of Machine Learning Research* 11, no. Feb (2010): 625–660.
60. Salakhutdinov, Ruslan, and Hugo Larochelle. "Efficient learning of deep Boltzmann machines." In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 693–700. 2010.
61. Hayashi, Fumio. "Econometrics. Princeton." *New Jersey, USA: Princeton University* (2000).

62. Yosinski, Jason, Jeff Clune, Yoshua Bengio, and Hod Lipson. "How transferable are features in deep neural networks?." In *Advances in neural information processing systems*, pp. 3320–3328. 2014.
63. Srivastava, Nitish, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. "Dropout: A simple way to prevent neural networks from overfitting." *The Journal of Machine Learning Research* 15, no. 1 (2014): 1929–1958.
64. Hinton, Geoffrey E., Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R. Salakhutdinov. "Improving neural networks by preventing co-adaptation of feature detectors." :1207.0580 (2012).
65. Anastasakis, L., and N. Mort. "The development of self-organization techniques in modelling: a review of the group method of data handling (GMDH)." *Research Report–University Of Sheffield Department Of Automatic Control And Systems Engineering* (2001).
66. Domhan, Tobias, Jost Tobias Springenberg, and Frank Hutter. "Speeding Up Automatic Hyperparameter Optimization of Deep Neural Networks by Extrapolation of Learning Curves." In *IJCAI*, vol. 15, pp. 3460–8. 2015.
67. Riedmiller, Martin, and Heinrich Braun. "A direct adaptive method for faster backpropagation learning: The RPROP algorithm." In *Neural Networks, 1993., IEEE International Conference on*, pp. 586–591. IEEE, 1993.
68. Boyd, Stephen, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.
69. Delalleau, Olivier, and Yoshua Bengio. "Shallow vs. deep sum-product networks." In *Advances in Neural Information Processing Systems*, pp. 666–674. 2011.
70. Montufar, Guido F., Razvan Pascanu, Kyunghyun Cho, and Yoshua Bengio. "On the number of linear regions of deep neural networks." In *Advances in*

neural information processing systems, pp. 2924–2932. 2014.

71. Ba, Jimmy, and Rich Caruana. "Do deep nets really need to be deep?." In *Advances in neural information processing systems*, pp. 2654–2662. 2014.

72. Maas, Andrew L., Awni Y. Hannun, and Andrew Y. Ng. "Rectifier nonlinearities improve neural network acoustic models." In *Proc. icml*, vol. 30, no. 1, p. 3. 2013.

73. Anthimopoulos, Marios, Stergios Christodoulidis, Lukas Ebner, Andreas Christe, and Stavroula Mougiakakou. "Lung pattern classification for interstitial lung diseases using a deep convolutional neural network." *IEEE transactions on medical imaging* 35, no. 5 (2016): 1207–1216.

74. Goodfellow, Ian J., David Warde-Farley, Mehdi Mirza, Aaron Courville, and Yoshua Bengio. "Maxout networks." :1302.4389 (2013).

75. MacLane, Saunders. "Some interpretations of abstract linear dependence in terms of projective geometry." *American Journal of Mathematics* 58, no. 1 (1936): 236–240.

76. Zaslavsky, Thomas. *Facing up to Arrangements: Face-Count Formulas for Partitions of Space by Hyperplanes: Face-count Formulas for Partitions of Space by Hyperplanes*. Vol. 154. American Mathematical Soc., 1975.

77. Birkhoff, Garrett. "Linear transformations with invariant cones." *The American Mathematical Monthly* 74.3 (1967): 274–276.

78. Goldin, Dina, Ricardo Mardales, and George Nagy. "In search of meaning for time series subsequence clustering: matching algorithms based on a new distance measure." In *Proceedings of the 15th ACM international conference on Information and knowledge management* (2006): 347–356.

79. Keogh, Eamonn, and Jessica Lin. "Clustering of time-series subsequences is meaningless: implications for previous and future research." *Knowledge and information systems* 8, no. 2 (2005): 154–177.

80. Rocach, L., and O. Maimon. "Clustering methods. Data mining and knowledge discovery handbook." *Springer US* (2005): 321.
81. Liu, Song, Makoto Yamada, Nigel Collier, and Masashi Sugiyama. "Change-point detection in time-series data by relative density-ratio estimation." *Neural Networks* 43 (2013): 72–83.
82. Keogh, Eamonn, Selina Chu, David Hart, and Michael Pazzani. "Segmenting time series: A survey and novel approach." In *Data mining in time series databases* (2004): 1–21.
83. Bellman, Richard. "Dynamic programming." *Science* 153, no. 3731 (1966): 34–37.
84. Carlin, Bradley P., Alan E. Gelfand, and Adrian FM Smith. "Hierarchical Bayesian analysis of changepoint problems." *Applied statistics* (1992): 389–405.
85. Keogh, Eamonn, Selina Chu, David Hart, and Michael Pazzani. "An online algorithm for segmenting time series." In *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on*, pp. 289–296. IEEE, 2001.
86. Jackson, Brad, Jeffrey D. Scargle, David Barnes, Sundararajan Arabhi, Alina Alt, Peter Gioumousis, Elyus Gwin, Paungkaew Sangtrakulcharoen, Linda Tan, and Tun Tao Tsai. "An algorithm for optimal partitioning of data on an interval." *IEEE Signal Processing Letters* 12, no. 2 (2005): 105–108.
87. Bai, Jushan. "Estimation of a change point in multiple regression models." *Review of Economics and Statistics* 79, no. 4 (1997): 551–563.
88. MacQueen, James. "Some methods for classification and analysis of multivariate observations." In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, no. 14, pp. 281–297. 1967.
89. Celeux, Gilles, and Gilda Soromenho. "An entropy criterion for assessing the number of clusters in a mixture model." *Journal of classification* 13,

no. 2 (1996): 195–212.

90. Cao, Liangyue. "Practical method for determining the minimum embedding dimension of a scalar time series." *Physica D: Nonlinear Phenomena* 110, no. 1–2 (1997): 43–50.

91. Makhzani, Alireza, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. "Adversarial autoencoders." *1511.05644* (2015).

92. Bezdek, James C., Robert Ehrlich, and William Full. "FCM: The fuzzy c-means clustering algorithm." *Computers & Geosciences* 10, no. 2–3 (1984): 191–203.

93. Parker, R. Gary, and Ronald L. Rardin. *Discrete optimization*. Elsevier, 2014.

94. Brucker, Peter. "On the complexity of clustering problems." *Optimization and operations research* (1978): 45–54.

95. Mahajan, Meena, Prajakta Nimbhorkar, and Kasturi Varadarajan. "The planar k-means problem is NP-hard." In *International Workshop on Algorithms and Computation*, pp. 274–285. Springer, Berlin, Heidelberg, 2009.

96. Hartigan, John A. "Clustering algorithms." (1975).

97. Ruder, Sebastian. "An overview of gradient descent optimization algorithms." (2016).

98. Gath, Isak, and Amir B. Geva. "Unsupervised optimal fuzzy clustering." *IEEE Transactions on pattern analysis and machine intelligence* 11, no. 7 (1989): 773–780.

99. Перова, И. Г., and Е. В. Бодянский. "Нечеткая классификация данных медико-биологических исследований в условиях дефицита информации." *Системы обработки информации* 11 (2015): 161–163.

100. Zadeh, Lotfi A. "Fuzzy sets." In *Fuzzy Sets, Fuzzy Logic, And Fuzzy Systems: Selected Papers by Lotfi A Zadeh*, pp. 394–432. 1996.

101. Zadeh, Lotfi A. "Fuzzy logic and approximate reasoning." *Synthese* 30, no. 3–4 (1975): 407–428.
102. Zadeh, Lotfi A. "Fuzzy logic." *Computer* 21.4 (1988): 83–93.
103. Jacobs, R A., Michael I. J., Steven J. N., Geoffrey E. H. "Adaptive mixtures of local experts." *Neural computation* 3, no. 1 (1991): 79–87.
104. Engl, H. W. "Discrepancy principles for Tikhonov regularization of ill-posed problems leading to optimal convergence rates." *Journal of optimization theory and applications* 52, no. 2 (1987): 209–215.
105. Engl, Heinz W., and Wilhelm Grever. "Using the L—curve for determining optimal regularization parameters." *Numerische Mathematik* 69, no. 1 (1994): 25–31.
106. Neubauer, Andreas. "An a posteriori parameter choice for Tikhonov regularization in Hilbert scales leading to optimal convergence rates." *SIAM journal on numerical analysis* 25, no. 6 (1988): 1313–1326.
107. Hawkins, D. M. "The problem of overfitting." *Journal of chemical information and computer sciences* 44.1 (2004): 1–12.
108. Nasrabadi, Nasser M. "Pattern recognition and machine learning." *Journal of electronic imaging* 16, no. 4 (2007): 049901.
109. Federal Reserve Bank of St. Louis, 10–Year Treasury Constant Maturity Minus 2–Year Treasury Constant Maturity, accessed 05.06.2019. Available online: <https://fred.stlouisfed.org/series/T10Y2Y>.
110. Rohloff, Kurt, and Richard E. Schantz. "High–performance, massively scalable distributed systems using the MapReduce software framework: the SHARD triple–store." In *Programming support innovations for emerging distributed applications* (2010): 1–5.
111. Garlan, David, and Mary Shaw. "An introduction to software architecture." In *Advances in software engineering and knowledge engineering*, pp.

1–39. 1993.

112. Krasner, Glenn E., and Stephen T. Pope. "A description of the model–view–controller user interface paradigm in the smalltalk–80 system." *Journal of object oriented programming* 1, no. 3 (1988): 26–49.

113. Shao, Gary, Francine Berman, and Richard Wolski. "Master/slave computing on the grid." In *Proceedings 9th Heterogeneous Computing Workshop (HCW 2000)*(Cat. No. PR00556), pp. 3–16. IEEE, 2000.

114. Szyperski, Clemens, Jan Bosch, and Wolfgang Weck. "Component–oriented programming." In *European Conference on Object–Oriented Programming*, pp. 184–192. Springer, Berlin, Heidelberg, 1999.

115. Batory, Don, and Sean O'malley. "The design and implementation of hierarchical software systems with reusable components." *ACM Transactions on Software Engineering and Methodology (TOSEM)* 1.4 (1992): 355–398.

116. Prosise, Jeff. *Programming Microsoft. NET*. Microsoft Press, 2002.

117. Inkpen, Kori M. "Drag–and–drop versus point–and–click mouse interaction styles for children." *ACM Transactions on Computer–Human Interaction (TOCHI)* 8, no. 1 (2001): 1–33.

ДОДАТОК А

Список публікацій здобувача за темою дисертації

Монографії:

1. Интеллектуальные методы прогнозирования [Текст]: монография / В. М. Синеглазов, Е. И. Чумаченко, В. С. Горбатюк. – К.: Освіта України, 2013. – 219 с. *(Особистий внесок: аналіз проблеми прогнозування нестационарних часових рядів та існуючих методів її вирішення).*

У виданнях іноземних держав–членів ЄС:

2. An algorithm for solving the problem of forecasting / Victor Sineglazov, Elena Chumachenko, Vladyslav Gorbatyuk // Aviation. – 2013. №1(17), p. 9–13. (Видання іноземної держави–члена ЄС, Литва. Входить до міжнародних наукометричних баз ESCI, DOAJ, EBSCOhost, Elsevier Bibliographic Databases: SCOPUS, El Compendex Gale, Google Scholar, ICONDA, Microsoft Academic, MyScienceWork, OpenAIRE2020, ProQuest). *(Особистий внесок: розробка вдосконаленої моделі прогнозування на основі нейронних мереж та методу групового урахування аргументів).*

3. Using a mixture of experts' approach to solve the forecasting task / Victor Sineglazov, Elena Chumachenko, Vladyslav Gorbatyuk // Aviation. – 2014. №3(18), p. 129–133. (Видання іноземної держави–члена ЄС, Литва. Входить до міжнародних наукометричних баз ESCI, DOAJ, EBSCOhost, Elsevier Bibliographic Databases: SCOPUS, El Compendex Gale, Google Scholar, ICONDA, Microsoft Academic, MyScienceWork, OpenAIRE2020, ProQuest). *(Особистий внесок: розробка спеціального регуляризуючого критерію, що застосовується для побудови локальних моделей).*

4. Forecasting aircraft miles flown time series using deep learning–based hybrid approach / Victor Sineglazov, Elena Chumachenko, Vladyslav Gorbatyuk //

Aviation. – 2018. № 1(22), p. 6–12. (Видання іноземної держави–члена ЄС, Литва. Входить до міжнародних наукометричних баз ESCI, DOAJ, EBSCOhost, Elsevier Bibliographic Databases: SCOPUS, El Compendex Gale, Google Scholar, ICONDA, Microsoft Academic, MyScienceWork, OpenAIRE2020, ProQuest). *(Особистий внесок: розробка вдосконаленого багаторядного алгоритму методу групового урахування аргументів).*

5. An algorithm for the corporate shares price forecasting / V. Sineglazov, O. Chumachenko, V. Gorbatiuk // Information Technology in Selected Areas of Management 2017. – 2018, p. 73–79. (Видання іноземної держави–члена ЄС, Польща. Входить до міжнародної наукометричної бази Scopus, Google Scholar). *(Особистий внесок: розробка загального підходу до побудови моделі).*

У фахових наукових виданнях України:

6. Soft clustering algorithm based on separating hypersurfaces / O. I. Chumachenko, V. S. Gorbatiuk // Electronics and control systems. – 2017. Volume 2, № 52, p. 11–15. (Наукове фахове видання з технічних наук. Входить до міжнародних наукометричних баз РІНЦ, Google Scholar, EBSCO, Index Copernicus, Наукова періодика України). *(Особистий внесок: розробка вдосконаленого методу м'якої кластеризації).*

7. Structural–parametric synthesis of the feedforward neural networks with sigmoid piecewise–type neurons / M. Z. Zgurovsky, O. I. Chumachenko, V. S. Gorbatiuk // Electronics and control systems. – 2018. Volume 4, № 58, p. 42–47. (Наукове фахове видання з технічних наук. Входить до міжнародних наукометричних баз РІНЦ, Google Scholar, EBSCO, Index Copernicus, Наукова періодика України). *(Особистий внесок: розробка нової моделі штучного нейрону sigmoid piecewise).*

8. Комплексування декількох алгоритмів під час розв'язання задачі прогнозування / О. І. Чумаченко, В. С. Горбатюк // Житомир: Вісник ЖДТУ.

Серія: Технічні науки. – 2016. №1 (76), С. 101–106. (Наукове фахове видання з технічних наук. Входить до міжнародних наукометричних баз Index Copernicus, DOAJ, Google Scholar, WorldCat, BASE, eLibrary, Наукова періодика України). *(Особистий внесок: розробка методу прогнозування, що базується на використанні декількох локальних прогнозуючих моделей).*

9. A method for building a forecasting model with dynamic weights / Victor Sineglazov, Elena Chumachenko, Vladyslav Gorbatyuk // Eastern–European Journal of Enterprise Technologies. – 2014. [S.l.], v. 2, n. 4(68), p. 4–8. (Наукове фахове видання з технічних наук. Входить до міжнародних наукометричних баз Scopus, CrossRef, Index Copernicus, DOAJ, Google Scholar, OpenAIRE, WorldCat, CORE, SIS, CAS, BASE, EBSCO, Cite Factor, Наукова періодика України). *(Особистий внесок: розробка алгоритму побудови прогнозуючої моделі, який застосовує динамічні ваги).*

10. Алгоритм решения задачи прогнозирования / Е.И. Чумаченко, В.С. Горбатюк // Штучний інтелект. – 2012. № 2, С. 24–31. (Наукове фахове видання з технічних наук. Входить до міжнародних наукометричних баз Google Scholar, Наукова періодика України). *(Особистий внесок: розробка алгоритму прогнозування на базі штучних нейронних мереж).*

11. Метод решения задачи прогнозирования на основе комплексирования оценок / В. М. Синеглазов, Е. И. Чумаченко, В. С. Горбатюк // Індуктивне моделювання складних систем. – 2012. Випуск 4, С. 214–223. (Наукове фахове видання з технічних наук. Входить до міжнародних наукометричних баз РІНЦ, Google Scholar, Наукова періодика України). *(Особистий внесок: дослідження існуючих методів прогнозування, що використовують декілька локальних моделей–нейронних мереж).*

12. Использование искусственных нейронных сетей для задачи прогнозирования / Е. И. Чумаченко, В. С. Горбатюк // Електроника и системы

управления. – 2012. № 1, С. 113–119. (Наукове фахове видання з технічних наук. Входить до міжнародних наукометричних баз РІНЦ, Google Scholar, EBSCO, Index Copernicus, Наукова періодика України). *(Особистий внесок: аналіз переваг та недоліків застосування штучних нейронних мереж в якості моделей прогнозування).*

Матеріали конференцій:

13. One approach for the forecasting task solution / Victor Sineglazov, Elena Chumachenko, Vladyslav Gorbatyuk // Безпека в авіації та космічні технології: тези доповідей V Всесвітнього конгресу «Авіація у XXI столітті». – 2012, р. 3.5.49–3.5.53.

14. Method for predicting failure risk of UAV navigation systems / Elena Chumachenko, Vladyslav Gorbatyuk // Methods and systems of navigation and motion control (MSNMC – 2012). Book of abstracts. – 2012, р. 63–65.

15. Алгоритм решения задачи прогнозирования / Е.И. Чумаченко, В.С. Горбатюк // Интеллектуальные системы принятия решений и проблемы вычислительного интеллекта (ISDMCI – 2012): тезисы докладов. – 2012, С. 428–430.

16. Интеллектуальная система прогнозирования рисков послеоперационных осложнений / В. М. Синеглазов, Е. И. Чумаченко, В. С. Горбатюк // Интеллектуальные системы принятия решений и проблемы вычислительного интеллекта (ISDMCI – 2013): тезисы докладов. – 2013, С. 289–291.

17. Applying Different Neural Network's Topologies to the Forecasting Task / Victor Sineglazov, Elena Chumachenko, Vladyslav Gorbatyuk // International Conference in Inductive Modelling (ICIM – 2013). Book of abstracts. – 2013, р. 217–220.

18. Forecasting the Demand for UAV Using Different Neural Networks Topology / E. I. Chumachenko, V. S. Gorbatiuk // The 2–nd International Conference, Actual Problems of Unmanned Air Vehicles Development Proceedings. – 2013, p. 62–64.

19. Комплексування декількох алгоритмів при вирішенні задачі прогнозування / О. І. Чумаченко, В. С. Горбатюк // Інформаційно–комп’ютерні технології – 2016: тези доповідей VIII Міжнародної науково–технічної конференції. – Житомир: ЖДТУ. – 2016, С. 95–96.

20. Метод побудови прогнозуючої моделі з динамічними параметрами / В. М. Синєглазов, О. І. Чумаченко, В. С. Горбатюк // Інформаційні технології та комп’ютерне моделювання: тези доповідей Міжнародної науково–практичної конференції. – Івано–Франківськ: Супрун В. П. – 2016, С. 23–26.

21. Object's movement prediction in 3D space using neural networks / O. Chumachenko, V. Gorbatiuk // 4th International Conference Methods and systems of navigation and motion control (MSNMC–2016). Book of abstracts. – 2016, p. 255–258.

22. A New Approach in Cluster Analysis / Victor Sineglazov, Elena Chumachenko, Vladyslav Gorbatiuk // 4th International Conference on Actual Problems of Unmanned Aerial Vehicles Developments (APUAVD–2017). Book of abstracts. – 2017, p. 223–227.

23. Нова модель штучного нейрону для побудови нейронних мереж у задачі апроксимації / О.І. Чумаченко, В.С. Горбатюк // Інформаційні технології та комп’ютерне моделювання: тези доповідей Міжнародної науково–практичної конференції. – Івано–Франківськ. – 2018, С. 296–299.

24. Алгоритм м’якої кластеризації на основі розділяючих гіперповерхонь / В.С. Горбатюк // Електроніка – 2018: тези доповідей XI

Міжнародної науково–технічної конференції молодих вчених. – 2018, С. 425–430.

25. An algorithm for the corporate shares price forecasting / V. Sineglazov, O. Chumachenko, V. Gorbatiuk // XVIII International Scientific Conference «Corporate Governance – Theory and practice». Book of abstracts. – 2018, p. 101–102.

26. Прогнозування часових рядів з використанням регуляризованої суміші експертів / В. М. Синєглазов, О. І. Чумаченко, В. С. Горбатюк // Інформаційні системи та технології ІСТ–2018: тези доповідей VII Міжнародної науково–технічної конференції. – 2018, р. 457–459.

ДОДАТОК Б

Впровадження результатів дисертації

Затверджую
 Головний інженер ДН «ВО Київприлад»
 Білий В.М.
 «12» 06 2018р



АКТ

впровадження результатів дисертаційної роботи
 Горбатюка Владислава Сергійовича
 «Прогнозування часових рядів на основі гібридних
 нейронних мереж» подану на здобуття наукового ступеня
 кандидата технічних наук

Комісія у складі: голови комісії, начальника конструкторського бюро Подольського В.П., начальника відділення Кошмана О.Г., провідного фахівця Нікіфорова Ю.П.

цим Актом засвідчує, що результати дисертаційного дослідження Горбатюка В. С., а саме інтелектуальна система прогнозування на основі гібридної нейронної мережі з нейронами типу SigmPL, були використані для прогнозування попиту на лінійку пристроїв релейного захисту МРЗС.

Впровадження системи дозволило визначити оптимальним чином замовлення на комплектуючі та витратні матеріали і тим самим скоротити витрати.

Голова комісії,

Начальник конструкторського бюро

Члени комісії:

Начальник відділення

Провідний фахівець

Подольський В.П.

Кошман О.Г.

Нікіфоров Ю.П.





"ЗАТВЕРДЖУЮ"

Проректор НАУ

з навчальної роботи

А.Г. Гудманян

2020 р.

**Акт впровадження
у навчальний процес Національного авіаційного університету результатів
кандидатської дисертаційної роботи Горбатюка Владислава Сергійовича
«Інформаційна технологія прогнозування нестационарних часових рядів на
основі нейронних мереж»**

Комісія у складі:

- голови комісії - Мачаліна І.О., д.т.н., проф., декана ФАЕТ;
та членів комісії: - Гумена М.Б., к.т.н., доц., відповідального з
освітньої роботи ФАЕТ;
Мухіної М.П., д.т.н., доц., проф. кафедри АКІК;
Яновського Ф.Й., д.т.н., проф. кафедри ЕРМІТ,

яка діє на підставі розпорядження декана факультету аеронавігації, електроніки та телекомунікацій від 15.09.2020 р. № 26, засвідчує, що результати дисертаційної роботи Горбатюка В.С. впроваджені на кафедрі авіаційних комп'ютерно-інтегрованих комплексів факультету аеронавігації, електроніки та телекомунікацій НАУ у вигляді навчального курсу «Нейротехнології в комп'ютерно-інтегрованих комплексах» для студентів 4-го курсу за напрямом 151 «Автоматизація та комп'ютерно-інтегровані технології».

Голова комісії

Мачалін І.О.

Члени комісії

Гумен М.Б.

Мухіна М.П.

Яновський Ф.Й.

ДОДАТОК В

Апробація результатів дисертації

ЗАТВЕРДЖУЮ
Проректор з навчальної роботи
Національного технічного
університету України
“Київський політехнічний інститут
імені Ігоря Сікорського”
к. филос. н. доц.

Анатолій МЕЛЬНИЧЕНКО
“___” _____ 20__ р.



ВИТЯГ

з протоколу № 3 від 30 09 2020 р. розширеного засідання
кафедри технічної кібернетики
Національного технічного університету України
“Київський політехнічний інститут імені Ігоря Сікорського”

БУЛИ ПРИСУТНІ:

- з кафедри технічної кібернетики:
Д. т. н., професор, Пархомей І. Р.
Д. т. н., професор, Жураковський Б. Ю.
Д. т. н., доцент, Чумаченко О. І.
К. т. н., професор, Поліщук М. М.
К. т. н., доцент, Ліхоузова Т. А.
К. т. н., доцент, Тимошин Ю. А.
К. т. н., доцент, Ткач М. М.
К. т. н., доцент, Пасько В. П.
К. т. н., доцент, Резніков С. А.

- з кафедри автоматики та управління в технічних системах:
Д. т. н., професор, Теленик С. Ф.
Д. т. н., професор, Ролік О. І.

СЛУХАЛИ:

1. Повідомлення здобувача наукового ступеня кандидата наук, аспіранта кафедри технічної кібернетики у 2015-2018 рр. Горбатюка Владислава Сергійовича за матеріалами дисертаційної роботи “Інформаційна технологія прогнозування нестационарних часових рядів на основі нейронних мереж”, поданої на здобуття наукового ступеня кандидата технічних наук за спеціальністю 05.13.06 – Інформаційні технології.

Тему дисертаційної роботи “Інформаційна технологія прогнозування нестаціонарних часових рядів на основі нейронних мереж” затверджено на засіданні Вченої ради факультету інформатики та обчислювальної техніки Національного технічного університету України “Київський політехнічний інститут імені Ігоря Сікорського” (протокол № 6 від “21” січня 2016 року) та перезатверджено на засіданні Вченої ради факультету інформатики та обчислювальної техніки Національного технічного університету України “Київський політехнічний інститут імені Ігоря Сікорського” (протокол № 1 від “31” серпня 2020 року).

Науковим керівником затверджений д.т.н., доцент, Чумаченко О. І.

2. Запитання до здобувача.

Запитання по темі дисертації ставили:

Д. т. н., професор, Пархомей І. Р.

Д. т. н., професор, Теленик С. Ф.

К. т. н., доцент, Тимошин Ю. А.

3. Виступи за обговореною роботою.

В обговоренні дисертації взяли участь:

Д. т. н., професор, Пархомей І. Р.

Д. т. н., професор, Теленик С. Ф.

Д. т. н., професор, Ролік О. І.

УХВАЛИЛИ:

ПРИЙНЯТИ такий висновок за дисертаційною роботою:

1. Актуальність теми дослідження. У сучасному світі інформаційні технології застосовуються для вирішення широкого спектру завдань, в тому числі у системах підтримки прийняття рішень, які майже завжди потребують прогнозування часових рядів різної природи (економічні, технічні, природні та інші). На основі прогнозів ці системи будують плани та/або пропонують відповідні рішення – таким чином, неправильний прогноз може призвести до неправильних планів та рішень. Усе це вказує на актуальність задачі прогнозування. Класичні підходи до розв’язання задачі прогнозування в основному базуються на використанні теорії ймовірності та математичної статистики, і зазвичай припускають, що модель прогнозованого об’єкту відома з точністю до параметрів (лінійна, поліноміальна, експоненціальна та інші), та припускають стаціонарність прогнозованого об’єкту, або деякий конкретний вигляд нестаціонарності. Використання технологій на основі штучного інтелекту дозволяє позбавитися деяких припущень та недоліків класичних підходів. Однак, більшість методів на основі штучного інтелекту припускають постійний характер прогнозованого об’єкту. У зв’язку з цим доцільною і актуальною є розробка нових методів, моделей та інформаційної технології прогнозування нестаціонарних часових рядів з використанням нейронних мереж та інтелектуальних технологій, які дозволять підвищити якість прогнозу за рахунок відмови від припущення постійного характеру прогнозованого об’єкту та підвищення стабільності прогнозуючої моделі при малій кількості наявних даних.

2. Зв'язок роботи з науковими програмами, планами, темами. Дисертаційне дослідження виконувалось на кафедрі технічної кібернетики відповідно до планів НДР, програм і договорів, що виконувались в НТУУ «КПІ ім. Ігоря Сікорського»:

- НДР «Методологія побудови інформаційних систем з інтелектуальною підтримкою прийняття рішень» (номер державної реєстрації 0111U002510);
- НДР «Розробка методів, алгоритмів структурно-параметричного синтезу гібридних нейронних мереж та створення на її основі інтелектуальних систем розв'язання прикладних задач» (номер державної реєстрації 0117U002595).

3. Наукова новизна отриманих результатів.

У дисертації вперше одержані такі нові наукові результати:

1. Уперше запропоновано модель штучного нейрону, яка відрізняється тим, що складається з трьох зважених суматорів та нової функції активації, яка залежить від трьох змінних, що дозволяє отримати модель, яка апроксимує певну просту кусково-лінійну функцію і при цьому дозволяє незалежно налаштовувати параметри гіперплощини, яка задає півпростори з різними лінійними функціями. За рахунок цих відмінностей нова модель має покращені апроксимуючі властивості.
2. Уперше розроблено метод розв'язання задачі прогнозування для нестаціонарних часових рядів, який базується на застосуванні методів кластеризації та побудові декількох локальних прогнозуючих моделей або застосуванні методу комбінації експертів, що відрізняється використанням спеціального регуляризуючого критерію. Завдяки використанню цього критерію зменшується вплив розмірності вибірки на якість вихідної прогнозуючої моделі та підвищується стійкість локальних моделей до перенавчання.
3. Уперше запропоновано метод кластеризації, що відрізняється використанням моделей роздільних гіперповерхонь. Цей алгоритм дозволяє вирішувати задачу кластеризації для певного класу критеріїв, і при цьому перетворює задачу кластеризації у задачу оптимізації диференційованої функції. Крім того, алгоритм теоретично дозволяє знаходити кластери, що розділені гіперповерхнею будь-якої складності.
4. Вдосконалено багаторядний метод групового урахування аргументів для вирішення задачі прогнозування шляхом застосування алгоритмів зворотного поширення помилки та вибіркового виключення, що дозволило покращити точність прогнозу.

4. Ступінь обґрунтованості наукових положень та висновків, сформульованих у дисертаційній роботі. Ступінь обґрунтованості отриманих у дисертаційній роботі наукових положень, висновків і рекомендацій, підтверджується правильністю постановки наукових і практичних завдань, комплексністю підходу до вибору методів дослідження, методичних прийомів, аналізом здобутків вітчизняних та зарубіжних науковців з даної тематики, а також апробацією основних положень і отриманих практичних результатів на науково-технічних конференціях.

Наукові положення, висновки та рекомендації дисертаційної роботи базуються на дослідженні існуючих підходів побудови систем прогнозування часових рядів, що базуються на використанні ШНМ і розробці нових моделей та методів структурного та параметричного синтезу систем прогнозування часових рядів на базі штучних нейронних мереж.

Наукові положення і висновки, отримані у дисертації, підкріплено порівняльними тестами результатів вирішення прикладних задач, отриманих шляхом використання розробленої інформаційної технології та шляхом використання відомих методів прогнозування нестационарних часових рядів.

5. Практична цінність результатів роботи. На основі розроблених методів та моделей створено програмні засоби для вирішення задачі побудови прогнозуючих моделей для часових рядів, які в подальшому було успішно впроваджені.

6. Впровадження та пропозиції з використання отриманих результатів. Розроблені в рамках дослідження алгоритми впроваджено в навчальний процес кафедри авіаційних комп'ютерно-інтегрованих комплексів факультету аеронавігації, електроніки та телекомунікацій Національного авіаційного університету, зокрема в матеріали курсу «Нейротехнології в комп'ютерно-інтегрованих комплексах». Акт впровадження від 25.09.2020.

Розроблена інформаційна технологія прогнозування нестационарних часових рядів на основі штучних нейронних мереж була впроваджена на державному підприємстві «ВО Київприлад», що дозволило визначити попит на комплектуючі та витратні матеріали і тим самим скоротити витрати. Акт впровадження від 12.06.2018.

7. Особиста участь автора в одержанні наукових та практичних результатів, що викладені в дисертаційній роботі. В опублікованих працях, написаних у співавторстві, авторі належить:

- розробка алгоритму побудови прогнозуючої моделі на основі використання методу групового урахування аргументів та штучних нейронних мереж
- розробка спеціального критерію, що застосовується для побудови локальних моделей у методі прогнозування часових рядів на основі комбінації експертів
- розробка алгоритму м'якої кластеризації на основі розділяючих гіперповерхонь
- розробка загального методу комплексування прогнозуючих моделей.

Дисертаційна робота виконана на кафедрі технічної кібернетики, факультету інформатики та обчислювальної техніки, НТУУ «Київський політехнічний інститут імені Ігоря Сікорського», науковий керівник д.т.н., доцент Чумаченко О.І. Робота є результатом самостійних досліджень Горбатюка В.С.

8. Апробація результатів дисертації. Основні результати дисертації доповідались та обговорювались на наступних конференціях та інших заходах: V Всесвітній конгрес «Авіація у XXI столітті» – «Безпека в авіації та космічні технології». – Київ: НАУ, 25-27 вересня, 2012 р.; Methods and systems of navigation and motion control (MSNMC – 2012). – Kyiv: NAU, October 9-12, 2012; Интеллектуальные системы принятия решений и проблемы вычислительного интеллекта (ISDMCI – 2012). – м. Євпаторія, 27–31 травня, 2012 р.; Интеллектуальные системы принятия решений и проблемы вычислительного интеллекта (ISDMCI – 2013). – м. Євпаторія, 20–24 травня, 2013 р.; International Conference in Inductive Modelling (ICIM – 2013). – Kyiv: IRTC, September 15-21, 2013; VIII Міжнародна науково-технічна конференція «Інформаційно-комп'ютерні технології – 2016». – Житомир: ЖДТУ, 22–23 квітня, 2016 р.; Міжнародна науково-практична конференція «Інформаційні технології та комп'ютерне моделювання». – м. Івано-Франківськ: Прикарпатський національний університет імені Василя Стефаника, 23-28 травня, 2016 р.; 4th International Conference «Methods and systems of navigation and motion control (MSNMC-2016)». – Kyiv:

NAU, October 18-20, 2016; 4th International Conference on Actual Problems of Unmanned Aerial Vehicles Developments (APUAVD-2017). – Kyiv: NAU, October 17 – 19, 2017; Міжнародна науково-практична конференція «Інформаційні технології та комп'ютерне моделювання». – м. Івано-Франківськ: Прикарпатський національний університет імені Василя Стефаника, 14-19 травня, 2018 р.; XI Міжнародна науково-технічна конференція молодих вчених «Електроніка – 2018». – м. Київ: НТУУ «КПІ ім. І. Сікорського», 3-5 квітня, 2018 р.; XVIII International Scientific Conference «Corporate Governance - Theory and practice». – Kraków: AGH University of Science and Technology, June 21-22, 2018; 7-ма Міжнародна науково-технічна конференція «Інформаційні системи та технології ICT-2018». – м. Харків – Коблеве: Харківський національний університет радіоелектроніки, 10-15 вересня, 2018 р.

9. Перелік публікацій за темою дисертації із зазначенням особистого внеску здобувача. За результатами досліджень опубліковано 26 наукових праць, у тому числі 1 монографія, 11 статей (4 з них у закордонних наукових виданнях країн, що входять до ЄС, і є у наукометричній базі даних Scopus, 5 – у фахових наукових виданнях України, що входять до бібліографічної бази даних CiteFactor, Index Copernicus та інших, 2 – у фахових виданнях України з технічних наук, які представлені у базах даних: DOAJ; Ulrich's Periodicals Directory; eLibrary; Google Scholar, ResearchBib, UIF), 14 тез доповідей на наукових конференціях.

Монографії:

1. Интеллектуальные методы прогнозирования [Текст]: монография / В. М. Синеглазов, Е. И. Чумаченко, В. С. Горбатиук. – К.: Освіта України, 2013. – 219 с. *Особистий внесок здобувача: класифікація постановок задачі прогнозування; дослідження задачі прогнозування ризиків та методів її вирішення.*

Статті у виданнях іноземних держав:

1. An algorithm for solving the problem of forecasting / Victor Sineglazov, Elena Chumachenko, Vladyslav Gorbatiuk // Aviation. – 2013. №1(17), p. 9-13. (Видання іноземної держави-члена ЄС, Литва). *Особистий внесок здобувача: розробка алгоритму побудови прогнозуючої моделі на основі використання методу групового урахування аргументів та штучних нейронних мереж.*

2. Using a mixture of experts' approach to solve the forecasting task / Victor Sineglazov, Elena Chumachenko, Vladyslav Gorbatiuk // Aviation. – 2014. №3(18), p. 129-133. (Видання іноземної держави-члена ЄС, Литва). *Особистий внесок здобувача: розробка спеціального критерію, що застосовується для побудови локальних моделей.*

3. An algorithm for the corporate shares price forecasting / V. Sineglazov, O. Chumachenko, V. Gorbatiuk // Information Technology in Selected Areas of Management 2017. – 2018, p. 73–79. (Видання іноземної держави-члена ЄС, Польща). *Особистий внесок здобувача: виконання порівняльних тестів на реальних вибірках.*

4. Forecasting aircraft miles flown time series using deep learning-based hybrid approach / Victor Sineglazov, Elena Chumachenko, Vladyslav Gorbatiuk // Aviation. – 2018. Issue 1, № (22), p. 6-12. (Видання іноземної держави-члена ЄС, Литва). *Особистий внесок здобувача: розробка методу поєднання глибинних нейронних мереж та методу групового урахування аргументів.*

Статті у наукових фахових виданнях України:

1. Алгоритм решения задачи прогнозирования / Е.И. Чумаченко, В.С. Горбатюк // Штучний інтелект. – 2012. № 2, С. 24-31. (Наукове фахове видання з технічних наук. Входить до міжнародної наукометричної бази Google Scholar). *Особистий внесок здобувача: збір та попередня обробка даних, оглядова частина статті.*
2. Метод решения задачи прогнозирования на основе комплексирования оценок / В. М. Синеглазов, Е. И. Чумаченко, В. С. Горбатюк // Индуктивное моделирование сложных систем. – 2012. Выпуск 4, С. 214-223. (Наукове фахове видання з технічних наук. Входить до міжнародної наукометричної бази Google Scholar). *Особистий внесок здобувача: проведення експериментів.*
3. Использование искусственных нейронных сетей для задачи прогнозирования / Е. И. Чумаченко, В. С. Горбатюк // Электроника и системы управления. – 2012. № 1, С. 113-119. (Наукове фахове видання з технічних наук. Входить до міжнародних наукометричних баз РІНЦ, Google Scholar, EBSCO, Index Copernicus). *Особистий внесок здобувача: огляд існуючих алгоритмів.*
4. Structural-parametric synthesis of the feedforward neural networks with sigmoid piecewise-type neurons / M. Z. Zgurovsky, O. I. Chumachenko, V. S. Gorbatiuk // Electronics and control systems. – 2018. Volume 4, № 58, p. 42-47. (Наукове фахове видання з технічних наук. Входить до міжнародних наукометричних баз РІНЦ, Google Scholar, EBSCO, Index Copernicus). *Особистий внесок здобувача: розробка алгоритму структурно-параметричного синтезу штучних нейронних мереж прямого поширення з нейронами типу sigmoid piecewise.*
5. Soft clustering algorithm based on separating hypersurfaces / O. I. Chumachenko, V. S. Gorbatiuk // Electronics and control systems. – 2017. Volume 2, № 52, p. 11-15. (Наукове фахове видання з технічних наук. Входить до міжнародних наукометричних баз Scopus, CrossRef, Index Copernicus, DOAJ, Google Scholar, OpenAIRE, WorldCat, CORE, SIS, CAS, BASE, EBSCO, Cite Factor). *Особистий внесок здобувача: розробка алгоритму кластеризації.*
6. A method for building a forecasting model with dynamic weights / Victor Sineglazov, Elena Chumachenko, Vladyslav Gorbatiuk // Eastern-European Journal of Enterprise Technologies. – 2014. [S.l.], v. 2, n. 4(68), p. 4-8. ISSN 1729-4061. (Наукове фахове видання з технічних наук. Входить до міжнародних наукометричних баз РІНЦ, Google Scholar, EBSCO, Index Copernicus). *Особистий внесок здобувача: збір даних та проведення експериментального тестування запропонованого підходу до задачі прогнозування.*
7. Комплексування декількох алгоритмів під час розв'язання задачі прогнозування / О. І. Чумаченко, В. С. Горбатюк // Житомир: Вісник ЖДТУ. Серія: Технічні науки. – 2016. №1 (76), С. 101-106. (Наукове фахове видання з технічних наук. Входить до міжнародних наукометричних баз Index Copernicus, DOAJ, Google Scholar, WorldCat, BASE, eLibrary). *Особистий внесок здобувача: розроблення загального алгоритму комплексування.*

Матеріали конференцій:

1. One approach for the forecasting task solution / Victor Sineglazov, Elena Chumachenko, Vladyslav Gorbatiuk // Безпека в авіації та космічні технології: тези

доповідей V Всесвітнього конгресу «Авіація у XXI столітті». – 2012, р. 3.5.49-3.5.53. Особистий внесок здобувача: огляд існуючих алгоритмів.

2. Method for predicting failure risk of UAV navigation systems / Elena Chumachenko, Vladyslav Gorbatiuk // *Methods and systems of navigation and motion control (MSNMC – 2012). Book of abstracts.* – 2012, р. 63-65. Особистий внесок здобувача: проведення та дослідження результатів експериментів.

3. Алгоритм решения задачи прогнозирования / Е.И. Чумаченко, В.С. Горбатюк // *Интеллектуальные системы принятия решений и проблемы вычислительного интеллекта (ISDMCI – 2012): тезисы докладов.* – 2012, С. 428-430. Особистий внесок здобувача: вступ, проведення порівняльних експериментів.

4. Интеллектуальная система прогнозирования рисков послеоперационных осложнений / В. М. Синеглазов, Е. И. Чумаченко, В. С. Горбатюк // *Интеллектуальные системы принятия решений и проблемы вычислительного интеллекта (ISDMCI – 2013): тезисы докладов.* – 2013, С. 289-291. Особистий внесок здобувача: постановка задачі прогнозування ризиків, аналіз переваг та недоліків існуючих методів.

5. Applying Different Neural Network's Topologies to the Forecasting Task / Victor Sinoglazov, Elena Chumachenko, Vladyslav Gorbatiuk // *International Conference in Inductive Modelling (ICIM – 2013). Book of abstracts.* – 2013, р. 217-220. Особистий внесок здобувача: огляд існуючих алгоритмів.

6. Forecasting the Demand for UAV Using Different Neural Networks Topology / E. I. Chumachenko, V. S. Gorbatiuk // *The 2-nd International Conference, Actual Problems of Unmanned Air Vehicles Development Proceedings October, 15-17, 2013, Kyiv, Ukraine.* pp. 62-64. Особистий внесок здобувача: постановка задачі прогнозування ризиків, аналіз переваг та недоліків існуючих методів.

7. Комплексування декількох алгоритмів при вирішенні задачі прогнозування / О. І. Чумаченко, В. С. Горбатюк // *Інформаційно-комп'ютерні технології – 2016: тези доповідей VIII Міжнародної науково-технічної конференції.* – Житомир: ЖДТУ, 2016. – С. 95-96. Особистий внесок здобувача: постановка задачі, розроблення загального алгоритму комплексування.

8. Метод побудови прогнозуючої моделі з динамічними параметрами / В. М. Синеглазов, О. І. Чумаченко, В. С. Горбатюк // *Інформаційні технології та комп'ютерне моделювання: тези доповідей Міжнародної науково-практичної конференції.* – Івано-Франківськ: Супрун В. П., 2016. – С. 23-26. Особистий внесок здобувача: аналіз даних, порівняльне тестування запропонованого підходу до задачі прогнозування.

9. Object's movement prediction in 3D space using neural networks / O. Chumachenko, V. Gorbatiuk // *4th International Conference Methods and systems of navigation and motion control (MSNMC-2016). Book of abstracts.* – 2016, р. 255-258. Особистий внесок здобувача: адаптація алгоритму до прогнозування 3D руху.

10. A New Approach in Cluster Analysis / Victor Sinoglazov, Elena Chumachenko, Vladyslav Gorbatiuk // *4th International Conference on Actual Problems of Unmanned Aerial Vehicles Developments (APUAVD-2017). Book of abstracts.* – 2017, р. 223-227. Особистий внесок здобувача: розробка алгоритму кластеризації.

11. Нова модель штучного нейрону для побудови нейронних мереж у задачі апроксимації / О. І. Чумаченко, В. С. Горбатюк // *Інформаційні*

технології та комп'ютерне моделювання: тези доповідей Міжнародної науково-практичної конференції. – Івано-Франківськ. - 2018. – С. 296-299. Особистий внесок здобувача: розробка нової моделі штучного нейрону.

12. Алгоритм м'якої кластеризації на основі розділяючих гіперповерхонь / В.С. Горбатюк // Електроніка – 2018: тези доповідей XI Міжнародної науково-технічної конференції молодих вчених. – 2018, С. 425-430. Особистий внесок здобувача: дослідження проблеми кластеризації та збір даних для експериментів.

13. An algorithm for the corporate shares price forecasting / V. Sineglazov, O. Chumachenko, V. Gorbatiuk // XVIII International Scientific Conference «Corporate Governance - Theory and practice». Book of abstracts. – 2018, p. 101-102. Особистий внесок здобувача: огляд існуючих методів.

14. Прогнозування часових рядів з використанням регуляризованої суміші експертів / В. М. Синеглазов, О. І. Чумаченко, В. С. Горбатюк // Інформаційні системи та технології ICT-2018: тези доповідей VII Міжнародної науково-технічної конференції. – 2018. Особистий внесок здобувача: вдосконалення класичного методу суміші експертів шляхом використання спеціальної регуляризації.

ВВАЖАТИ, що дисертаційна робота Горбатюка В.С. “Інформаційна технологія прогнозування нестаціонарних часових рядів на основі нейронних мереж”, що подана на здобуття наукового ступеня кандидата технічних наук, за своїм науковим рівнем та практичною цінністю, змістом та оформленням повністю відповідає вимогам, що їх пред'являють до кандидатських дисертацій та паспорту спеціальності 05.13.06 – Інформаційні технології.

РЕКОМЕНДУВАТИ дисертаційну роботу “Прогнозування часових рядів на основі гібридних нейронних мереж”, подану Горбатюком Владиславом Сергійовичем на здобуття наукового ступеня кандидата технічних наук, до захисту у спеціалізованій раді Д 26.002.29 за спеціальністю 05.13.06 – Інформаційні технології.

Завідувач кафедри технічної кібернетики
Д. т. н., професор



Ігор ПАРХОМЕЙ

Вчений секретар
кафедри технічної кібернетики
К. т. н., доцент



Тетяна ЛІХОУЗОВА