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

УДК 004.942:519.216.3

БЕЛАС АНДРІЙ ОЛЕГОВИЧ

ДИСЕРТАЦІЯ

МОДЕЛІ І МЕТОДИ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ ДАНИХ ДЛЯ ПРОГНОЗУВАННЯ НЕЛІНІЙНИХ НЕСТАЦІОНАРНИХ ПРОЦЕСІВ

12 - Інформаційні технології

122 - Комп’ютерні науки

Подається на здобуття наукового ступеня доктора філософії

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

_____ А.О. Белас

Науковий керівник

Бідюк Петро Іванович, доктор технічних наук, професор.

Київ - 2022

АНОТАЦІЯ

Белас А. О. Моделі і методи інтелектуального аналізу даних для прогнозування нелінійних нестаціонарних процесів. - Кваліфікаційна праця на правах рукопису.

Дисертація на здобуття наукового ступеня доктора філософії з галузі знань 12 Інформаційні технології за спеціальністю 122 Комп'ютерні науки. – Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Київ, 2022.

Нелінійні нестаціонарні процеси, представлені у вигляді часових рядів, можуть описувати динаміку процесів в технічних, економічних та інших системах та мати численні застосування в різних галузях.

Задача моделювання та прогнозування нелінійних нестаціонарних фінансово-економічних процесів є критично важливою для кожного підприємства, що здійснює торгівельну або інвестиційну діяльність, оскільки є невід'ємною частиною планування закупівель, використання ресурсів, прийняття стратегічних рішень.

На сьогодні підходи до вирішення поставленої задачі методами інтелектуального аналізу даних перебувають у стані активного дослідження, а загальноприйнятої методики їх практичного застосування немає.

Метою дисертаційного дослідження є підвищення адекватності моделей нелінійних нестаціонарних фінансово-економічних процесів і точності прогнозів їх розвитку шляхом застосування сучасних методів інтелектуального аналізу даних та їх комбінуванням з регресійними моделями.

В дисертації отримані такі наукові результати:

1. **Вперше** запропоновано, сформульовано і описано загальну методику для моделювання і прогнозування нелінійних нестаціонарних процесів на основі математичних моделей інтелектуального аналізу даних з використанням статистичних даних, в тому числі критеріальну базу, методи обробки і підготовки даних до моделювання і прогнозування, що є основою для створення системи підтримки прийняття рішень для аналізу нелінійних нестаціонарних процесів.

2. **Вперше** запропоновано метод побудови моделей нелінійних процесів, який відрізняється застосуванням окремих процедур для оптимізації структури лінійної та нелінійної складових моделі з їх наступним адитивним об'єднанням в єдину модель, що забезпечує підвищення адекватності моделі і точності оцінок прогнозів у цілому.

3. **Вперше** запропоновано використання нових для вибраного типу процесів моделей та їх модифікацій на основі методів інтелектуального аналізу даних, а саме згорткових нейронних мереж, для формального опису і прогнозування ННП, побудовано відповідні нові моделі та прогнози.

4. **Удосконалено** існуючий підхід з використанням класичних статистичних моделей на основі регресійного підходу шляхом комбінування їх з методами інтелектуального аналізу даних;

5. **Удосконалено** метод оцінювання параметрів лінійних моделей для адаптації на основі методу Монте-Карло для марківських ланцюгів.

Практична цінність дисертаційної роботи полягає в тому, що: розроблено моделі і методи інтелектуального аналізу даних для прогнозування нелінійних нестаціонарних процесів.

Результати роботи впроваджено у навчальний процес інституту прикладного системного аналізу НТУУ «КПІ імені Ігоря Сікорського».

Всі результати дисертаційного дослідження доведено до високого практичного інженерного рівня та впроваджено у АТ «ПУМБ» з метою покращення процесів прийняття рішень та побудови стратегій розвитку.

Створено систему підтримки прийняття рішень, що побудована на основі оцінок прогнозів за допомогою розробленої загальної методики з використанням методів інтелектуального аналізу даних. За допомогою розробленої системи і методики фахівці ПУМБ будують моделі і прогнози для фінансово-економічних процесів на основі аналізу економетричних даних, якими володіє банк в своїх системах.

Всі теоретичні і практичні результати дисертаційної роботи у повній мірі опубліковано у фахових вітчизняних та закордонних наукових виданнях, що входять до відповідного встановленого переліку, а також виконано їх належну апробацію на міжнародних наукових конференціях і семінарах.

У дисертаційному дослідженні розв’язана задача підвищення адекватності математичних моделей нелінійних нестаціонарних фінансово-економічних процесів та точності відповідних прогнозів, шляхом застосування сучасних методів інтелектуального аналізу даних до статистичних даних, представлених у вигляді часових рядів.

В роботі вибрано, розглянуто і описано класи нелінійних нестаціонарних процесів, що використані для моделювання та прогнозування, а також виконано опис математичних моделей та підходів, що на базі статистичних даних у вигляді часових рядів, використовуються для опису їх динаміки. Вибрано типи процесів для дослідження, такі як: інтегровані, гетероскедастичні процеси, процеси Леві, процеси зі стохастичним трендом, процеси логістичного типу.

В роботі вибрано клас фінансово-економічних процесів, сформульовано задачу їх прогнозування. Однак розроблену методику можливо застосовувати і в інших системах (технічних, медичних, тощо) з відповідною визначеною динамікою. Для прогнозування розглянуто підхід предиктивної аналітики з використанням методів інтелектуального аналізу даних. Методологічною основою роботи є сучасна аналітична методологія SEMMA. Зібрано статистичні дані для експериментів на базі продажів магазину Walmart,

продажів протидіабетичних препаратів в Австралії, продажу пального в США для застосування розробленої методики на реальних статистичних даних.

Для первинного аналізу даних запропоновано підходи статистичного та графічного аналізу, а також статистичні тести для визначення нелінійності та нестаціонарності процесу: вибрано тест Уайта для перевірки на нелінійність та тест KPSS для перевірки на нестаціонарність як основні. Розглянуто методи для виявлення та обробки аномальних та пропущених значень.

Для виділення шумової складової з часового ряду розглянуто фільтр Калмана та метод експонеційного згладжування. Підходи цифрової фільтрації варто застосовувати обережно, не завжди як необхідний етап процесу побудови моделі, а як можливий варіант, обов'язково перевіряючи в кінці процесу моделювання якість отриманих прогнозів з та без використання попередньої фільтрації.

В роботі запропоновано метод побудови моделей нелінійних процесів, що відрізняється застосуванням окремих процедур для оптимізації структури лінійної та нелінійної складових моделі з їх наступним адитивним об'єднанням в єдину модель, що забезпечує підвищення адекватності моделі і точності прогнозів у цілому. Розглянуто різні методи комбінування оцінок прогнозів різних моделей, вибрано метод бустінгу як основний для використання в роботі.

Описано підхід для виділення та моделювання лінійної складової процесу на базі регресійних моделей, а також підхід з використання отриманої моделі AR для формування оцінок прогнозів, розглянуто і вибрано критерії для оцінки адекватності отриманих моделей, такі як BIC. Для оцінок якості прогнозів побудовано критеріальну базу на базі критеріїв MSE, MAE, RMSLE.

Для опису нелінійної складової розглянуто авторегресивний підхід на базі ARIMA, з алгоритмом автоматичної побудови моделі, а також підходи на базі рекурентних (RNN) і згорткових (CNN) нейронних мереж, проаналізовано їх переваги та недоліки. Для нейронних мереж проаналізовано

підходи до оптимізації параметрів моделі, запропоновано алгоритм Adam, як найбільш ефективний. Розглянуто різні підходи для багатокрокового прогнозування за допомогою нейронних мереж, вибрано підхід із застосуванням multi-output мереж як основний.

Розглянуто необхідність адаптивної побудови моделей для прогнозування нелінійних нестаціонарних процесів, основні принципи адаптації таких моделей. Розглянуто відповідні підходи до адаптації як для лінійних моделей, так і для нейронних мереж. Для адаптації лінійних моделей удосконалено метод оцінювання параметрів на основі методу Монте-Карло для марківських ланцюгів.

Проведено порівняльний аналіз отриманих прогнозів з результатами використання відомих підходів та методів. За результатами всіх практичних експериментів було показано, що використання розробленого підходу AR-CNN дозволяє отримувати адекватні моделі та точні прогнози при відносній простоті побудові та невеликих обчислювальних втратах.

За матеріалами дисертації опубліковано 8 робіт, з яких 5 – це статті у журналах і збірниках наукових праць, що входять до переліку фахових видань затверджених МОН України за спеціальністю дисертації або у періодичних виданнях іноземних держав (2 з них включені до міжнародної наукометричної бази Scopus), та 3 – публікації у матеріалах конференцій (у тому числі, міжнародних).

Ключові слова: математичне моделювання, прогнозування, інтелектуальний аналіз даних, фінансово-економічні процеси, нелінійні процеси, нестаціонарні процеси, часові ряди, знаходження аномалій, фільтрація, регресійні моделі, нейронні мережі, рекурентні нейронні мережі, згорткові нейронні мережі, системний аналіз, адаптивне моделювання.

SUMMARY

Belas A. Models and methods of intellectual data analysis for forecasting nonlinear nonstationary processes. − Qualifying scientific work, the manuscript.

PhD thesis in the field of knowledge 12 Information Technology in specialty 122 Computer Science. – National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”, Kyiv, 2022.

Nonlinear nonstationary processes, represented in the form of time series data, can describe the dynamics of processes in technical, economic, and other systems and have numerous applications in various industries.

The task of modeling and forecasting nonlinear nonstationary financial and economic processes is critically important for every enterprise engaged in trading or investment activities since it is an integral part of procurement planning, resource usage, and strategic decision-making.

Now approaches to solving this problem using intellectual data analysis or machine learning are in a state of active research, and there is no generally accepted methodology for their practical implementation.

The aim of the thesis is to increase the adequacy of models of nonlinear nonstationary financial and economic processes and the accuracy of forecasts of their development by applying modern methods of intellectual data analysis and combining them with regression models.

In the thesis the following new scientific results were obtained:

1. **For the first time**, a general methodology for modeling and forecasting nonlinear nonstationary processes based on mathematical models of intellectual data analysis using statistical data was proposed, formulated, and described, including a criterion base, methods of data processing and preparation for modeling and forecasting, which is the basis for creating a decision support system for the analysis of nonlinear nonstationary processes.

2. **For the first time**, a method of building models of nonlinear processes is proposed, which is characterized by the use of separate procedures for optimizing the structure of linear and nonlinear components of the model with their subsequent additive combination into a single model, which ensures an increase in the adequacy of the model and the accuracy of forecast estimates as a whole.

3. **For the first time**, the use of new models for the selected type of processes and their modifications based on the methods of intellectual data analysis, namely convolutional neural networks, was proposed for the formal description and forecasting of NNP, and corresponding new models and forecasts were built.

4. The existing approach using classical statistical models based on the regression approach **has been improved** by combining them with methods of intellectual data analysis;

5. The method of estimating the parameters of linear models for adaptation based on the Monte Carlo method for Markov chains **has been improved**.

The results of the work are implemented and were used in the educational process of the Institute for Applied System Analysis at the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute".

All the results of the dissertation research were brought to a high practical engineering level and implemented in JSC "FUIB" in order to improve decision-making processes and build development strategies. A decision-making support system has been created, which is built on the basis of forecast assessments using the developed general methodology using methods of intellectual data analysis. With the help of the developed system and methodology, FUIB specialists build models and forecasts for financial and economic processes based on the analysis of econometric data that the bank has in its systems.

All theoretical and practical results of the dissertation work have been fully published in specialized domestic and foreign scientific publications that are included in the corresponding established list, and they have also been properly approved at international scientific conferences and seminars.

In the dissertation study, the problem of increasing the adequacy of mathematical models of nonlinear nonstationary financial and economic processes and the accuracy of corresponding forecasts is solved by applying modern methods of intellectual data analysis to statistical data presented in the form of time series.

In the work, the classes of nonlinear nonstationary processes used for modeling and forecasting was selected, considered, and described, as well as a description of mathematical models and approaches that are used to describe their dynamics based on statistical data in the form of time series. Types of processes are selected for research, such as integrated, heteroskedastic processes, Levy processes, processes with a stochastic trend, and logistic-type processes.

In the work, a class of financial and economic processes was selected, and the problem of their forecasting was formulated. However, the developed methodology can be applied in other systems (technical, medical, etc.) with the corresponding defined dynamics. For forecasting, the approach of predictive analytics using methods of machine learning is considered. The methodological basis of the work is the modern analytical methodology of SEMMA. Collected statistical data for experiments based on sales of the Walmart store, sales of antidiabetic drugs in Australia, and sales of fuel in the USA to apply the developed methodology to real statistical data.

For exploratory data analysis, statistical and graphical analysis approaches, as well as statistical tests for determining process nonlinearity and nonstationarity were proposed: White's test for nonlinearity and KPSS test for nonstationarity were chosen as the main ones. Methods for the detection and processing of anomalies and missing values were considered.

To extract the noise component from the time series, the Kalman filter and the exponential smoothing method were considered. Digital filtering approaches should be used carefully, not always as a necessary stage of the model building process, but as a possible option, necessarily checking at the end of the modeling

process the quality of the obtained forecasts with and without the use of preliminary filtering.

The thesis proposes a method of building models of nonlinear processes, which is distinguished by the use of separate procedures for optimizing the structure of linear and nonlinear components of the model with their subsequent additive combination into a single model, which ensures an increase in the adequacy of the model and the accuracy of forecasts in general. Different methods of combining estimates of forecasts of different models were considered, and the boosting method was chosen as the main one for use in the work.

An approach for the selection and modeling of the linear component of the process based on regression models was described, as well as an approach for using the obtained AR model to form forecast estimates, criterion for assessing the adequacy of the obtained models, such as BIC, was considered and selected. A criterion base based on the MSE, MAE, and RMSLE was built to assess the quality of forecasts.

To describe the nonlinear component, an autoregressive approach based on ARIMA, with an algorithm for automatic model building, as well as approaches based on recurrent (RNN) and convolutional (CNN) neural networks were considered, and their advantages and disadvantages were analyzed. For neural networks, approaches to optimization of model parameters were analyzed, and the Adam algorithm was proposed as the most effective. Different approaches for multi-step forecasting using neural networks were considered, and the approach using multi-output networks was chosen as the main one.

The need for adaptive construction of models for forecasting nonlinear nonstationary processes, the main principles of adaptation of such models were considered. Appropriate approaches to adaptation for both linear models and neural networks were considered. To adapt linear models, the parameter estimation method based on the Monte Carlo method for Markov chains has been improved.

A comparative analysis of the obtained forecasts with the results of the use of known approaches and methods was carried out. According to the results of all practical experiments, it was shown that the use of the developed AR-CNN approach allows for obtaining adequate models and accurate forecasts with relative ease of construction and small computational losses.

Keywords: mathematical modeling, forecasting, intellectual data analysis, machine learning, financial and economic processes, nonlinear processes, nonstationary processes, time series, anomaly detection, filtering, regression models, neural networks, recurrent neural networks, convolutional neural networks, system analysis, adaptive modeling.

List of principal publications of the applicant:

1. Белас О.М., Бідюк. П.І., Белас А.О. Порівняльний аналіз авторегресійних підходів та рекурентних нейронних мереж для моделювання і прогнозування нелінійних нестаціонарних процесів. *Information Technology and Security*. 2019 January-June.Vol. 7, No 1. P. 91-99.

В роботі здобувачем розглянуто найбільш вживані підходи до розв'язання задачі прогнозування нелінійних нестаціонарних процесів на даний момент часу, які вважаються найбільш ефективними для роботи з даними, представленими у вигляді послідовностей: авторегресійні моделі та рекурентні нейронні мережі. Виконаний їх порівняльний аналіз, виявлено переваги та недоліки кожного з методів. Сформульована необхідність у розробці нового підходу.

2. Бідюк П.І., Белас А.О. Вибір критерію якості для оцінювання прогнозів нелінійних нестаціонарних процесів. *Наукові вісті КПІ*. 2021. № 2. С. 38-45.

У роботі здобувачем показано, що оптимізуючи модель стосовно деякого вибраного критерію, модель стає гіршою відносно іншого критерію, тому важливо розуміти яку метрику необхідно використовувати для оптимізації та оцінювання якості прогнозу у поставленій задачі.

Сформульовано критеріальну базу для оцінки якості прогнозів нелінійних нестаціонарних процесів, а також підхід до вибору критерію якості з урахуванням особливостей поставленої задачі прогнозування.

3. Белас О.М., Белас А.О. Загальна методика прогнозування нелінійних нестаціонарних процесів на основі математичних моделей з використанням статистичних даних. *Системні дослідження та інформаційні технології.* 2021. № 1. С. 79-86.

В роботі здобувачем сформульовано й описано загальну методику прогнозування нелінійних нестаціонарних процесів та її основні етапи. Показана досліджена методика для автоматичної побудови регресійних моделей, сформульовані проблеми існуючої методики для побудови моделей на основі нейронних мереж. Сформульована проблема єдиного критерію якості прогнозів. Побудовані нові моделі для прогнозування вибраних процесів і у формі регресії та нейронних мереж типу LSTM, показані їх переваги та недоліки.

4. Danilov V., Belas A., Bidyuk P., Gozhyj O., Kalinina I., Jirov O. Adaptive forecasting and risk estimation. *Системні дослідження та інформаційні технології.* 2020. № 1. С. 34-53.

У роботі здобувачем запропоновано системну методологію для побудови системи підтримки прийняття рішень для адаптивного математичного моделювання та прогнозування сучасних економічних та фінансових процесів. Методологію застосовано до розв'язання задачі оцінювання кредитного ризику, що базується на таких принципах системного аналізу: ієрархічна структура системи з урахуванням ймовірнісних та статистичних невизначеностей; наявність процедур адаптації моделі, генерування кількох альтернатив прийняття рішень та відстеження обчислювальних процесів на всіх етапах обробки даних за допомогою відповідних наборів критеріїв статистичної якості (відомих або нещодавно введених).

5. Belas A., Bidyuk P. Convolutional neural networks for modeling and forecasting nonlinear nonstationary processes. *ScienceRise*. 2021. No 3. P. 12-20.

У роботі здобувачем запропоновано методологію використання згорткових нейронних мереж для моделювання та прогнозування нелінійних нестаціонарних процесів, представлених у вигляді даних часових рядів. На реальних статистичних даних побудовано модель за допомогою використання згорткових нейронних мереж. Дослідження показує результати, у яких згорткові мережі перевершують загальновживані методи з точки зору точності та складності. Вдалося побудувати більш точну модель із набагато меншою кількістю параметрів. Це вказує на те, що одновимірні згорткові нейронні мережі можуть бути цілком розумним вибором для вирішення задачі прогнозування часових рядів.

6. Belas A. O., Gavrylenko O. V., Bidyuk P. I. Bayesian modeling and forecasting for nonlinear nonstationary processes. Проблеми інфокомунікацій: матеріали Третьої всеукраїнської науково-технічної конференції, м.Полтава, Мінськ 12-2019.

Здобувачем розроблено програму байєсівського моделювання і прогнозування для дослідження характеристик нелінійних нестаціонарних процесів.

7. Белас А.О., Бідюк П.І. Застосування фільтра Калмана для моделювання та прогнозування нелінійних нестаціонарних процесів. Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів управління : тези доп. 10-ї міжнар. наук.-техн. конф. (Баку–Харків–Жиліна, 9-10 квітня 2020 р.). Т. 2, секції 3, 4. Військ. акад. збройних сил Азербайджанської Республіки (та ін.). Харків : Петров В. В., 2020. 92 с.

Здобувачем розроблено алгоритм оптимальної фільтрації, виконано імітаційне моделювання фільтра.

8. Белас А.О. Згорткові нейронні мережі для моделювання та

прогнозування нелінійних нестаціонарних процесів. Радіоелектроніка і молодь у XXI столітті: матеріали 24-й Міжнародного молодіжного форуму, м.Харків, 7 – 9 квітня 2020 р., секція 6. Зб. матеріалів 24-го Міжнар. молодіж. форуму, Харків: ХНУРЕ, 2020. Т. 11. 68 с.

Здобувачем запропоновано підхід для моделювання та прогнозування нелінійних нестаціонарних процесів за допомогою згорткових нейронних мереж, розроблено відповідну аналітичну програму, проведено експерименти, побудовано нові моделі для прогнозування.

ЗМІСТ

АНОТАЦІЯ 2
SUMMARY 7
ЗМІСТ 15
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ 18
ВСТУП 19
РОЗДІЛ 1 25
АКТУАЛЬНІСТЬ ТА ЗАГАЛЬНА МЕТОДИКА 25
1.1. Актуальність задачі моделювання та прогнозування нелінійних нестаціонарних процесів 25
1.2. Аналітичні підходи для розв’язання задачі прогнозування нелінійних нестаціонарних процесів 29
1.3. Загальна методика прогнозування нелінійних нестаціонарних процесів на основі методів інтелектуального аналізу даних з використанням статистичних даних 33
1.4. Існуючі програмні засоби для аналізу та прогнозування нелінійних нестаціонарних процесів 37
1.4.1. SAS 37
1.4.2. Мова програмування R 39
1.4.3. Мова програмування Python 41
Висновки до розділу 44
РОЗДІЛ 2 45
МАТЕМАТИЧНІ МОДЕЛІ, МЕТОДИ ТА ПІДХОДИ ДЛЯ ОПИСУ, АНАЛІЗУ ТА ОБРОБКИ НЕЛІНІЙНИХ НЕСТАЦІОНАРНИХ ПРОЦЕСІВ 45
2.1. Математичні моделі, що використовуються для опису динаміки фінансово-економічних процесів 45
2.2. Встановлення належності стохастичних процесів до певного класу 55
2.2.1. Статистичний та графічний аналіз 55

2.2.2. Нейромережевий тест Уайта для перевірки нелінійності часового ряду 58
2.2.3. Тест KPSS для перевірки часового ряду на нестаціонарність ... 64
2.3. Методи обробки пропущених та аномальних значень 66
2.4. Перетворення та фільтрація даних з метою вилучення шумових складових 74
Висновки до розділу 80
РОЗДІЛ 3 81
МАТЕМАТИЧНІ МОДЕЛІ І МЕТОДИ ДЛЯ МОДЕЛЮВАННЯ І ПРОГНОЗУВАННЯ НЕЛІНІЙНИХ НЕСТАЦІОНАРНИХ ПРОЦЕСІВ 81
3.1. Регресійні моделі для виділення та моделювання лінійної складової процесу 81
3.2. Прогнозування за допомогою AR моделі 91
3.3. Критерії адекватності моделі AR для виділення та моделювання лінійної складової процесу 93
3.4. Авторегресивний підхід до моделювання і прогнозування нестаціонарних процесів 95
3.5. Нейронні мережі 99
3.6. Алгоритми оптимізації параметрів для навчання нейронної мережі 113
3.7. Рекурентні нейронні мережі 121
3.8. Згорткові нейронні мережі 136
3.9. Прогнозування за допомогою нейронних мереж 146
3.10 Методи комбінування оцінок прогнозів. Комбінування лінійної та нелінійної частини процесу 149
3.11. Вибір критерію якості для оцінювання прогнозів нелінійних нестаціонарних процесів 151
3.12. Адаптивна побудова моделей прогнозування 164
3.13. Метод Монте-Карло марківських ланцюгів для адаптивної побудови моделей 167

Висновки до розділу 172
РОЗДІЛ 4..... 174
ЕКСПЕРИМЕНТИ З ПОБУДОВИ МОДЕЛЕЙ ТА ПРОГНОЗІВ ВИБРАНИХ ПРОЦЕСІВ..... 174
4.1. Прогнозування попиту на товари Walmart..... 174
4.2. Побудова моделей з продажу ліків 184
4.3. Побудова моделей процесу з продажу палива..... 194
Висновки до розділу 204
ВИСНОВКИ 206
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ..... 208

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ННП – нелінійні нестаціонарні процеси

LOESS – локальна поліноміальна регресія

АКФ – автокореляційна функція

ЧАКФ – часткова автокореляційна функцію

AR(p) – авторегресійна модель порядку p.

МНК – метод найменших квадратів

SSE – сума квадратів залишків моделі

AIC – інформаційний критерій Акайке

BIC – інформаційний критерій Байєса-Шварца

ARMA – модель авторегресії з ковзним середнім

ARIMA – модель авторегресії з інтегрованим ковзним середнім

MSE – середньоквадратична похибка моделі

SGD – стохастичний градієнтний спуск

LSTM – нейронна мережа с коротко- та довгостроковою пам’яттю

CNN – згорткова нейронна мережа

RNN – рекурентна нейронна мережа

AR-CNN – комбінована модель авторегресії та згорткової нейронної мережі

RMSE – корінь із середньоквадратичної похибки моделі

MAE – середня абсолютна похибка

MAPE – середня абсолютна похибка у відсотках

RMSLE – середньоквадратична логарифмічна похибка

СППР - система підтримки прийняття рішень

МКМЛ - метод Монте-Карло марківських ланцюгів

ВСТУП

Актуальність роботи. Нелінійні нестаціонарні процеси, представлені у вигляді часових рядів, можуть собою описувати динаміку процесів в технічних, економічних та інших системах і мають численні застосування в різних галузях. Задача моделювання та прогнозування нелінійних нестаціонарних фінансово-економічних процесів є критично важливою для кожного підприємства, що займається торгівлею або інвестиційною діяльністю оскільки є невід'ємною частиною планування закупівлі, використання ресурсів, прийняття стратегічних рішень. Прогнозування є одним з найважливіших елементів ефективної організації управління підприємствами, оскільки якість прийнятих рішень багато в чому визначається якістю прогнозування їх наслідків. В роботі вибрано і розглянуто фінансово-економічні процеси, проте розроблену методику можливо застосовувати і в інших системах з відповідною визначеною динамікою. У прогнозуванні використовуються математичні методи, іноді досить складні та різноманітні залежно від конкретного завдання. Швидкий науково-технічний прогрес робить прогнозування швидким, точним та ефективним механізмом аналізу. На даному етапі підходи до розв'язання поставленої задачі методами інтелектуального аналізу даних знаходяться в стані активного дослідження, причому загальноприйнятої методики не існує.

Мета і задачі наукового дослідження. Метою дисертаційної роботи є підвищення адекватності моделей нелінійних нестаціонарних фінансово-економічних процесів і точності прогнозів їх розвитку шляхом застосування сучасних методів інтелектуального аналізу даних та їх комбінуванням з регресійними моделями.

Для досягнення поставленої мети необхідно розв'язати такі задачі:

- сформулювати задачу моделювання і прогнозування нелінійних нестаціонарних процесів, вибрати типи процесів для дослідження, зібрати статистичні дані;

- сформулювати і описати загальну методику та її основні етапи для моделювання і прогнозування нелінійних нестаціонарних процесів на основі математичних моделей з використанням статистичних даних;

- вибрати нелінійні елементи математичних моделей для опису нелінійних складових досліджуваних процесів, виконати огляд існуючих методів і моделей для формалізованого опису нелінійних нестаціонарних процесів, виконати їх порівняльний аналіз;

- сформулювати критеріальну базу для дослідження процесів на нелінійність та нестаціонарність;

- сформулювати критеріальну базу для оцінки якості оцінок прогнозів нелінійних нестаціонарних процесів, а також підхід до вибору критерію якості з урахуванням особливостей поставленої задачі прогнозування;

- проаналізувати наявні моделі і методи для моделювання і прогнозування нелінійних нестаціонарних процесів, визначити їх переваги та недоліки;

- запропонувати шляхи подальшого удосконалення існуючих моделей та методів за допомогою їх комбінування;

- запропонувати та використати нові методи інтелектуального аналізу даних стосовно розв’язання поставлених задач;

- застосувати методику на реальних даних, виконати експериментальне дослідження та провести порівняльний аналіз отриманих прогнозів з результатами відомих моделей та методів.

Об’єкт дослідження: нелінійні нестаціонарні фінансово-економічні процеси, представлені статистичними даними.

Предмет дослідження: математичні моделі та методи моделювання і прогнозування нелінійних нестаціонарних процесів на основі регресійного

підходу та методів інтелектуального аналізу даних (рекурентні та згорткові нейронні мережі різного типу).

Методи дослідження: математична статистика, чисельні методи, методи оптимізації та дослідження операцій, теорія прийняття рішень, методи регресійного аналізу, методи аналізу часових рядів, методи інтелектуального аналізу даних, методи обчислювального інтелекту.

У ході дослідження використовувалась мова програмування Python, фреймворк для чисельних методів оптимізації та нейронних мереж PyTorch.

Наукова новизна роботи. Наукова новизна дисертаційної роботи полягає у такому:

- **вперше** запропоновано, сформульовано і описано загальну методику для моделювання і прогнозування нелінійних нестаціонарних процесів на основі математичних моделей інтелектуального аналізу даних з використанням статистичних даних, в тому числі критеріальну базу, методи обробки і підготовки даних до моделювання і прогнозування, що є основою для створення СППР для аналізу ННП;

- **вперше** запропоновано метод побудови моделей нелінійних процесів, який відрізняється застосуванням окремих процедур для оптимізації структури лінійної та нелінійної складових моделі з їх наступним адитивним об’єднанням в єдину модель, що забезпечує підвищення адекватності моделі і точності оцінок прогнозів у цілому;

- **вперше** запропоновано використання нових для вибраного типу процесів моделей та їх модифікацій на основі методів інтелектуального аналізу даних, а саме згорткових нейронних мереж, для формального опису і прогнозування ННП, побудовано відповідні нові моделі та прогнози;

- **удосконалено** існуючий підхід з використанням класичних статистичних моделей на основі регресійного підходу шляхом комбінування їх з методами інтелектуального аналізу даних;

- **удосконалено** метод оцінювання параметрів лінійних моделей для адаптації на основі методу Монте-Карло для марківських ланцюгів.

Обґрунтованість і достовірність наукових результатів забезпечується коректним застосуванням математичного апарату та відповідних обчислювальних експериментів, з використанням сучасного програмного забезпечення. Забезпечується також використанням методів інтелектуального аналізу даних, математичної статистики, методів аналізу часових рядів, методів регресійного аналізу, а також процедурами контролю точності проміжних та остаточних результатів, що оцінювались на незалежних тестових вибірках.

Практичне значення отриманих результатів.

Результати роботи впроваджено у навчальний процес інституту прикладного системного аналізу НТУУ «КПІ імені Ігоря Сікорського».

Всі результати дисертаційного дослідження доведено до високого практичного інженерного рівня та впроваджено у АТ «ПУМБ» з метою покращення процесів прийняття рішень та побудови стратегій розвитку. Створено систему підтримки прийняття рішень, що побудована на основі оцінок прогнозів за допомогою розробленої загальної методики з використанням методів інтелектуального аналізу даних. За допомогою розробленої системи і методики фахівці ПУМБ будують моделі і прогнози для фінансово-економічних процесів на основі аналізу економетричних даних, якими володіє банк в своїх системах.

Всі теоретичні і практичні результати дисертаційної роботи у повній мірі опубліковано у фахових вітчизняних та закордонних наукових виданнях, що входять до відповідного встановленого переліку, а також виконано їх належну апробацію на міжнародних наукових конференціях і семінарах.

Особистий внесок здобувача. Всі основні результати дисертаційної роботи отримані автором особисто. Всі 5 фахових публікацій написано у співавторстві, а здобувачеві належать такі результати.

В роботі [1] здобувачем розглянуто найбільш вживані підходи до розв’язання задачі прогнозування нелінійних нестаціонарних процесів на даний момент часу, які вважаються найбільш ефективними для роботи з даними, представленими у вигляді послідовностей: авторегресійні моделі та рекурентні нейронні мережі. Виконаний їх порівняльний аналіз, виявлено переваги та недоліки кожного з методів. Сформульована необхідність у розробці нового підходу.

В роботі [2] здобувачем сформульовано й описано загальну методику прогнозування нелінійних нестаціонарних процесів та її основні етапи. Показана досліджена методика для автоматичної побудови регресійних моделей, сформульовані проблеми існуючої методики для побудови моделей на основі нейронних мереж. Сформульована проблема єдиного критерію якості прогнозів. Побудовані нові моделі для прогнозування вибраних процесів і у формі регресії та нейронних мереж типу LSTM, показані їх переваги та недоліки.

У роботі [3] запропоновано системну методологію для побудови системи підтримки прийняття рішень для адаптивного математичного моделювання та прогнозування сучасних економічних та фінансових процесів. Методологію застосовано до розв’язання задачі оцінювання кредитного ризику, що базується на таких принципах системного аналізу: ієрархічна структура системи з урахуванням ймовірнісних та статистичних невизначеностей; наявність процедур адаптації моделі, генерування кількох альтернатив прийняття рішень та відстеження обчислювальних процесів на всіх етапах обробки даних за допомогою відповідних наборів критеріїв статистичної якості (відомих або нещодавно введених).

У роботі [4] показано, що оптимізуючи модель стосовно деякого вибраного критерію, модель стає гіршою відносно іншого критерію, тому важливо розуміти яку метрику необхідно використовувати для оптимізації та оцінювання якості прогнозу у поставленій задачі. Сформульовано

критеріальну базу для оцінки якості прогнозів нелінійних нестаціонарних процесів, а також підхід до вибору критерію якості з урахуванням особливостей поставленої задачі прогнозування.

У роботі [5] запропоновано методологію використання згорткових нейронних мереж для моделювання та прогнозування нелінійних нестаціонарних процесів, представлених у вигляді даних часових рядів. На реальних статистичних даних побудовано модель за допомогою використання згорткових нейронних мереж. Дослідження показує результати, у яких згорткові мережі перевершують загальновживані методи з точки зору точності та складності. Вдалося побудувати більш точну модель із набагато меншою кількістю параметрів. Це вказує на те, що одновимірні згорткові нейронні мережі можуть бути цілком розумним вибором для вирішення задачі прогнозування часових рядів.

Апробація результатів дисертації. Результати та основні положення роботи подавалися та обговорювалися на:

- Міжнародній науково-технічній конференції “Проблеми інформатизації”, (Полтава-Мінськ), 12-2019 [6];
- Міжнародній науково-технічній конференції “Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів управління”, (Баку – Харків – Жиліна), 9 – 10 квітня 2020 р [7];
- 24-му Міжнародному молодіжному форумі «РАДІОЕЛЕКТРОНІКА І МОЛОДЬ У XXI СТОЛІТТІ» [8].

Публікації. За матеріалами дисертації опубліковано 5 робіт, всі з яких - статті у журналах і збірниках наукових праць, що входять до переліку фахових видань затверджених МОН України за спеціальністю дисертації або у періодичних виданнях іноземних держав.

РОЗДІЛ 1
АКТУАЛЬНІСТЬ ТА ЗАГАЛЬНА МЕТОДИКА

1.1. Актуальність задачі моделювання та прогнозування нелінійних нестаціонарних процесів

Нелінійні нестаціонарні процеси, представлені у вигляді часових рядів, можуть собою описувати динаміку процесів в різних системах і має численні застосування в таких галузях:

- в енергетиці, для опису процесу виробництва та формування попиту на електроенергію, газ та інші ресурси (Рис. 1.1);

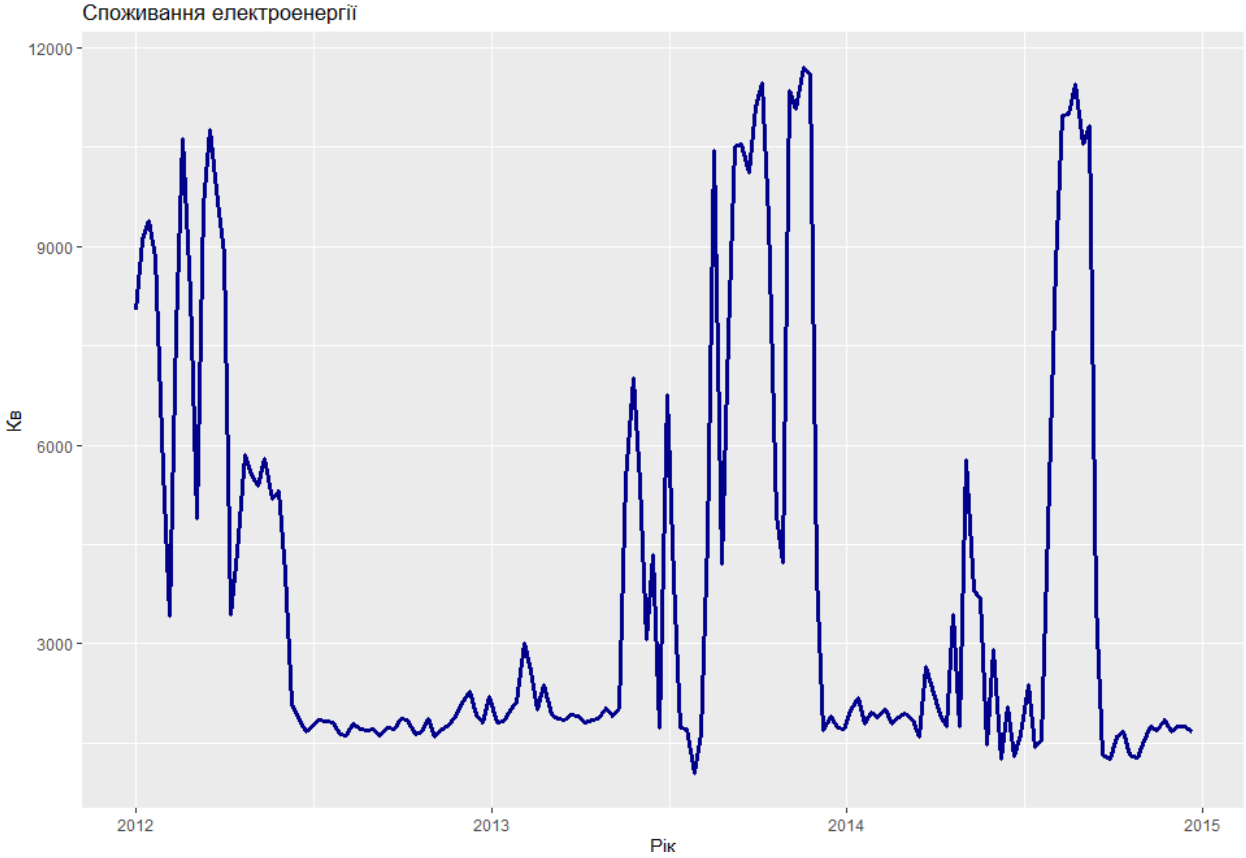


Рис. 1.1 Приклад часового ряду. Споживання електроенергії в Австралії (штат Тасманія)

- в технічних системах, для аналізу навантажень в мережі, пропускної здатності каналів, ємності буферів, для опису сигналів з датчиків системи

інтернету речей [9] на транспортних засобах, що може бути корисно для передбачення майбутніх критичних станів в системі та прийняття рішення про превентивні ремонти;

- в соціально-економічних системах для опису процесу формування попиту на товари, продажів у великому та малому бізнесі, активності на веб-сайтах торгівельних підприємств, опису динаміки обсягів виробництва та накопичення продукції на складах, що має важливу роль в формуванні бюджетів підприємств та оцінювання альтернативних економічних стратегій [10];

- в фінансових системах та інвестиційній діяльності для опису динаміки процесів на фінансових ринках, таких як ціни на акції та всі можливі похідні інструменти (опціони, деривативи та ін.), що є важливим для прогнозування та менеджменту ризиків довільної природи (Рис. 1.2);

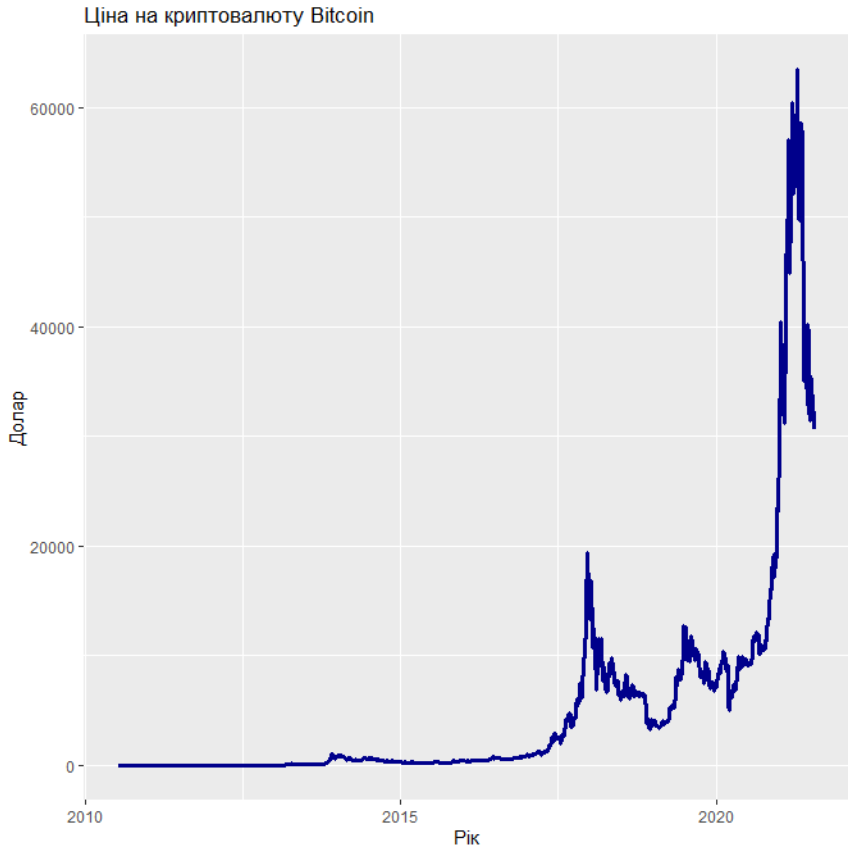


Рис. 1.2 Приклад часового ряду. Ціна на криптовалюту Bitcoin

- в медицині власне люди є великим джерелом біосигналів: активність мозку (ЕЕГ), серцева діяльність (ЕКГ), м’язова напруга (ЕМГ), дані з таких носіїв, як пульсометр, активність на основі акселерометра, пристрої моніторингу сну, індекси стресу - все це дуже важливі життєві сигнали які описуються нелінійними нестаціонарними процесами і їх також дуже важливо аналізувати (Рис. 1.3);

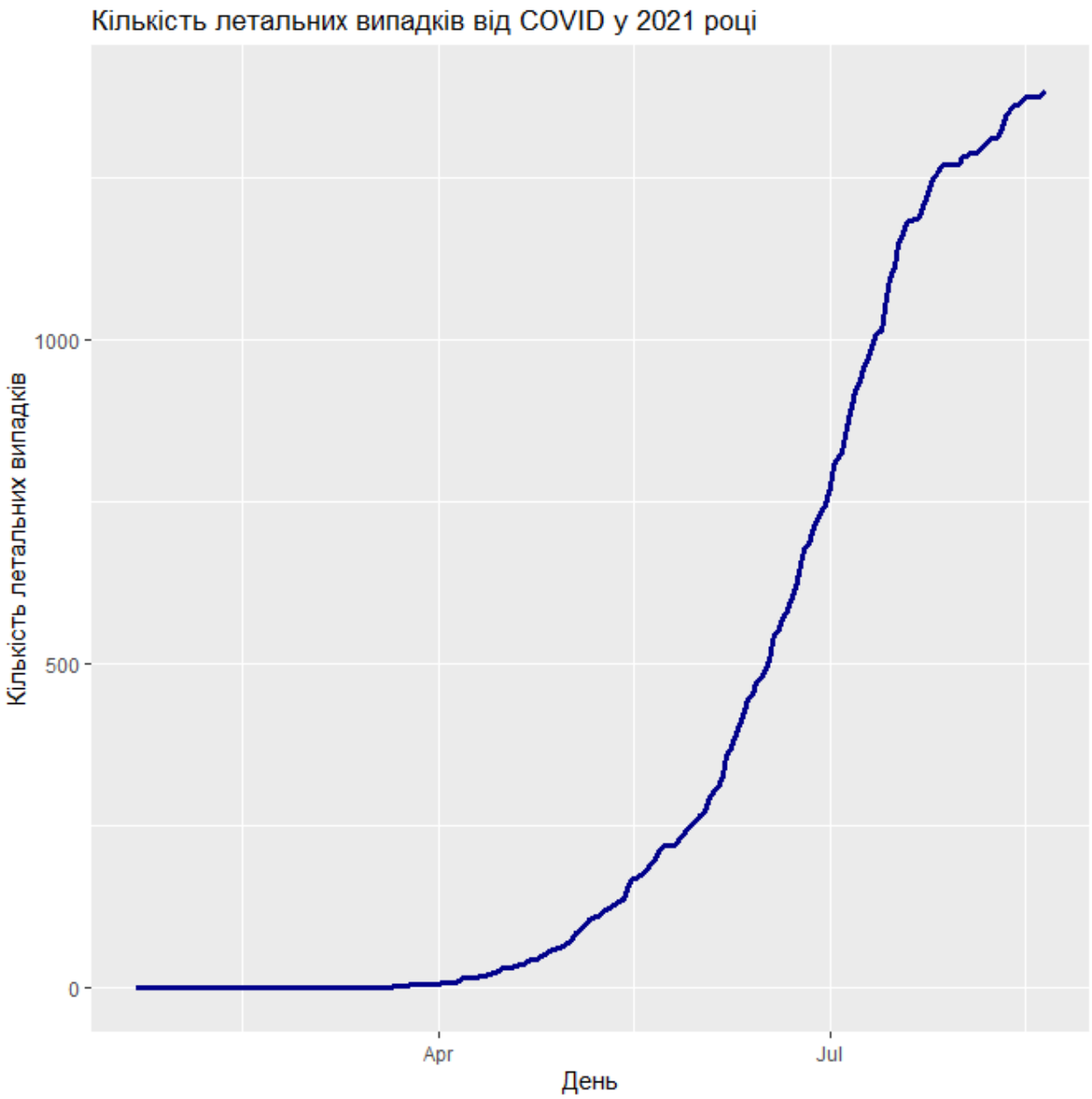


Рис. 1.3 Приклад часового ряду. Кількість летальних випадків від COVID у 2021 році

- в екології, оскільки планета Земля та космічні тіла навколо неї є джерелами сигналів - ми вимірюємо кількість та інтенсивність сонячних плям, зміни температури в різних регіонах, швидкість вітру, швидкість астероїдів та багато іншого, що також є прикладами вибраних процесів (Рис 1.4).

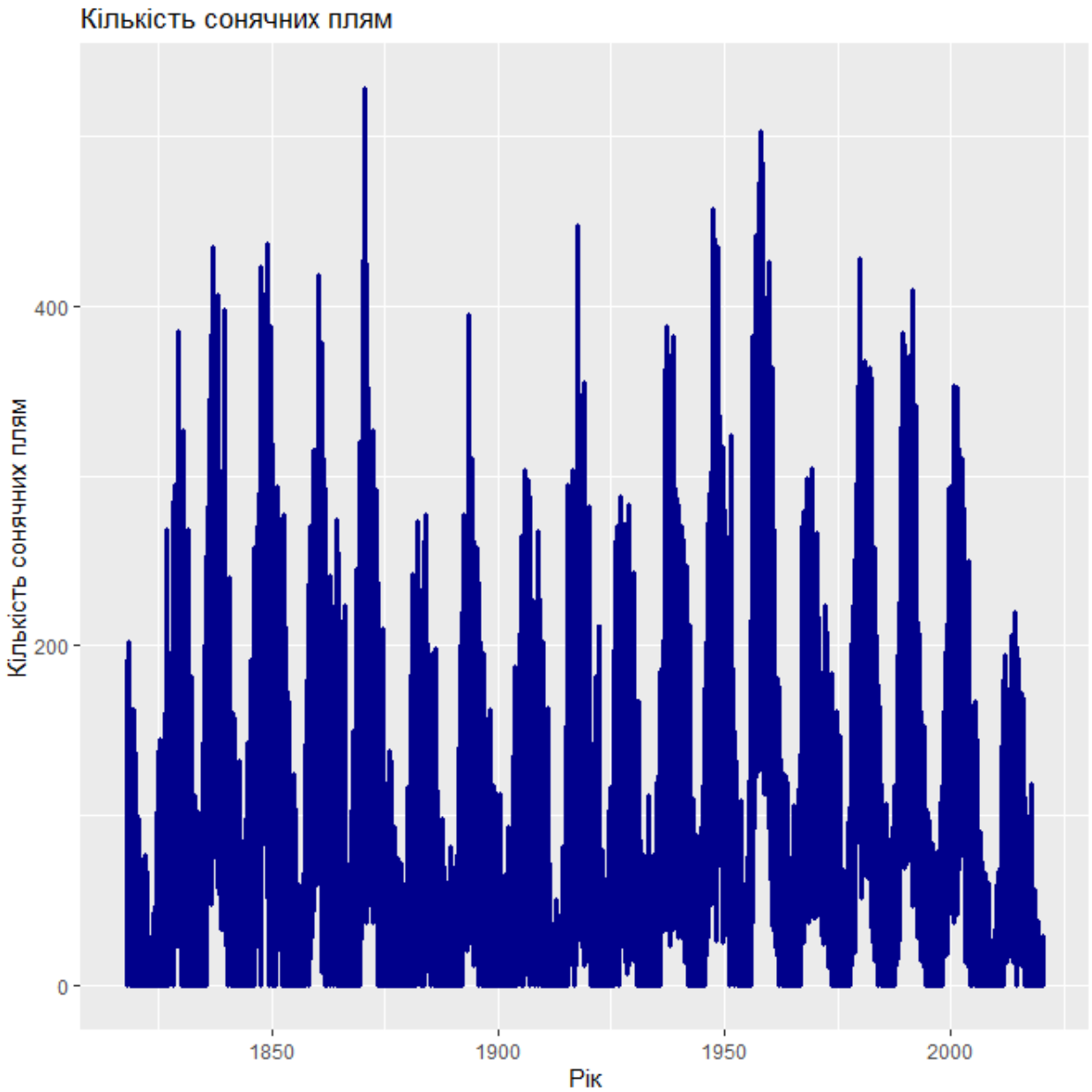


Рис. 1.4 Приклад часового ряду. Кількість сонячних плям

В даній роботі буде вибрано і розглянуто фінансово-економічні процеси, проте розроблену методику можливо застосовувати і в інших системах з відповідною визначеною динамікою.

Всі підприємства, що займаються роздрібною торгівлею мають необхідність в прийнятті рішень про планування закупівель та використання ресурсів. Ці рішення є досить важливі, проблема в тому, що якщо закупити недостатньо товарів, це може привести до дефіциту товарів на полицях і до втраченої вигоди, якщо товари будуть користуватися великим попитом. Занадто багато замовлених товарів вимагає відповідних витрат на зберігання, а при недостатньому попиті зайві товари можуть зіпсуватися, що призведе до збитків для компанії. Тому постає задача прогнозування попиту для підримки прийняття рішень щодо закупівель.

В усіх наведених процесах та задачах, бачимо, що прогнозування - невід'ємна складова аналітичної роботи, так як дозволяє розглянути найбільш ймовірний розвиток подій, а також визначати, які заходи впливу приведуть до відповідних результатів. Прогнозування також є одним з найважливіших елементів ефективної організації управління підприємствами, оскільки якість прийнятих рішень багато в чому визначається якістю прогнозування їх наслідків.

Отже, задача прогнозування нелінійних нестаціонарних процесів має численну кількість застосувань як в технічних так і в економічних системах і є дуже актуальною на сьогоднішній день для підтримки прийняття рішень в управлінні цими системами.

1.2. Аналітичні підходи для розв’язання задачі прогнозування нелінійних нестаціонарних процесів

Історично формувалися різні підходи для розв’язання задач прийняття рішень в складних системах:

1. Експертний. Для прийняття відповідних рішень експерт або група експертів з великим досвідом в вибраній галузі на базі своїх знань, досвіду та логічного мислення приходили до розв’язання поставленої задачі, як то:

медична діагностика, прийняття стратегічних рішень в бізнесі, керування технічними системами тощо. Такий підхід являє собою прогнозування на основі суджень, тобто, прогнозування на основі суб'єктивних оцінок, інтуїції, глибоких знань в конкретній галузі та іншої інформації, що відноситься до прогнозованого процесу. Однак, з плином часу було відзначено суттєві недоліки використання лише цього підходу через в певній мірі бездоказовість прийнятих рішень, неможливість системно їх вивчати та покращувати використовуючи наукові математичні методи [11]. Тому із потреби використовувати широкий науковий математичний апарат а також системні підходи для розв’язання задач з прийняття рішень виникли наступні ідеї.

2. Прескриптивна аналітика. Для використання математичних методів для розв’язання задач необхідно мати відповідні статистичні дані. Збираючи статистичні дані для обраних процесів з подальшим їх історичним аналізом отримав назву “прескриптивна аналітика”, а в застосуванні її до відповідних задач бізнесу - “Business Intelligence” [12]. В процесі використання методів прексриптивної аналітики, а саме статистичного аналізу, було відзначено високу ефективність прийняття рішень на основі даних [13]. Однак, зі стрімким ростом кількості даних стало неможливо обробляти їх існуючими засобами. Проте зі стрімким ростом кількості даних відбувся стрімкий ріст інформаційних технологій та систем [14], а тому стало можливо використовувати програмні засоби для реалізації математичних методів для аналізу вибраних процесів. З об’єднанням підходів програмної інженерії, математичних методів, та практик відповідної галузі вибраних процесів отримали міждисциплінарну науку, що є потужним сучасним інструментом для прийняття рішень - Data Science [13] (Рис 1.5).

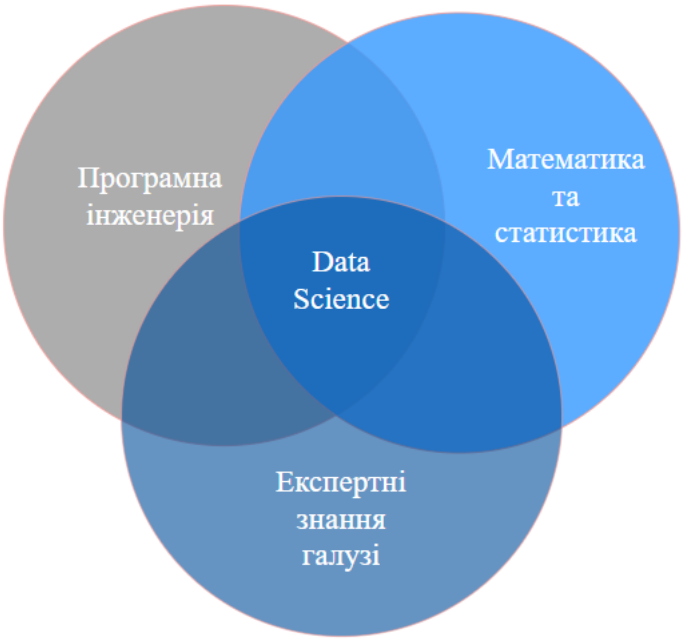


Рис. 1.5 Міждисциплінарна наука - Data Science

Використання інформаційних технологій також сприяло росту методів інтелектуального аналізу даних, які вдалося використати в Data Science для переходу на наступний рівень прийняття рішень.

3. Предиктивна аналітика. З відзначеною ефективністю підходу прийняття рішень на основі даних математичними методами та подальшим розвитком підходів та технологій виникла ідея не просто аналізувати історичні дані з подальшим експертним прийняттям рішень, а поглибити цю ідею використовуючи математичні методи для прогнозування - методи інтелектуального аналізу даних [15], або Machine Learning [16]. Різниця підходів з використанням методів інтелектуального аналізу даних від класичного програмування або експертного прийняття рішень полягає в тому, що у випадку класичного програмування чи в експертній системі на основі даних та заздалегідь сформованих експертних правил відбувається прогнозування та приймаються ті чи інші рішення. Система що використовує методи інтелектуального аналізу даних, може за допомогою історичних даних

та правильних відповідей щодо майбутнього відтворити правила або алгоритм для їх прогнозування та прийняття рішень (Рис. 1.6).

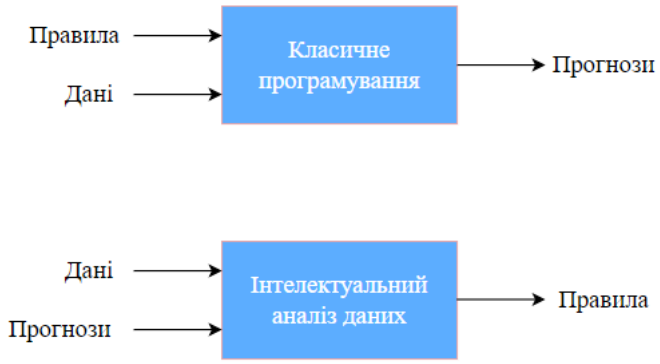


Рис. 1.6 Різниця підходів з використанням методів інтелектуального аналізу від класичного програмування

Тепер ми можемо математично не лише проаналізувати, що було в минулому, а виконати прогнозування, що значно спрощує прийняття рішень та покращує їх якість. Знаючи, або з достатньою точністю та надійністю прогнозуючи, майбутнє можна прогнозувати наслідки тих чи інших рішень, що значно покращує процес їх прийняття (Рис 1.7).

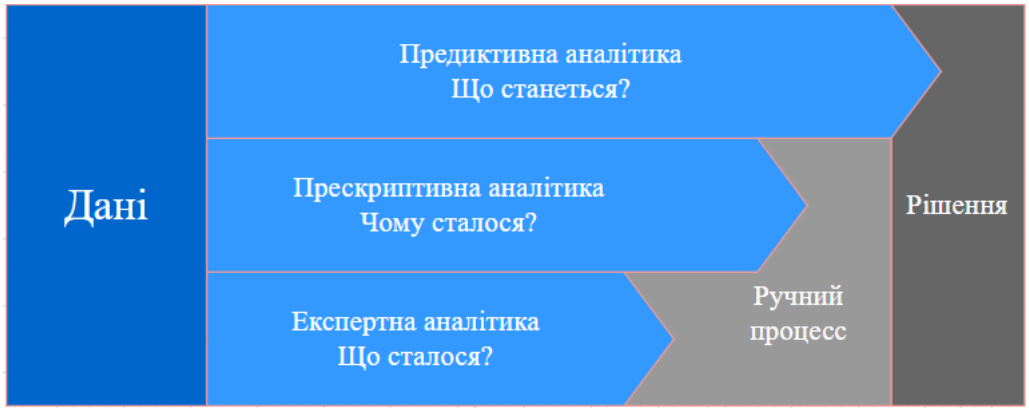


Рис. 1.7 Процес прийняття рішень

Отже, в даній роботі буде розглянуто саме підхід предиктивної аналітики з використанням методів інтелектуального аналізу даних для задачі прогнозування нелінійних нестаціонарних процесів.

1.3. Загальна методика прогнозування нелінійних нестаціонарних процесів на основі методів інтелектуального аналізу даних з використанням статистичних даних

Основи першої методики побудови моделей нелінійних нестаціонарних процесів, представлених у вигляді часових рядів були запропоновані Боксом і Дженкінсом у роботі [17]. Модифікована авторами методика побудови математичної моделі процесу, складається з декількох кроків і поклала основу процесу моделювання та прогнозування часових рядів. З розвитком математичного моделювання, появи нових задач і методів їх розв’язання, методика модифікувалась та покращувалась.

На сьогоднішній день існує кілька методик побудови таких математичних моделей: KDD, SEMMA, CRISP-DM [18].

В даній роботі пропонується використовувати сучасну аналітичну методологію SEMMA (Sampling, Exploring, Modifying, Modeling, Assessing) [19]. Дана методологія добре орієнтована на побудову моделей процесів, для яких, як правило, набагато складніше поставити експеримент та отримати інформативні експериментальні дані в достатньому обсязі. Також методологія є найбільш орієнтованою саме на процес дослідження, більше фокусуючись на побудові моделей, а не на бізнес-розумінні або інтеграції з технічними системами.

Розглянемо основні етапи запропонованої методики, та їх особливості.

1. Sampling – відбір і завантаження даних в проект. На даному етапі розробляється стратегія збору, зберігання та розмітки даних, проводиться ретельний пошук та відбір необхідних даних для розв’язання задачі, завантаження їх у систему, розмітка даних, приведення до найбільш зручного вигляду, об’єднання даних в остаточну базу даних та знань до роботи.

2. Exploring – розуміння суті даних, пошук трендів, аномалій та взаємозв’язків. Для цього використовується візуалізація та статистичний аналіз

даних, що первинно дозволяє виявити аномальні значення або викиди в даних, а також візуально оцінити наявність тренду, нелінійностей чи інших особливостей або взаємозв'язків в даних. Далі викиди або аномальні значення можна перевірити за допомогою відповідних статистичних тестів. Виконується аналіз даних на можливу наявність нелінійностей за допомогою множини статистичних критеріїв. Використовуються тести на тренд та на гетероскедастичність для розуміння належності досліджуваного процесу до певного класу нестаціонарності [20].

3. Modifying – підготовка даних до моделювання, модифікування та обробка даних даних шляхом створення, вибору та перетворення змінних, застосування ймовірнісної або числової фільтрації. Остаточне перетворення вихідних даних, розбиття вибірки на навчальну і тестову. Правильний вибір тестової вибірки є важливою задачею та запорукою коректного навчання моделі.

Формально, можна виділити наступні операції на цьому етапі:

– коригування даних полягає у заповненні пропусків, зменшенні викидів, роботі з аномальними значеннями, що виходять за основний діапазон значень змінних;

- ортогональні перетворення та цифрова або ймовірнісна фільтрація даних з метою вилучення шумових складових (за необхідності);

– нормування даних: їх логарифмування, стандартизація, або приведення до зручного діапазону їх зміни, наприклад, від 0 до 1; від – 1 до +1; від +10 до –10 і т.і.;

- робота з категоріальними змінними, перетворення їх до числового вигляду або відповідна обробка для можливості подання відповідних даних до моделювання;

- робота з часовими ознаками, виділення додаткових часових факторів з наявних дат, наприклад: рік, місяць, сезон, день тижня, свята або вихідні, тощо.

Також можливе формування лагових ознак з інших змінних, що будуть описувати їх стан певну кількість часу тому.

4. Modeling – побудова вибраних математичних моделей, що дають бажаний результат. На даному етапі пропонується виконати розділення моделі на лінійну та нелінійну складову, виділення та оцінювання структури, оптимізація параметрів лінійної моделі, аналіз залишків лінійної моделі, підбір і побудова на залишках моделі, що буде описувати нелінійну складову. Далі виконується комбінування моделей та формування повної моделі методом бустінгу [21].

5. Assessing – використання критеріїв для оцінки якості прогнозів отриманих в процесі моделювання та дослідження, побудова графіків. Вибір кращої з оцінених моделей-кандидатів. Застосування моделі до розв’язання основної задачі – прогнозування, керування, поглибленого дослідження процесу, моніторинг якості роботи моделі та остаточне встановлення її придатності (Рис. 1.8).

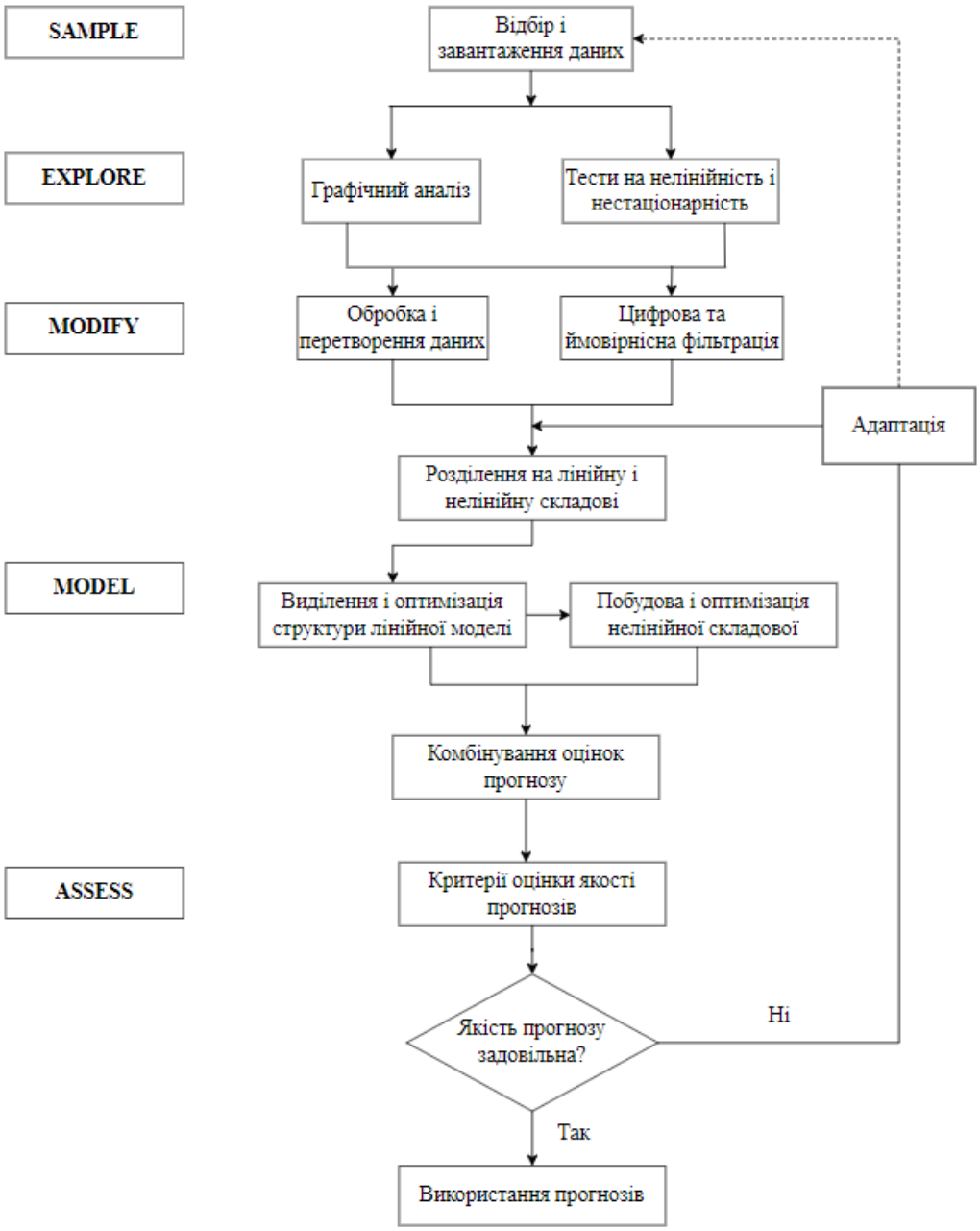


Рис. 1.8 Загальна методика прогнозування ННП на базі методології SEMMA

Важливо зазначити, що наведений процес є ітеративним і практичні задачі вимагають проходження його етапів багато разів. Наприклад, після оцінки якості прогнозів, можна зробити висновки, щодо покращення якості даних чи їх підготовки, або вибору додаткових факторів для залучення в модель, або зміни власне структури моделі з відповідним проходженням всіх етапів знову, доки якість моделей та прогнозів не буде задовільною.

1.4. Існуючі програмні засоби для аналізу та прогнозування нелінійних нестаціонарних процесів

Як було зазначено, для використання методів інтелектуального аналізу на великих обсягах даних необхідно мати як відповідні обчислювальні потужності, так і зручні програмні засоби, які можуть обробляти довільну інформацію та можуть зручно бути використані для побудови різноманітних моделей.

Розглянемо порівняльний аналіз найбільш популярних програмних рішень в області побудови моделей на сучасному ринку інформаційних технологій.

Порівнювати інструменти будемо як за загальними рисами, так і за наявністю зручних та потужних інструментів на кожному етапі визначеної нами загальної методики.

1.4.1. SAS

Система SAS об’єднує в собі як новітні технології інтеграції, обробки та збереження даних, так і найпотужніші аналітичні інструменти [22].

Компанія SAS представляє як готові рішення для розв’язання конкретних бізнес задач, як наприклад: SAS Credit Scoring for banking [23] або SAS Intelligent Planning Suite [24], так і аналітичні інструменти для роботи з будь-якими даними, як наприклад SAS Enterprise Miner [25]. В рамках дослідницької роботи розглянемо саме інструмент для роботи з даними та побудови моделей SAS Enterprise Miner. SAS Enterprise Miner - це інтегрований компонент системи SAS, призначений для виявлення інформації, необхідної для прийняття рішень на основі статистичних даних. Розроблений спеціально для пошуку та аналізу прихованих закономірностей в даних SAS Enterprise Miner включає в себе ефективні методи

інтелектуального аналізу даних і зручний графічний інтерфейс. Розглянемо можливості SAS Enterprise Miner на кожному етапі запропонованої загальної методики.

Sampling. Розглянемо можливості роботи з даними в SAS.

При роботі з даними SAS може працювати з різними типами даних:

- файли у різних форматах: *. dat, *. txt, *. csv, тощо;

- підключення до різних баз та систем обробки даних: Microsoft SQL Server, Oracle, DBase, Microsoft Excel, тощо;

Великою перевагою SAS рішень є широка можливість інтеграцій з різними системами. Однак, варто зазначити, що в рамках дослідження це не є суттєвою вимогою, оскільки зазвичай робота йде зі звичайними текстовими файлами. Після завантаження даних в систему, дані перетворюються в SAS набір.

Exploring. Для візуального аналізу даних в системах SAS є дуже багато зручних інструментів, що дозволяють побудувати різноманітні графіки, порахувати статистики, провести значний ряд тестів майже в автоматичному режимі без написання коду.

Modifying. В SAS представлено багато інструментів для коригування, фільтрації та перетворення даних, що покривають більшу частину аналітичних потреб в зручному графічному інтерфейсі.

Modeling. В SAS Enterprise Miner реалізовано багато потужних методів для моделювання та прогнозування. Але, незважаючи на наявність великої кількості популярних перевірених методів, найбільш сучасні моделі інтелектуального аналізу даних не будуть одразу після їх винаходу доступні в програмному комплексі SAS. Зі стрімким розвитком сфери інтелектуального аналізу даних щотижня створюються нові підходи, моделі та можливості, і, як правило, постачальникам із закритим кодом потрібен час для їх копіювання, тож треба буде чекати поки компанія реалізує їх та додасть в програму відповідний функціонал.

Assessing. В SAS реалізовані необхідні підходи для оцінки якості моделей. Варто зазначити дуже широкий функціонал для створення різноманітних графічних звітів для валідації моделей та для підтримки прийняття рішень.

Отже, загальними перевагами SAS є автоматизація та зручність великої кількості процесів завдяки функціональному графічному інтерфейсу, проте, таким чином користувач втрачає певну гнучкість та іноваційність на деяких етапах, наприклад, при виборі та налаштуванні моделей та параметрів. Дуже суттєвим фактором також є доволі висока ціна програмних рішень SAS.

1.4.2. Мова програмування R

R — це популярна мова та середовище для статистичних обчислень і графічного аналізу. R забезпечує широкий спектр статистичних (лінійне та нелінійне моделювання, класичні статистичні тести, аналіз часових рядів, класифікація, кластеризація) та графічних методів, що активно розвиваються [26].

Розглянемо можливості R на кожному етапі запропонованої загальної методики.

Sampling. Розглянемо можливості роботи з даними в R.

При роботі з даними R також може працювати з різними типами даних, як з файлами, так і з базами даних.

Для досліджень багато пакетів R містять популярні набори даних. Наприклад, base R поставляється з пакетом datasets, який пропонує широкий спектр простих, іноді дуже відомих наборів даних [27].

Для стандартних типів файлів, таких як .csv, base R має набір стандартних функцій для завантаження. Для більш складних типів, як наприклад .xls, існують відповідні пакети, наприклад readxl [28].

Для підключення до різних баз даних також існує широка екосистема, наприклад DBI [29].

Отже, R також має велику екосистему для завантаження різних типів даних. Варто зазначити зручну функціональність з наявності великої кількості популярних наборів даних, що вже інтегровані в систему.

Exploring. Мова програмування R відома своїми графічними можливостями. В R є багато зручних інструментів, що дозволяють побудувати різноманітні графіки для візуального аналізу даних, наприклад ggplot2 [30]. В R також реалізована можливість порахувати різноманітні статистики, та проводити значний ряд статистичних тестів навіть в base R. Варто зазначити, що для використання всієї потужності цих пакетів, потрібне написання коду.

Modifying. В R представлено багато інструментів для коригування, фільтрації та перетворення даних, що покривають більшу частину аналітичних потреб. Для підготовки даних також потрібне написання коду, без графічних інструментів, проте відповідні пакети є дуже зручними для використання і більшість необхідних операцій для підготовки даних там вже автоматизована. Варто відзначити для цього етапу екосистему tidyverse [31] та бібліотеку dplyr [32].

Modeling. В R екосистемі в різних пакетах реалізовано багато потужних методів для моделювання та прогнозування. Варто відзначити пакет forecast [33] для роботи з часовими рядами та пакет caret [34] для широкого спектру задач машинного навчання. Але, незважаючи на наявність великої кількості популярних методів, в R дуже поверхнево розвинена екосистема для роботи з нейронними мережами. Багато сучасних інноваційних підходів, що з’являються або взагалі не реалізовані, або реалізовані через python інтерфейс. Наприклад, один з найпопулярніших інструментів для роботи з нейронними мережами - система Torch, має лише python інтерфейс - pytorch [35]. Альтернативою є менш гнучкий, проте доволі потужний пакет tensorflow [36], але в R він реалізований через доступ до python інтерфейсу, що не є дуже

зручно і часто приводить до помилок з типами даних, оскільки в R масиви нумеруються з 1, а в python з 0.

Assessing. В R реалізовані необхідні підходи для оцінки якості моделей [37]. Варто також зазначити широкий функціонал для створення різноманітних графічних звітів для використання моделей за допомогою shiny [38].

Отже, загальними перевагами R є розвинена, безкоштовна, відкрита екосистема на всіх етапах аналітичного процесу, особливо при підготовці даних та візуалізації. Однак, в R бракує набору інструментів для повноцінної роботи з нейронними мережами.

1.4.3. Мова програмування Python

Python - високорівнева мова програмування загального призначення з динамічної строгою типізацією і автоматичним управлінням пам'яттю, що орієнтована на підвищення продуктивності розробника, читання коду і його якості, а також на забезпечення переносимості написаних на ньому програм [39]. Python широко використовується для веб розробки, розробки ігор, інфраструктурних задач, та, звичайно, для аналітики. Python також є безкоштовною та відкритою мовою.

Розглянемо можливості Python на кожному етапі запропонованої загальної методики.

Sampling. Розглянемо можливості роботи з даними в Python.

При роботі з даними Python також може працювати з різними типами даних, як з файлами, так і з базами даних. Дуже потужним і популярним інструментом для роботи з даними є бібліотека pandas [40]. Pandas дозволяє підключатись як і до різних баз даних локально (SQL, SAS) чи у хмарі (Google BigQuery), так і працювати з різноманітними форматами вхідних даних, як то CSV (та інші текстові файли), JSON, HTML, XML, Excel, тощо.

Отже, Python як і R має багато можливостей для завантаження різних типів даних.

Exploring. Мова програмування Python має багато графічних можливостей. В Python є багато зручних інструментів, що дозволяють побудувати різноманітні графіки для візуального аналізу даних, наприклад matplotlib [41], seaborn [42]. В Python також реалізована можливість порахувати різноманітні статистики після завантаження даних використовуючи вже зазначену бібліотеку Pandas, та проводити значний ряд статистичних тестів підключаючи бібліотеку SciPy [43]. Окрім тестів, SciPy містить модулі для оптимізації, генетичних алгоритмів, лінійної алгебри, розв'язування звичайних диференціальних рівнянь та інших задач, які зазвичай вирішуються в науці і в інженерних розробках. Варто зазначити, що для використання всієї потужності цих пакетів, також потрібне написання коду.

Отже, Python, як і R має багато можливостей для аналізу даних.

Modifying. В Python також представлено багато інструментів для коригування, фільтрації та перетворення даних, що покривають більшу частину аналітичних потреб. Базовий Python не дає такої можливості для чисельного програмування, як R, проте, підключивши бібліотеку NumPy [44], можна отримати багато можливостей для вирішення обраних задач, наприклад, для обробки великих багатовимірних масивів і матриць, разом з великою кількістю високорівневих математичних функцій для операцій з цими масивами. Варто зазначити, що бібліотека NumPy, як і Pandas використовує на низькому рівні скомпільований на C код, що робить обчислення доволі швидкими та ефективними. Для підготовки даних також потрібне написання коду, проте відповідні бібліотеки є дуже зручними для використання і більшість необхідних операцій для підготовки даних там вже автоматизована. Варто відзначити для цього етапу зазвичай використовується

вищезазначена бібліотека Pandas. Завдяки цьому не треба завантажувати, знати та аналізувати багато різних бібліотек.

Modeling. В Python екосистемі в різних бібліотеках реалізовано багато потужних методів для моделювання та прогнозування. Варто відзначити бібліотеку scikit-learn [45], яка підтримує більшість класичних та сучасних методів інтелектуального аналізу даних. Багато статистичних моделей, наприклад, для часових рядів представлено в бібліотеці statsmodels [46]. Але, на відміну від R чи SAS, в Python широко розвинена екосистема для роботи з нейронними мережами. Багато сучасних інноваційних підходів, реалізовано в бібліотеці PyTorch [35]. Альтернативою є також доволі потужний пакет TensorFlow [36]. Також завдяки популярності Python як мови програмування [47], зазвичай всі інноваційні моделі та підходи доволі швидко з’являються реалізованими саме на Python.

Assessing. В Python реалізовані необхідні підходи для оцінки якості моделей у бібліотеці scikit-learn. Варто також зазначити широкий функціонал для інтеграції моделей з різними промисловими системами, оскільки Python, на відміну від R є мовою програмування загального призначення, яка широко використовується в розробці програмних систем.

Отже, Python як і R є розвиненою, безкоштовною, гнучкою, відкритою мовою програмування в якій є своя екосистема на всіх етапах аналітичного процесу. Однак, на відміну від R Python має переваги в інноваційності, через більшу популярність мови програмування, більш розвинену екосистему для нейронних мереж та більше можливостей інтеграції з промисловими системами. В дослідженні для проведення експериментів використана саме мова програмування Python з відповідними зазначеними вище бібліотеками.

Висновки до розділу

Нелінійні нестаціонарні процеси, представлені у вигляді часових рядів, можуть собою описувати динаміку процесів в технічних, економічних та інших системах і мають численні застосування в різних галузях.

Прогнозування є одним з найважливіших елементів ефективної організації управління підприємствами, оскільки якість прийнятих рішень багато в чому визначається якістю прогнозування їх наслідків.

Задача прогнозування нелінійних нестаціонарних процесів має численну кількість застосувань і є дуже актуальною на сьогоднішній день для підтримки прийняття рішень в управлінні цими системами.

В даній роботі буде вибрано і розглянуто фінансово-економічні процеси, проте розроблену методику можливо застосовувати і в інших системах з відповідною визначеною динамікою. Для прогнозування буде розглянуто підхід предиктивної аналітики з використанням методів інтелектуального аналізу даних.

Методологічною основою роботи є сучасна аналітична методологію SEMMA. Дана методологія добре орієнтована на процес дослідження, більше фокусуючись на побудові моделей, а не на бізнес-розумінні або інтеграції з технічними системами.

В дослідженні для проведення експериментів використана мова програмування Python, оскільки є розвиненою, безкоштовною, гнучкою, відкритою мовою програмування в якій є своя екосистема на всіх етапах аналітичного процесу.

РОЗДІЛ 2

МАТЕМАТИЧНІ МОДЕЛІ, МЕТОДИ ТА ПІДХОДИ ДЛЯ ОПИСУ, АНАЛІЗУ ТА ОБРОБКИ НЕЛІНІЙНИХ НЕСТАЦІОНАРНИХ ПРОЦЕСІВ

2.1. Математичні моделі, що використовуються для опису динаміки фінансово-економічних процесів

Для опису динаміки фінансово-економічних процесів будемо користуватися експериментальними статистичними даними у вигляді часових рядів. Під часовим рядом даних будемо розуміти сукупність вимірів змінної, отриманих з однаковим дискретним часовим інтервалом, який називають періодом дискретизації вимірів [48]. Наприклад, ми можемо розглянути часовий ряд як сукупність $\{y_t\} = y_1, y_2, y_3, \ldots$, де y_1 - значення змінної, отримане в перший часовий інтервал, y_2 - в другий, і т.д. $t = 1, 2, 3, \ldots$ - дискретна ціла змінна, що відповідає за номер спостереження в часі. Періодом дискретизації може виступати довільний часовий інтервал, наприклад: секунда, година, день, місяць, квартал, тощо.

Розглянемо детальніше властивості та характеристики моделей, які застосовують для опису динаміки досліджуваних процесів.

Під стаціонарністю часового ряду мають на увазі “стабільність” його структури в часі, тобто властивості ряду не залежать від часу. Розглянемо більш точне визначення стаціонарності.

Ряд $\{y_t\}$ називається строго (сильно) стаціонарним, якщо сумісний розподіл m спостережень $y_{t_1}, y_{t_2}, \ldots, y_{t_m}$ не залежить від зсуву по часу, тобто співпадає з розподілом $y_{t_1+k}, y_{t_2+k}, \ldots, y_{t_m+k}$ для будь-яких k. Однак строге

визначення стаціонарності є дуже складним для застосування на практиці, та його важко перевірити емпірично, тому зазвичай при дослідженні процесів використовують більш слабкий варіант стаціонарності, в якому аналізуються тільки перші два моменти ряду, а не весь розподіл [49].

Формально стохастичний процес із скінченним середнім та дисперсією називають слабко стаціонарним, якщо для всіх k і $k-s, (k = 0, 1, 2, ...; s = 0, 1, 2, ...)$ виконуються наступні умови:

– математичне сподівання є константою і не залежить від часу:

$$E\,[y(k)] = E\,[y(k-s)] = \mu = const\,; \tag{2.1}$$

– дисперсія залишається постійною для всього часового інтервалу, на якому розглядається процес:

$$E\left\{[y(k)-\mu]^2\right\} = E\left\{[y(k-s)-\mu]^2\right\} = \sigma_y^2 = const\,; \tag{2.2}$$

– коваріація залишається незмінною в часі для всього часового інтервалу:

$$\mathrm{cov}[y(k), y(k-s)] = \mathrm{cov}[y(k-j), y(k-j-s)] = \gamma(s) = const\quad . \tag{2.3}$$

Очевидно доводиться, що зі строгої стаціонарності випливає слабка стаціонарність за умови обмеженості першого і другого моменту розподілу. Надалі процес будемо вважати нестаціонарним, якщо не виконується хоча б одна з вищенаведених умов (2.1) – (2.3), тобто хоча б одна із наведених статистичних характеристик процесу змінюється в часі [48]. Саме таке означення зазвичай зустрічається в фінансовій літературі та на практиці [49] (Рис. 2.1).

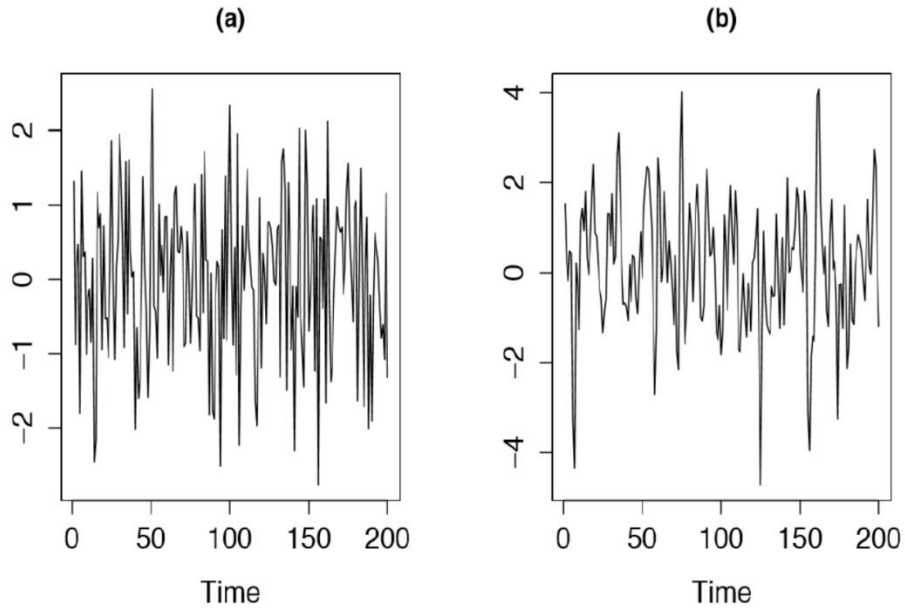


Рис. 2.1 Приклад стаціонарного процесу

Зазначимо також, що основні статистичні характеристики процесу – математичне сподівання, дисперсія та коваріація – можуть бути розраховані на основі конкретних значень часового ряду (характеристики вибірки) або на основі математичної моделі, яка описує процес (теоретичні характеристики).

В рамках даної роботи будемо розглядати процеси з двома видами нестаціонарності - інтегровані та гетероскедастичні процеси.

У випадку, коли математичне сподівання змінюється в часі, такий процес називають інтегрованим процесом, або процесом з трендом, або процесом з одиничними коренями (відповідного характеристичного рівняння) (Рис. 2.2 -2.3).

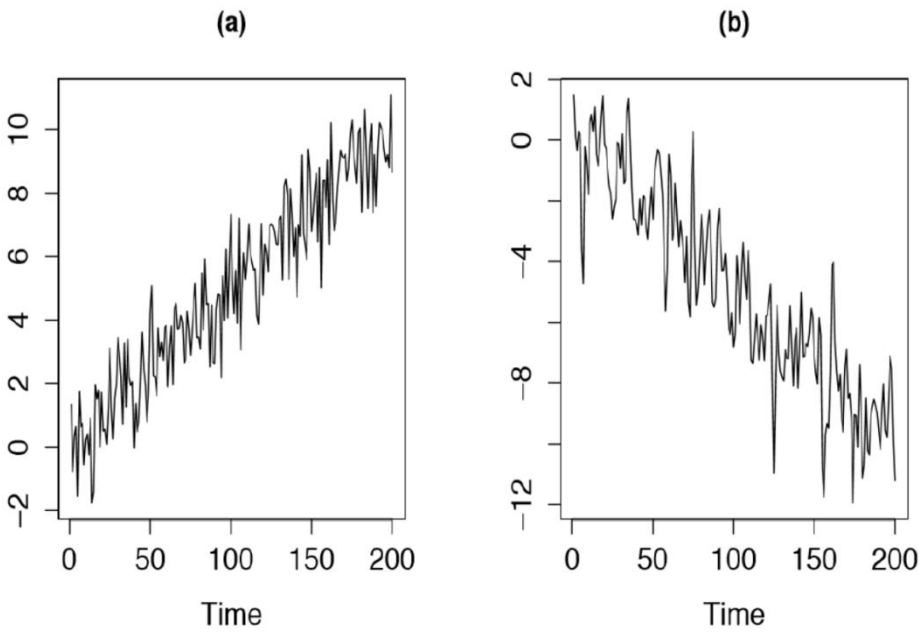


Рис. 2.2 Приклад нестаціонарного процесу з лінійним трендом

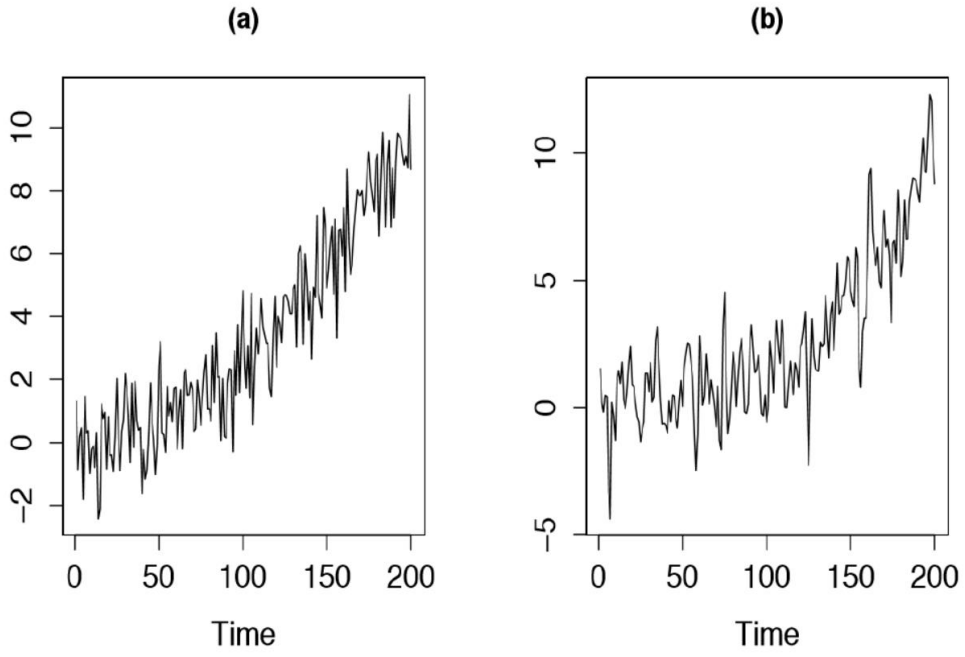


Рис. 2.3 Приклад нестаціонарного процесу з нелінійним трендом

Процеси зі змінною дисперсією називають гетероскедастичними (Рис. 2.4), а – процеси із сталою дисперсією - гомоскедастичними.

Формально для гетероскедастичного процесу можна записати

$$\mathrm{var}[\varepsilon(k)] = \sigma_{\varepsilon}^{\ 2} \neq const \tag{2.4}$$

Гетероскедастичніть означає, що дисперсія процесу зменшується чи збільшується в часі, або є більш складною функцією часу.

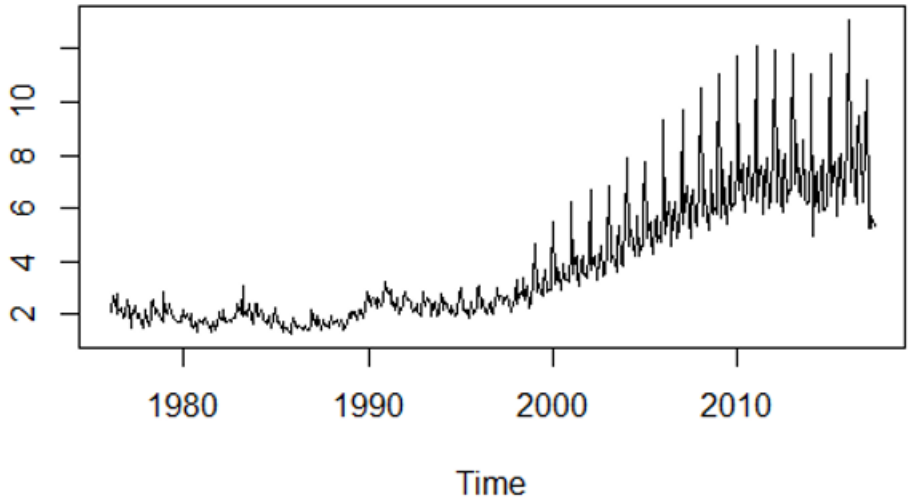


Рис. 2.4 Гетероскедастичний процес

В літературі зазвичай розглядають саме лінійні та стаціонарні процеси через те, що їх значно простіше аналізувати та прогнозувати [50]. Однак, варто зазначити, що на практиці фінансово-економічні процеси зазвичай мають нелінійну нестаціонарну природу. Часто виявляється, що проста лінійна модель часового ряду зазвичай залишає певні аспекти економічних і фінансових даних без пояснення [51].

В загальному сенсі, моделлю, що описує стохастичний процес представлений у вигляді часового ряду можемо називати функцію поточного стану від попередніх:

$$y_t = f(y_{t-1}, y_{t-2}, \ldots, y_1) \tag{2.5}$$

Якщо функція f є лінійною, тобто, в загальному сенсі

$$y_t = a_0 + \sum_{i=1}^{t-1} a_i y_{t-i}, \tag{2.6}$$

де, a_i - константні дійсні числа, то будемо називати такий процес лінійним, а відповідний часовий ряд таким, що може бути описаний лінійною моделлю [49]. Моделі подібного типу називаються авторегресійними, тобто такими, що описують регресію змінної відносно себе і будуть розглянуті далі.

Дійсно, лінійні моделі прості та інтуїтивно зрозумілі для дослідження, бо всі характеристики просто моделюються та прогнозуються. Також вони зручні і для використання на практиці завдяки простоті структури, за рахунок якої відбувається суттєвий виграш щодо обчислювальних витрат, а тому є можливість її оперативної адаптації до характеристик процесу в реальному часі. Однак, оскільки дана модель являє собою звичайну лінійну регресію, побудовану на попередніх значеннях, то вона не здатна враховувати нелінійний вплив як внутрішніх так і зовнішніх факторів, комбінований вплив факторів, а також не може бути адекватно побудована в разі наявності колінеарних факторів.

Варто також зазначити, що лінійні моделі не можуть допускати сильної асиметрії в даних, вони не підходять для даних, що характеризуються раптовими та нерегулярними стрибками, нехтують нелінійною залежністю, корисною для передбачення, а також вони не підходять для рядів, які не є оборотними в часі. Більше того, нездатність розпізнати і впоратися з наявними нелінійностями в механізмі аналізу часового ряду часто може призводять до поганих оцінок параметрів і моделей, які взагалі втрачають важливі залежності в часових рядах [51].

Отже, нелінійними процесами будемо вважати такі, в яких в функції f наявна нелінійність будь-якого типу.

Загалом, задача формального опису та встановлення наявності та визначення типу нелінійності не є остаточно вирішеною та залишається предметом дослідження. Багато нелінійних моделей часових рядів було запропоновано в статистичній літературі, такі як білінійні моделі Грейнджера та Андерсена [52], порогова авторегресивна модель Тонга (TAR) [53], модель залежності від стану Прістлі [54], і модель перемикання Маркова-Гамільтона [55]. Основна ідея, що лежить в основі цих нелінійних моделей - те, що вони дозволяють умовному середньому розвиватися з часом відповідно до деякої простої параметричної нелінійної функції [49].

Наприклад, наявність нелінійного детермінованого тренду в процесі можна визначити шляхом оцінювання рівняння:

$$y(k) = a_0 + c_1\, k + c_2\, k^2 + \ldots + c_m k^m\ , \qquad (2.7)$$

яке представляє собою поліном порядку m відносно часу.

Якщо хоча б один із коефіцієнтів $c_i, i = 1,\ldots,m$ є статистично значимим, то гіпотеза щодо відсутності тренду відхиляється. Якщо тренд відносно швидко змінює напрямок свого розвитку і важко знайти його адекватний функціональний опис, то використовуються моделі випадкових трендів, засновані на комбінаціях випадкових величин [48].

Останнім часом ряд нелінійних моделей були запропоновані з використанням прогресу в обчислювальних можливостях та комп’ютерах. Прикладами таких моделей є нелінійне моделювання простору станів [56], функціональна авторегресивна модель [57], нелінійна адитивна авторегресивна модель [58], а також багатовимірний сплайн адаптивної регресії [59].

Основна ідея цих розширень полягає у використанні методів моделювання для опису еволюції умовного розподілу часового ряду або

використання даних для поглибленого дослідження нелінійних характеристик ряду. Однак у всіх вищезазначених підходів до опису нелінійностей є недолік, що полягає в описі конкретного типу нелінійності, а не нелінійності в загальному сенсі. Вважається, що в загальному вигляді нелінійні процеси не можуть бути описані, оскільки вони будуть мати дуже багато параметрів, або не можуть бути застосовані [49].

Однак існує ефективний альтернативний підхід для опису нелінійних процесів довільної природи - технологія штучних нейронних мереж. Сучасний апарат штучних нейронних мереж здатний представити будь-яку функціональну залежність згідно теоремі про універсальну апроксимацію [60], [61], а отже він автоматично оцінює структуру математичної моделі з подальшим налаштуванням параметрів вибраним методом оптимізації. Методи, з використанням нейронних мереж, істотно відрізняються від розглянутих вище, оскільки нейронні мережі являють собою системи, здатні навчатися на досвіді і адаптуватися до змін середовища. Нейронні технології мають такі переваги: можливість нелінійного моделювання, стійкість до інформаційних перешкод і здатність до узагальнення на основі прикладів. Нейромережі дозволяють будувати доволі точні та потужні моделі, навіть там, де потрібний якісний аналіз взаємозв’язків факторів, що впливають на результат. Цей метод вже багаторазово застосовували до опису широкого класу процесів з метою розробки прогнозів та стратегій керування [15].

В даній роботі пропонується використовувати саме апарат штучних нейронних мереж для моделювання процесів різної природи з нелінійностями та нестаціонарностями. Незважаючи на теоретичну універсальність запропонованої методики, в даній роботі ми будемо розглядати саме процеси вибраних типів, що найбільш характерні для фінансово-економічної сфери, а саме:

- інтегровані процеси (процеси з трендами, в т.ч і вищих порядків, що призводять до нелінійності) - нестаціонарні процеси, що містять детермінований тренд;

- гетероскедастичні процеси - нестаціонарні процеси, що призводять як мінімум до квадратичної [48];

- процеси Леві [62] - процеси з випадковими викидами (Рис. 2.5);

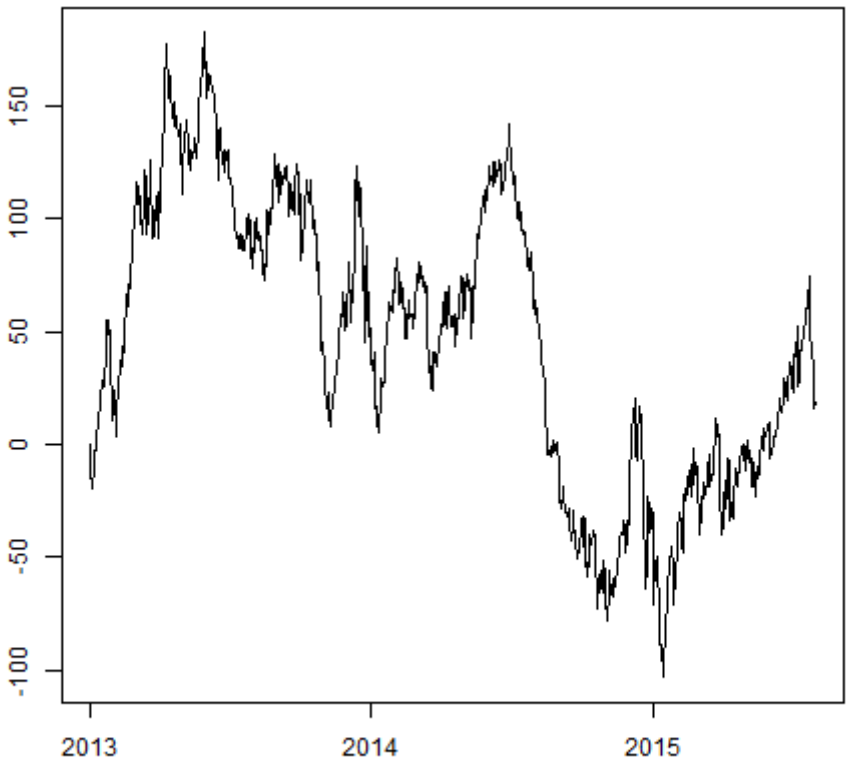


Рис. 2.5 Приклад процесу Леві

- процеси зі стохастичним трендом, що являють собою лінійну або нелінійну комбінацію випадкових процесів [64] (Рис. 2.6);

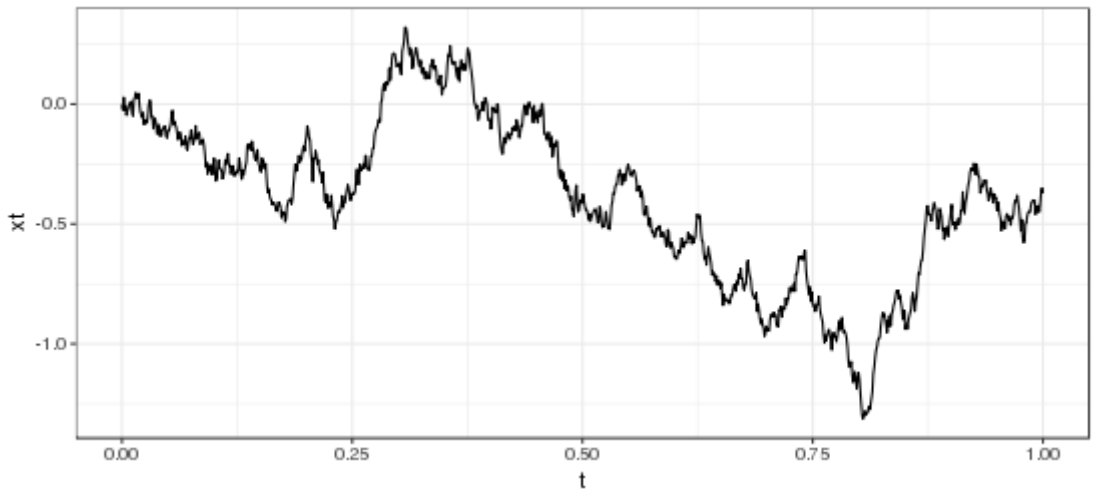


Рис. 2.6 Приклад процесу зі стохастичним трендом

- процеси логістичного типу [49].

Загальну схему класів фінансово-економічних процесів зображено на рис. 2.7

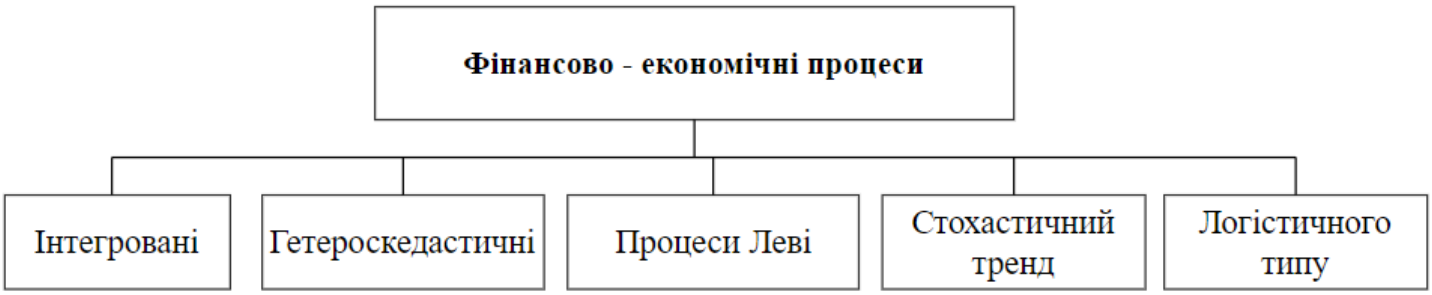


Рис. 2.7 Класи фінансово-економічних процесів

2.2. Встановлення належності стохастичних процесів до певного класу

На другому етапі запропонованої нами загальної методики необхідно зробити первинний аналіз даних для пошуку трендів, аномалій та взаємозв'язків, а також дослідити наданий процес до належності до певного класу за допомогою відповідних статистичних тестів. Розглянемо основні теоретичні підходи та практичні інструменти на цьому етапі.

2.2.1. Статистичний та графічний аналіз

При первинному дослідженні необхідно візуально дослідити часовий ряд, а також розглянути основні його статистичні характеристики.

Графічний аналіз - побудова та аналіз основних візуалізацій часового ряду досить часто попередньо дозволяє припустити наявність лінійного чи нелінійного тренду, сезонності, гетероскедастичності чи наявності аномальних значень (викидів).

Для часових рядів очевидним графіком для початку є графік значень відносно часу. Спостереження наносять на графік відносно часу спостереження, а послідовні спостереження об'єднуються прямими лініями. На малюнку 2.8 нижче показано тижневе пасажирське навантаження економ-класу авіакомпанією Ansett Airlines між двома найбільшими містами Австралії [63].

Рис. 2.8 Пасажирське навантаження економ-класу авіакомпанією Ansett Airlines

З візуального аналізу одразу видно деякі особливості, які необхідно враховувати або з’ясовувати додаткову інформацію щодо процесу, що моделюється для коректної обробки відповідного ряду.

Покажемо, як можна проаналізувати деякі викиди або аномальні значення для цього ряду за допомогою аналізу додаткової інформації.

- У 1989 році був період, коли пасажирів не перевозили — це було через промисловий конфлікт.

- У 1992 році був період зниження навантаження. Це було пов'язано з випробуванням, під час якого деякі сидіння економ-класу були замінені на сидіння бізнес-класу.

- Значне збільшення пасажиропотоку відбулося в другій половині 1991 року.

- На початку кожного року спостерігаються значні падіння навантаження. Це пов’язано з ефектами свята.

- Існують довгострокові коливання рівня ряду, які збільшуються протягом 1987 року, зменшуються в 1989 році і знову збільшуються в 1990 і 1991 роках.

- Є деякі періоди відсутніх спостережень.

Всі подібні аномалії потрібно пояснювати або перевіряти за допомогою відповідних статистичних тестів та займатися подальшою їх відповідною обробкою. Будь-якій моделі потрібно буде враховувати всі ці особливості, щоб ефективно прогнозувати пасажирське навантаження в майбутньому [64].

Для виявлення аномалій також часто допомагає аналіз основних статистичних характеристик ряду, таких як мінімальне та максимальне значення (часто використовуються і інші перцентилі, наприклад, 25й і 75й), медіана, вибіркове середнє. Наприклад, значне відхилення медіани від середнього, або значна різниця між максимумом та 75-м перцентилем може свідчити про наявність аномальних значень в ряді.

Розглянемо приклад для нестаціонарних процесів на прикладі продажу антидіабетичних ліків в Австралії [65] (Рис. 2.9).

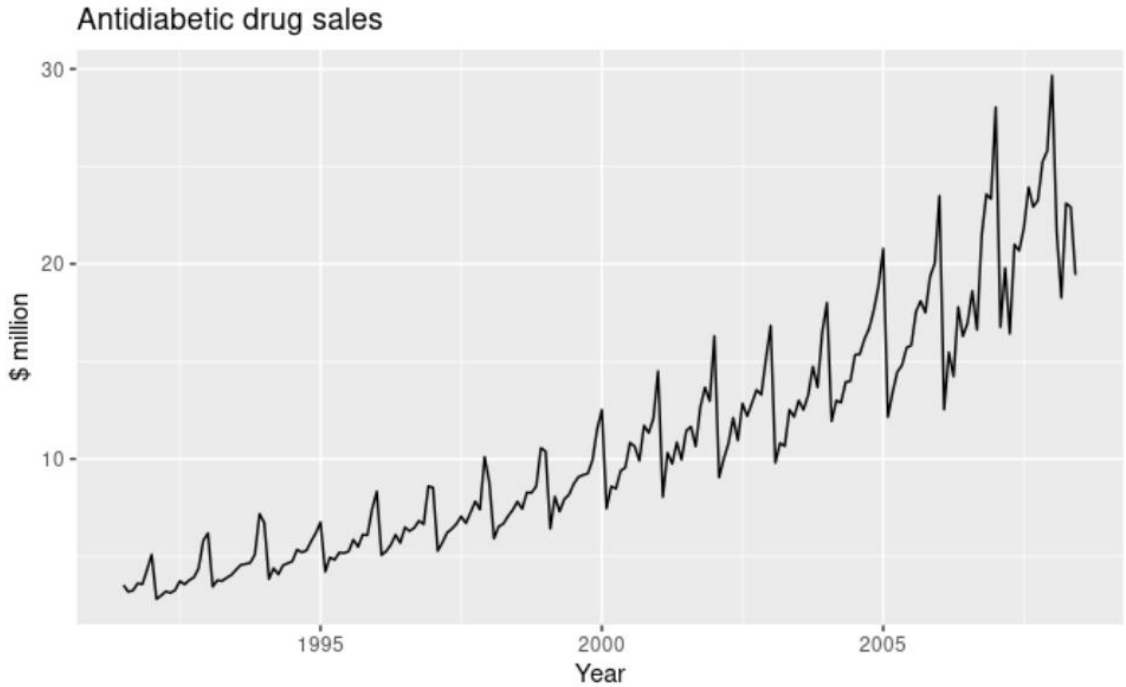


Рис. 2.9 Продаж антидіабетичних ліків в Австралії

Тут простежується чітка тенденція до зростання, можна припустити наявність тренду в ряді, а отже відповідну його нестаціонарність. Існує також сильна сезонна закономірність, яку видно за повторюючимися візуальними елементами ряду. Більш того, ці елементи збільшуються у міру зростання ряду, що може сигналізувати про зростаючу дисперсію. Раптове падіння на початку кожного року спричинене державною схемою субсидування, яка робить рентабельним для пацієнтів накопичувати ліки наприкінці календарного року.

Такі ознаки ряду також необхідно пояснювати та перевіряти. Будь-які прогнози цього ряду повинні відображати сезонну закономірність і той факт, що тенденція повільно змінюється [64].

2.2.2. Нейромережевий тест Уайта для перевірки нелінійності часового ряду

Розглянемо статистичні тести для перевірки часового ряду на нелінійність.

Розглянемо F-тест, який теоретично можна застосовувати у випадку коли можна набрати кілька груп спостережень для одного і того ж процесу. Цей тест досліджує відношення відхилення середніх значень від прямої регресії до відхилення значень ряду від групових середніх:

$$\hat{F} = \frac{\frac{1}{m-2}\sum_{i=1}^{m} n_i(\bar{y}_i - \hat{y}_i)^2}{\frac{1}{n-m}\sum_{i=1}^{m}\sum_{j=1}^{n_i}(y_{ij} - \bar{y}_i)^2}, \qquad (2.8)$$

де y_i - середнє значення для і-ї групи даних; y_i - середнє для лінійної апроксимації даних; m- число груп даних; n_i - число вимірів в і-й групі; n - загальне число вимірів.

Якщо статистика F з m-2 і n-m ступенями свободи досягає або перевищує рівень значущості, то гіпотеза лінійності має бути відхилена. Недоліком такого підходу є те, що для його застосування необхідно мати кілька (не менше трьох) груп даних по тому самому процесу, які можна отримати тільки проводячи повторні експерименти. Очевидно, це не завжди можливо [48].

Квадратичну нелінійність можна встановити, наприклад, за допомогою вибіркових нелінійних кореляційних функцій.

$$r_{yx^2}(s) = r_{y(k)x^2(k-s)} = \frac{1}{N}\frac{\sum_{k=s+1}^{N}\left\{[y(k)-\bar{y}][x(k-s)-\bar{x}]^2\right\}}{\sigma_y\sigma_x^2}, \quad s = 0,1,2,3,\ldots, \tag{2.9}$$

Якщо значення такої функції, розраховане за вибірковими статистичними даними, суттєво (статистично значимо) відрізняється від нуля, то досліджуваний процес містить квадратичну нелінійність [48].

В літературі існує ряд більш розвинутих підходів щодо дослідження часового ряду на нелінійність, а саме: тести Кінана [66], Тсея [67], RESET тест Ремз [68], динамічний тест Уайта [69], тести МакЛеода-Лі [70], BDS тест [71].

Більшість існуючих тестів, мають як перший крок вилучення лінійної структури за допомогою певного оцінювального фільтра. Як правило, до часового ряду застосовується певна авторегресивна модель, а тест потім застосовуються до оцінених залишків. Щоб автоматизувати цю процедуру, в симуляції використовується конкретне значення для параметру авторегресії p, зазвичай p = 1 або 2, але при роботі з реальними даними ми виберемо p за критерієм SIC, що веде до послідовного вибору p [72].

Кілька тестів включають регресію залишків лінійної моделі на конкретні функції, що обираються для захоплення істотних ознак можливих

нелінійностей. Нульова гіпотеза відкидається, якщо ці функції істотно корелюють з залишками. Коли нульова гіпотеза відхиляється, може надаватися неявна альтернативна модель, прогнози якої перевершують прогнози лінійної моделі. Ці прогнози не обов'язково мають бути оптимальними, просто краще за відповідну лінійну модель. Тести не засновані на моделях, які передбачають такі прогнози: тест МакЛеода-Лі (на основі автокореляції квадратів залишків) [70], тест Брокка-Дехерта-Шейнкмана (BDS) (що виникає внаслідок розгляду хаотичних процесів) [71], а також тест Bispectrum [73].

Проте, як вже було зазначено, подібний підхід до розгляду нелінійності має той недолік, що він зазвичай може виявити певний тип нелінійності, що перевіряється, а не нелінійність в загальному сенсі.

В данній роботі пропонується розглянути тест Уайта заснований на нейронних мережах. Нейромережевий тест заснований на апроксимаційній здатності останніх методів моделювання нейронних мереж і розроблений вченими-когнітивистами. Цей тест визначає статистично чи було б вигідним додавання «прихованих елементів» до лінійної нейронної мережі. В роботі [74] виконані тести, що порівнюються з альтернативними завдяки використанню різноманітних нелінійних синтетичних рядів, включаючи білінійні, порогові авторегресивні моделі, моделі нелінійної ковзної середньої, а також тести застосовувалися до фактичних економічних часових рядів. Результати досліджень свідчать про те, що даний тест може відігравати цінну роль в оцінці адекватності моделі. Тест нейронної мережі Уайта має високу потужність, і показав себе на багатьох з перевірених економічних рядів,що демонструють потенційні нелінійності.

Останнім часом дуже популярним для дослідження в аналізі даних є клас моделей заснованих на нейронних мережах - моделях, поява яких була натхненна певними особливостями способу обробки інформації в мозку [75].

Базовою моделлю є «одношарова мережа прямого зв’язку», зображена на рис. 2.10.

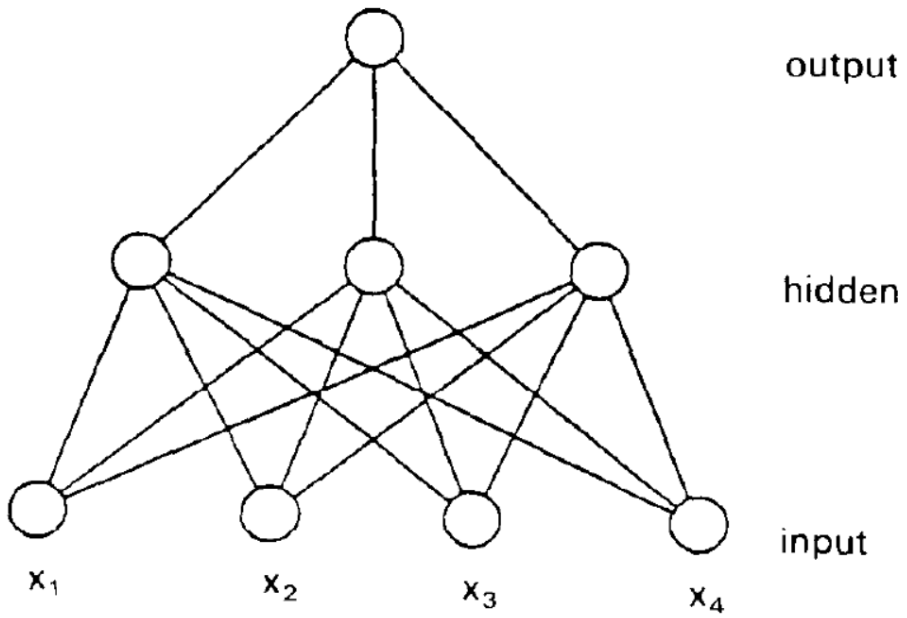


Рис. 2.10 Одношарова мережа прямого зв’язку

У цій мережі вхідні елементи (“нейрони”) посилають сигнали - елементи вхідних даних x_i, i= l, ... , n, уздовж зв'язків (“з'єднань”), які послаблюють або посилюють вихідні сигнали з коефіцієнтом α_{ij} (“ваги” або параметри нейронної мережі). Кожний елемент j з проміжного або “прихованого” прошарку “бачить” сигнали $x_i\alpha_{ji}$, і обробляє їх якимось характерним способом. Як правило, елементи прихованих шарів підсумовують сигнали, що надходять, а потім застосовують до них певну нелінійну функцію “активації”. Отримані сигнали потім переходять до виходу, який підсумовує те, що він бачить для отримання результату. В загальному вигляді, для базової одношарової нейронної мережі можемо записати:

$$f(x, w) = \beta_0 + \sum_{j=1}^{q} \beta_j \varphi\left(\sum_{i=1}^{n} x_i \alpha_{ij}\right), n, q \in N, \qquad (2.10)$$

де w - множина параметрів або “ваг” нейронної мережі, $w = \{\beta_0, \ldots, \beta_q, \alpha_{11}, \ldots, \alpha_{nq}\}$, $\varphi: R \to R$, наприклад, $\varphi(x) = \frac{1}{1+e^{-x}}$, доволі популярна, так звана, сигмоїдна функція активації.

Як було показано у [76], [77] функції f, що моделюються таким чином, належать до сімейства гнучких функціональних форм, тобто можуть моделювати дуже різноманітні функціональні залежності, в тому числі і різні типи нелінійних зв’язків. У [61], [78], [79] показали, що для широких класів нелінійних функцій, функції, що моделюються такою нейронною мережею можуть забезпечити як завгодно точні наближення до довільних функцій в різноманітних нормованих функціональних просторах (наприклад, неперервні функції на компакті з рівномірною нормою, функції в L_p просторі та функції в просторах Соболєва з нормою Соболєва), за умови, що число нейронів (параметрів) достатньо велике. Таким чином, функції вказаної форми здатні апроксимувати довільне нелінійне відображення. Значний практичний досвід показує, що коли відображення є досить гладким, прийнятні для обчислень значення для кількості параметрів можуть забезпечити досить хороші наближення. Наприклад, в роботі [80] добре наблизили детермінований хаос з використанням сигмоїди та використанням п'яти прихованих нейронів, а у [81] добре наблизили хаос Макея-Гласса п'ятьма прихованими одиницями.

Подібні апроксимаційно-теоретичні проблеми виникають у контексті проекції переслідування [82-85]. Фактично дану функціональну залежність можна розглядати як обмежену проекцію функції переслідування, в якій функції φ задані апріорно.

Тест нейронної мережі на нелінійність використовує таку мережу з одним прихованим шаром, доповнена з'єднаннями від входу до виходу.

Таким чином, коли підтверджується нульова гіпотеза про лінійність, тоді оптимальні ваги нейронної мережі β_j, позначимо їх як β_j^* будуть нульові,

що призводить до так званної “афінної” нейромережі, що фактично моделює звичайну лінійну регресійну модель. Запропонований нейромережевий тест власне перевіряє гіпотезу щодо $\beta_j^* = 0, j = 1, \ldots, q$, для вибраних q та α_{ij}. Даний тест має статистичну потужність, коли $\sum_{j=1}^{q} \beta_j \varphi(\sum_{i=1}^{n} x_i \alpha_{ij})$ здатна виділити структуру з $e_t^* = y_t - x_t\theta^*$, де x_t - вхідні змінні, y_t - прогнозовані значення, θ^* - вектор параметрів, отриманий процедурою оптимальної оцінки за допомогою методу найменших квадратів.

Застосування методу множників Лагранжа призводить до тестування гіпотези:

$$H_0: E(\varphi_t e_t^*) = 0 \text{ проти } H_a: E(\varphi_t e_t^*) \neq 0, \tag{2.11}$$

де $\varphi_t \equiv (\varphi(x_t\Gamma_1), \ldots, \varphi(x_t\Gamma_q)), \text{і } \Gamma = (\Gamma_1, \ldots, \Gamma_q)$, що вибираються випадково як пріор, незалежно від послідовності $\{x_t\}$. В роботі [74] φ_t називають “фантомними активаціями прихованого шару”.

При конструкції статистики замінимо e_t^* на залишки лінійної моделі, отриманої за методом найменших квадратів, тобто $e_t^* = y_t - x_t\theta^*$, що приводить до наступної статистики:

$$M_n = (\frac{1}{\sqrt{n}} \sum_{t=1}^{n} \varphi_t e_t^*) W_n^{-1} (\frac{1}{\sqrt{n}} \sum_{t=1}^{n} \varphi_t e_t^*), \tag{2.12}$$

де W_n є конзистентною оцінкою дисперсії $W^* = D(\frac{1}{\sqrt{n}} \sum_{t=1}^{n} \varphi_t e_t^*)$

Статистика M буде прямувати до статистики хі-квадрат при виконанні нульової гіпотези. Детальне доведення, розгляд і аналіз даної статистики, а також опис критичних значень описаний в [74].

Такий тест на базі нейронної мережі показує себе так само або краще за існуючі альтернативні тести і може бути широко застосований для визначення загальної нелінійності в практичних економічних часових рядах [74].

2.2.3. Тест KPSS для перевірки часового ряду на нестаціонарність

Більш об’єктивним способом показати нестаціонарність ніж графічний аналіз є використання статистичних тестів. Розглянемо відповідні тести для перевірки ряду на нестаціонарність.

Доволі популярним і рекомендованим підходом в літературі є використання тестів Дікі-Фуллера (простого або розширеного) [86], [87]. Однак, системне дослідження [88], що проводилося на 14 різних часових рядах показало, що вказані тести не виявили нестаціонарність в усіх рядах крім одного. Ці результати залишились незміненими, навіть дозволяючи автокореляцію помилок за допомогою розширених тестів [89] або тестової статистики Філіпса [90], [91]. Подібні результати отримані для багатьох інших макроекономічних часовий рядів. Частковий перелік емпіричних досліджень, що дають ці висновки можна знайти в роботі [92].

Загальний висновок, який можна зробити з цих емпіричних досліджень, такий, що багато або більшість агрегованих економічних часових рядів містять нестаціонарність хоча б у вигляді наявного тренду. Однак важливо зазначити, що в цьому емпіричному дослідженні тренду основною є нуль гіпотеза, що підлягає перевірці, і спосіб перевірки класичної гіпотези, що гарантує, що нульова гіпотеза буде прийнята, якщо немає сильних доказів проти цього. Тому альтернативним поясненням загальної неможливості відхилити гіпотезу про наявність нестаціонарності або те, що більшість економічних часових рядів не є дуже інформативними про наявність нестаціонарності, чи еквівалентно, що стандартні тести виявлення нестаціонарності не є доволі потужними проти відповідних альтернативних гіпотез. Кілька останніх досліджень на цю тему стверджують, що причиною дійсно є слабка потужність тестів. Наприклад, у роботі [92] надані докази того, що Дікі-Фуллер тести мають низьку потужність проти стабільних авторегресивних альтернатив, а у роботі [93] показано, що

вони також мають низьку потужність проти частково інтегрованих альтернатив.

В даній роботі пропонується використовувати тест Квятковського-Філіпса-Шмідта-Шина, або KPSS тест [94], в якому перевіряється нульова гіпотеза, що полягає в тому, що часовий ряд є стаціонарним, і ми шукаємо докази того, що нульова гіпотеза хибна.

У цьому тесті використано параметризацію, яка забезпечує правдоподібність представлення як стаціонарних, так і нестаціонарних змінних і яка веде закономірно до перевірки гіпотези стаціонарності. Зокрема, вибирається така репрезентація компонентів, у якій досліджуваний часовий ряд записується як сума детермінованого тренду, випадкового блукання та стаціонарної помилки.

$$y_t = \alpha t + r_t + \varepsilon_t, \text{де } r_t = r_{t-1} + u_t, \tag{2.13}$$

де y_t - досліджуваний часовий ряд, αt - компонент, що відповідає за детермінований тренд, r_t - випадкове блукання, u_t, ε_t - незалежні та однаково розподілені випадкові величини за $N(0,\sigma^2)$. Початкове значення r_0 фіксоване і виступає в ролі вільного члена.

Нульова гіпотеза стаціонарності відповідає гіпотезі про те, що дисперсія випадкового блукання дорівнює нулю, тобто $\sigma_u^2 = 0$. За додатковими припущеннями, що випадкове блукання є нормальним, а стаціонарна похибка є нормальним білим шумом, одностороння статистика тесту множників Лагранжа для гіпотези стаціонарності є такою ж, як локальний найкращий інваріант (статистика тесту LBI і випливає з роботи [95]. Тобто, розглянемо e_t, t = 1, 2, …, T - залишки регресійної моделі з часовим трендом. Тому, можемо записати часткові суми залишків, як:

$$S_t = \sum_{i=1}^{t} e_i \,, t = 1,2,\ldots,T \tag{2.14}$$

Запишемо також дисперсію помилок цієї регресивної моделі:

$$\sigma_{\varepsilon}^{2} = \frac{1}{T}\sum_{i=1}^{T} e_i^2 \tag{2.15}$$

Тоді, статистика для тесту множників Лагранжа, а також і статистика для локального найкращого інваріанту буде:

$$\mathrm{LM} = \sum_{t=1}^{T} \frac{S_t^2}{\sigma_{\varepsilon}^2} \tag{2.16}$$

Більш того, у випадку, якщо ми хочемо перевірити нульову гіпотезу рівня стаціонарності замість просто наявності тренду (наприклад, для трендів вищих порядків), ми просто визначаємо e як залишок від регресії у лише для вільного члена (тобто $e_t = y_t - \bar{y}$) замість того як вказано вище, а решта побудови тестової статистики не змінюється.

Однак припущення, що помилка є білим шумом не підтверджується в деяких практичних прикладах. Тому в тесті KPSS пропонується за принципами, розглянутими в [90] і [91] зробити виведення асимптотичного розподілу статистики за загальних умов стаціонарності помилки, і тому пропонується модифікована версія статистики множників Лагранжа, що буде справедливою асимптотично за цих загальних умов [94].

Тест є одностороннім вгору, відповідні критичні значення також наведені в роботі [94]. Відповідно, малі значення p-values (наприклад, менше 0,05) свідчать про наявність нестаціонарності.

2.3. Методи обробки пропущених та аномальних значень

Реальні дані часто містять пропущені та аномальні значення. Пропущеним вважаємо значення, відсутнє в вибраний момент часу з певних причин. Аномальними значеннями, або викидами вважаються значення,

отримані внаслідок помилок у даних: неточностей вимірювання або збору статистичних даних, округлення, неправильного запису, наявності шуму в даних, присутності об'єктів з “інших” вибірок (наприклад, показання датчика, що зламався). Більшість методів з аналізу та моделювання часових рядів не можуть коректно працювати коли в даних є пропуски або викиди. Пропущені дані не дозволяють арифметичних операцій з ними. Аномальні значення будуть створювати значний вплив на роботу моделей і оцінку параметрів, а також можуть впливати на коректну роботу тестів на нелінійність та нестаціонарність, оскільки аномальні значення порушують типову поведінку процесів в даних і відповідно можуть давати хибні припущення щодо характеристик ряду. Навіть звичайні статистичні характеристики, такі як середнє значення, будуть значно зміщені в сторону аномалії. Виявлення та коректна обробка таких даних іноді може бути доволі складною, і є великою окремою темою для дослідження.

Класичний підхід для виявлення аномалій полягає в застосуванні статистичних тестів, що базуються на статистиках, які є чутливими до значних відхилень. По суті, ці методи використовують статистичну перевірку гіпотез для певного рівня значущості, щоб визначити, чи є дане спостереження аномальним. Іншими словами, тест використовується для оцінки відхилення нульової гіпотези (H0) для попередньо визначеного рівня значимості, на користь альтернативної гіпотези (H1).

Одним із історично перших таких тестів є критерій Ірвіна [96], що призначений для оцінки значень вибірки на грубі помилки при нормальному розподілу випадкової величини.

Пізніше було розроблено тест Граббса [97], [98] для загального пошуку аномалій в одновимірній вибірці. Тест також передбачає, що основний розподіл даних є нормальним. Тест Граббса визначає наступні гіпотези:

- H_0: у вибірці немає викидів
- H_1: у вибірці наявний хоча б один викид

Тестова статистика за Граббсом виглядає наступним чином:

$$C = \frac{max_t|x_t - \bar{x}|}{s}, \tag{2.17}$$

де $\bar{x}$ і s вибіркове середнє значення і дисперсія відповідно деякого часового ряду X. Для двостороннього тесту гіпотеза відсутності викидів відхиляється на рівні значущості α, якщо

$$C > \frac{(N-1)}{\sqrt{N}} \sqrt{\frac{(t_{\frac{\alpha}{2N},N-2})^2}{N-2+(t_{\frac{\alpha}{2N},N-2})^2}}, \tag{2.18}$$

де $t_{\frac{\alpha}{2N},N-2}$ позначає верхню критичну границю для t розподілу з N-2 ступенями свободи і рівнем значущості $\frac{\alpha}{2N}$. Для односторонніх тестів α/(2N) стає α/N. Найбільше спостереження у часовому ряді, яке є більшим за тестову статистику позначається як аномалія. На практиці ми спостерігаємо, що часто існує більше однієї аномалії у даних часових рядів, отриманих від виробництва. Можливо, для виявлення декількох аномалій, можна ітеративно застосувати тест Граббса. Видалення найбільшої аномалії на кожній ітерації зменшує значення N; однак тест Граббса не оновлює значення, отримані з таблиць t-розподілу. Отже, тест Граббса не підходить для виявлення декількох викидів у даних часових рядів.

Існують і інші підходи подібного типу [48], що намагаються встановити розподіл тестової статистики, яка для нульової гіпотези має бути розподілена нормально, або відхилити цю гіпотезу, якщо виявляється що обчислена статистика їй протирічить.

Тест Extreme Studentized Deviate (ESD) [99] (і його узагальнена версія [100] вже можна використовувати для виявлення множинних аномалій у

часовому ряді. У гіршому випадку кількість аномалій може становити не більше 49,9% від загальної кількості точок у заданому часовому ряді. На практиці наші спостереження, засновані на реальних даних, показують, що цього достатньо. ESD обчислює наступну тестову статистику для k найбільш екстремальних значень в наборі даних:

$$C_k = \frac{max_k|x_k - \bar{x}|}{s}, \tag{2.19}$$

Потім тестову статистику порівнюють з критичним значенням, що обчислюється за допомогою рівняння:

$$\lambda_k = \frac{(n-k)t_{p,n-k-1}}{\sqrt{(n-k-1+t^2_{p,n-k-1})(n-k+1)}}, \tag{2.20}$$

щоб визначити, чи є значення аномальним. Якщо значення дійсно аномальне, воно видаляється з набору даних, а критичне значення перераховується з решти даних. ESD повторює цей процес k разів, при цьому кількість аномалій дорівнює найбільшому k так, що $C_k > \lambda_k$. У випадку тесту Граббса, вищезазначене призведе до передчасного закінчення тесту; однак, ESD триватиме доки тест не буде виконано для k викидів.

Однак, на практиці, головна вимога до всіх вищенаведених тестів – нормальність розподілу вибірки, загалом, не виконується через наявні нелінійні та нестаціонарні елементи, а також автокореляції в даних. Тому ця проблема знаходиться зараз в активному дослідженні.

Одним з популярних способів вирішення цієї проблеми – застосувати деяку модель і представити ряд у вигляді:

$$y(k) = f\big(X(k)\big) + \varepsilon(k), \tag{2.21}$$

де y(k) – значення часового ряду в момент k, X(k) – всі значення часового ряду до моменту k, тоді, при правильно підібраній моделі, ε(k) – нормально розподілена випадкова величина. Після побудови такої моделі, можна аналізувати вже ряд з залишків ε(k), застосовуючи до нього вищезазначені тести (оскільки він має нормальний розподіл), і визначати аномальні значення вихідного ряду за відповідними аномальними значеннями ряду залишків. В якості моделі застосовують локальну поліноміальну регрессію (LOESS) [101], або, для випадку нестаціонарних процесів з трендами чи сезонністю, попередньо застосовують процедуру STL декомпозиції [102].

Однак, в даних підходах проблемою є те, що наявність викидів у вхідних даних може впливати на побудову моделі, що може призводити до таких потенційних наслідків:

- Наявність викидів може призвести до некоректно вибраної структури моделі, а відповідно і некоректних результатів тесту
- Навіть якщо модель визначена належним чином, викиди у часовому ряді все ще можуть призвести до зміщення оцінок параметрів а, отже, може вплинути на ефективність визначення аномалій. Типова складність цього підходу полягає в тому, що обидва типи та розташування викидів можуть змінюватись на різних ітераціях оцінки параметрів моделі.
- Деякі викиди можуть не бути ідентифіковані через ефект «маскування», в якому вони стануть частиною процесу відповідно до побудованої моделі.

Тому розглядаються комбіновані підходи, які намагаються використовувати модельний підхід з одночасним вирішенням вищезазначених проблем. Такі підходи запропоновані, наприклад, в роботах [103], [104].

Підводячи певний підсумок, можна зазначити, що визначення та корекція аномалій є дуже складною темою для окремого дослідження, більш

глибокий аналіз якої виходить за межі цієї роботи. Тому для практичного використання спробуємо визначити деякі емпіричні статистики та графічні методи, що зможуть в більшості випадків вказати на наявність проблемних точок в даних.

Розглядаючи графічні методи, варто зазначити, що доволі інформативною виступає звичайний частотний аналіз гістограми часового ряду. Для довільного згенерованого ряду, що не містить аномальних значень, побудуємо гістограму (рис. 2.11)

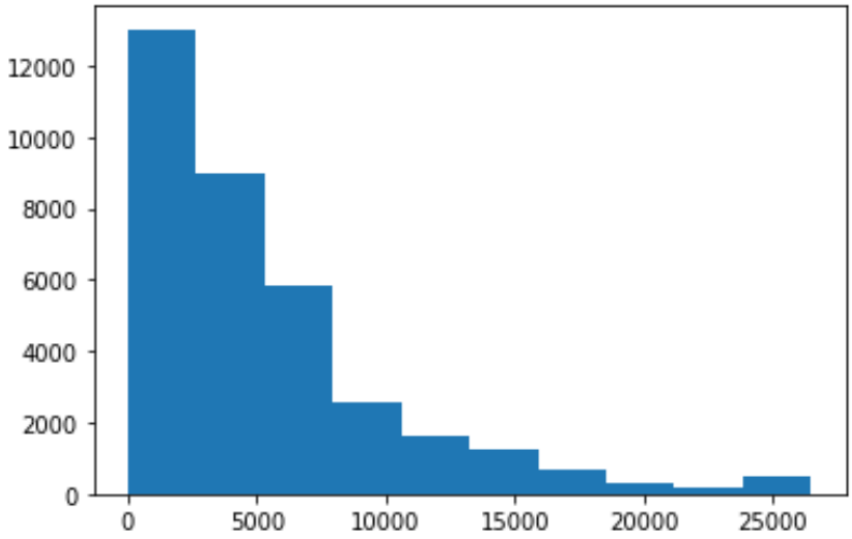


Рис. 2.11 Гістограма для звичайного часового ряду

Проте, якщо додати в цей ряд аномальне значення, наприклад 8000000, то отримаємо наступний графік (рис. 2.12)

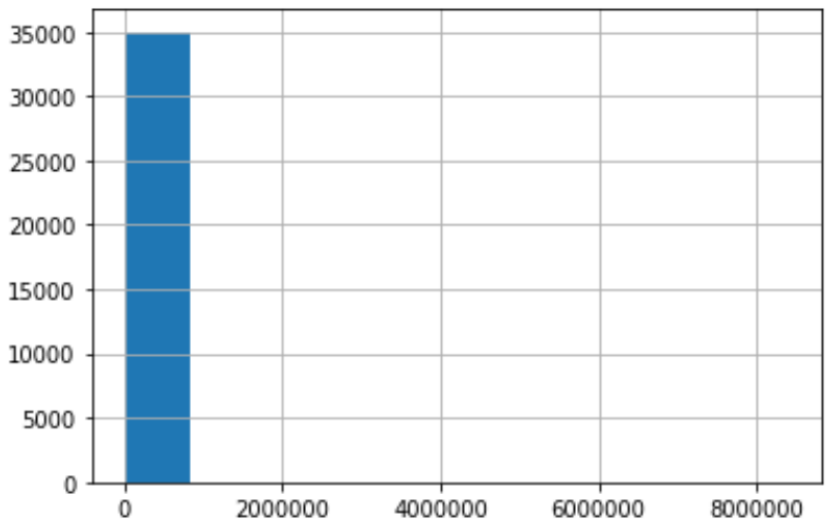


Рис. 2.12 Гістограма часового ряду, в якому наявні аномальні значення

Подібні, «порожні», гістограми можуть свідчити про наявність аномальних значень у вибірці і на практиці є дуже зручним і наочним інструментом.

Методики, що базуються на статистичних тестах і розглянуті раніше, використовують середнє значення і стандартне відхилення. Загальновідомо, що ці показники дуже чутливі до аномальних даних [105]. Для статистичного аналізу, замість використаних у вищезазначених тестах статистик будемо використовувати перцентилі, оскільки вони значно менше чутливі до викидів [106]. Так, наприклад, у вищенаведеному прикладі з додаванням аномального значення, середнє значення значно збільшиться, а медіана майже не зміниться.

Наприклад у роботі [107] пропонується вважати аномальними значення, що виходять за межі $[q_{25}-1{,}5(q_{75}-q_{25});\ q_{75}+1{,}5(q_{75}-q_{25})]$, або $[q_{25}-3(q_{75}-q_{25});\ q_{75}+3(q_{75}-q_{25})]$ для екстремально аномальних значень, де q_i – відповідний перцентиль. Однак, наші практичні експерименти і експерименти з [64] показують, що це доволі груба оцінка. Наприклад, на згенерованому гауссовому розподілі, згідно цих правил, 4% точок буде визначено як викиди і приблизно 1 на 20000 як екстремальні викиди. Тому в даній роботі пропонується використати дещо іншу оцінку інтервалу в $[q_{25}-2(q_{90}-q_{10});$

$q_{75}+2(q_{90}-q_{10})$], що на практиці дає кращі результати [64], і для гауссового розподілу такий інтервал визначить менше ніж 1 точку на 3 мільйони як аномальну. Для визначених аномалій пропонується обробка за допомогою техніки вінсоризації [108], за якою вони будуть замінені на відповідні верхні чи нижні границі для їх пошуку.

Варто також зазначити, що проста заміна викидів, не замислюючись про те, чому вони виникли, є небезпечною практикою. Аномалії можуть надати корисну інформацію про процес, який створив дані, і яку слід враховувати при прогнозуванні. Наприклад, завдяки аналізу аномалій можна отримати алгоритм детектування «неправильного функціонування устаткування», тобто, система прогнозування може давати сигнали у разі некоректної експлуатації приладу, і навіть під час роботи у дуже рідкісних режимах.

Аналогічно варто замислюватись і про пропущені значення. Відсутні дані можуть виникати з багатьох причин, і варто подумати, чи спричинить відсутність певну упередженість в моделі прогнозування. Наприклад, припустимо, що ми вивчаємо дані про продажі для магазину, і пропущені значення трапляються у святкові дні, коли магазин зачинений. Внаслідок цього наступний день може збільшити продажі. Якщо ми не врахуємо це в нашій моделі прогнозування, ми, швидше за все, будемо занижувати продажі в перший день після святкових днів, але завищувати продажі в наступні дні. Один із способів вирішення подібних ситуацій — використання моделі динамічної регресії з фіктивними змінними, які вказують, чи є цей день державним святом чи наступним після нього. Жоден автоматизований метод не може впоратися з такими ефектами, оскільки вони залежать від конкретного контексту прогнозування.

В інших ситуаціях відсутність може бути випадковою. Наприклад, можливо, хтось забув записати дані про продажі, або пристрій запису даних вийшов з ладу. Якщо хронометраж відсутніх даних не є інформативним для

проблеми прогнозування, то відсутні значеннями можна обробляти легше. Поширеними техніками обробки, що можуть бути рекомендовані є:

- Заміна на останнє спостереження
- Заміна на наступне спостереження
- Для стаціонарних даних – лінійна інтерполяція
- Для нестаціонарних даних – STL декомпозиція [102].

Аналогічно до проблеми аномальних значень, більш складна інтерполяція відсутніх значень є окремою проблемою для відповідного дослідження [109-112].

2.4. Перетворення та фільтрація даних з метою вилучення шумових складових

Для зменшення впливу випадкових збурень та похибок вимірів на статистичні дані за допомогою певних перетворень та фільтрації існують різні концепції.

Одним із можливих підходів до врахування статистичних невизначеностей є представлення моделі досліджуваного процесу у просторі станів та оптимальне оцінювання стану за допомогою фільтра Калмана [113].

Якщо застосувати основну ідею загального фільтру Калмана до фільтрації часового ряду, то фільтр Калмана представляє часовий ряд наступним чином:

$$y(t) = x(t) + \varepsilon(t), x(t) = \theta x(t-1) + \omega(t), \qquad (2.22)$$

де y(t) - спостереження в статистичних даних в момент часу t, x(t) - “реальний” стан об’єкта в момент часу t, ε(t), ω(t) - випадкові величини (білий шум з нульовим середнім та дисперсіями Q і R відповідно), при тому, x(t) також виражається через попередні стани, з деяким параметром θ, та також наявною шумовою складовою.

Проте на відміну від звичайної лінійної моделі, особливість фільтра Калмана в тому, що це адаптивний алгоритм, що динамічно на кожному кроці (на кожному наступному значенні ряду) оновлює значення своїх параметрів за певним рекурсивним алгоритмом, що послідовно виконує прогнозування та відповідне коригування значень часового ряду [114].

При застосуванні рекурентної адаптивної процедури, в загальному випадку, для часового ряду рівняння фільтрації буде мати вигляд [48]:

$$\hat{x}(t) = \hat{x}(t-1) + K(t)[y(t) - \hat{x}(t-1)], \tag{2.23}$$

$$\mathrm{K(t)} = \frac{\mathrm{P(t-1)+Q}}{\mathrm{P(t-1)+Q+R}}, \mathrm{P(k)} = \mathrm{P(k-1)} + \mathrm{Q} \tag{2.24}$$

У роботі [7] була розроблена концепція адаптивного моделювання фінансово-економічних процесів, яка заснована на одночасному використанні регресійних моделей і оптимального фільтра Калмана, виконаний порівняльний аналіз отриманих результатів із регресійними моделями без попередньої обробки чи фільтрації. В роботі подаються результати моделювання і прогнозування ВВП України. Наведені дані показують, що вдалося досягти покращення прогнозування ВВП з середньої абсолютної процентної похибки 7,98%, отриманої за допомогою авторегресійної моделі до 5,62%, отриманої за допомогою попереднього використання фільтра Калмана для обробки даних. Застосування оптимального фільтра для попередньої обробки даних дає можливість зменшити похибки оцінок прогнозів. Побудовані функції прогнозування можуть бути успішно використані для оцінювання короткострокових прогнозів.

Фільтр Калмана працює за припущення, що зв’язок між прихованим станом і спостережуваним станом є лінійним, і було показано, що модель працює досить добре. Але якщо змінити це співвідношення на нелінійне,

прогнози стають менш хороші. Фільтр Калмана є оптимальним алгоритмом для лінійних систем, але коли існує нелінійний зв'язок між змінною прихованого стану та змінною, що спостерігається, ці моделі, як правило, не працюють достатньо ефективно [115]. В практичних дослідженнях було виявлено також інші причини, пов'язані з проблемами визначення адекватної моделі, оцінок початкових умов, припущень щодо розподілу та статистичних характеристик шумових компонент, що також можуть впливати на розбіжність алгоритму оптимальної фільтрації [48].

Іншим можливим підходом до перетворення даних для вилучення шумових складових є використання експоненційного згладжування для прогнозування і коригування значень ряду. Експоненціальне згладжування було запропоновано у [116] і стало поштовхом до створення нових успішних методів прогнозування. Прогнози, створені за допомогою методів експоненційного згладжування, є середніми зваженими показниками минулих спостережень, причому вагові коефіцієнти зменшуються експоненціально в міру старіння спостережень. Іншими словами, чим новіше було спостереження, тим вище пов'язана вага. Ця структура генерує надійні прогнози швидко та для широкого діапазону часових рядів, що є великою перевагою та має велике значення для застосування в промисловості.

Найпростіший з методів експоненціального згладжування називається простим експоненційним згладжуванням. Цей метод підходить для прогнозування даних без тренду чи сезонної закономірності.

При простому наївному прогнозуванні, в припущенні, що ми нічого не можемо сказати про процес, всі прогнози на майбутнє дорівнюють останньому спостережуваному значенню ряду:

$$\hat{y}(T + h) = y(T), h = 1,2,3\ldots \tag{2.25}$$

Отже, наївний метод передбачає, що останнє спостереження є єдиним

важливим, а всі попередні спостереження не дають інформації на майбутнє. Такий підхід ще можна розглядати як середньозважене значення, де вся вага приділяється останньому спостереженню.

Припускаючи, що всі попередні значення ряду вкладають однаково важливу інформацію, можемо використати метод усереднення, в якому усі майбутні прогнози дорівнюють простому середньому спостережуваних даних:

$$\hat{y}(T+h) = \frac{1}{T}\sum_{t=1}^{T} y(t)\,, h = 1,2,3\ldots. \qquad (2.26)$$

Отже, метод усереднення припускає, що всі спостереження мають однакову важливість, і надає їм однакові ваги при створенні прогнозів.

Однак, має сенс отримати більш збалансований підхід між цими двома крайнощами. Наприклад, може бути доцільно надати більшу вагу до останніх спостережень, ніж до спостережень з далекого минулого. Саме така концепція лежить в основі простого експоненційного згладжування. Значення розраховуються з використанням середньозважених показників, де вагові коефіцієнти зменшуються експоненціально, оскільки спостереження надходять із подальшого минулого, тому найменші вагові коефіцієнти пов’язані з найстарішими спостереженнями:

$$\begin{aligned}\hat{y}(T) = \alpha y(T) + \alpha(1-\alpha)y(T-1) + \alpha(1-\alpha)^2 y(T-2) + \ldots = \\ = \alpha y(T) + (1-\alpha)\hat{y}(T-1), \qquad (2.27)\end{aligned}$$

де $0 < \alpha < 1$ - згладжуючий параметр. Значення ряду можна отримати ітеративною процедурою, починаючи з першого значення, і обраховуючи зважені середні попередніх значень. Що стосується першого значення ряду, його зазвичай залишають незмінним, або іноді коригують за формулою:

$$\hat{y}(0) = \frac{y(1)+y(2)+y(3)}{3} \tag{2.28}$$

Такий метод вибору першого значення забезпечує хорошу відповідність між згладженим і вихідним рядами для перших рівнів. Якщо останні згладжені рівні ряду, розраховані цим методом з певним параметром, починають різко відрізнятися від відповідних значень вихідного ряду, необхідно змінити значення параметру α на інше [50].

У практичних задачах обробки економічних часових рядів рекомендують вибирати параметр згладжування в інтервалі від 0,1 до 0,3 [117]. В окремих випадках пропонують визначати величину α, виходячи з довжини згладжуваного ряду [118]:

$$\alpha = \frac{2}{m+1}, \tag{2.29}$$

де m — довжина згладжуваного ряду.

Однак більш надійним та об’єктивним способом отримання значень для невідомих параметрів є їх оцінка за даними спостережень використовуючи методи оптимізації. Можна обчислити залишки моделі і спробувати розв’язати задачу мінімізації квадратичної похибки:

$$\sum_{i=1}^{m} e(i)^2 \to \min, e(t) = y(t) - \hat{y}(t), \tag{2.30}$$

Однак, на відміну від звичайного випадку оптимізації для задачі лінійної регресії (де у нас є формули, обчислені за методом найменших квадратів, наприклад, які повертають значення коефіцієнтів регресії, що мінімізують похибку), тут виникає проблема нелінійної мінімізації, і для її вирішення потрібно знайти відповідний підхід [64].

За функціональними особливостями даний підхід з його недоліками дуже

схожий на фільтр Калмана, тому у [119] було запропоновано включати в рівняння компоненти тренду та сезонності. Такі моделі отримали назву моделі Хольта-Вінтерса. Усі можливі комбінації експоненційного згладжування з можливим адитивним чи мультиплікативним врахуванням тренду чи сезонності полягли в основу класифікації [120]. У роботі [121] розглядається також так званий “послаблений” адитивний тренд, а в роботі [122] було запропоновано також “послаблений” мультиплікативний тренд.

Однак, в запропонованій в даній роботі методиці, для обробки нелінійних та нестаціонарних компонент пропонується використання нейронних мереж, тому процедура цифрової фільтрації необхідна лише для видалення шумової складової, а отже пропонується використовувати найпростіший варіант експоненційного згладжування.

Застосування процедури цифрової фільтрації для попередньої обробки даних дійсно дає можливість зменшити похибки оцінок прогнозів [7]. Але, як було вказано раніше, подібні підходи до фільтрації не завжди можуть давати стабільний результат через низку причин (непередбачуваність “шумової” компоненти, її розподілу та характеристик, можливе видалення необхідних для подальшого моделювання нелінійних характеристик через фільтрацію, проблеми оцінки параметрів) [115], [48]. Тому підходи цифрової фільтрації варто застосовувати обережно, не завжди як необхідний етап процесу побудови моделі, а як можливий варіант, обов’язково перевіряючи в кінці процесу моделювання якість отриманих прогнозів з та без використання попередньої фільтрації.

Висновки до розділу

В даному розділі вибрано класи фінансово-економічних процесів, що будуть використані для моделювання та прогнозування, а також виконано опис математичних моделей та підходів, що можуть на базі статистичних даних у вигляді часових рядів, використовуватись для опису динаміки нелінійних нестаціонарних процесів.

Запропоновано підходи статистичного та графічного аналізу, а також строгі статистичні тести для визначення нелінійності та нестаціонарності процесу. Для визначення нелінійності пропонується використовувати нейромережевий тест Уайта, для нестаціонарності - тест KPSS.

Розглянуто методи для виявлення та обробки аномальних та пропущених значень. Запропоновано деякі емпіричні статистики та графічні методи, що зможуть в більшості випадків вказати на наявність проблемних точок в даних, а також формальні статистичні критерії, такі як ESD.

Для виділення шумової складової з часового ряду розглянуто фільтр Калмана та метод експонеційного згладжування, проаналізовано їх переваги та недоліки. Застосування процедури цифрової фільтрації для попередньої обробки даних дійсно дає можливість зменшити похибки оцінок прогнозів, але подібні підходи не завжди можуть давати стабільний результат через низку причин. Тому підходи цифрової фільтрації варто застосовувати обережно, не завжди як необхідний етап процесу побудови моделі, а як можливий варіант, обов’язково перевіряючи в кінці процесу моделювання якість отриманих прогнозів з та без використання попередньої фільтрації.

РОЗДІЛ 3

МАТЕМАТИЧНІ МОДЕЛІ І МЕТОДИ ДЛЯ МОДЕЛЮВАННЯ І ПРОГНОЗУВАННЯ НЕЛІНІЙНИХ НЕСТАЦІОНАРНИХ ПРОЦЕСІВ

3.1. Регресійні моделі для виділення та моделювання лінійної складової процесу

В загальному сенсі, моделлю, що описує стохастичний процес представлений у вигляді часового ряду можемо називати функцію поточного стану від попередніх:

$$yt = f(yt\text{-}1, yt\text{-}2, \ldots, y1) \tag{3.1}$$

Для виділення лінійної складової будемо користатися наступним описом лінійної моделі порядку p у загальному вигляді:

$$y_t = a_0 + \sum_{i=1}^{p} a_i y_{t-i} + \varepsilon_t, \tag{3.2}$$

де, ai - константні дійсні числа, ε_t - залишки моделі, що не можуть бути описані лінійно.

Моделі подібного типу називаються авторегресійними, тобто такими, що описують регресію змінної відносно себе і позначаються AR(p). Тобто, для виділення і опису лінійної компоненти процесу будемо використовувати авторегресивну модель AR(p).

Для побудови такої моделі на першому етапі необхідно визначити кількість авторегресійних компонент (порядок p моделі). Для цього

використовується кореляційний аналіз [49].

Подібно до того, як кореляція вимірює ступінь лінійного зв'язку між двома змінними, автокореляція вимірює лінійний зв'язок між лаговими значеннями часового ряду, наприклад між поточним часовим рядом, і рядом на одне спостереження в минулому.

Наприклад, для визначення ступеню лінійного впливу попереднього спостереження на поточне, можна обчислити коефіцієнт кореляції між поточним рядом і рядом з першим лагом, або з попереднім його станом:

$$\rho_{y_t,y_{t-1}} = \frac{Cov(y_t,y_{t-1})}{\sqrt{Var(y_t)Var(y_{t-1})}} \tag{3.3}$$

Як відомо з теорії математичної статистики, цей коефіцієнт вимірює ступінь лінійної залежності між двома змінними, і може набувати значень $-1 \leq \rho_{x,y} \leq 1$, де $\rho_{x,y} = \rho_{y,x}$, і $\rho_{x,y} = 0$ буде означати відсутність лінійного зв'язку, або про такі змінні кажуть, що вони не корелюють. $\rho_{x,y} = \pm 1$ вказує на прямий чи обернений лінійний зв'язок.

Аналогічно, коефіцієнт кореляції можна обрахувати і для позаминулого значення ряду відносно поточного і для всіх інших. Позначивши коефіцієнт автокореляції за $r_k = \rho_{y(t),y(t-k)}$ в загальному вигляді маємо:

$$r_k = \frac{\sum_{t=k+1}^{T}(y_t-\bar{y})(y_{t-k}-\bar{y})}{\sum_{t=1}^{T}(y_t-\bar{y})^2}, \tag{3.4}$$

Де $\bar{y}$ - середнє значення ряду, T - довжина ряду.

Для візуального аналізу зручно значення автокореляції для різних лагів зображати на графіку (рис.3.1):

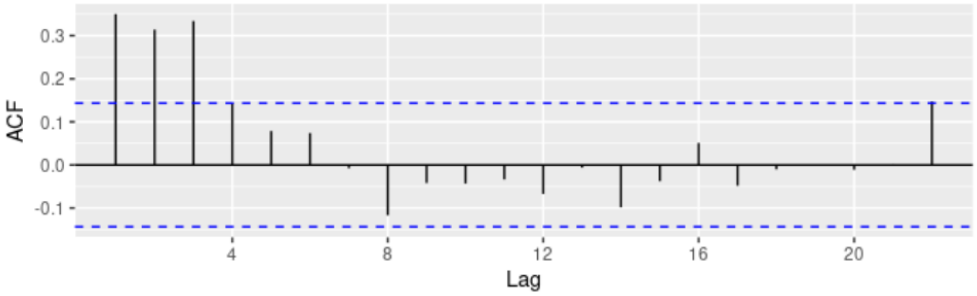


Рис.3.1 Графік автокореляційної функції

На графіку для кожного лагу видно за відповідним стовпчиком значення відповідного коефіцієнту автокореляції. Власне, автокореляційна функція (АКФ) дозволяє визначити коефіцієнт кореляції для будь-якого лагу.

Однак, якщо y(t) і y(t-1) корелюють, то y(t-1) і y(t-2) також будуть мати певну залежність. Однак тоді буде з'являтися залежність між y(t) і y(t-2), але не через її реальну залежність або будь-яку нову інформацію, що міститься в y(t-2), яку можна використовувати для прогнозування y(t), а лише тому, що вони обидва пов'язані з y(t-1).

Наприклад, для АКФ процесу AR(1) маємо:

$$\rho_0 = 1, \rho_1 = a_1, \rho_2 = a_1^2, \ldots, \rho_s = a_1^s, \tag{3.5}$$

тобто дійсно, значення y(t) і y(t-2) корельовані, незважаючи на те, що y(t-2) у моделі відсутнє. З цього прикладу також видно, що:

$$\rho_2 = \rho_1 * \rho_1 = \rho_1^2 \tag{3.6}$$

Очевидно, що такі "непрямі" кореляції є в АКФ будь-якого лінійного авторегресивного процесу. Загалом це спричиняє те, що АКФ недостатньо коректно визначає порядок авторегресії.

Щоб подолати цю проблему, ми можемо використовувати часткову

автокореляційну функцію (ЧАКФ). Вона вимірює співвідношення між y(t) та y(t-k) після усунення наслідків проміжних лагів. Перша часткова автокореляція ідентична першій автокореляції, тому що між ними нема чого видалити.

В загальному випадку кожну часткову автокореляцію можна оцінити як відповідний коефіцієнт в авторегресійній моделі, тобто ϕ_k - к-й коефіцієнт часткової автокореляції буде відповідати $a_к$ моделі AR(k). На практиці існують більш ефективні алгоритми обчислень значень ЧАКФ, ніж відповідна побудова усіх необхідних авторегресивних моделей, при тому, вони дають ті самі результати [64].

На практиці ЧАКФ обчислюють за наступними виразами [48]:

$$\phi_1 = \rho_1, \phi_2 = \frac{\rho_2 - \rho_1^2}{1 - \rho_1^2}, \phi_s = \frac{\rho_s - \sum_{j=1}^{s-1} \phi_{s-1,j} \rho_{s-j}}{1 - \sum_{j=1}^{s-1} \phi_{s-1,j} \rho_j} \quad (3.7)$$

Аналогічно, ЧАКФ зручно зображати на графіку (рис. 3.2):

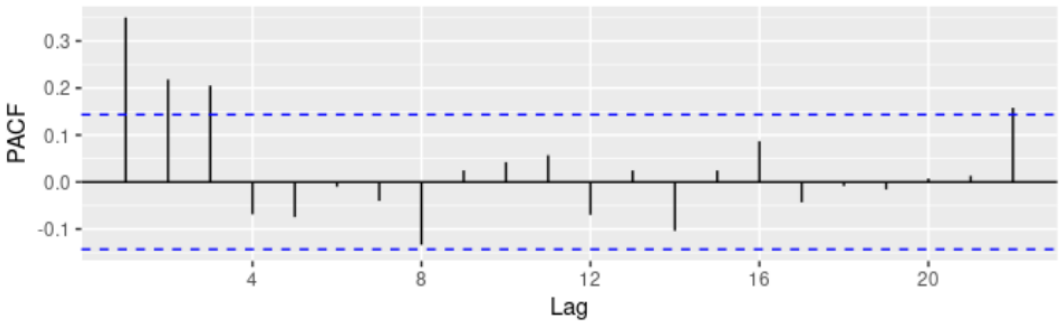


Рис.3.2 Графік ЧАКФ

Бачимо, що значення для першого лагу не змінилося відносно рис 3.1, але для наступних зменшилося, внаслідок видалення накопичувального впливу.

На практиці для оцінки значущості відповідних кореляційних коефіцієнтів використовують критичне значення, що дорівнює [64]:

$$r = \frac{\pm 1.96}{\sqrt{T}} \tag{3.8}$$

Відповідний порядок моделі AR(p) визначається за номером останнього компоненту (лагу), значення ЧАКФ якого перевищує критичне значення. Його можна зручно визначати за допомогою графічного аналізу наведених графіків, або за допомогою відповідної автоматизованої програмної процедури. Наприклад, для наведеного на рис. 3.2 процесу, буде вибрана модель AR(3), оскільки останній лаг, значення ЧАКФ якого перевищувала критичне значення (позначене пунктирною лінією) був саме 3й лаг.

Модель для виділення та опису лінійної компоненти такого процесу буде мати загальний вигляд:

$$y(t) = a_0 + a_1 y(t-1) + a_2 y(t-2) + a_3 y(t-3) + \varepsilon \tag{3.9}$$

Для визначення невідомих коефіцієнтів a_i та подальшого використання моделей для прогнозування будемо використовувати методи і моделі регресійного аналізу.

Регресійний аналіз – набір статистичних методів дослідження залежностей між залежною змінною y і однією або декількома незалежними змінними x_1, x_2, ..., x_N. Незалежні змінні інакше називають регресорами, предикторами або ознаками, а залежні змінні – вихідними, цільовими або прогнозованими.

Іншим способом визначення порядку моделі є побудова великої кількості моделей різних порядків, після побудови яких вони перевіряються на адекватність за різними статистичними критеріями, що будуть розглянуті далі. Для побудови великої кількості моделей застосовують процедуру поступової (покрокової) регресії. У статистиці поступова (покрокова)

побудова регресії — це метод побудови регресійних моделей, у якому вибір прогнозних змінних здійснюється автоматичною процедурою. На кожному кроці розглядається змінна для додавання або віднімання з набору пояснювальних змінних на основі деякого заздалегідь заданого критерію [123].

Якщо є велика кількість можливих предикторів, теоретично неможливо перебрати всі можливі моделі. Наприклад, 40 можливих предикторів призводить до $2^{40} > 1$ трильйону можливих моделей. Отже, необхідна стратегія для обмеження кількості моделей, які підлягають дослідженню.

Підхід, який працює досить добре на практиці [64], - це покрокова регресія назад, що в загальному випадку складається з наступних кроків:

1. Побудова починається з моделі, яка містить усі потенційні предиктори.

2. Видалення одного предиктора. Модель зберігається, якщо вона покращує показник адекватності моделі.

3. Процедура повторюється до тих пір, поки покращення критерію не зупиниться.

Якщо кількість потенційних предикторів занадто велика, то покрокова регресія назад не працюватиме, і замість неї можна використовувати покрокову регресію вперед. Ця процедура починається з моделі, яка включає лише вільний член (перетин, зміщення). Предиктори додаються по одному, і той, який найбільше покращує критерій адекватності, зберігається в моделі. Процедуру повторюють до тих пір, поки не буде досягнуто зупинки поліпшення критерію.

В якості альтернативи для зворотного або прямого напрямку початковою моделлю може бути модель, яка включає підмножину потенційних предикторів. У цьому випадку необхідно включити додатковий крок. Для процедури зворотного ходу ми також повинні розглянути можливість додавання предиктора до кожного кроку, а для процедури

прямого ходу ми також повинні розглянути можливість відкидання предиктора з кожним кроком. Такі підходи називаються гібридними процедурами.

Важливо усвідомлювати, що будь-який поетапний підхід не гарантовано призведе до найкращої моделі, але майже завжди веде до якісної, адекватної моделі.

Розглянемо тепер підходи регресійного аналізу для визначення невідомих параметрів - коефіцієнтів регресії.

У найпростішому випадку для однієї незалежної змінної лінійна регресія є лінійною функцією між залежною змінною y і предиктором x:

$$y = a_0 + a_1 x \tag{3.10}$$

Таке співвідношення називається теоретичною лінійною регресійною моделлю; a_0 і a_1 – теоретичні параметри (теоретичні коефіцієнти) регресії.

Для визначення значень теоретичних коефіцієнтів регресії необхідно знати та використовувати всі значення змінних x та y генеральної сукупності, що практично неможливо. Отже, потрібно побудувати так зване емпіричне рівняння регресії, що будується з вибірки обмеженого обсягу, де коефіцієнти є оцінками теоретичних коефіцієнтів регресії:

$$\hat{y} = \hat{a}_0 + \hat{a}_1 x, \tag{3.11}$$

де $\hat{a}_0$ і $\hat{a}_1$ – оцінки невідомих параметрів a_0 і a_1.

Задачі лінійного регресійного аналізу полягають у тому, щоб за наявними статистичними даними (x_i,y_i), i =1,2, ..., n, для змінних x і y:

а) отримати найкращі оцінки $\hat{a}_0$ і $\hat{a}_1$ невідомих параметрів a_0 і a_1;

б) перевірити статистичні гіпотези про параметри моделі;

в) перевірити, наскільки добре модель узгоджується зі статистичними даними (перевірка адекватності моделі).

Для того, щоб показати, що кожне індивідуальне значення y_i відхиляється від відповідного умовного математичного сподівання, у модель додають випадкову шумову складову ε_i:

$$y_i = E(y|x = x_i) + \varepsilon_i = a_0 + a_1 x_i + \varepsilon_i \qquad (3.12)$$

Отже, індивідуальні значення y_i подають у вигляді суми двох компонент – детермінованої ($a_0 + a_1 x_i$) та випадкової (ε_i) (Рис. 3.3).

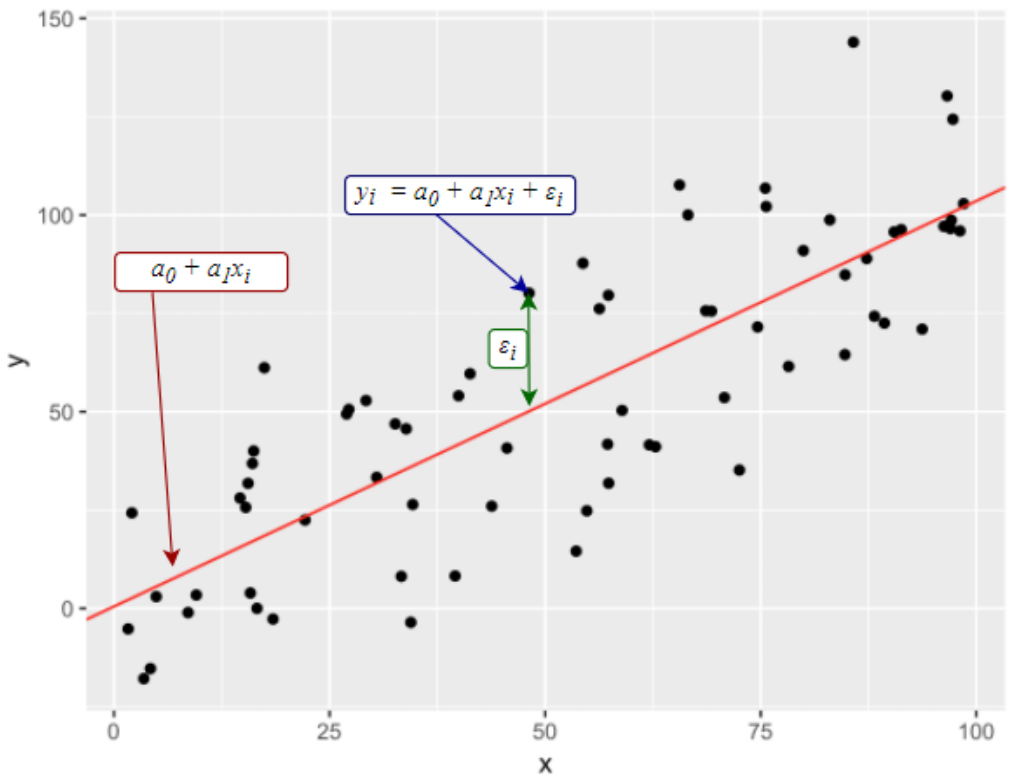


Рис. 3.3 Графічне зображення побудови лінійної регресійної моделі

Задача полягає в тому, щоб за заданим набором даних (x_i, y_i), i=1, 2, ..., n, знайти такі значення оцінок невідомих параметрів a_0 і a_1, щоб побудована

лінія регресії була найкращою в деякому сенсі серед усіх інших подібних прямих.

Для вирішення цієї задачі використовують метод найменших квадратів (МНК), що має свою назву через процедуру мінімізації відповідної міри якості.

Мірою якості знайдених оцінок є умова мінімізації суми квадратів помилок:

$$\mathrm{L}(\hat{a}_0, \hat{a}_1) = \sum_{i=1}^{n} \varepsilon_i^2 = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} (y_i - \hat{a}_0 - \hat{a}_1 x_i)^2 \qquad (3.13)$$

З рівняння видно, що квадратична функція $\mathrm{L}(\hat{a}_0, \hat{a}_1)$ неперервна, опукла та обмежена знизу ($\mathrm{L} \geq 0$), а отже має мінімум.

Необхідною умовою існування мінімуму неперервно-диференційованої функції двох змінних є рівність нулю її часткових похідних:

$$\begin{cases} \frac{\partial L}{\partial a_0} = -2 \sum (y_i - \hat{a}_0 - \hat{a}_1 x_i) = 0 \\ \frac{\partial L}{\partial a_1} = -2 \sum (y_i - \hat{a}_0 - \hat{a}_1 x_i) x_i = 0 \end{cases} \qquad (3.14)$$

Виконавши перетворення, отримаємо систему нормальних рівнянь:

$$\begin{cases} n\hat{a}_0 + \hat{a}_1 \sum x_i = \sum y_i \\ \hat{a}_0 \sum x_i + \hat{a}_1 \sum x_i^2 = \sum x_i y_i \end{cases} \qquad (3.15)$$

Розв’яжемо систему відносно її невідомих оцінок параметрів $\hat{a}_0$ і $\hat{a}_1$:

$$\hat{a}_1 = \frac{\sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i - n \sum_{i=1}^{n} x_i y_i}{(\sum_{i=1}^{n} x_i)^2 - n \sum_{i=1}^{n} x_i^2}; \ \hat{a}_0 = \frac{\sum_{i=1}^{n} x_i \sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i^2 \sum_{i=1}^{n} y_i}{(\sum_{i=1}^{n} x_i)^2 - n \sum_{i=1}^{n} x_i^2}; \qquad (3.16)$$

В загальному випадку коли в моделі необхідно враховувати декілька вхідних змінних, модель лінійної регресії має вигляд:

$$y_i = a_1 x_{i1} + \cdots + a_m x_{im} + \varepsilon_i, i = 1, \dots, n \qquad (3.17)$$

Для зручності запису і обчислень рівняння регресії подають у матричній формі у наступному вигляді:

$$Y = X\alpha + \varepsilon, \qquad (3.18)$$

де Y = [y_1,…,y_n] - вектор прогнозованої змінної, α = [a_1,…,a_n] - шуканий вектор параметрів моделі - коефіцієнтів лінійної регресії, матриця X має вигляд:

$$\begin{bmatrix} 1 & \cdots & x_{1,m} \\ \vdots & \ddots & \vdots \\ 1 & \cdots & x_{n,m} \end{bmatrix} \qquad (3.19)$$

Аналогічно проводячи обчислення, як у попередньому прикладі з однією змінною, необхідно брати часткові похідні для параметрів моделі. Тоді з невиродженості матриці X можна знайти вектор коефіцієнтів МНК [124]:

$$\alpha = (X^T X)^{-1} X^T y \qquad (3.20)$$

Такий підхід називається оцінкою за методом найменших квадратів, оскільки він дає найменше значення для суми квадратів помилок. Знаходження найкращих оцінок коефіцієнтів часто називають “навчанням” моделі. Таким чином була отримана лінія, показана на рис. 3.3.

Існують і інші підходи, які можуть бути використані для оцніки параметрів лінійної моделі [48], [125], [126].

Для задачі моделювання часових рядів, предикторами стають лагові значення часового ряду, вигляд і кількість яких була визначена за ЧАКФ раніше.

3.2. Прогнозування за допомогою AR моделі

Розглянемо, як використовувати побудовану модель AR(p) для прогнозування на один та на декілька кроків вперед.

Структура моделі AR(p) така, що дозволяє виконувати прогнозування на один крок без додаткових перетворень. Використовуючи попередні значення ряду і підставляючи їх в рівняння моделі можна отримати оцінку прогнозу цільової змінної. Такий спосіб прогнозування на один крок з постійним додаванням нової, доступної, точно відомої на момент часу виконання прогнозу інформації для прогнозування наступних кроків є найбільш точним і коректним [64]. Але на практиці іноді необхідно знайти оцінку прогнозу одразу на декілька кроків вперед, не знаючи майбутніх значень ряду чи інших предикторів, тоді необхідно застосувати деякі попередні перетворення різницевих рівнянь, що лежать в основі моделі AR.

Розглянемо деякі можливі підходи до обчислення прогнозованих значень на декілька кроків вперед.

Наприклад, розглянемо рівняння для моделі AR(1):

$$y(t) = a_0 + a_1 y(t-1) + \varepsilon(t) \tag{3.21}$$

Збільшимо незалежну змінну від часу на одну одиницю і запишемо рівняння моделі знову:

$$y(t+1) = a_0 + a_1 y(t) + \varepsilon(t+1) \tag{3.22}$$

Якщо коефіцієнти a_0, a_1 відомі, то можна знайти умовне математичне сподівання на основі відомої інформації до моменту t включно:

$$E_t[y(t+1)] = E_t[y(t+1)|y(t), y(t-1), \dots, \varepsilon(t), \varepsilon(t-1), \dots] = a_0 + a_1 E_t[y(t)] = a_0 + a_1 y(t) \quad (3.23)$$

оскільки y(t) є відомою константою.

Аналогічно, запишемо рівняння для часу t + 2

$$y(t+2) = a_0 + a_1 y(t+1) + \varepsilon(t+2) \quad (3.24)$$

знайдемо умовне математичне сподівання:

$$E_t[y(t+2)] = a_0 + a_1 E_t[y(t+1)] = a_0 + a_1 E_t[a_0 + a_1 y(t)] = a_0 + a_0 a_1 + a_1^2 y(t) \quad (3.25)$$

Отже, для загального випадку прогнозування на h кроків вперед можна записати:

$$E_t[y(t+h)] = a_0(\sum_{i=0}^{h-1} a_i^j) + a_1^h y(t) = a_0 \sum_{i=0}^{h-1} a_i^j + a_1^h y(t)$$
(3.26)

Отримане рівняння буде функцією прогнозування на довільне число кроків вперед. Прогноз представляє собою збіжний процес, якщо $a_1<1$, тобто

$$\lim_{h\to\infty} E_t[y(t+h)] = \frac{a_0}{1-a_1} \quad (3.27)$$

де a_1 – знаменник геометричної прогресії.

Для процесу AR(p) довільного порядку, функція прогнозування може бути узагальнена наступним чином [48]:

$$E_t[y(t+h)] = a_0 + \sum_{i=1}^{p} a_i E_k[y(t+h-i)] \quad (3.28)$$

3.3. Критерії адекватності моделі AR для виділення та моделювання лінійної складової процесу

Авторегресивна модель AR(p) має як основу класичну лінійну модель множинної регресії, що накладає на неї ряд статистичних вимог щодо розподілу і властивостей залишків (ε) [48], [64], [125], [50]. Загалом, всі вимоги зводяться до того, що модель адекватно описує вхідний процес, якщо залишки моделі є білим шумом, що можна перевірити, наприклад за критерієм Льюнга-Бокса [127]. Незалежність значень залишків та відсутність між ними автокореляцій досліджується часто за допомогою критерія Дарбіна-Уотсона [128], [129]. Рівність нулю математичного сподівання залишків можна перевірити за критерієм Стьюдента [130].

Однак, в даній роботі, розглядаються нелінійні нестаціонарні процеси, які зазвичай зустрічаються на практиці у фінансово-економічних процесах, а тому лінійна модель в більшості випадків не може адекватно описати динаміку таких процесів, а використовується в даній роботі лише для опису лінійної складової для подальшого використання в комбінованій моделі. Тому використання подібних критеріїв не має сенсу, але існує необхідність застосування певного числового критерію адекватності для вибору оптимальної структури лінійної складової з отриманих можливо за різними методами пошуку оптимальних параметрів чи вибору різного порядку моделі AR.

Позначимо суму квадратів залишків моделі за SSE:

$$SSE = \sum_{t=1}^{T} e_t^2 , e_t = y_t - \hat{y}_t, \quad (3.29)$$

де T - довжина часового ряду, що використовувався для моделювання процесу, $\hat{y}_t$ - прогнози моделі AR для кроку t, e_i - залишки моделі AR.

Інформаційний критерій Акайке визначається як:

$$AIC = T\ln(\frac{SSE}{T}) + 2(p + 1), \quad (3.30)$$

де p - порядок моделі AR(p), T - довжина часового ряду, що використовувався для моделювання процесу.

Різні джерела використовують дещо різні визначення для AIC, проте всі вони будуть призводити до вибору однієї моделі. Частина рівняння з p+1 виникає тому, що є p+1 параметр в моделі: p коефіцієнтів для предикторів, один параметр для зміщення (вільний член або “перетин”). Ідея такого критерію полягає в тому, щоб “штрафувати” точність моделі (SSE) кількістю параметрів, які необхідно оцінити. Модель з мінімальним значенням AIC часто є найбільш адекватною моделлю.

Схожим критерієм адекватності моделі є інформаційний критерій Байєса або критерій Байєса-Шварца:

$$BIC = T\ln(\frac{SSE}{T}) + \ln(T)(p + 1) \quad (3.31)$$

Модель, обрана BIC, або така сама, як модель, обрана AIC, або модель з меншою кількістю параметрів. Це пояснюється тим, що BIC сильніше штрафує кількість параметрів, ніж AIC.

В статистичній літературі часто надають перевагу BIC, оскільки він має властивість, якщо за наявності справжньої відомої моделі процесу, BIC вибере саме цю модель, маючи достатньо даних. Однак насправді справжня відома

модель рідко існує, якщо взагалі існує, і навіть якщо існує справжня відома модель процесу, вибір цієї моделі не обов'язково дасть найкращі прогнози (оскільки оцінки параметрів можуть бути неточними) [64]. Однак, для великих вибірок, асимптотичні властивості кращі у критерія Байєса-Шварца [48]. Оскільки в сучасних практичних задачах для моделювання та прогнозування фінансово-економічних процесів зазвичай працюють з великими обсягами даних, в даній роботі буде використано саме цей критерій.

3.4. Авторегресивний підхід до моделювання і прогнозування нестаціонарних процесів

Можна застосувати підхід на базі регресивних моделей для того щоб використати залишки - минулі помилки в моделі авторегресії для її коригування та моделювання відповідних залежностей. Для цього використовують інструмент ковзного середнього. Такий підхід є дуже глибоко проаналізованим і широко вживаним [125]. В даній роботі буде використано авторегресивні підходи для порівняльного аналізу з пропонованою методикою.

Загальна формула для зваженого ковзного середнього має вигляд:

$$MA(k) = \frac{\sum_{i=1}^{N} w_i y(k-i+1)}{\sum_{i=1}^{N} w_i} \tag{3.32}$$

де w_i – вагові коефіцієнти, y − часовий ряд вхідних даних.

В загальній формулі для простого арифметичного ковзного середнього всі вагові мають однакову вагу (одиничну).

Розглянемо авторегресію з ковзним середнім. Модель авторегресії з ковзним середнім (ARMA(p,q)) має вигляд:

$$y(k) = a_0 + \sum_{i=1}^{p} a_i y(k-i) + \sum_{j=1}^{q} b_j \varepsilon(k-j) + \varepsilon(k) \qquad (3.33)$$

Тобто процедура побудови авторегресійної моделі, описана в попередньому розділі застосовується вже до ряду залишків ε(k) для моделювання існуючих в похибках залежностей, пошуку значення q і побудови відповідної моделі.

Для прогнозування за допомогою моделей ARMA(p,q) також застосовуються аналогічні до AR(p) моделей підходу.

Розглянемо, наприклад, функцію прогнозування для процесу ARMA(2,1), який описується наступним рівнянням:

$$y(k) = a_0 + a_1 y(k-1) + a_2 y(k-2) + \varepsilon(k) + \beta_1 \varepsilon(k-1) \qquad (3.34)$$

Для наступного моменту часу можна записати:

$$y(k+1) = a_0 + a_1 y(k) + a_2 y(k-1) + \varepsilon(k+1) + \beta_1 \varepsilon(k) \qquad (3.35)$$

Умовне математичне сподівання для y(k+1) має вигляд:

$$E_k[y(k+1)] = a_0 + a_1 y(k) + a_2 y(k-1) + \beta_1 \varepsilon(k) \qquad (3.36)$$

де ε(k) розглядається як відома величина на момент k включно. При цьому $E_k[\varepsilon(k+j)] = 0, \forall j > 0$.

Для моменту часу k + 2 маємо умовне математичне сподівання:

$$E_k[y(k+2)] = a_0(1+a_1) + (a_1^2 + a_2) y(k) + a_1 a_2 y(k-1) + a_1 \beta_1 \varepsilon(k) \qquad (3.37)$$

З отриманих формул для умовного математичного сподівання можемо встановити, що рекурсивна формула для прогнозу на довільне число s кроків може бути записана у наступному вигляді:

$$E_k[y(k+s)] = a_0 + a_1 E_k[y(k+s-1)] + a_2 E_k[y(k+s-2)]$$
(3.38)

Якщо корені характеристичного рівняння знаходяться всередині одиничного кола, то оцінка прогнозу асимптотично збігається до безумовного середнього значення

$$\lim_{s\to\infty} E_k[y(k+s)] = \frac{a_0}{1-a_1-a_2} \quad (3.39)$$

а для довільного процесу АРКС(p,q) оцінку умовного прогнозу можна записати як [48]:

$$E_k[y(k+s)] = a_0 + \sum_{i=1}^{p} a_i E_k[y(k+s-i)] \quad (3.40)$$

Розглянемо також авторегресію з інтегрованим ковзним середнім (ARIMA). Якщо для процесу, що моделюється, можна виділити порядок тренду, то відповідну модель можна записати наступним чином:

$$y'^{(k)} = a_0 + \sum_{i=1}^{p} a_i y'(k-i) + \sum_{j=1}^{q} b_j \varepsilon(k-j) + \varepsilon(k). \quad (3.41)$$

Відповідні процеси називають процесами авторегресії з інтегрованим ковзним середнім – ARIMA(p, q, d), де d – число одиничних коренів характеристичного рівняння (відповідне характеристичне рівняння може мати одиничні корені, тобто один або більше коренів характеристичного рівняння

можуть набувати значення «1»). Процеси цього класу нестаціонарні – вони мають тренд, порядок якого визначається числом одиничних коренів. Якщо d = 1, то тренд лінійний; якщо d = 2, то тренд квадратичний і т. ін. Порядок тренду d визначають за допомогою відповідних статистичних тестів на наявність тренду та послідовної процедури диференціювання вхідного ряду [64]. Тобто, дані перевіряються на нестаціонарність за допомогою формальних тестів (наприклад, тест KPSS). В разі якщо тест вказує на наявність нестаціонарності, виконується процес диференціювання ряду (застосування оператору перших різниць). До диференційованого ряду застосовується обраний тест ще раз. Ця процедура виконується до тих пір, поки ряд не стане стаціонарним. Кількість застосованих операторів диференціювання – порядок тренду заданого ряду - це буде параметр ‘d’ в ARIMA моделі.

Загальну методику для автоматичної побудови авторегресивних моделей для нестаціонарних процесів досліджено і описано у праці [131].

Цей алгоритм отримав назву за прізвищами авторів - алгоритм Хайдмана-Хандакара і в загальному вигляді його можна описати наступним чином:

1. Параметр d визначається процедурою поступового диференціювання із застосуванням KPSS тесту [94].

2. Після визначення параметру d, до ряду відповідну кількість разів застосовується процедура його диференціювання для приведення ряду до стаціонарного.

3. Параметри p і q вибираються шляхом мінімізації інформаційного критерію Акайке з використанням гібридної процедури для покрокової регресії. Для того, щоб уникнути перебору численної кількості моделей, процедура пошуку виглядає наступним чином:

А. Будуються наступні моделі:

- ARIMA(0,d,0)
- ARIMA(2,d,2)
- ARIMA(1,d,0)

- ARIMA(0,d,1)

B. Найкраща за критерієм Акайке модель з попереднього кроку визначається як “поточна найкраща модель”

C. Поки не може бути знайдена модель за меншим значенням AIC виконується зміна параметрів p і q на ±1 від поточної найкращої. Найкраща розглянута модель (або поточна модель, або одна з цих варіацій) стає новою “поточною найкращою моделлю”.

Отже, перевагами авторегресивних моделей, є простота структури, за рахунок якої відбувається суттєвий виграш щодо обчислювальних витрат, а тому є можливість її оперативної адаптації до характеристик процесу в реальному часі. Однак, оскільки дана модель являє собою звичайну множинну лінійну регресію, побудовану на лагових значеннях, то вона не здатна враховувати нелінійний вплив як внутрішніх так і зовнішніх факторів, комбінований вплив факторів, а також не може бути адекватно побудована в разі наявності колінеарних факторів. Тому цей підхід є доволі ефективним для виділення лінійної складової процесу, однак для виділення нелінійної складової пропонується використати інший підхід, а саме підхід на базі нейронних мереж.

3.5. Нейронні мережі

Ідея штучних нейронних мереж з’явилася як спроба побудувати електронну або програмну модель, що буде аналогією нейронної структури мозку людини, який головним чином навчається з досвіду.

Основним елементом людського мозку є певні клітини, відомі як нейрони, які здатні запам'ятовувати, думати та застосовувати попередній досвід до кожної дії, що радикально відрізняє їх від інших клітин тіла.

У 1943 році нейрофізіолог Воррен Маккаллок і логік Уолтер Піттс

об’єдналися, щоб розробити математичну модель штучного нейрона. У своїй статті [132] вони заявили, що: через характер нервової діяльності “все або нічого”, нейронні події та відносини між ними можна розглядати за допомогою пропозиційної логіки. Виявлено, що поведінку кожної мережі можна описати цими термінами.

Людський нейрон складний, має власні компоненти, підсистеми і механізми управління і передає інформацію через велику кількість електрохімічних зв'язків. Існує близько сотні різних класів нейронів. Водночас нейрони та зв'язки між ними утворюють двійковий, нестабільний та асинхронний процес, що відрізняється від обчислювального процесу традиційних комп'ютерів. Штучні нейронні мережі моделюють лише найважливіші елементи складного мозку, що надихає вчених та розробників на нові способи вирішення проблеми.

Розглянемо біологічний нейрон (рис. 3.4). Нейрон (нервова клітина) складається з тіла клітини – соми, і двох типів зовнішніх деревоподібних відгалужень: аксона і дендритів. Тіло клітини вміщує ядро, що містить інформацію про властивості нейрону, і плазму, яка продукує необхідні для нейрону матеріали. Нейрон отримує сигнали (імпульси) від інших нейронів через дендрити (приймачі) і передає сигнали, згенеровані тілом клітки, вздовж аксона (передавач), що наприкінці розгалужується на волокна. На закінченнях волокон знаходяться синапси.

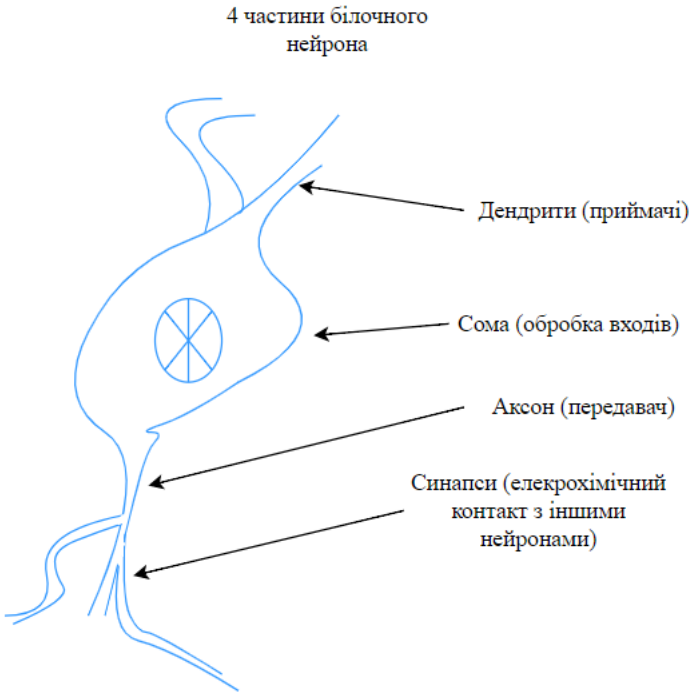


Рис.3.4 Схема біологічного нейрона

Маккалок і Піттс [132] зрозуміли, що спрощену модель реального нейрона можна представити за допомогою простого додавання та застосування деякого порогового значення, як показано на рис.3.5.

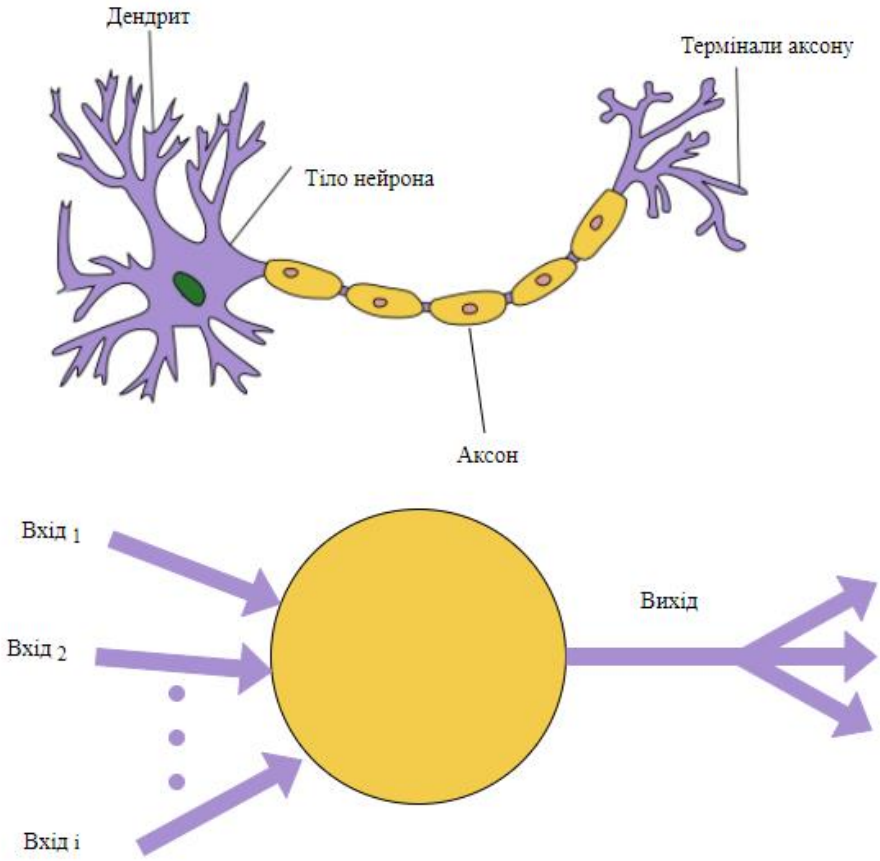


Рис. 3.5 Людський та штучний нейрон

Розробку даної ідеї продовжив Розенблатт. Розенблатт розробив штучний нейрон, щоб надати йому здатність до навчання. Він працював над створенням першого фізичного пристрою, який фактично використовував ці принципи, Mark I Perceptron, або “перцептрон Розенблатта”. Розенблатт у своїй роботі [133] писав про свій винахід: “Ми зараз станемо свідками народження такої машини – машини, здатної сприймати, розпізнавати й ідентифікувати навколишнє середовище без будь-якого навчання чи контролю людини”. Перцептрон був побудований і зміг успішно розпізнавати прості форми.

Професор Массачусетського технологічного інституту Марвін Мінський разом із Сеймуром Пепертом ретельно дослідили винахід

Розенблатта у своїй роботі [134]. Вони показали, що один рівень, побудований з цих пристроїв не в змозі вивчити деякі прості, але важливі математичні функції (наприклад, XOR). У тій же роботі вони також показали, що використання кількох шарів пристроїв дозволить усунути ці обмеження. На жаль, лише перша з цих ідей отримала широке поширення. В результаті світова академічна спільнота майже повністю відмовилася від нейронних мереж протягом наступних двох десятиліть.

Сила людського розуму залежить від кількості основних компонентів, різноманітності зв'язків між ними, а також генетичного програмування та навчання.

Експериментальні дослідження показують, що біологічні нейрони структурно складніші, ніж спрощене їх пояснення, і набагато складніше, ніж існуючі штучні нейрони, які є елементами сучасних штучних нейронних мереж. Оскільки нейрофізіологія дає вченим глибше розуміння роботи нейронів, а обчислювальні технології постійно розвиваються, розробники мереж мають необмежений простір поліпшення біологічних моделей мозку.

У 1980-х роках більшість моделей вже були побудовані з другим шаром нейронів, таким чином уникаючи проблеми, яку виявили Мінській та Пеперт. І справді, нейронні мережі широко використовувалися протягом 80-х і 90-х років для реальних практичних проектів. Однак знову неповне розуміння певних теоретичних питань стримувало розвиток нейронних мереж. Теоретично додавання лише одного додаткового шару нейронів було достатньо для того, щоб будь-яка математична функція була апроксимована цими нейронними мережами, але на практиці такі мережі часто були занадто великими та занадто повільними, щоб бути корисними. Хоча у 1990-х роках дослідники показали, що для досягнення практичних хороших результатів потрібно використовувати ще більше шарів нейронів, лише в 2010-х роках цей принцип став ширше оцінюватися та застосовуватися. Нейронні мережі тепер нарешті реалізують свій потенціал завдяки використанню більшої

кількості шарів у поєднанні з можливостями для цього завдяки вдосконаленню комп'ютерного обладнання, збільшення доступності даних та вдосконаленим алгоритмічним налаштуванням, що дозволяють швидше навчати нейронні мережі [135].

Деякі функціональні, структурні та апроксимаційні особливості нейронних мереж були розглянуті у розділі з тестуванням процесів на нелінійність. Опишемо загальну структуру та особливості нейронних мереж.

Нехай ми маємо задачу по відтворенню довільної функціональної залежності за певним набором вхідних даних. Набір вхідних даних має вигляд пар з вектору вхідних даних $\overrightarrow{x_i} = [x_1, x_2, \ldots, x_n]$, та відповідного значення вихідної змінної y_i. Тобто, нейронна мережа повинна бути здатна апроксимувати невідому функціональну залежність $y = f(\vec{x})$ і власне потім виконувати відповідне функціональне перетворення для прогнозування. Приклад довільної функціональної залежності, яку повинна вміти апроксимувати нейронна мережа у випадку однієї вхідної змінної зображено на рис. 3.6.

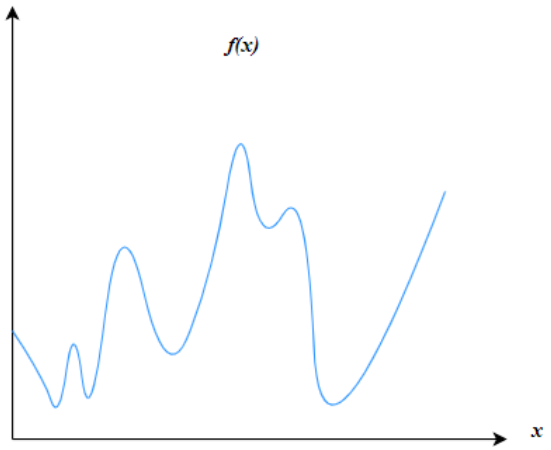


Рис. 3.6 Приклад довільної функціональної залежності

Розглянемо штучний нейрон. Базовий модуль нейронних мереж штучний нейрон моделює чотири основні функції природного нейрона (рис. 3.7).

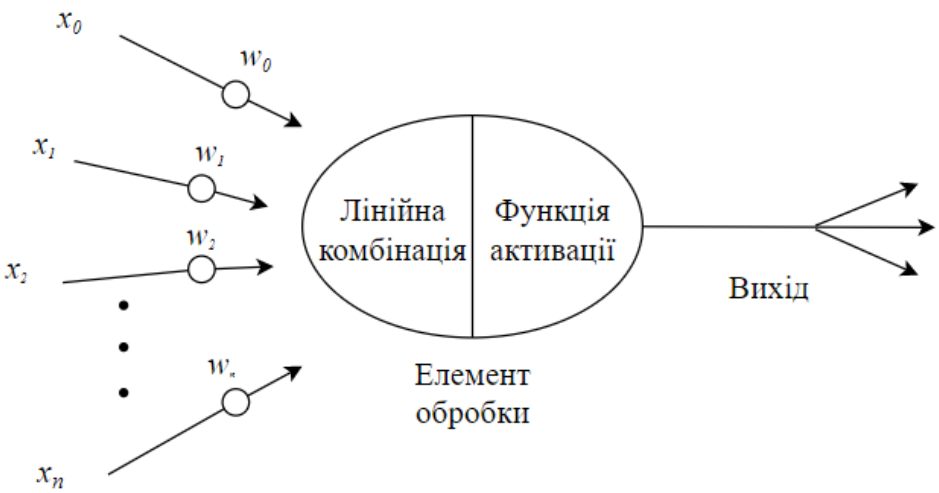


Рис. 3.7 Базовий штучний нейрон

Вхідні сигнали x_n зважені відповідними ваговими коефіцієнтами з'єднання w_n додаються, проходять через передатну або активаційну функцію, обраховують результат і виводяться в якості прогнозу.

Тобто, такий нейрон може моделювати наступне функціональне перетворення:

$$f(x, w) = \varphi(\sum_{i=1}^{n} x_i w_i), \tag{3.42}$$

де w - множина параметрів або “ваг” нейронної мережі, $\varphi: R \rightarrow R$, наприклад, $\varphi(x) = \frac{1}{1+e^{-x}}$, доволі популярна, так звана, сигмоїдна функція активації (рис3.8).

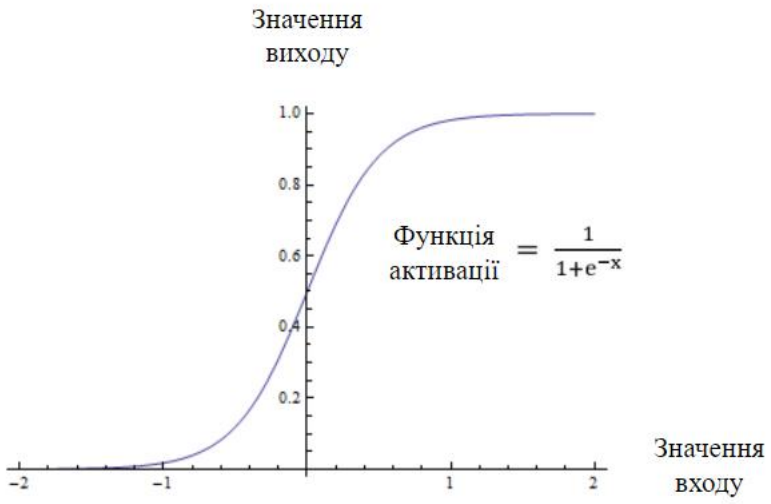


Рис. 3.8 Сигмоїдна функція активаціії

В нейромережах в якості функцій активації можуть бути використані сигмоїда, синус, гіперболічний тангенс та ін. Необхідно, щоб функція активації була нелінійною та обмеженою.

Однак, сучасні дослідження [136] з використанням багатошарових нейронних мереж показали проблематику класичних функцій активації, як то сигмоїда чи гіперболічний тангенс, що полягає в тому числі і в проблемі сповільнення процесу навчання глибоких мереж і запропонували використовувати для активації функцію ReLU (рис. 3.9):

$$\mathrm{ReLU}(x) = \begin{cases} 0, x \leq 0 \\ x, x > 0 \end{cases}, \text{або } \mathrm{ReLU}(x) = \max(0, x) \tag{3.43}$$

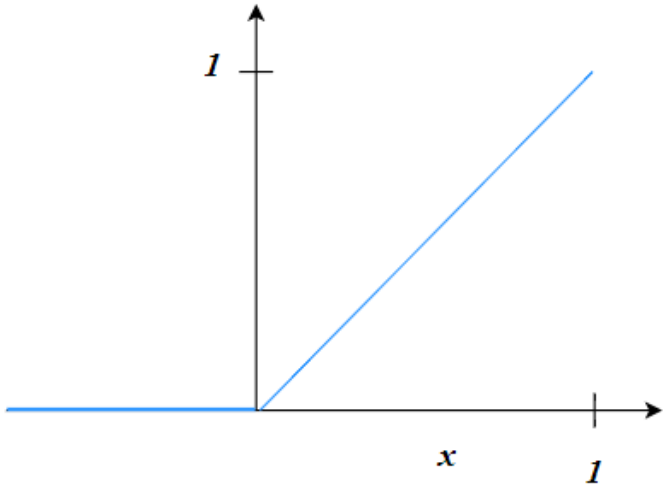


Рис 3.9 Функція активації ReLU

У зазначених роботах, а також у [135] детально розглянута мотивація переходу до такої функції активації а також всі переваги та особливості такого підходу.

Щоб мати можливість комбінувати підходи до обробки вхідних даних з різними вагами, створюють декілька нейронів і об’єднують їх в один шар, виходи з якого потім знову комбінуються у вихід аналогічним чином (рис. 3.10).

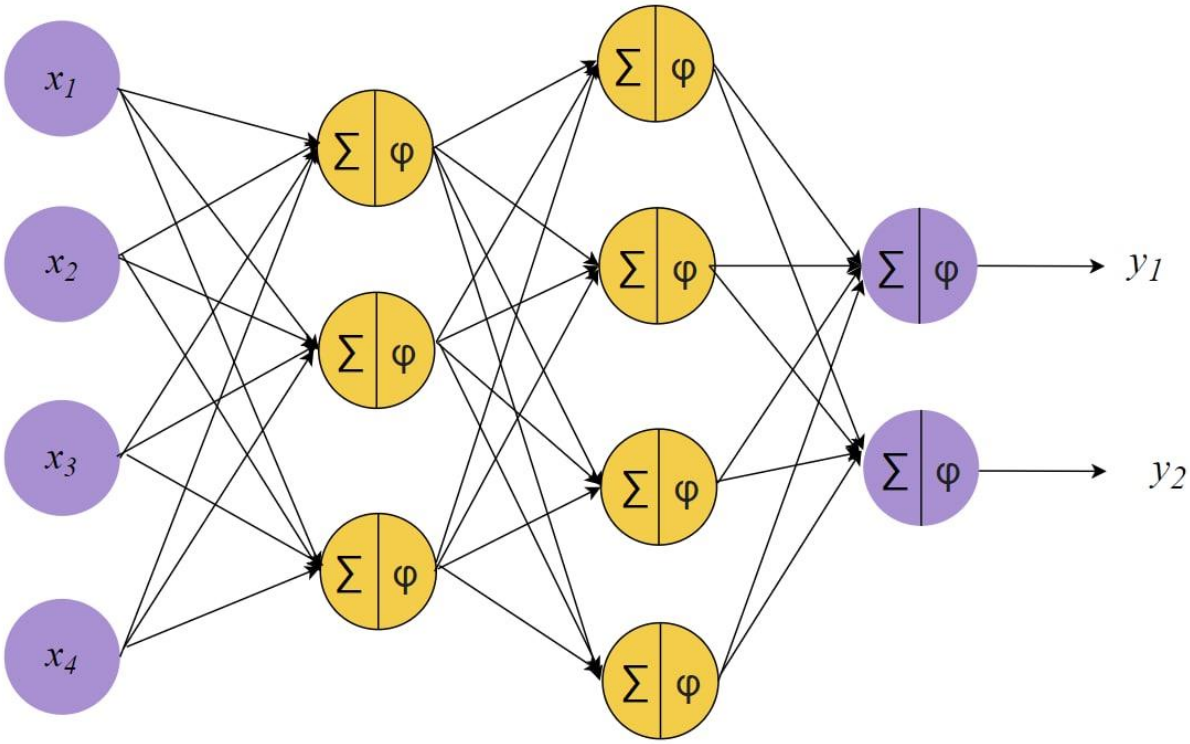


Рис 3.10 Комбінація декількох нейронів

В загальному вигляді, для базової одношарової нейронної мережі можемо записати:

$$f(x, w) = \beta_0 + \sum_{j=1}^{q} \beta_j \varphi(\sum_{i=1}^{n} x_i \alpha_{ij}), n, q \in N, \quad (3.44)$$

де w - множина параметрів або “ваг” нейронної мережі, $w = \{\beta_0, \ldots, \beta_q, \alpha_{11}, \ldots, \alpha_{nq}\}$, φ - функція активації.

Як вже було зазначено, сучасні нейронні мережі для моделювання більш складних та глибоких зв’язків між змінними можуть використовувати багато прихованих шарів. Формальний запис для багатошарової нейронної мережі стає дуже складним, тому зазвичай використовують графічний схематичний опис (рис 3.10), маючи на увазі використання базових операцій з вагами та функціями активації аналогічно до одношарової мережі.

Виявляється, що такою процедурою багаторазового комбінування лінійних операцій (множення і сумування ваг на вхідні дані) з проміжним застосуванням нелінійностей (активації) можна отримати настільки гнучку функцію, що її теоретично можна використовувати для вирішення будь-якої проблеми, просто змінюючи її ваги. Тобто, якщо розглядати нейронну мережу як математичну функцію, то вона виявляється функцією, яка є надзвичайно гнучкою залежно від її ваг. Математичний доказ, який називається теоремою універсальної апроксимації, показує, що ця функція теоретично може вирішити будь-яку задачу з будь-яким рівнем точності. Тобто теорема про універсальну апроксимацію є теоремою існування, тобто математичним доказом можливості апроксимації будь-якої неперервної функції за допомогою нейронної мережі [137].

Базуючись на дослідженнях Колмогорова [60], у роботі [61] було доведено, що дійсно, довільна функціональна залежність f(x) із заданим рівнем точності ε може бути апроксимована деякою функцією F(x), тобто:

$$|F(\vec{x}) - f(\vec{x})| < \varepsilon, \text{де}$$
$$F(\vec{x}) = \sum_{i=1}^{m} \alpha_i \, \varphi(\sum_{j=1}^{n} w_{ij} + b_i), \qquad (3.45)$$

де функція φ деяка обмежена, непостійна, монотонно зростаюча неперервна функція. Очевидно, що структура функції F підходить до описаної вище загальної структури побудови нейронної мережі.

Інтуїтивне пояснення цього видатного результату полягає в наступному. Для одного нейрону, розглянемо його вихід σ(wx+b), де $\sigma(z) = \frac{1}{1+e^{-z}}$. Для параметрів нейрону, наприклад w=8, b=-4, будемо мати наступний вихід (Рис.3.11):

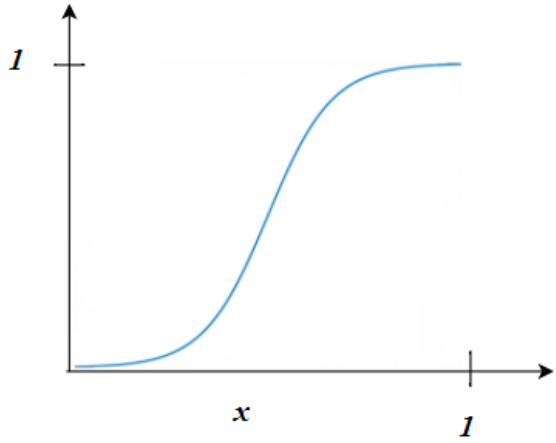


Рис.3.11 Вихід нейрону з параметрами w=8, b=-4

Однак, налаштовуючи значення параметрів, наприклад до w=999, b=-400 ми зможемо отримати наступний вихід (рис.3.12):

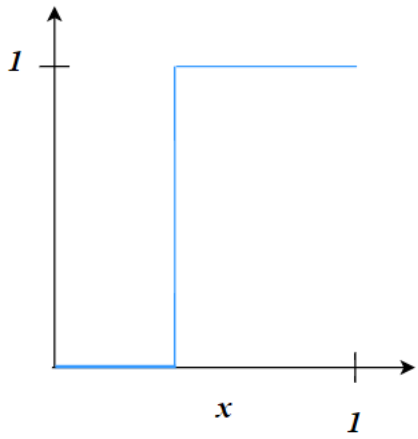


Рис.3.12 Вихід нейрону з параметрами w=999, b=-400

Використовуючи вже декілька нейронів, скомбінувавши їх виходи можемо отримати вихід наступного вигляду (рис. 3.13):

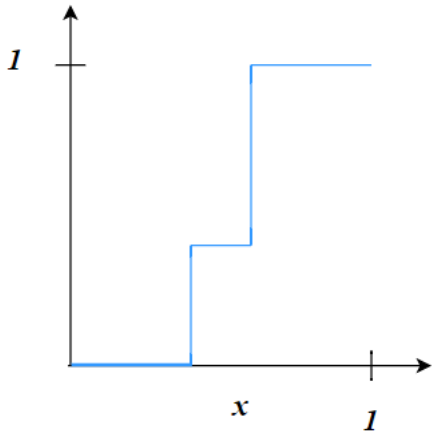


Рис. 3.13 Комбінація декількох нейронів

Налаштовуючи значення параметрів, можна регулювати відповідно висоту та ширину “сходинок”.

Тепер для деякої довільної функції, спробуємо апроксимувати її за допомогою нейронної мережі з 5-ма нейронами (рис. 3.14):

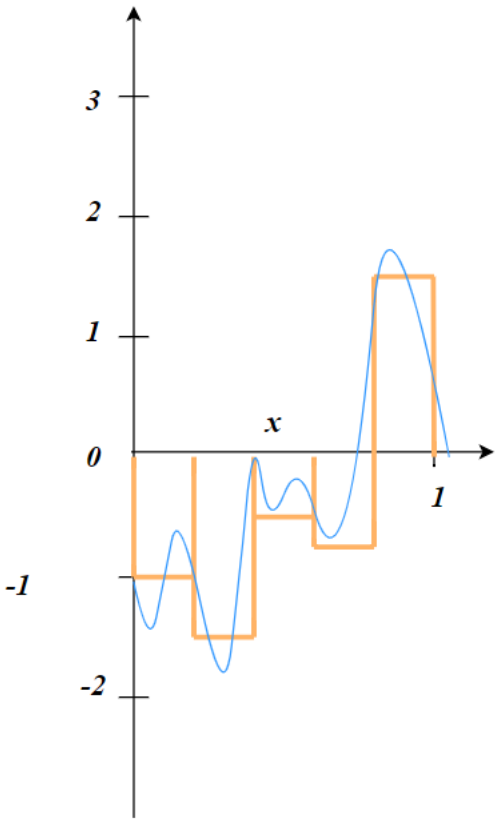


Рис. 3.14 Апроксимація функції нейронною мережею

Як бачимо, збільшуючи кількість нейронів-”сходинок” можна точніше апроксимувати шукану функціональну залежність. Цікаво, що даний процес, в якійсь мірі схожий на схему побудови інтегралу з математичного аналізу. Кількість нейронів в мережі обирається з метою мінімізації похибки прогнозування, але важливо, що похибка мінімізується не на навчальній а на тестовій вибірці. Детальніше цей процес буде розглянутий далі.

Отже, нейронна мережа є дуже параметризованою функцією, що в загальному вигляді може описувати довільну функціональну залежність при правильному підборі параметрів. Для того, щоб знайти “правильні” параметри (ваги) нейронної мережі виконується так зване “навчання” нейронної мережі, в процесі якого користуються алгоритмами оптимізації.

3.6. Алгоритми оптимізації параметрів для навчання нейронної мережі

Розглянемо, як виглядає процедура навчання нейронної мережі.

1. Спочатку виконується ініціалізація параметрів (ваг) нейронної мережі. Загалом це відбувається завдяки простої випадкової ініціалізації, однак сучасні дослідження показали, що навіть при випадковій ініціалізації важливо дотримуватись декількох властивостей для параметрів. Одним з таких модифікованих способів була запропонована ініціалізація Ксав'є [138], що дозволила суттєво прискорити процес навчання та покращити його якість. Ця ініціалізація показує себе найкращим чином при використанні функцій активації по типу гіперболічного тангенсу чи сигмоїди. Однак в цій же роботі були розглянуті недоліки цих функцій активації для навчання глибоких нейронних мереж, що сприяло появі функції ReLU, що вимагала нового підходу для ініціалізації ваг. Тому, в 2015 році у [139] були розглянуті і інші переваги і застосування функції ReLU, а також була запропонована ініціалізація Хе, що являється найбільш підходящою для цієї функції активації схемою ініціалізації параметрів.

2. Для кожного запису в даних, виконується так званий прямий прохід (forward propagation) нейронною мережею - обрахування прогнозів за вибраною архітектурою і ініціалізованими параметрами. Як було зазначено раніше, цей етап можна спрощено описати як комбінацію з лінійних перетворень, що можна описати матричним добутком матриці ваг на вхідні дані з подальшим застосуванням нелінійного перетворення у вигляді функції активації. Формально, для j-го виходу для n-го прошарку це можна записати як

$$y_j^{(n)} = \varphi(\sum_{i=1}^{k} y_i^{(n-1)} w_{ij}^{(n)}), \tag{3.46}$$

Де k-кількість нейронів в n-1 прошарку, $y_i^{(n-1)}$- вихід попереднього прошарку ($y_i^{(0)} = x_i$), $w_{ij}^{(n)}$ - значення відповідних ваг (параметрів), φ - нелінійна функція активації.

Графічно це перетворення виглядає наступним чином (рис.3.15):

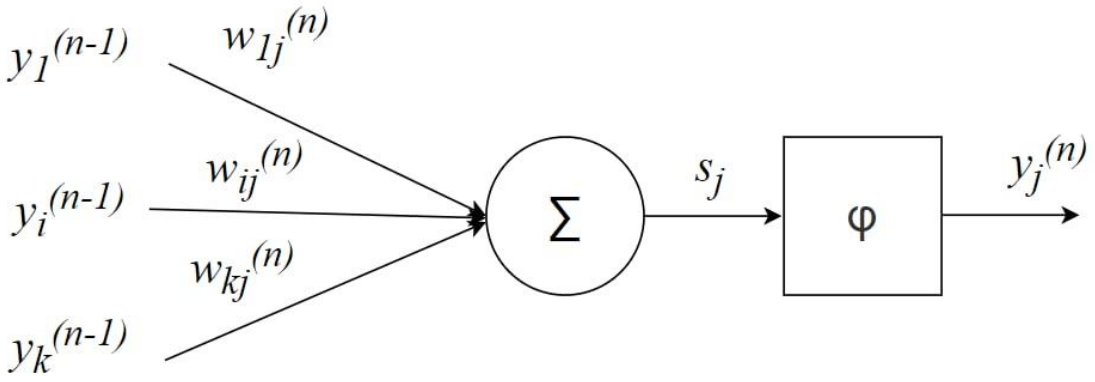


Рис.3.15 Перетворення forward propagation

3. За допомогою обраного критерію якості (який ще для процесу навчання нейронної мережі називають функцією втрат), обраховується його значення. Проблематика вибору критерію якості буде розглянута далі, на даному етапі для опису алгоритму, буде вибраний критерій MSE:

$$\mathrm{MSE}(w) = L(w) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i(w))^2, \tag{3.47}$$

де y_i - реальні значення вихідної змінної, $\hat{y}_i(w)$ - прогнозовані нейронною мережею з вагами w значення вихідної змінної, N - обсяг вибірки.

4. Для функції втрат розраховуються також градієнти відповідно кожного параметру. Для одношарової мережі можна напряму обчислити значення градієнту, для багатошарових мереж для обрахування градієнтів

глибоких шарів використовується метод зворотнього розповсюдження помилки (back propagation) [137], що базується на застосуванні правила взяття похідної від композиції функцій, що наприклад, для розглянутого випадку для j-го виходу для n-го прошарку буде виглядати наступним чином:

$$\frac{\partial L}{\partial w_{ij}^{(n)}} = \frac{\partial L}{\partial y_j^{(n)}} \frac{\partial y_j^{(n)}}{\partial w_{ij}^{(n)}} \tag{3.48}$$

Цю процедуру обчислення градієнтів можна виконувати поступово, починаючи з останнього прошарку і поступово отримавши шукані градієнти для всіх параметрів. Рекурсивна процедура пошуку детально проаналізована у [126]. Для обчислення похідних використовуються різні способи наближення за чисельними методами [135].

5. Виконується крок вибраного алгоритму оптимізації, що оновлює значення параметрів відповідно до знайдених градієнтів.

6. Процедура повторюється знову, починаючи з кроку (2), поки не буде досягнутий критерій зупинки алгоритму. Зазвичай цим критерієм являється фіксована кількість ітерацій алгоритму, проте може бути також досягнення певного порогу якості (тобто коли $L(w) < \varepsilon$).

Найпростішим способом оновлення значення параметрів є алгоритм градієнтного спуску [126]. Його ідея полягає в застосуванні властивості градієнту показувати напрям зростання функції. Відповідно, тоді параметри треба змінювати в протилежному напрямі з деяким кроком η (крок буде вказувати на те, наскільки сильно оновлюються параметри). Тому, формулу для оновлення параметрів W при застосуванні алгоритму градієнтного спуску можна записати наступним чином:

$$\mathrm{W}(\mathrm{n}+1) = \mathrm{W}(\mathrm{n}) - \eta \nabla_{\mathrm{w}} \mathrm{L}(\mathrm{W}(\mathrm{n})) \tag{3.49}$$

Оскільки нам потрібно розрахувати градієнти для всього набору даних, щоб виконати лише одне оновлення, градієнтний спуск може бути дуже повільним і навіть неможливим до застосування для наборів даних, які не поміщаються в пам’ять обчислювальної машини. Градієнтний спуск також не дозволяє оновлювати модель адаптивно, в реальному часі.

Стохастичний градієнтний спуск (Stochastic gradient descent - SGD) виконує оновлення параметрів для кожного запису в наборі даних окремо:

$$W(n+1) = W(n) - \eta \nabla_w L(W(n); x^{(i)}; y^{(i)}) \tag{3.50}$$

Градієнтний спуск виконує надлишкові обчислення для великих наборів даних, оскільки він повторно обчислює градієнти для подібних прикладів перед кожним оновленням параметрів. SGD усуває цю надмірність, виконуючи одне оновлення за раз. Тому оновлення параметрів зазвичай відбувається набагато швидше, але навчання відбувається набагато довше, SGD вже можна використовувати для адаптивного навчання в реальному часі. Однак SGD виконує часті оновлення з високою дисперсією, що спричиняє значні коливання функції втрат [140]. У той час як градієнтний спуск збігається до зони мінімуму, в якій розміщуються параметри, коливання SGD, з одного боку, дають змогу йому перейти до нових і потенційно кращих локальних мінімумів. З іншого боку, це в кінцевому підсумку ускладнює зближення до точного мінімуму, оскільки SGD буде продовжувати “перестрибувати” шукані значення. Проте було показано [141], що коли ми повільно зменшуємо крок навчання η, SGD показує таку ж поведінку конвергенції, що й звичайний градієнтний спуск, майже напевно зближаючись до локального або глобального мінімуму для неопуклої та опуклої оптимізації відповідно.

Тому у сучасних системах, використовують переваги обох підходів використовуючи стохастичний градієнтний спуск по міні-батчам, виконуючи

оновлення для кожного міні-батчу з n записів з навчальної вибірки.

$$W(n+1) = W(n) - \eta \nabla_w L(W(n); x^{(i:i+n)}; y^{(i:i+n)}) \quad (3.51)$$

Таким чином, цей підхід дозволяє балансувати між швидкістю навчання нейронної мережі, а також наявною пам'яттю за допомогою вибору розміру батчу (поширені розміри міні-батчу коливаються від 50 до 256, але можуть відрізнятися для різних застосувань). Також такий варіант SGD дозволяє навчати нейронні мережі зменшуючи дисперсію оновлення параметрів, що може призвести до більш стабільної збіжності. Тому в сучасних застосуваннях градієнтний спуск за міні-батчами, як правило, є базовим алгоритмом вибору під час навчання нейронної мережі, і термін SGD зазвичай використовується також, коли використовуються міні-батчі.

Метод моментів [142] дозволяє прискорити градієнтний спуск у потрібному напрямку і ще зменшити коливання функції втрат. Ідея методу полягає в застосуванні аналогії з фізичними явищами матеріального світу. Матеріальні точки мають інерцію руху, їх напрям руху не можна змінити миттєво, і їх рух по схилу донизу буде описуватись не лише швидкістю, а й прискоренням. Метод моментів полягає в тому, щоб не просто рухатись строго протилежно напрямку градієнта, а й ще продовжувати рух в тому напрямі, в якому рухалися раніше. Формулу для оновлення параметрів за методом моментів можна представити наступним чином:

$$W(n+1) = W(n) - \vartheta_n,$$
$$\vartheta_n = \gamma \vartheta_{n-1} + \eta \nabla_w L(W(n)), \quad (3.52)$$

де γ - параметр методу моментів, він завжди менше за 1 (зазвичай визначається як 0.9), і саме він визначає, яку частину попереднього градієнту ми хочемо взяти на поточному кроці, а на яку частину будемо використовувати

новий градієнт.

Існує також модифікація Нестерова для методу моментів [143], [144], що полягає у обчисленні градієнта функції в іншій точці з також врахуванням моменту:

$$\mathrm{W(n+1) = W(n) - \vartheta_n}$$
$$\vartheta_n = \gamma\vartheta_{n-1} + \eta\nabla_{\mathrm{w}}\mathrm{L}(\mathrm{W(n)} - \gamma\vartheta_{n-1}) \qquad (3.53)$$

Метод моментів спочатку обчислює поточний градієнт, а потім робить великий стрибок у напрямку оновленого накопиченого градієнта; метод Нестерова спочатку робить великий стрибок у напрямку попереднього накопиченого градієнта, вимірює градієнт, а потім робить корекцію, що призводить до повного оновлення параметрів. Це випереджувальне оновлення не дозволяє нам рухатися занадто швидко і призводить до підвищення швидкості реагування, що, наприклад, значно підвищило продуктивність рекурентних нейронних мереж для ряду завдань [145], що будуть розглядатися в даній роботі.

В розглянутих вище методах, однакова швидкість навчання застосовується до всіх оновлень параметрів. Якщо представлені дані розріджені, тобто деякі приклади зустрічаються доволі часто, а деякі - рідко, тоді, можливо, не є ефективним оновлювати їх відповідні параметри в однаковій мірі, а краще виконувати більші оновлення для прикладів, що рідко зустрічаються. Наприклад, цілком може так скластися, що деякі ваги вже близькі до своїх локальних мінімумів, і за цими координатами потрібно рухатися повільно і обережно, а інші ваги ще на середині відповідного схилу, і їх можна змінювати набагато швидше. До того ж, саме собою регулювання швидкості навчання може виявитися досить витратною процедурою. Для того щоб мати можливість адаптувати швидкість навчання для різних параметрів автоматично, були створені адаптивні способи оптимізації. Незважаючи на те,

що у них все одно є свої власні метапараметри, ці методи, як правило, поводяться краще на розріджених даних і більш стабільні, ніж зміна єдиної швидкості навчання у чистому вигляді.

Перший з адаптивних методів оптимізації - Adagrad [146] базується на наступній ідеї: ступінь зміни повинна бути меншою у тих параметрів, які відповідають за опис ознак для прикладів в вхідних даних, які в більшій ступені присутні в вхідних даних, і, відповідно, ступінь зміни буде більшою в тих параметрів, які відповідають за ознаки для опису прикладів, які рідко зустрічаються у вхідних даних. Параметри з найбільшою частковою похідною від втрати мають відповідно велике зниження їх швидкості навчання, при цьому параметри з малими частковими похідними мають відносно невелике зниження швидкості навчання. Формулу для оновлення параметрів за цим методом можна записати наступним чином:

$$\mathrm{W}(\mathrm{n}+1;\mathrm{i}) = \mathrm{W}(\mathrm{n};\mathrm{i}) - \frac{\eta}{\sqrt{\mathrm{G}_{\mathrm{n};\mathrm{ii}}+\varepsilon}} * \mathrm{g}_{\mathrm{n},\mathrm{i}}, \tag{3.54}$$

де i - номер відповідного параметру, оскільки тепер, для кожного параметру своя система оновлення. $\mathrm{g}_{\mathrm{n},\mathrm{i}} = \nabla_{\mathrm{w}}\mathrm{L}(\mathrm{W}(\mathrm{n};\mathrm{i}))$ - аналогічно до SGD, але тепер для кожного параметру відповідне значення градієнту. Новим елементом у формулі оновлення є корінь у знаменнику. ε - параметр для забезпечення стабільності і відсутності ділення на нуль. $G_n \epsilon R^{dxd}$ - діагональна матриця, кожний елемент i,i якої є сумою квадратів градієнтів відносно параметру i до кроку n.

Власне цей механізм коригування значення оновлення, що полягає у накопиченні квадратів градієнтів у знаменнику є водночас і основною слабкістю Adagrad. Оскільки кожен доданок є додатнім, накопичена сума продовжує зростати під час навчання. Це, у свою чергу, призводить до зменшення швидкості навчання і в кінцевому підсумку стає нескінченно

малою, після чого алгоритм більше не може отримувати додаткові знання. Наступні алгоритми спрямовані на усунення цього недоліку.

Алгоритм RMSProp [147] модифікує AdaGrad для кращої роботи шляхом зміни схеми накопичення градієнта з простої суми на експоненційне зважене ковзне середнє. RMSProp використовує експоненційне згладжування, щоб відкинути (зробити затухаючою) історію з далекого минулого і краще використовувати більш нові значення градієнтів. Формула оновлення параметрів для алгоритму RMSProp буде виглядати наступним чином:

$$W(n+1;i) = W(n;i) - \frac{\eta}{\sqrt{E[g^2]_{n;i} + \varepsilon}} * g_{n,i},$$

$$E[g^2]_n = \lambda E[g^2]_{n-1} + (1-\lambda) g_n^2 \tag{3.55}$$

Параметр згладжування λ зазвичай обирають рівним 0.9.

Adam (adaptive momentum estimation) [148] у контексті попередніх алгоритмів - це варіант комбінації RMSProp і методу моментів з метою об’єднати переваги обох підходів, з кількома відмінностями. По-перше, в Adam включено момент з застосуванням експоненційного зважування. По-друге, Adam включає поправки зміщення до оцінок обох моментів першого порядку (середнє - момент) і моменти другого порядку (нецентрована дисперсія) для врахування їх коректної ініціалізації. Формула оновлення параметрів за алгоритмом Adam буде виглядати наступним чином:

$$W(n+1) = W(n) - \frac{\eta}{\sqrt{\hat{v}_n} + \varepsilon} * \hat{m}_n, \tag{3.56}$$

де $\hat{m}_n$ і $\hat{v}_n$ відповідні виправлені оцінки щодо їх зміщення для першого і другого моменту:

$$\hat{m}_n = \frac{m_n}{1-\beta_1^n}$$
$$\hat{v}_n = \frac{v_n}{1-\beta_2^n}, \tag{3.57}$$

де $\beta_1^0 = 0.9$, $\beta_2^0 = 0.999$ - параметри для корекції зміщеності оцінок.

В роботі [148] показано, що при нульовій початковій ініціалізації відповідних параметрів, вони будуть зміщені до нуля, особливо під час початкових кроків алгоритму навчання, і коли швидкості оновлення невеликі. Власне самі моменти будуть виглядати наступним чином:

$$m_n = \beta_1 m_{n-1} + (1-\beta_1) g_n$$
$$v_n = \beta_2 v_{n-1} + (1-\beta_2) g_n^2 \tag{3.58}$$

Сучасні практичні дослідження [135], [149] показують, що для більшості задач, алгоритм Adam сходиться туди, куди потрібно, і також є найшвидшим. Adam є прямим розширенням і гарним поєднанням переваг алгоритмів SGD з моментом і RMSProp, і мабуть в більшості випадків буде найкращим вибором на даний момент. В даній роботі в якості алгоритму оптимізації параметрів мережі буде використано саме Adam.

3.7. Рекурентні нейронні мережі

З стрімким поширенням в дослідницьких і практичних задачах апарату моделювання за допомогою нейронних мереж, які стали дуже привабливими через свою універсальність і доступними у зв’язку з розвитком обчислювальної техніки, було розроблено численну кількість різних архітектур нейронних мереж для вирішення різних типів задач. Підходи для оптимального підбору параметрів, описані вище, залишаються незмінними,

але існують різні способи представити шукану функціональну залежність у різних формах, що називаються архітектурами. З теоретичної точки зору, вимоги теореми про універсальну апроксимацію залишатимуться незмінними, проте більш складні типи даних, як-то тексти, зображення, аудіо, стає набагато простіше обробляти для подальших обчислень. Досліджувані в даній роботі процеси зазвичай представляються статистичними даними у вигляді послідовностей (часових рядів). Для обробки даних у вигляді послідовностей були розроблені рекурентні нейронні мережі [150]. Історично рекурентні нейронні мережі почали розробляти для більш ефективної обробки текстової інформації, тому для опису деяких підходів для більш наочного пояснення будемо користуватись саме текстовими прикладами.

Щоб перейти від багатошарового перцептрону до рекурентних мереж, потрібно скористатися перевагами однієї з ранніх ідей, знайдених у машинному навчанні та статистичних моделях: спільне використання параметрів у різних частинах моделі. Спільний доступ до параметрів дає можливість розширити та застосувати модель до прикладів різних форм (різні довжини) та узагальнити їх. Якби ми мали окремі параметри для кожного значення індексу часу ми не могли узагальнити довжину послідовності, що спостерігається під час тренування, а також не мали б статистичної сили на різних довжинах послідовностей і в різних позиціях у часі. Такий обмін особливо важливий, коли конкретна частина інформації може зустрічатися в кількох позиціях послідовності. Наприклад, розглянемо два речення “Я поїхав до Непалу в 2009 році” та “У 2009 році, Я поїхав до Непалу”. Якщо ми попросимо модель машинного навчання прочитати кожне речення і витягти рік, в якому оповідач їздив до Непалу, ми б хотіли, щоб модель впізнала 2009 рік як шукану інформацію, незалежно чи є вона в шостому слові або другому слові речення. Припустимо, що ми навчили класичну нейронну мережу - багатошаровий перцептрон, яка обробляє речення фіксованої довжини. Така мережа з прямим зв'язком буде мати окремі параметри для кожної вхідної

змінної, отже потрібно буде вивчити всі правила мови окремо на кожній позиції в реченні. Для порівняння, рекурентна нейронна мережа поширює однакові ваги через кілька часових кроків [126].

Розглянемо тепер, як будується така архітектура. Як було зазначено раніше, для наочності і зрозумілості, зазвичай складні архітектури нейронних мереж зображають в графічно-схематичному форматі, розуміючи під базовими операціями матричний добуток (комбінація лінійних перетворень) з подальшим застосуванням функції активації (нелінійне перетворення). Для розгляду рекурентних мереж скористаємось наступною схемою: кожна фігура буде представляти шари нейронної мережі: прямокутник для вхідного прошарку, коло для прихованого (внутрішнього) шару та трикутник для вихідного прошарку (рис 3.16).

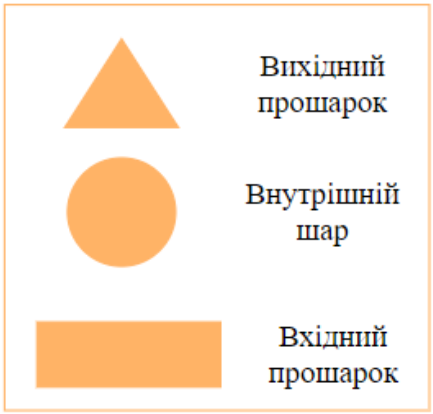


Рис 3.16 Схематичні позначення

Тоді, класичний перцептрон з одним прихованим прошарком буде зображатись як (рис. 3.17):

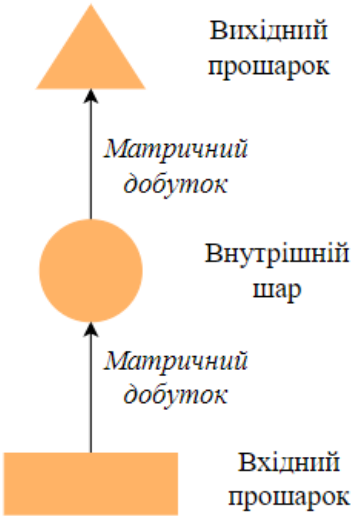


Рис. 3.17 Класичний перцептрон

Розглянемо тепер, як ми можемо використати ідею з поширенням ваг для аналізу, наприклад текстової інформації. Аналогія, що призвела до появи вибраної архітектури полягає в аналізі того, як людина читає. Для цього розглянемо, наприклад, задачу прогнозування наступного слова в реченні. На вхід будемо подавати довільну кількість слів, а на виході спробуємо спрогнозувати наступне слово. Потенційно довільна кількість вхідних слів була б першою проблемою для класичної нейронної мережі. І розглянутий вище приклад з різним порядком слів також міг би бути проблемою. Спробуємо спочатку вирішити проблему з порядком слів. Для цього припустимо, що на вхід ми поки що приймаємо лише 3 слова. Тоді багатошаровий перцептрон, що вирішує дану задачу, виглядатиме наступним чином (рис.3.18):

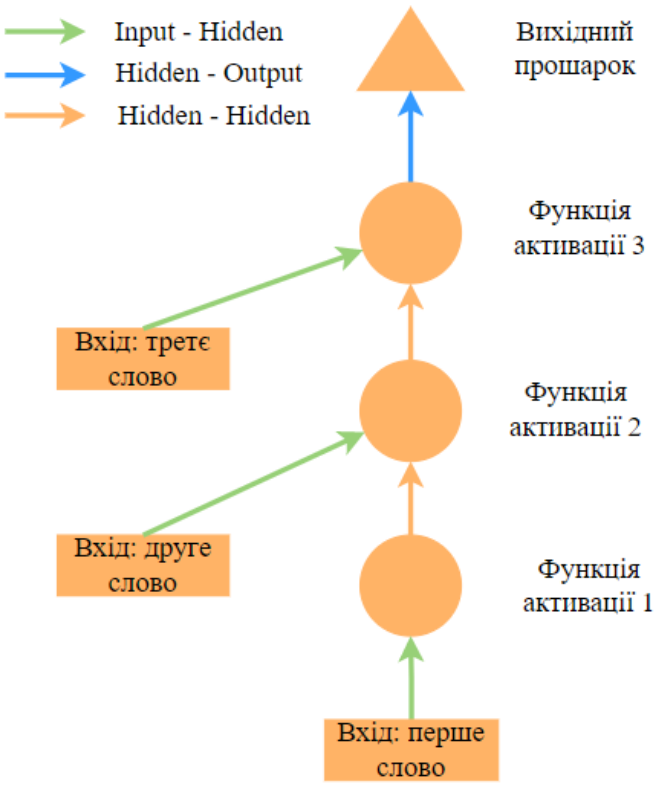


Рис.3.18 Багатошаровий перцептрон

Модель матиме 3 різні матриці: одна для обробки вхідної інформації (зелена стрілка, матриця ІН), одна для додавання вхідної інформації в існуючий контекст (жовта стрілка, матриця НН), одна для створення остаточного прогнозу (синя стрілка, матриця НО). Модель бачить перше слово, обробляє його матрицею ІН, щоб зрозуміти, що це за слово. Далі матрицею НН вона додає вже перетворене у векторний вигляд слово до поки що неіснуючого контексту (бо це перше слово). Потім модель бачить друге слово, і також оброблює його матрицею ІН, розуміючи його. За допомогою матриці НН, друге слово у векторному вигляді додається до вже існуючого контексту, формуючи краще представлення про весь текст. З теоретичної точки зору, це все ще багатошаровий перцептрон, з дещо незвичним способом

подачі і обробки даних і використанням спільних матриць ваг для різних прошарків.

Тепер можна узагальнити цей приклад, оброблюючи таким чином будь-яку кількість вхідних слів, отримуючи схему (рис. 3.19):

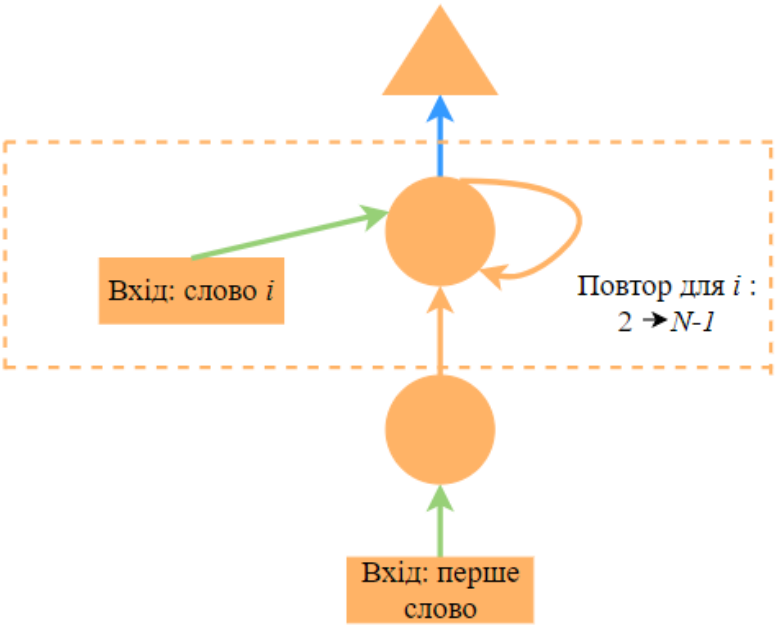


Рис. 3.19 Узагальнююча схема багатошарового перцептрона

Аналогічно цей підхід можна поширити і до даних, що представлені у вигляді часових рядів. Подаючи кожний момент часу нові сигнали ми можемо обробляти їх і додавати до контексту, що потенційно дозволить знаходити і розпізнавати нелінійні та нестаціонарні паттерни в даних.

Проте ця мережа мала свої недоліки, для покращення яких були розроблені більш розвинені нейронні мережі.

Для того, щоб не починати кожний новий батч з випадкових ваг, з’явилася ідея пам’яті. Наприклад, якщо для прогнозу наступного слова буде речення “Хмари на небі”, моделі не потрібно більше ніякого контексту -

досить очевидно, що в кінці речення мова йде про небо. У таких випадках, де проміжок між необхідною інформацією і місцем, де вона потрібна для застосування невеликий, рекурентні нейронні мережі можуть навчитися використовувати інформацію, отриману раніше (рис. 3.20).

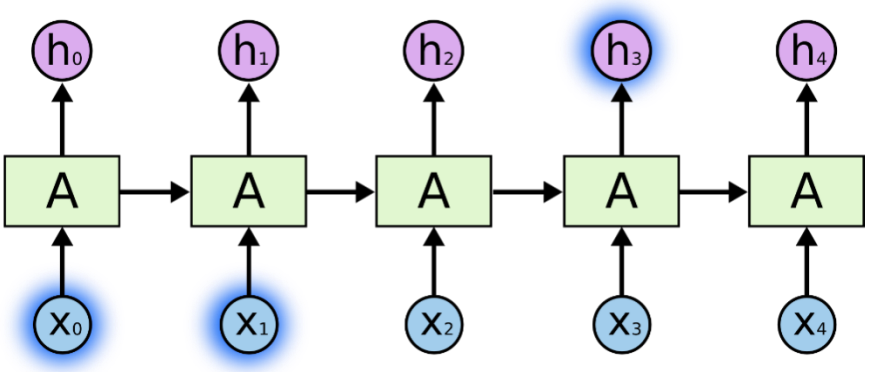


Рис. 3.20 Загальна схема рекурентної нейронної мережі

Але також бувають випадки, коли нам потрібен більш широкий контекст. Припустимо, потрібно передбачити останнє слово в тексті “Я виріс у Франції ... Я вільно розмовляю французькою”. Недавня інформація підказує, що наступне слово, ймовірно, назва мови, але якщо ми хочемо уточнити, яка саме мова, нам потрібен попередній контекст аж до інформації про Францію на початку речення. Іноді проміжок між необхідною інформацією і місцем, де вона потрібна, стає дуже великим.

На жаль, при зростанні проміжку, рекурентні мережі стають нездатні навчитися поєднувати інформацію (рис. 3.21).

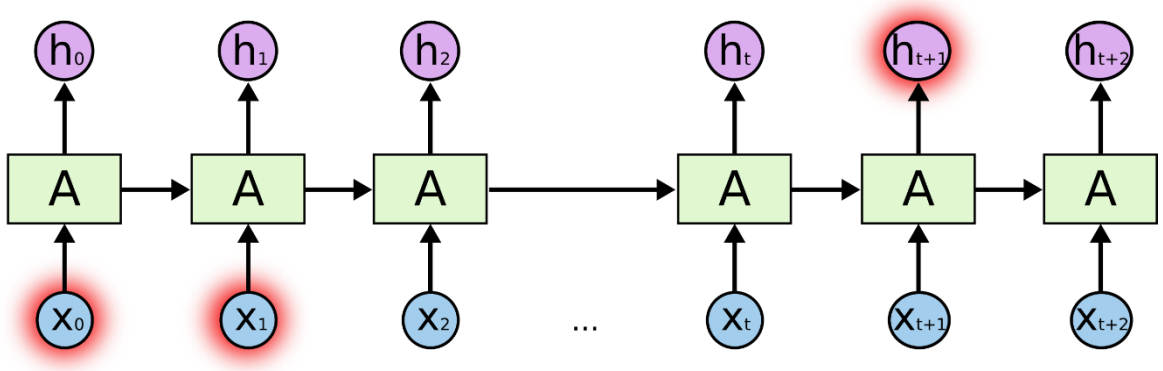


Рис.3.21 Рекурентні мережі з великими проміжками

Математична проблема вивчення довгострокових залежностей у рекурентних мережах полягає в тому, що градієнти, які поширюються протягом багатьох прошарків мають тенденцію або зникати (у більшості випадків), або вибухати (рідше, але з великою шкодою для оптимізації) [126]. Теоретично, рекурентні нейронні мережі здатні обробляти такі довготривалі залежності, можливо ретельно підібрати їх параметри, щоб вирішувати конкретні проблеми такого типу. Але, на жаль, на практиці, звичайні нейронні мережі зазвичай не здатні розв’язати таку проблему [151], [152]. Частково ця проблема була вирішення завдяки появі активації ReLU, та нової схеми ініціалізації Хе, що були розглянуті раніше, проте зрозумівши проблему пам’яті і просто змінивши певним чином архітектуру ми зможемо значно спростити процес навчання мережі.

Для того, щоб більш розумно комбінувати існуючий контекст з новою інформацією, з’явилася ідея з використанням додаткової внутрішньої нейронної мережі, що вибирає яку частину ми беремо з попереднього контексту, а яку з нового слова. Ці ідеї сприяли розробці нової архітектури нейронної мережі - нейронної мережі типу LSTM (Long Short Term Memory) [153].

Мережі з довго-короткостроковою пам'яттю (Long Short Term Memory) – зазвичай просто називають “LSTM” – особливий вид рекурентних

нейронних мереж, здатних до навчання довгостроковим залежностям. Вони дають можливість отримати високоякісні результати для різних проблем і в даний момент широко застосовуються також для моделювання нелінійних нестаціонарних процесів [154], [155].

LSTM спеціально спроектовані таким чином, щоб уникнути проблеми довгострокових залежностей. Запам'ятовувати інформацію на тривалий період часу - це практично їх типова поведінка.

Як було зазначено, рекурентні нейронні мережі мають циклічну форму повторюваних модулів нейронної мережі. Скористаємось тепер більш детальним графічним представленням, оскільки в мережі LSTM будуть застосовуватись перетворення всередині прошарку. Стандартна рекурентна нейронна мережа буде мати наступну структуру (рис. 3.22).

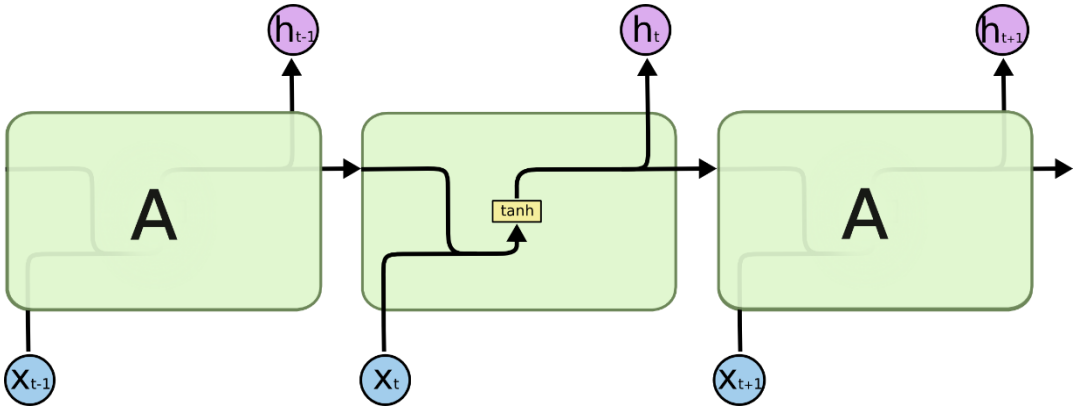


Рис.3.22 Структура повторюваного модуля РНМ

LSTM мають ланцюгову структуру, як і класичні РНМ, але повторюваний модуль має іншу побудову. Замість одного нейронного шару їх чотири, причому вони взаємодіють особливим чином (рис. 3.23)

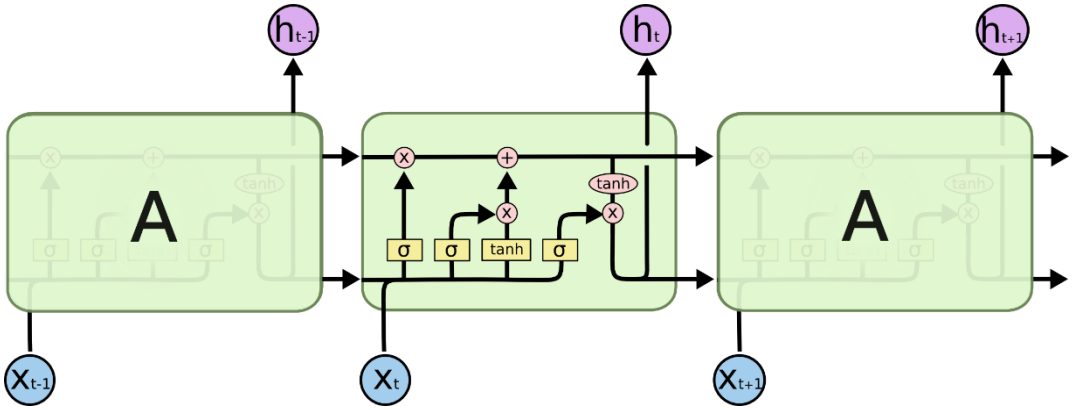


Рис. 3.23 Структура повторюваного модуля LSTM

У діаграмі вище кожна лінія передає цілий вектор від виходу одного вузла до входів інших. Рожеві кола представляють поточкові оператори, такі як додавання векторів, в той час, як жовті прямокутники - це навчені шари нейронної мережі. Лінії, що зливаються позначають конкатенацію, в той час як розгалужені лінії позначають, що їх вміст копіюється, і копії відправляються в різні місця.

Ключ до LSTM – клітинний стан – горизонтальна лінія, що проходить крізь верхню частину діаграми. Клітинний стан подібний до стрічки конвеєра. Ця стрічка рухається прямо вздовж всього ланцюга тільки з невеликими лінійними взаємодіями. Інформація передається до наступних шарів мережі по ній без змін (рис. 3.24). Це основний механізм для реалізації пам’яті.

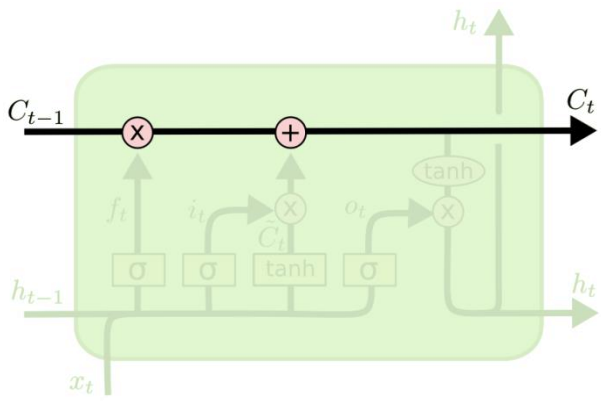


Рис. 3.24 Клітинний стан LSTM

LSTM має здатність видаляти або додавати інформацію до клітинного стану, проте ця здатність ретельно регулюється структурами, що називаються вентилями. Вони забезпечують механізм регуляції додавання нової інформації, комбінуючи її коректним чином з існуючою пам’яттю.

Вентилі - це спосіб вибірково пропускати інформацію. Вони складені з сигмоїдного шару нейромережі і операції поточкового множення (рис. 3.25).

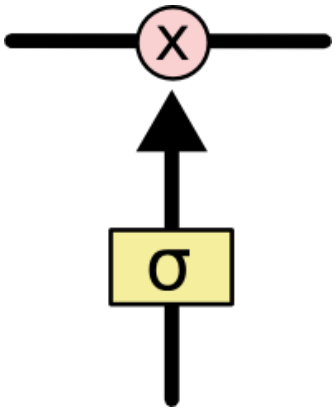


Рис.3.25 Приклад вентилю LSTM

Сигмоїдний шар подає на вихід числа між нулем і одиницею, описуючи таким чином, наскільки кожен компонент повинен бути пропущений крізь вентиль. Нуль - "нічого не пропускати", один - "пропускати все".

LSTM має три таких вентиля, щоб захищати і контролювати клітинний стан.

Першим кроком в LSTM треба вирішити яку інформацію збираємося видалити з клітинного стану. Це рішення приймається сигмоїдним шаром, що називається “забуваючим вентилем”. Забуваючий вентиль “дивиться” на h(t-1) і x(t) і подає на вихід число між 0 і 1 для кожного числа в клітинному стані C(t-1). Одиниця означає “збережи це повністю”, в той час як нуль означає “позбудься цього повністю” (рис. 3.26).

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \tag{3.59}$$

де σ – сигмоїдна функція активації

W_f – матриця ваг даного шару f в момент часу t

b_f – вектор вільних членів шару f

h_{t-1} – вихід h з попереднього кроку – вхід на цьому кроці

x_t – основний вхід нейронної мережі на кроці t.

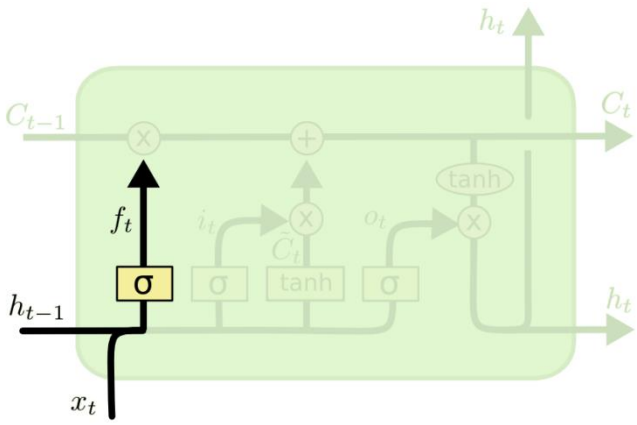


Рис. 3.26 Забуваючий вентиль LSTM

Наступним кроком треба вирішити, яку нову інформацію збираємося зберегти в клітинному стані. Цей крок складається з двох частин. По-перше, сигмоїдний шар, що називається "вхідним вентилем", вирішує, які значення буде оновлено. Далі, шар гірпеболічного тангенса створює вектор кандидатів на нові значення C(t), який може бути доданий до клітинного стану. На наступному кроці з'єднаємо ці дві частини, щоб створити оновлення для стану (рис. 3.27).

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i)$$
$$\tilde{C}_t = tanh(W_C[h_{t-1}, x_t] + b_C) \quad (3.60)$$

де tanh – функція гіперболічного тангенсу активації шару нейронної мережі

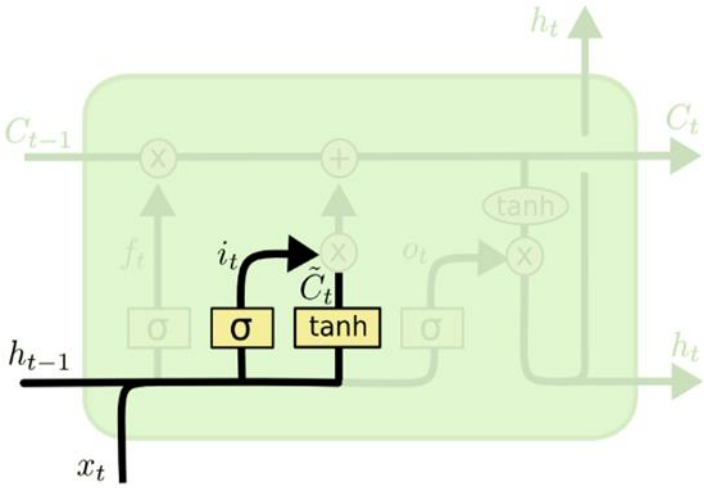


Рис. 3.27 Вхідний вентиль LSTM

Далі необхідно оновити попередній клітинний стан, C(t-1), новим клітинним станом C(t). (рис. 3.28).

$$C_t = f_t \times C_{t-1} + i_t \times \widetilde{C}_t \tag{3.61}$$

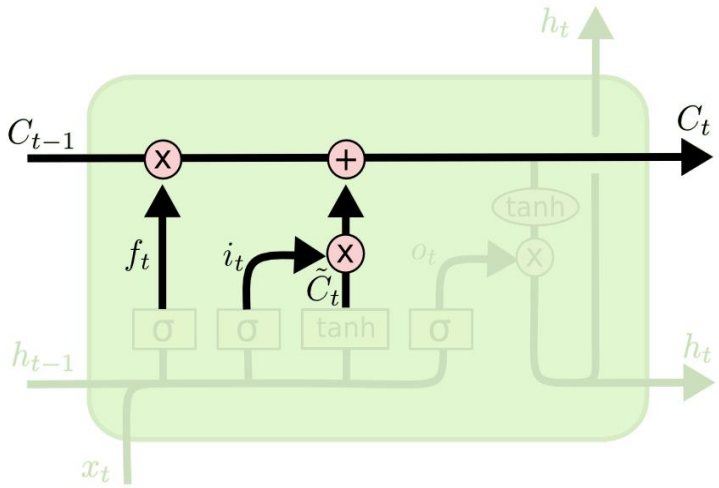


Рис. 3.28 Оновлення клітинного стану LSTM

Нарешті, потрібно вирішити, який результат збираємося подати на вихід. Цей результат заснований на клітинному стані, але буде його відфільтрованою версією. Спочатку запускаємо сигмоїдний шар, який вирішує, які частини клітинного стану ми збираємося відправити на вихід. Потім пропускаємо клітинний стан крізь гіперболічний тангенс (tanh) (щоб вмістити значення в проміжок від -1 до 1) і множимо його на вихід сигмоїдного вентиля, так що відправляємо на вихід тільки ті частини, які бажаємо (рис. 3.29).

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o),$$
$$h_t = o_t \tanh(C_t), \tag{3.62}$$

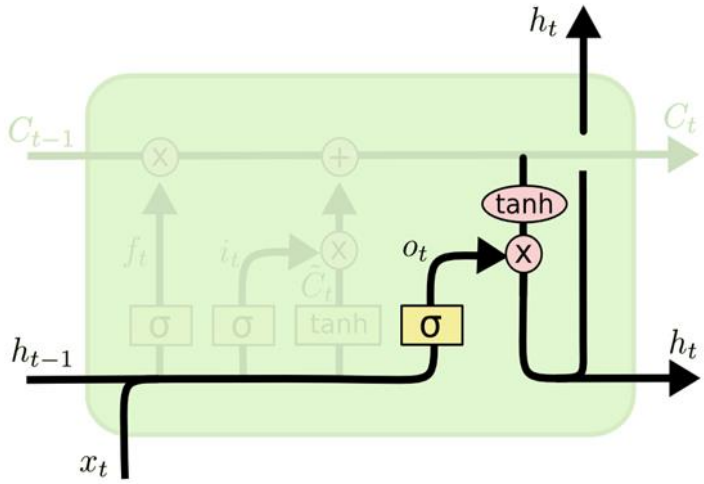
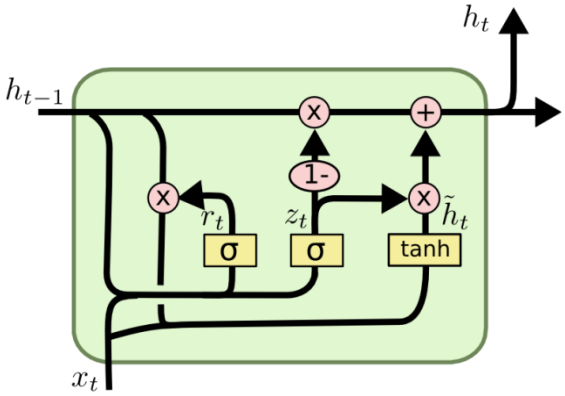


Рис 3.29 Формування виходу LSTM шару

Іншим способом вирішення поставлених проблем є модифікація LSTM - вентильна рекурентна одиниця (Gated Recurrent Unit) або GRU [156] (рис 3.30). Вона поєднує забуваючі і вхідні вентилі в один “відновлювальний вентиль” (“update gate”). Вона також зливає клітинний стан з прихованим шаром і вносить деякі інші зміни. Модель, що виходить в результаті, простіше, ніж звичайна модель LSTM і також доволі поширена на практиці [142].



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$
$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Рис.3.30 Gated Recurrent Unit

Отже, рекурентні нейронні мережі типу LSTM, вирішують задачу моделювання послідовностей, до того ж враховують нелінійний або комбінований вплив зовнішніх факторів, оскільки нейронні мережі за побудовою є універсальними апроксиматорами. Однак побудова таких мереж пов’язана з великими обчислювальними витратами. Крім того, цей підхід не може бути застосований для дуже довгих послідовностей, що є проблемою для сучасних задач з використанням великих даних [149].

За результатами цього аналізу встановлено необхідність розробки нового підходу, що дозволив би ефективно з обчислювальної точки зору моделювати великі послідовності, враховуючи нелінійний або комбінований вплив зовнішніх факторів. Для вирішення цієї проблеми пропонується розглянути згорткові нейронні мережі.

3.8. Згорткові нейронні мережі

Згорткові нейронні мережі (Convolutional Neural Networks - CNN) були введені Яном Лекуном у [157] для створення моделей комп’ютерного зору, оскільки вони здатні вловлювати найдрібніші деталі (локальні візерунки) на

зображеннях або навіть на об’ємних 3D даних. Архітектура мережі отримала свою назву через наявність операції згортки, суть якої полягає в тому, що кожен фрагмент зображення множиться на основну матрицю згортки елемент за елементом, а результат підсумовується і записується в аналогічній позиції вихідного зображення. У мережевій архітектурі закладено апріорне знання предметної області комп'ютерного зору: піксельне зображення тісно пов'язане з сусіднім (локальна кореляція) і об'єкт на зображенні можна знайти в будь-якій частині зображення.

Основна ідея CNN полягає в тому, що зображення подається безпосередньо на вхід нейронної мережі. А мережа автоматично вивчає і визначає ієрархію необхідних функцій. Таким чином, можемо отримати мережу, яка є більш точною, ніж мережа, побудована на традиційних підходах, не використовуючи складні процеси ручної генерації ознак [158]. Нейронні мережі зі згорткою вирішують одразу дві задачі. По-перше, вони вивчають нелокальні залежності – вони здатні знаходити певні зв’язки та певні шаблони не лише пов’язані з їх локальним значенням. По-друге, згорткові мережі, порівняно з рекурентними мають перевагу в значному зменшенні кількості параметрів. CNN майже не потребують попередньої обробки даних та створення додаткових ознак, що робить їх дуже ефективними.

Ця ідея була використана для розпізнавання символів і чисел у роботі [159]. Але після одноразового успішного використання нейронних мереж згорток вони не набули популярності. І лише в роботі [160] було значно відновлено інтерес до згорткових нейронних мереж після демонстрації розпізнавання зображень із вражаюче високою точністю класифікації в ImageNet Large Scale Visual Recognition Challenge. У цьому конкурсі нейронні мережі були застосовані до набору даних, який містив понад мільйон зображень з Інтернету, в якому представлено понад 1000 різних класів. Цей

успіх зробив революцію в застосуванні глибоких мереж комп’ютерного зору в різних напрямках і в багатьох різних програмах комп’ютерного зору [161].

Після того було створену велику кількість сучасних архітектур для згорткових нейронних мереж, таких як ResNet або DenseNet [162]. Тож можемо спробувати застосувати їх до ще простіших одновимірних даних, у яких ми можемо замінити двовимірні згортки одновимірними. CNN високоефективні, швидкі, їх можна оптимізувати і добре працюють як для класифікації, так і для регресійного аналізу [135].

Розглянемо як CNN успішно застосується на даних для розпізнавання зображень, а потім покажемо, як можна застосувати цю архітектуру для даних представлених у вигляді часових рядів.

Вхідні дані визначаються як тензор 3-го або 2-го рангу, залежно від кількості каналів, присутніх у даних. Канали означають кількість змінних, які описують колір пікселя. Тобто, у разі кодування RGB, матимемо 3 канали. Однак, щоб зменшити обсяг даних, досить часто вхідні дані переводяться в 1 чорно-білий канал. Отримане значення операції згортки для кожного пікселя відфільтрованого зображення обчислюється на основі квадратної зони навколо нього за допомогою ядра згортки (kernel). Рис. 3.31 демонструє це перетворення:

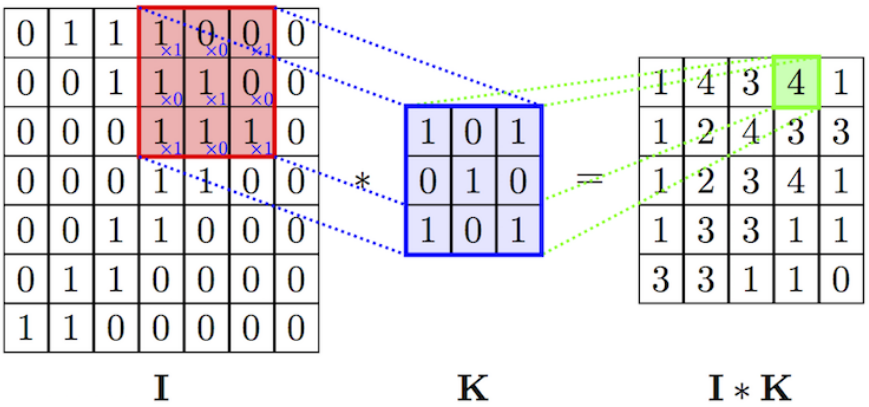


Рис. 3.31 Операція згортки

Формула для операції згортки буде виглядатиме наступним чином [126]:

$$(I * K)(r,s) = \sum_{i=-u}^{u} \sum_{j=-v}^{v} K(i,j) I(r+i, s+j), \tag{3.63}$$

Де (I*K)(r,s) - значення результуючої матриці в точці (r,s)

K - ядро згортки (kernel), матриця розмірності (2u+1, 2v+1),

I - вхідна матриця, I(i,j) - значення пікселя (I,j) для вхідного зображення.

Як бачимо з рис.3.31 та формули (3.63) вище, у згортковій операції ми робимо поелементне множення матриці, просуваючи матрицю ядра згортки по вхідній матриці. Матриця ядра – це матриця з параметрами моделі, які необхідно вивчити. Вона ініціалізується випадковим чином, а потім змінюється під час навчання в результаті процедури оптимізації, такої як стохастичний градієнтний спуск.

Варто зазначити, що операція згортки може бути записана в термінах звичайних лінійних перетворень, що використовуються в класичних нейронних мережах, тож до них також застосовна теорема про універсальну апроксимацію [163].

Розглянемо зображення розміром 3x3 (рис. 3.32):

A	B	C
D	E	F
G	H	J

Рис. 3.32

Застосуємо до нього згортку з ядром 2x2 наступного вигляду (рис. 3.33):

Рис. 3.33

Застосовуючи згортку, отримаємо наступний результат (рис. 3.34):

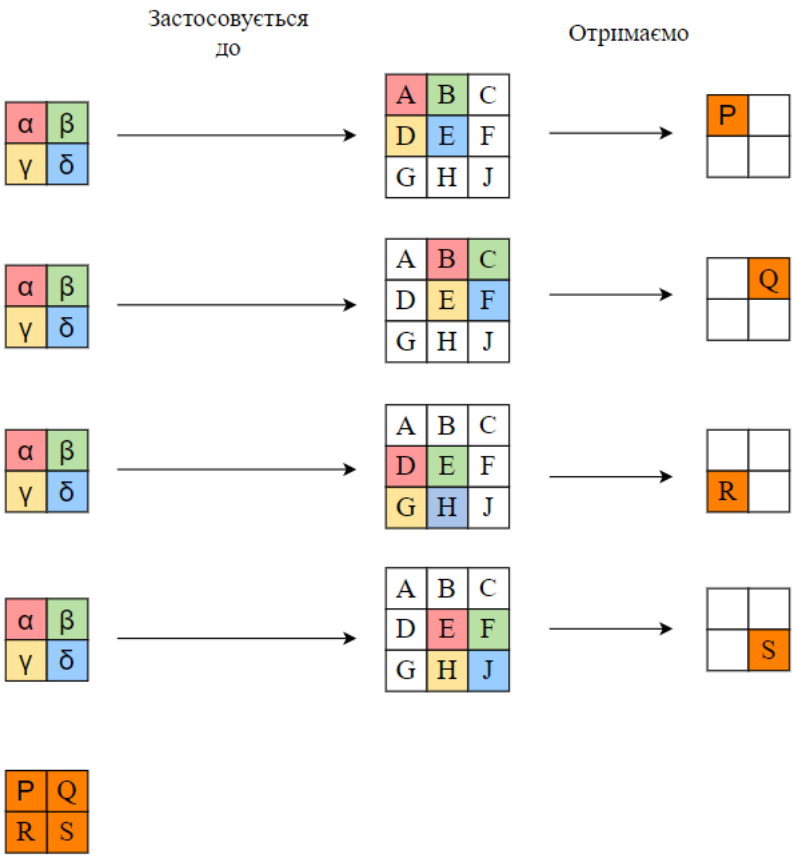


Рис. 3.34 Процес обрахування згортки

У вигляді системи рівнянь, маємо:

$$\alpha A + \beta B + \gamma D + \delta E + b = P$$

$$\alpha B + \beta C + \gamma E + \delta F + b = Q$$

$$\alpha D + \beta E + \gamma G + \delta H + b = R$$

$$\alpha E + \beta F + \gamma H + \delta J + b = S$$

(3.64)

Зазначимо, що член зміщення, b, однаковий для кожного рівняння перетворення зображення. Можна розглядати зміщення як частину ядра, так само, як ваги (α, β, γ, δ) є частиною ядра.

Отже, згортку можна представити як особливий вид множення матриці, як показано на рис. 3.35:

$$\begin{bmatrix} \alpha & \beta & 0 & \gamma & \delta & 0 & 0 & 0 & 0 \\ 0 & \alpha & \beta & 0 & \gamma & \delta & 0 & 0 & 0 \\ 0 & 0 & 0 & \alpha & \beta & 0 & \gamma & \delta & 0 \\ 0 & 0 & 0 & 0 & \alpha & \beta & 0 & \gamma & \delta \end{bmatrix} \times \begin{bmatrix} A \\ B \\ C \\ D \\ E \\ F \\ G \\ H \\ J \end{bmatrix} + \begin{bmatrix} b \\ b \\ b \\ b \end{bmatrix} = \begin{bmatrix} \alpha A+\beta B+0C+\gamma D+\delta E+0F+0G+0H+0J+b \\ 0A+\alpha B+\beta C+0D+\gamma E+\delta F+0G+0H+0J+b \\ 0A+0B+0C+\alpha D+\beta E+0F+\gamma G+\delta H+0J+b \\ 0A+0B+0C+0D+\alpha E+\beta F+0G+\gamma H+\delta J+b \end{bmatrix} = \begin{bmatrix} \alpha A+\beta B+\gamma D+\delta E+b \\ \alpha B+\beta C+\gamma E+\delta F+b \\ \alpha D+\beta E+\gamma G+\delta H+b \\ \alpha E+\beta F+\gamma H+\delta J+b \end{bmatrix} = \begin{bmatrix} P \\ Q \\ R \\ S \end{bmatrix}$$

Рис. 3.35 Згортка, як матрична операція

Матриця ваг така ж, як у традиційних нейронних мереж. Однак ця вагова матриця має дві особливі властивості:

1. Нулі, показані сірим кольором, не підлягають навчанню. Це означає, що вони залишаться нульовими протягом усього процесу оптимізації.

2. Деякі ваги рівні, і хоча їх можна навчати (тобто змінювати), вони повинні залишатися рівними. Вони називаються спільними вагами (аналогічно до ідеї поширення спільних матриць ваг у рекурентних мережах).

Нулі відповідають пікселям, яких ядро не охоплює. Кожен рядок вагової матриці відповідає одному застосуванню ядра.

Для зображень довільного розміру, запишемо формулу операції згортки і отримаємо лінійне перетворення даних, просто особливого вигляду [135]:

$$y_{i,j}^{s} = \sum_{-d<=a,b<=d} W_{a,b} x_{i+a,j+b}^{s} \quad (3.65)$$

Де x^s - карта ознак в прошарку s,

$Y_{i,j}^{s}$ - результат згортки на прошарку s,

$X_{i,j}^{s}$ - вхід прошарку s,

W - матриця ваг розмірністю 2d+1

При побудові згорткового шару можна використовувати кілька фільтрів, за допомогою послідовного використання декількох різних матриць ядра згортки в результаті чого на вихідних зображеннях можна отримати нові канали, які зазвичай називають картами ознак, оскільки з різними ядрами вони дійсно вивчають ті чи інші ознаки в зображеннях [158]. Також, аналогічно до класичних нейронних мереж (перцептрон), можна утворювати глибокі, багатошарові згорткові нейронні мережі, застосовуючи операцію згортки послідовно, вивчаючи таким чином з даних якомога складні взаємозв’язки. Тобто в цілому ознаки можуть змінюватися, як показано на рис. 3.36.

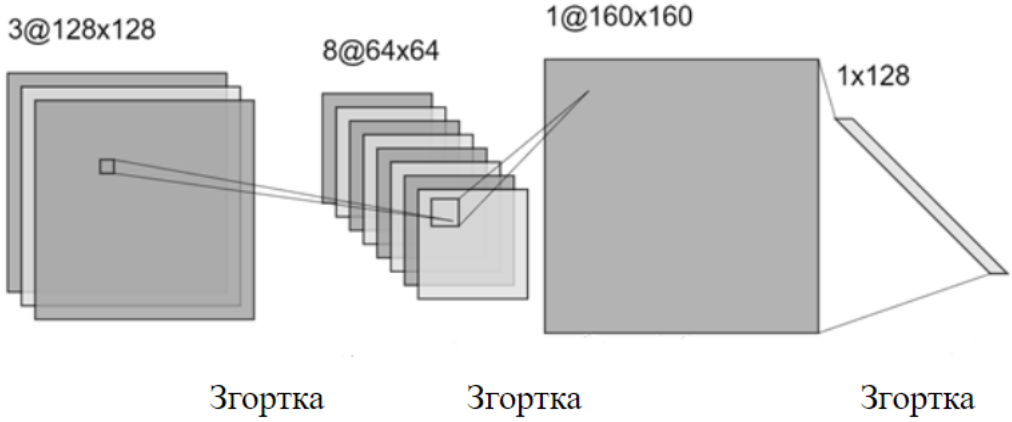


Рис. 3.36 Багатошарова згорткова мережа

Формула для багатошарової згорткової мережі буде виглядати наступним чином [126]:

$$y_i^s = b_i^s + \sum_{j=1}^{m_{s-1}} K_{i,j}^s y_i^{s-1}, \tag{3.66}$$

Де y_i^s - значення в прошарку s

b_i^s - зміщення для i-го значення в прошарку s

m_{s-1} - розмір (s-1)-го прошарку

$K_{i,j}^s$ - значення ядра в заданій точці.

Таким чином, бачимо, як згорткові нейронні мережі можуть бути застосовані до проблем комп’ютерного зору і працюють особливо добре, завдяки їх здатності працювати конволюційно, витягуючи ознаки з локальних груп вхідних даних та забезпечуючи модульність та ефективність представлення даних. Ті самі властивості, завдяки яким CNN досягають

успіху в комп’ютерному зору, також роблять їх дуже актуальними для обробки послідовностей.

Можемо спробувати використовувати CNN для задачі прогнозування даних часових рядів, у яких можемо замінити 2D згортки одновимірними. Тому можемо спробувати використати всі переваги, описані вище, і досягти чудової продуктивності, нелінійної та комбінованої взаємодії між змінними з допустимою швидкістю та складністю. Час можна розглядати як просторовий вимір, як-от висота або ширина 2D-зображення. Такі 1D-CNN можуть бути конкурентоспроможними з RNN у певних проблемах обробки послідовностей, як правило, при значно менших обчислювальних витратах. Невеликі 1D-CNN можуть запропонувати швидку альтернативу RNN для таких завдань, як прогнозування часових рядів. Шари згортки, введені раніше, являли собою 2D згортки, витягуючи двовимірні взаємодії тензорів зображень і застосовуючи ідентичне перетворення до кожної групи. Таким же чином можемо використовувати 1D згортки, витягуючи локальні 1D групи (підпослідовності) з послідовностей (рис. 3.37).

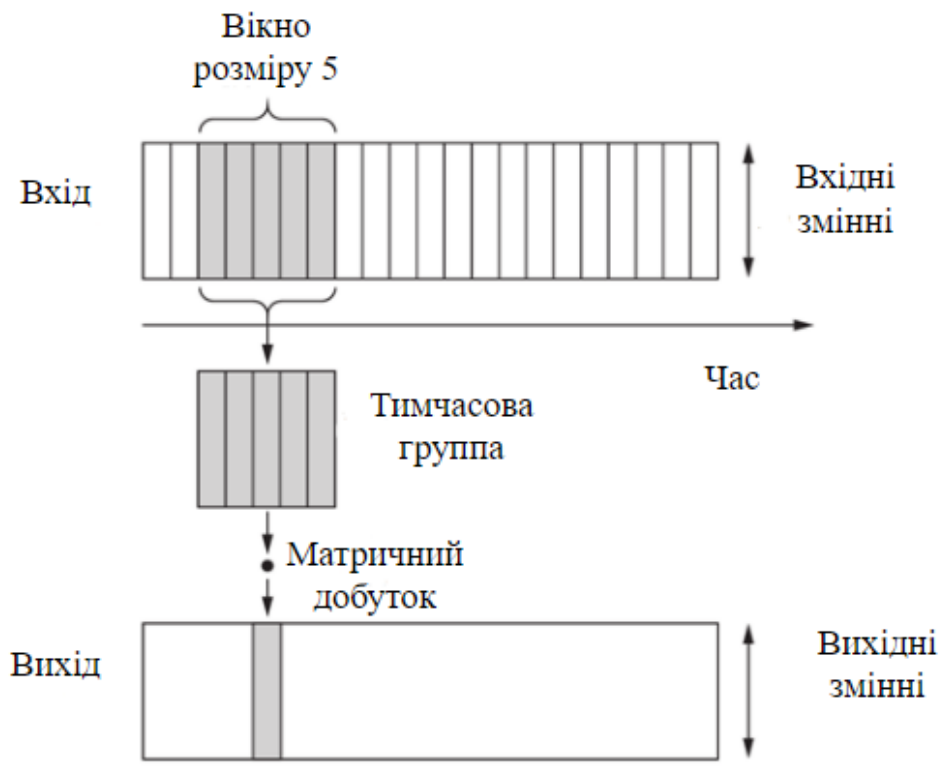


Рис. 3.37. 1D згортки

Як працює одновимірна згортка: кожен вихідний часовий крок отримується з тимчасової групи у вхідній послідовності.

Такі одновимірні згортки можуть розпізнавати локальні шаблони в послідовностях. Оскільки для кожної групи виконується одне й те саме вхідне перетворення, шаблон, засвоєний у певній позиції, пізніше може бути розпізнаний в іншій позиції, що робить перетворення 1D-CNN інваріантним (для тимчасових перетворень). Наприклад, 1D-CNN для обробки часових рядів з використанням вікон згортки розміром 5 повинна мати можливість вивчати фрагменти шаблону довжиною 5 або менше, і повинна мати можливість розпізнавати їх у будь-якому іншому контексті у вхідному ряді.

Отже, для моделювання і прогнозування нелінійної частини нелінійних нестаціонарних процесів пропонується використання математичних моделей на основі згорткових нейронних мереж.

3.9. Прогнозування за допомогою нейронних мереж

Структура нейронних мереж така, що дозволяє виконувати прогнозування на один крок без додаткових перетворень. Використовуючи попередні значення ряду і підставляючи їх в модель можна отримати оцінку прогнозу цільової змінної, відбувається звичайний прохід вперед мережею – forward propagation [137].

Але на практиці іноді необхідно знайти оцінку прогнозу одразу на декілька кроків вперед, не знаючи майбутніх значень ряду чи інших факторів, тоді існують такі підходи:

1. Використання прогнозних значень в якості майбутніх значень ряду. Даний підхід є найбільш очевидним і простим для використання, однак таке виконання прогнозів за допомогою інших прогнозів поступово призводить до росту дисперсії похибок [48], а тому не є рекомендованим на практиці, особливо для довгострокових прогнозів.

2. Використання sequence-to-sequence моделей, коли ми навчаємо мережу прогнозувати не один а одразу декілька кроків. Такий підхід використовує ідеї з моделей, що застосовуються у сучасних перекладачах: рекурентна нейронна мережа отримує на вхід і віддає на вихід текст довільної довжини (рис 3.38). Цей підхід досить успішно зарекомендував себе в сфері обробки текстових даних [126], але він застосований лише коли ми використовуємо саме рекурентні нейронні мережі.

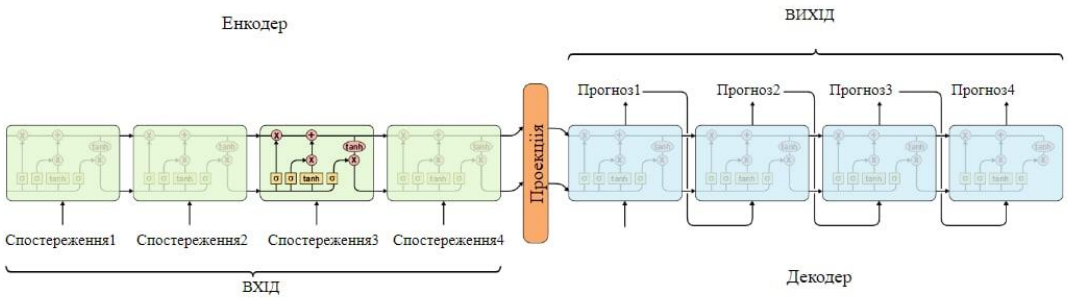


Рис. 3.38. Прогнозування sequence-to-sequence моделлю

3. Використання декількох моделей, кожна з яких вчиться прогнозувати свій крок. В даному підході будується стільки моделей, на скільки кроків необхідно прогнозувати. Відповідна модель за номером k буде отримуючи той самий набір вхідних даних, що і інші моделі, буде прогнозувати саме на k-й крок вперед, без попередніх і наступних. Такий підхід є цілком ефективним з точки зору якості прогнозів, проте побудова великої кількості моделей для великої кільості кроків може бути важкою з обчислювальної точки зору.

4. Використання multi-output нейронних мереж, коли можна в мережі робити декілька виходів. Можна залишити ідею з попереднього пункту, вдосконаливши за допомогою тої особливості нейронної мережі, що вона може мати декілька виходів, утворюючи так звані multi-output або multi-task мережі (рис. 3.39)

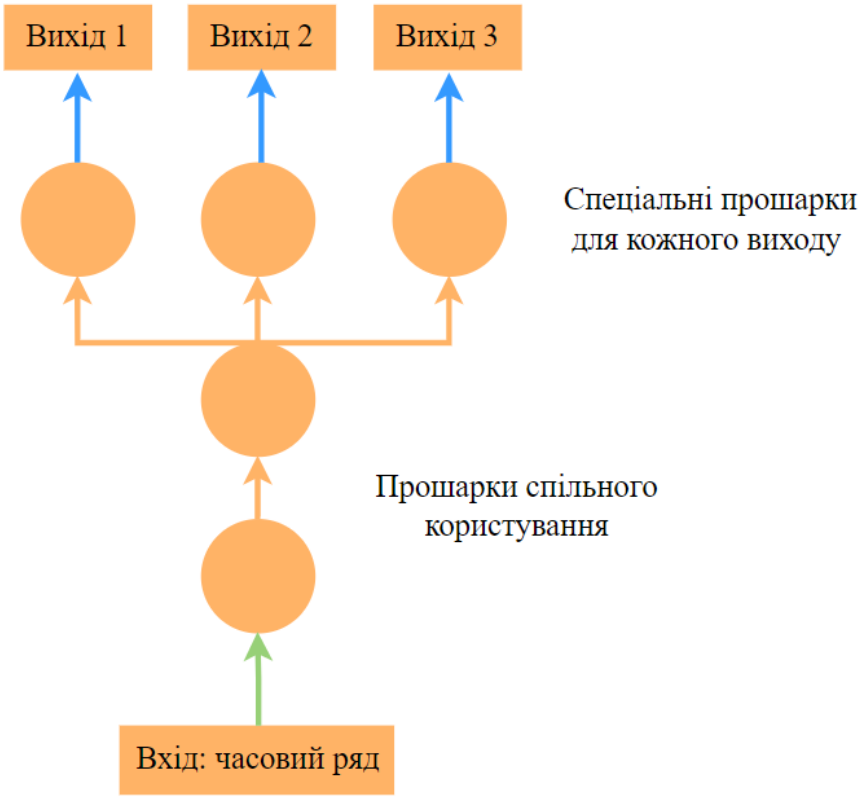


Рис. 3.39 Multi-output нейронна мережа

Така нейронна мережа може виконувати ті ж завдання, що і декілька нейронних мереж - скільки завдань буде поставлено, і зроблено відповідних виходів, стільки мережа і буде намагатись виконати. При цьому, такий підхід часто є більш ефективним за використання декількох мереж, не тільки через виграш у обчислювальних витратах, а й через те, що додаткові задачі в нейронній мережі допомагають в розв’язанні інших задач, через так звані “ваги спільного користування” [149]. В нашому випадку відповідні нові задачу (виходи) - це відповідні кроки прогнозування. Саме такий підхід для виконання довгострокових прогнозів пропонується використовувати в роботі.

3.10 Методи комбінування оцінок прогнозів. Комбінування лінійної та нелінійної частини процесу

Оскільки в основі запропонованої методики моделювання і прогнозування нелінійних нестаціонарних процесів лежить розділення процесу на лінійну і нелінійну складові, необхідно розглянути особливості подальшого комбінування відповідних оцінок прогнозів.

Існують такі методи комбінування оцінок прогнозів [21]:

1. Метод ансамблювання полягає в незалежній побудові декількох моделей з їх подальшим усередненням (простим або зваженим за допомогою значень похибок прогнозів). Цей метод є найпростішим в реалізації, але може не враховувати різні переваги, недоліки і особливості моделей при побудові і комбінуванні.

2. Метод беггінгу (bagging) за основу бере метод ансамблювання, але різні моделі будуються на різних підвибірках даних. В даному методі часто для побудови моделей використовують один і той же алгоритм, а різні особливості моделей та прогнозів отримуються завдяки тому, що моделі під час навчання працювали з різними наборами даних. Метод є також простим для використання, але зазвичай потрібен саме тоді, коли необхідно зробити ансамбль зі схожих за природою моделей.

3. Метод стекінгу (stacking) - це метод оптимізації вагових коефіцієнтів при комбінуванні оцінок прогнозів в ансамблі за допомогою побудови додаткової моделі, що за допомогою прогнозів з побудованих моделей буде прогнозувати необхідні значення ряду. Метод є доволі складним для використання і використовуються тоді, коли в методі ансамблювання необхідно якомога точніше підібрати коефіцієнти.

4. Метод бустінгу (boosting) полягає в тому, що моделі будуються послідовно, кожна наступна модель будується на похибках попередньої, а прогноз отримується як сума прогнозів. Такий підхід дозволяє комбінувати

різні за природою моделі, враховуючи їх особливості. В роботі для комбінування оцінок прогнозів лінійної та нелінійної частини процесу пропонується використовувати саме метод бустінгу.

Для побудови моделей методом бустінгу в запропонованій методиці спочатку виконується побудова AR моделі для виділення лінійної складової процесу, обчислюються залишки моделі. На залишках моделі виконується навчання нейронної мережі для опису нелінійної нестаціонарної складової (рис 3.40).

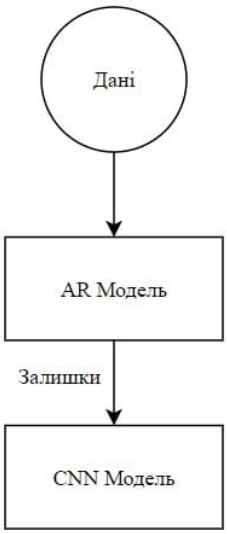


Рис. 3.40 Побудова комбінованої моделі методом бустінгу

Для прогнозування за допомогою комбінованої моделі окремо виконується прогноз за AR моделлю та нейронною мережею. Оцінки прогнозів додаються для отримання остаточного прогнозу (рис. 3.41).

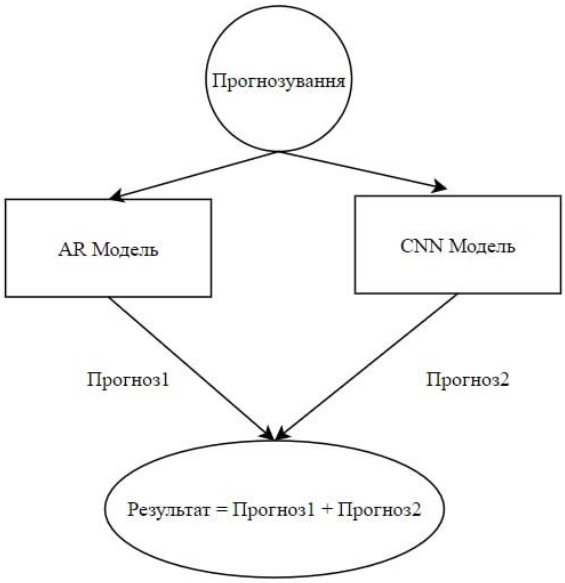


Рис. 3.41 Комбінування оцінок прогнозів лінійної та нелінійної складової

3.11. Вибір критерію якості для оцінювання прогнозів нелінійних нестаціонарних процесів

Одним із важливих етапів запропонованої в даній роботі загальної методики є використання критеріїв для оцінювання якості прогнозів отриманих в процесі моделювання та дослідження і, відповідно, вибір кращої з оцінених моделей-кандидаток. Лише після цього можливе застосування моделі для розв’язання основної задачі – прогнозування, керування та поглибленого дослідження процесу.

Можна очікувати, що найкраща модель зможе отримати найкращі результати за усіма доступними метриками, проте досить часто це не так. Як правило, на початку процесу оптимізації структури і параметрів моделі дійсно більшість критеріїв мають тенденцію до поліпшення, проте через деякий час вони досягають точки, коли поліпшення однієї метрики може призвести до погіршення стану іншої. У таких випадках стоїть задача вибору пріоритетного

критерію оцінювання якості прогнозів, що наразі є дуже актуальною та невирішеною для ряду задач.

Важливим моментом процесу прогнозування є об’єктивне визначення якості отриманого прогнозу. Зазвичай дослідників цікавить, наскільки добре алгоритм прогнозування працює з даними, до яких раніше не застосовувався. Це показує, як модель поводитиметься з реальними майбутніми даними на практиці. Це так звана властивість узагальнення алгоритмів машинного навчання, що є власне найважливішою його метою. Точність прогнозів можна визначити лише з урахуванням того, наскільки якісно працює модель з новими даними, які не використовувались при навчанні моделі. Тому наявну вибірку даних доцільно розділити на навчальну та тестову. На навчальній вибірці виконується навчання – оцінювання структури і параметрів методу прогнозування, та реалізується так званий “історичний” прогноз, який дає змогу встановити якість однокрокового прогнозу на цій ділянці ряду.

Прогноз на тестовій частині вибірки даних використовується для оцінювання його точності та вибору найкращої моделі, що має властивості узагальнення. Оскільки тестові дані не використовуються при навчанні моделі, вони повинні показати, наскільки вдало модель прогнозує нові дані. Таким чином запобігають так званого ефекту “перенавчання”. У різних емпіричних дослідженнях рекомендують залишати для перевірки 5–40 % значень ряду даних. Хоча під час аналізу коротких рядів доцільно використовувати значно більший відсоток ряду для оцінювання структури та параметрів моделі. Також це значення залежить від тривалості вибірки та того, настільки далеко необхідно прогнозувати. В ідеалі тестова вибірка повинна відповідати величині необхідного максимального горизонту прогнозу. Після вибору математичної моделі прогнозують значення поза вибіркою даних. Підходи до поділу вибірки на навчальну та тестову детально розглянуто у джерелах [48], [64], [126]. У цій роботі будемо вважати, що критерії якості обчислюють для тестового набору даних, окремого від тих, на

яких система навчалася. Зауважимо також, що метрики якості прогнозування відрізняються від так званих “залишків прогнозів”. По-перше, залишки розраховуються на навчальній вибірці, тоді як похибки оцінок прогнозів обчислюються на тестовому наборі. По-друге, залишки ґрунтуються на однокрокових прогнозах, тоді як похибки прогнозів можуть містити багатокрокові прогнози.

Для встановлення найкращої моделі при розв’язанні певної задачі використовують різні метрики для оцінювання якості прогнозів. Розглянемо найпоширеніші статистичні критерії якості прогнозу та їх призначення.

Досить часто проста сума похибок прогнозів дорівнює нулю, оскільки похибки мають різні знаки, а тому необхідно використовувати інші міри для похибок. Наприклад, можна використовувати квадрати похибок. Отримана формула за змістом схожа на стандартне відхилення, але прогнозних значень відносно реальних.

Середньоквадратична похибка моделі (англійською RMSE – Root Mean Squared Error) моделі є дуже популярною метрикою й обчислюється так:

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{k=1}^{N}[\hat{y}(k) - y(k)]^2}{N}}, \qquad (3.67)$$

де $\hat{y}(k)$ – прогнозоване за моделлю значення; $y(k)$ – реальне вимірювання; N – довжина вибірки.

RMSE може набувати значень $[0,+\infty)$. Чим вона менше, тим кращим вважають прогноз. Варто зазначити, що всі описані нижче властивості стосуються також спорідненої середньоквадратичної помилки MSE (Mean Squared Error):

$$\mathrm{MSE} = \frac{1}{N}\left[y(k) - \hat{y}(k)\right]^2, \mathrm{RMSE} = \sqrt{\mathrm{MSE}}.$$
$$\mathrm{MSE}(a) > \mathrm{MSE}(b) \Leftrightarrow \mathrm{RMSE}(a) > \mathrm{RMSE}(b). \quad (3.68)$$

MSE є евклідовою відстанню прогнозів до реальних значень. Також цю метрику можна отримати за методом максимальної правдоподібності. Для цього розглянемо прогнозну модель як

$$\mathrm{y} = \mathrm{a_w}(\mathrm{x}) + \varepsilon, \varepsilon \sim N(0, \sigma^2), \quad (3.69)$$

де y – фактичні виміри; $a_w(x)$ – модель зі змінними x і параметрами w; ε – нормально розподілена випадкова величина (шум) із заданою фіксованою дисперсією σ^2.

Для оцінювання параметрів w моделі aw (x) випишемо функцію правдоподібності моделі:

$$\mathrm{p}(\mathrm{y} \mid \mathrm{x}, \mathrm{w}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(\mathrm{y} - \mathrm{a_w}(\mathrm{x}))^2}{2\sigma^2}\right]. \quad (3.70)$$

Застосуємо метод максимальної правдоподібності

$$\log L(\mathrm{w}) = \log \prod_{\mathrm{i}=1}^{m} \mathrm{p}(\mathrm{y_i} \mid \mathrm{x_i}, \mathrm{w_i}) = \sum_{\mathrm{i}=1}^{m}\left[-\frac{1}{2}\log(2\pi\sigma^2) - \frac{(\mathrm{y_i} - \mathrm{a_w}(\mathrm{x_i}))^2}{2\sigma^2}\right] \to \max, \quad (3.71)$$

де m – розмір вибірки.

Звідси отримуємо задачу мінімізації MSE:

$$\frac{1}{2\sigma^2}\sum_{\mathrm{i}=1}^{m}(\mathrm{y_i} - \mathrm{a_w}(\mathrm{x_i}))^2 \to \min, \quad (3.72)$$

де MSE мінімізує як зміщення точкової оцінки, так і дисперсію.

Якщо втрати під час прогнозування, пов'язані з завищенням фактичного майбутнього значення, врівноважуються заниженням, то ідеальний прогноз повинен бути незміщеним. У такому разі MSE має прямувати до нуля.

Зміщення точкової оцінки визначається як

$$bias(\hat{\theta}_{\mathrm{m}}) = E(\hat{\theta}_{\mathrm{m}}) - \theta \ , \tag{3.73}$$

де θ – випадкова величина.

Оскільки MSE вимірює очікувану розбіжність (у контексті квадратів помилок), то маємо:

$$\mathrm{MSE} = E[(\hat{\theta}_{\mathrm{m}} - \theta)^2] = Bias(\hat{\theta}_{\mathrm{m}})^2 + Var(\hat{\theta}_{\mathrm{m}}) \ . \tag{3.74}$$

Вказана вище значна кількість способів отримання критерію якості MSE пояснює його широку популярність.

Основні властивості MSE:

1. Штраф стосовно великих похибок збільшується, а для малих зменшується. Менші похибки (наприклад, менші за 1) матимуть ще менший внесок у загальну похибку після піднесення до квадрата, тоді як більші похибки матимуть набагато більшу вагу після піднесення до квадрата.

2. Чутливість до викидів. Аналогічно до попереднього пункту наявність викидів (аномально великих значень даних) призведе до дуже великих значень метрики, що також вплине на поведінку алгоритму оптимізації.

3. Легка оптимізація. Структура функції, завдяки використанню квадрата, легко диференційована, тому різні алгоритми (наприклад,

градієнтний спуск) зможуть із нею працювати. Хоча метрики MSE та RMSE споріднені, їхні градієнти відрізнятимуться:

$$\frac{\partial \mathrm{MSE}}{\partial a} = \frac{2}{m}\sum_{i=1}^{m}(a(x_i) - y_i), \frac{\partial \mathrm{RMSE}}{\partial a} = \frac{1}{m\mathrm{RMSE}}\sum_{i=1}^{m}(a(x_i) - y_i), \quad (3.75)$$

де a(x) – прогнозна модель зі змінними x; m – розмір вибірки.

Тобто градієнти відрізняються в 2/RMSE рази.

4. Оптимальне розв’язання задачі мінімізації RMSE у класі констант – це середнє арифметичне ряду, як показано в [164].

Ці властивості також стосуються метрики R^2, оскільки її формулу можна виразити через MSE:

$$R^2 = 1 - \frac{SSE}{SST} = 1 - \frac{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\frac{1}{N}\sum_{i=1}^{N}(y_i - \bar{y})^2} = 1 - \frac{MSE}{MSE(\bar{y})}, \quad (3.76)$$

де $\bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i$ – модель, яка прогнозує все середнім значенням навчальної вибірки.

Тобто метрика R^2 показує, наскільки наша модель (за критерієм MSE) краща за наївну модель, що прогнозує все середнім значенням. Якщо наша модель не припускається помилок, то R^2 = 1; якщо наша модель така ж, як і наївна модель, то R^2 = 0. Однак вона може бути й гірша, тому діапазон можливих значень для метрики R^2: (−∞,1], і вважається, що чим ця метрика більша, тим краще.

Отже, RMSE, MSE, R^2 – дуже популярні метрики. Їх доцільно застосовувати у випадках, коли не можна дозволити собі велику похибку в жодному зі спостережень. Інакше кажучи, нам може бути зручно мати трохи

більшу похибку на багатьох зразках, якщо ніколи не отримаємо занадто велику похибку на жодному з них. Наприклад, якщо з погляду постановки задачі похибка прогнозу на 200 понад удвічі гірша, ніж похибка на 100, то слід використовувати RMSE чи MSE.

Замість квадратів можна використовувати модулі похибок, що також розв'яже проблему різних знаків у значеннях похибки (залежно від перепрогнозу чи недопрогнозу). Це також доволі популярний підхід і на практиці часто застосовується разом з MSE. Отримана метрика називатиметься середньою абсолютною похибкою, або MAE (Mean Absolute Error) й обчислюватиметься за формулою:

$$\mathrm{MAE} = \frac{\sum_{k=1}^{N} |\hat{y}(k) - y(k)|}{N} \tag{3.77}$$

де $\hat{y}(k)$ – прогнозоване за моделлю значення; $y(k)$ – реальне вимірювання; N – довжина вибірки.

MAE може набувати значень $[0,+\infty)$ та є дуже інтерпретабельною, оскільки вимірюється в тих самих одиницях, що й вихідний ряд. Чим меншою є MAE, тим кращим вважають прогноз. Зазначимо, що математично MAE є Мангеттенською метрикою, введеною Германом Мінковським [165]. Також можна отримати теоретичне підґрунтя для використання цієї метрики як функції похибки за допомогою методу максимальної правдоподібності. Нехай модель будують як і раніше, але похибки (шум) будемо вважати розподіленими за Лапласом:

$$y = a_w(x) + \varepsilon, \varepsilon \sim laplace(0, \alpha) \tag{3.78}$$

Для оцінювання параметрів w моделі a_w (x) випишемо функцію правдоподібності моделі:

$$p(y \mid x, w) = \frac{\alpha}{2} \exp[-\alpha \mid y - a_w(x) \mid] \tag{3.79}$$

Застосування методу максимальної правдоподібності відповідає максимізації

$$\log L(w) = \log \prod_{i=1}^{m} p(y_i \mid x_i, w) = \sum_{i=1}^{m} \log \left[\frac{\alpha}{2} - \alpha \mid y_i - a_w(x_i) \mid \right] \to \max \tag{3.80}$$

або (відкидаючи константи)

$$\alpha \sum_{i=1}^{m} \mid y_i - a_w(x_i) \mid \to \min \tag{3.81}$$

Тож максимізація функції правдоподібності еквівалентна мінімізації MAE. Основними властивостями MAE є:

1. Похибки зважуються однаково, похибка на 2 гірша за похибку на 1 удвічі.

2. Значно стійкіша до викидів, ніж MSE. Якщо ми впевнені, що в даних це саме викиди (помилкової природи), а не певні великі значення, цікаві з погляду дослідження, – варто використовувати MAE, а не MSE.

3. Складність оптимізації. MAE не диференційована в нулі (коли прогнози рівні фактичним значенням), і, залежно від розподілу ряду, це може зробити різні наближення для MAE кращими за інші.

4. Оптимальне розв’язання задачі мінімізації MAE в класі констант – це медіана вихідного ряду, як показано в [164]. Для задач “переривчастого

попиту”, коли в ряді багато нульових значень, небезпечно використовувати MAE, оскільки мінімізація за нею прямуватиме до того, щоб вважати нулем усі прогнози (бо вони будуть прямувати до медіани).

Отже, MAE є також популярною метрикою. Її варто використовувати в протилежних для RMSE випадках. Тобто коли не хочемо підкреслювати великі похибки, а навпаки – вважаємо похибки однаково важливими за своїми обсягами. MAE більш стійка до викидів, ніж RMSE.

Ідея впровадження відносної метрики полягає в наступному. Наприклад, ми досліджуємо задачу прогнозування курсу акцій на біржі. У разі, коли реальний курс був $1, а ми спрогнозували $2, з погляду задачі ми сильно помилилися, а якщо реальний курс був $100, а ми спрогнозували $101, то ми насправді майже вгадали. Однак MAE всюди буде однакова. Тому було запропоновано розглянути різні відносні або відсоткові метрики, які є незалежними до одиниці вимірювання ряду. Оскільки така міра характеризує відносну якість прогнозу, то її використовують переважно для порівняння точності прогнозів різнорідних об’єктів (процесів) прогнозування. Однак вона є також зручною для порівняльного аналізу якості прогнозування того самого процесу різними методами, оскільки відносна міра є чіткою та зрозумілою для дослідника та користувача [48], [64].

Найпопулярніша з таких метрик, середня абсолютна похибка у відсотках MAPE (Mean Absolute Percentage Error), обчислюється за формулою

$$\mathrm{MAPE} = \frac{100\,\%}{N} \sum_{k=1}^{N} \left| \frac{\hat{y}(k) - y(k)}{y(k)} \right| \tag{3.82}$$

де $\hat{y}(k)$ – прогнозоване за моделлю значення; $y(k)$ – реальне вимірювання; N – довжина вибірки. MAPE може набувати значень $[0,+\infty)$; чим менше значення метрики, тим кращий прогноз.

Основні властивості MAPE:

1. Похибки зважуються однаково, похибка в 20 % вдвічі гірша за похибку в 10 %.

2. Метрика стійка до викидів у даних.

3. Складність оптимізації.

4. Оптимальне розв'язання задачі мінімізації MAE в класі констант – це зважена медіана вихідного ряду, як показано в [164].

Проте ця метрика має значний недолік: якщо реальні значення будуть нульовими чи доволі близькими до нуля, то значення MAPE прямуватиме до нескінченності. Про це необхідно пам'ятати при застосуванні цього критерію якості прогнозу. Як запропоновано в [48], для того щоб виконати обчислення даного критерію в таких випадках, нульові значення необхідно пропускати з відповідним коригуванням значення N. Можливо, цей підхід дає змогу наближено проаналізувати якість прогнозування, проте ми не враховуємо нульові значення під час оцінювання якості, а прогнозування саме нульових значень може бути суттєвим, якщо навіть не принциповим. Тому в багатьох випадках, в яких в ряді є нульові або близькі до нуля значення, дана метрика не може бути застосованою взагалі. Також за великого діапазону значень ряду, наявності "стрибків" угору чи вниз у цій метриці можна отримувати дуже великі значення, що робить її незастосовною. Наприклад, прогнозований дохід на завтра – $ 1000000, а фактичний дохід став $ 0,01. У такому разі MAPE буде 9999999900 %, і такі випадки – часте явище на практиці.

Також, як зазначено в [166], негативні похибки штрафуються сильніше за позитивні, тому було запропоновано метрику SMAPE [167]:

$$\mathrm{SMAPE} = \frac{1}{N}\sum_{k=1}^{N}\frac{|\hat{y}(k)-y(k)|}{(|y(k)|-|\hat{y}(k)|)/2}100 \tag{3.83}$$

де $\hat{y}(k)$ – прогнозоване за моделлю значення; $y(k)$ – фактичний вимір; N – довжина вибірки.

Проте вона також не розв’язує проблему нулів і, насправді не вирішує проблему симетричності, як було показано в [168]. У роботі [169] показано, що MAPE і SMAPE не рекомендовані до використання, і було запропоновано метрику MASE.

$$\mathrm{MASE} = \frac{1}{N}\sum_{i=1}^{N}|q_i|,$$
$$q_i = \frac{y(i)-\hat{y}(i)}{\frac{1}{N-1}\sum_{k=2}^{N}|y(k)-y(k-1)|} \tag{3.84}$$

MASE дійсно вирішує ряд поставлених задач, проте є дуже важкою важкою для використання й інтерпретації, тому може бути використана при виконанні наукових досліджень, проте рідко буває корисною на практиці. У роботах [170], [171] запропоновано різні складні метрики для вирішення цієї проблеми. Наприклад, SMAE [170] розв’язує проблему нулів, проте залишається проблема з різними рівнями, особливо, якщо дані нестаціонарні.

Отже, MAPE є ідейно цікавою для обрахунку відносної похибки та простої інтерпретації, проте її можна застосовувати, лише тоді, коли дані знаходяться далеко від нуля та не мають великого розкиду значень. Деякі з цих проблем можуть бути вирішені за допомогою інших відносних метрик, вказаних вище, проте вони є доволі складними для використання та інтерпретації і не вирішують всіх вищевказаних проблем.

Середньоквадратична логарифмічна похибка (Root Mean Squared Logarithmic Error) моделі пропонується як метрика, що використовується тоді, коли необхідно мінімізувати саме відношення до реальних значень (аналогічно до того, що намагається зробити MAPE).

RMSLE обчислюють за формулою

$$\mathrm{RMSLE} = \sqrt{\frac{1}{N}\sum_{k=1}^{N}[\log(y(k)+1) - \log(\hat{y}(k)+1)]^2}\,, \tag{3.85}$$

де $\hat{y}(k)$ – прогнозоване за моделлю значення; $y(k)$ – фактичний вимір; N – довжина вибірки.

Додавання одиниці (чи будь-якої константи) дає змогу врахувати випадки, коли цільова змінна дорівнює нулю (дає можливість обчислити логарифм). RMSLE може набувати значень [0,+∞); чим менше значення, тим кращий прогноз.

Теоретичне обґрунтування RMSLE не відрізняється від RMSE, оскільки є тим самим RMSE, але над логарифмованим рядом, тобто

$$\mathrm{RMSLE} = \mathrm{RMSE}(\log(y(k)+1), \log(\hat{y}(k)+1))\,. \tag{3.86}$$

Логарифмування ряду також має згладжувальний ефект, зводячи значення ряду приблизно до одного рівня, що може допомогти за наявності експоненційних трендів.

Але головна ідея полягає в тому, що логарифм частки буде різницею логарифмів, а саме:

$$\mathrm{RMSLE} = \sqrt{\frac{1}{N}\sum_{k=1}^{N}\left(\log\frac{y(k)+1}{\hat{y}(k)+1}\right)^2}\,, \tag{3.87}$$

роблячи RMSLE такою, що мінімізує відносну похибку, зберігаючи всі переваги класичної RMSE.

Основні властивості RMSLE:

1. Штраф до великих помилок збільшується, однак завдяки логарифмуванню вплив великих значень певним чином зменшується, тому метрику можна вважати збалансованою.

2. Нечутливість до викидів завдяки логарифмуванню. Це є перевагою, крім тих випадків коли треба викиди власне знайти.

3. Легка оптимізація завдяки тому, що можна оптимізувати MSE над логарифмованим рядом.

4. Оптимальне розв’язання задачі мінімізації RMSLE в класі констант – це середнє арифметичне логарифмованого ряду аналогічно до RMSE.

Отже, RMSLE є ідеальним варіантом, коли необхідно мінімізувати відносну помилку. Метрика не має недоліків MAPE, дуже зручна для практичного використання, має класичне теоретичне обґрунтування. Під час використання RMSLE варто пам’ятати про її нечутливість до викидів.

Таким чином, оптимізація моделі для деякого обраного критерію, погіршує її відносно іншого, тому важливо розуміти яку метрику необхідно використовувати для оптимізації та оцінки якості прогнозу в конкретній поставленій задачі.

Отже, для мінімізації абсолютної похибки рекомендовано використовувати метрики RMSE (MSE, R^2) та MAE, в залежності від необхідності роботи з викидами. Метрику RMSLE рекомендовано для вирішення задач мінімізації відносної метрики, вирішуючи показану проблематику метрики MAPE для цього класу задач.

3.12. Адаптивна побудова моделей прогнозування

Нелінійні нестаціонарні процеси динамічно змінюються у часі, а тому у випадку необхідності довготривалої або багатокрокової роботи системи необхідно, щоб вона могла адаптуватися до змін.

Адаптація моделі може відбуватися одразу після появи нових спостережень, при виявленні зміщення розподілу даних, або при суттєвій зміні якості прогнозів.

Для вдосконалення прогнозів на основі математичних моделей необхідно розробити нові відповідні модельні структури, які б адекватно описували досліджувані процеси та забезпечували можливість обчислення якісних прогнозів. Одним із найперспективніших сучасних підходів до моделювання та прогнозування є так званий системний підхід, який передбачає застосування принципів системного аналізу в рамках спеціалізованої системи підтримки прийняття рішень (СППР) [172, 173].

Усі методи в роботі реалізовані в адаптивних версіях, що робить систему більш гнучкою для нових даних і здатною попередньо виявити, а потім і боротися з деякими типами можливих невизначеностей (факторів негативного впливу на обчислювальні процедури, що використовуються в СППР, різного роду та походження). Структурна невизначеність полягає в тому, що при використанні СППР структура моделі практично завжди повинна оцінюватися за допомогою даних. Це означає, що елементи структури моделі майже завжди приймають лише наближені значення. Коли модель будується для прогнозування, ми створюємо кілька моделей-кандидаток і вибираємо найкращу з них, використовуючи набір критеріїв якості (адекватності) моделі. Ймовірнісна невизначеність та невизначеність щодо розподілу полягає у тому, що використання випадкових величин призводить до необхідності оцінки фактичних розподілів ймовірностей та їх застосування в обчислювальних процедурах логічного висновку. Зазвичай

спостережуване значення відомо лише приблизно, хоча ми знаємо межі фактичних значень. Відповідні розподіли ймовірностей дуже корисні для опису досліджуваних процесів у таких ситуаціях. Маючи справу з дискретними результатами, ми призначаємо ймовірності конкретним результатам за допомогою функції розподілу. Вона показує, яку «вагу» (або масу) призначити кожному спостереженню чи вимірюванню. Відповіддю на питання про цінність конкретного результату буде його маса. Аксіоми ймовірності Колмогорова допомагають глибше зрозуміти, що відбувається. Якщо дві або більше змінних аналізуються одночасно, необхідно побудувати та використовувати багатовимірні розподіли. Багатовимірні розподіли дозволяють оцінювати умовні ймовірності за допомогою процедур перенормування, коли це необхідно.

Для того, щоб мати можливість працювати з такими та іншими можливими невизначеностями усі методи в роботі спираються на принципи системного аналізу. В роботі пропонується використовувати такі принципи системного аналізу: системної координації функцій, процедурної повноти, залежності взаємної інформації, багатоцільового узагальнення та багатофакторної адаптації та ін. [174].

Принцип системної координації функцій передбачає, що всі методи, підходи та алгоритми (функції), реалізовані в системі, повинні бути структурно та функціонально узгоджені та взаємозалежні. Таким чином можна створити та практично впровадити унікальну системну методологію аналізу статистичних даних у рамках сучасної СППР та суттєво підвищити якість проміжних та кінцевих результатів.

Системний принцип процедурної повноти гарантує, що розроблена система забезпечить можливість вчасно та в потрібному місці виконати всі необхідні обчислювальні функції, спрямовані на збір даних (редагування, нормалізацію, фільтрацію та оновлення), формалізацію постановки проблеми,

побудову моделі, обчислення прогнозів, а також для виконання оцінки якості моделі та прогнозних оцінок на її основі.

Відповідно до принципу залежності взаємної інформації результати обчислень, що генеруються кожною процедурою, повинні відповідати форматам і вимогам інших процедур. Ця вимога реалізується за допомогою відповідних проектних рішень для створеної системи.

Відповідно до системного принципу багатоцільового узагальнення всі функціональні модулі розробленої системи повинні мати необхідний ступінь узагальнення, що забезпечує можливість досягнення високоякісних результатів розв’язання набору можливих задач, що належать до одного класу (це може бути високоякісне прогнозування), і створення альтернативних рішень щодо майбутньої еволюції лінійних або нелінійних нестаціонарних процесів). До таких проблем можна віднести: накопичення необхідних даних та їх попередню обробку; оцінка структури та параметрів набору математичних моделей-кандидаток; побудова функцій прогнозування на розроблених моделях та розрахунок відповідних прогнозів; вибір найкращих результатів обчислення з використанням відповідних наборів критеріїв якості.

Системний принцип багатофакторної адаптації спрямований на можливість вирішення задач адаптації обчислювальних процедур до задач моделювання різноманітних процесів різної складності в залежності від повноти наявної інформації та вимог користувача. Адаптація виконується в процесі оцінювання структури та параметрів моделі, тобто вся процедура ідентифікації досліджуваного процесу складається з набору адаптивних процедур, спрямованих на досягнення основної мети дослідження: побудова адекватної моделі та обчислення якісних прогнозів.

Розглянемо, як можна застосувати принципи системного аналізу для того щоб працювати з різними типами невизначеностей при адаптивній побудові версій моделей, що застосовуються в роботі, у двох основних напрямках адаптації: адаптація структури та параметрів моделі.

Для лінійної складової для структурної адаптації (структура моделі в запропонованій методиці визначається даними) використовується автоматичний пошук кращої структури за допомогою кореляційного аналізу (для оцінки лагу), алгоритму Хайдмана-Хандакара (для пошуку кращої структури), та відповідних статистичних критеріїв (AIC, BIC). Для адаптації параметрів використовується повторна оцінка за методом найменших квадратів, а у випадку постійної адаптації можна використовувати і рекурсивну версію [48] та Монте Карло для марковських ланцюгів [175].

Нейронні мережі є адаптивними моделями за їх процесом побудови і навчання. Адаптація нейронної мережі обраної структури (RNN, CNN), оскільки нейронна мережа є багатопараметричною моделлю, відбувається шляхом постійного “донавчання” нейронної мережі (incremental learning) [149]. З отриманням нових спостережень, вони додаються до навчальної вибірки і відбувається чергова ітерація алгоритму навчання - оптимізації параметрів нейронної мережі (back-prop, Adam).

3.13. Метод Монте-Карло марківських ланцюгів для адаптивної побудови моделей

У попередньому пункті було розглянуто необхідність адаптивної побудови моделей, а також можливі способи такої побудови. Для лінійних моделей запропоновано метод Монте-Карло марківських ланцюгів (МКМЛ). Розглянемо основні ідеї та схему побудови моделей за цим методом.

У пункті 3.1 було розглянуто процес побудови моделі лінійної регресії, для оцінки параметрів якої було застосовано підхід, що називається оцінкою за методом найменших квадратів.

Застосуємо тепер байєсівський підхід для оцінки невідомих параметрів. Однією з переваг байєсівського підходу є її висока здатність до адаптації -

здатність оцінювати параметри моделі рекурсивно, коли нова інформація надходить постійно і поступово.

В загальному випадку, розглядаючи адаптивний процес постійної побудови моделі, на деякому кроці k нової побудови (надходження нових даних) ідею МКМЛ можна записати наступним чином [175]:

$$\hat{\theta}_k = \hat{\theta}_{k-1} + \alpha\Delta\hat{\theta}(k), \tag{3.88}$$

де $\hat{\theta}_k$ - оцінка деякого параметру θ моделі на кроці k,

α - фіксоване дійсне число,

$\Delta\hat{\theta}(k)$ - приріст значення оцінки, який необхідно оцінити.

Бачимо, що оцінка параметру на наступному кроці залежить лише від оцінки на попередньому кроці, що задовольняє так званій марківській вимозі до моделювання стохастичних процесів марківськими ланцюгами. Щоб оцінити приріст оцінки, необхідна схема генерації даних з відповідного розподілу параметру, для чого використовується метод Монте-Карло. Звідси і загальна назва методу Монте-Карло для марківських ланцюгів.

Проблемними в даній побудові є питання переоцінки розподілу параметрів за надходження нових спостережень та питання генерації даних з відповідного розподілу для оцінки невідомих параметрів. Розглянемо, як ці питання вирішуються за допомогою МКМЛ. Для цього необхідно застосувати байевський підхід для побудови моделей.

Байєсівська побудова регресійної моделі в загальному випадку буде виглядати наступним чином:

$$y_i \sim N(a_1x_{i1} + \cdots + a_mx_{im}, \sigma), i = 1, \dots, n, \tag{3.89}$$

де N - нормальний розподіл з відповідними параметрами

В матричному вигляді, можемо записати:

$$y \sim N(X\alpha, \sigma_e I), \alpha \sim N(\alpha_0, K_{\alpha,0}), \tag{3.90}$$

де α_0 - вектор математичних сподівань на нульовому кроці,

$K_{\alpha,0}$ - відповідна коваріаційна матриця

Під нульовим кроком мається на увазі попередня оцінка без спостережень, або попередня оцінка за МНК, що була зроблена раніше, і яка буде уточнюватися з надходженням нових даних і поступовою адаптацією.

У випадку, коли і спостережувані змінні, і залежна змінна є нормально розподіленими випадковими величинами (хоча в загальному випадку це не завжди так), можна використовувати МНК, однак зручність у використанні МКМЛ полягає в тому, що можна не робити повного перерахунку всіх матричних операцій, а просто оновити значення деяких параметрів розподілу, користуючись так званою властивістю спряжених апріорних розподілів з сімейства нормального, гамма та їх обернених розподілів, яка математично встановлює, що нормально розподілений апріорний розподіл є спряжений з нормально розподіленою підтримкою (likelihood) і продукує нормально розподілений апостеріорний розподіл [176].

Розглянемо, на прикладі регресії з однією незалежною змінною, як буде виглядати процес адаптації за МКМЛ з використанням властивості спряжених апріорних розподілів у припущенні нормального розподілу змінних і параметрів. Маємо регресійну модель наступного вигляду:

$$y = X\alpha + \varepsilon = a + bx + \varepsilon \tag{3.91}$$

Тоді, можемо записати вектор α_0:

$$\alpha_0 = \begin{bmatrix} a_0 \\ b_0 \end{bmatrix} \tag{3.92}$$

І коваріаційну матрицю $K_{\alpha,0}$:

$$K_{\alpha,0} = \begin{bmatrix} \sigma_{a,0}^2 & 0 \\ 0 & \sigma_{b,0}^2 \end{bmatrix} \tag{3.93}$$

При надходженні кожної нової пари спостережень (x_1, y_1), (x_2, y_2) (потрібно хоча б 2 точки, щоб задати пряму), застосовуючи вищенаведену властивість, можемо отримати формулу для нових значень параметрів розподілу (вже на кроці 1).

$$\alpha_1 = K_{\alpha,1}(K_{\alpha,0}^{-1}\alpha_0 + K_y^{-1}\mu_y) \tag{3.94}$$

$$K_{\alpha,1} = (K_{\alpha,0}^{-1} + K_y^{-1})^{-1} \tag{3.95}$$

$$\mu_y = [\frac{x_1 y_2 - x_2 y_1}{x_1 - x_2} \quad \frac{y_1 - y_2}{x_1 - x_2}]^T \tag{3.96}$$

$$K_y = \begin{bmatrix} [(\frac{x_1}{x_1 - x_2})^2 + (\frac{x_2}{x_1 - x_2})^2]\sigma_e^2 & 0 \\ 0 & 2(\frac{1}{x_1 - x_2})^2 \sigma_e^2 \end{bmatrix} \tag{3.97}$$

Однак, в загальному випадку, при роботі з нелінійними нестаціонарними процесами, ми маємо справу з ймовірнісною невизначеністю, або з невизначеністю щодо розподілу, тобто ми не можемо гарантувати незмінний нормальний розподіл змінних і параметрів, а тому не можемо завжди користуватися властивістю вище, а також теоретично не є коректним і постійне використання МНК, оскільки порушуються необхідні

для його використання умови. Для таких випадків в МКМЛ існує схема апроксимації невідомого розподілу за допомогою генерації даних для апостеріорного розподілу, навіть у випадках, коли ми власне не знаємо вигляд шуканого розподілу.

Нехай нам потрібно оцінити невідомий розподіл деякого параметру α. Для цього пропонується згенерувати велику кількість точок з цього розподілу, з яких потім можна за законом великих чисел [176] апроксимувати необхідні нам параметри розподілу, як то математичне сподівання. Але оскільки апостеріорний розподіл параметрів невідомий, необхідний інструмент для такої генерації. Потужним інструментом для вирішення поставленої задачі є алгоритм Метрополіса-Гастінгса [177]. Основні кроки алгоритму можна описати наступним чином:

1. Алгоритм починає роботу з деякого початкового значення α_0, яке на початку може бути довільним випадковим числом.

2. Вибирається нове значення α*, що пропонується як згенероване з шуканого розподілу. Генерація α* базується на деякій пропозиційній функції $g(\alpha^*,\alpha_0)$. Як вказано у [177], функція g має бути симетричною, і зазвичай вибирається як Гаусівська. Даний процес спирається на отримані нові спостереження, враховуючи їх вплив власне за теоремою Байєса.

3. Обраховується коефіцієнт прийняття β:

$$\beta = \frac{f(y|\alpha^*)f(\alpha^*)}{f(y|\alpha_0)f(\alpha_0)} \tag{3.98}$$

4. Береться випадкове число $0 \le u \le 1$, якщо $u \le \beta$, нова згенерована точка α* приймається і стає α_1, інакше відхиляється і алгоритм залишається на попередній точці.

5. Кроки 2,3,4 повторюються до тих пір, поки не буде згенеровано достатню кількість спостережень.

Після генерації деяка кількість початкових значень відхиляється, через те, що на перших кроках згенеровані значення ще не є стабільними, і алгоритм Метрополіса-Гастінгса стверджує, що ті точки, що залишилися, виходять з апостеріорного розподілу $f(\alpha|y, X)$.

При наступному надходженні нових даних у схемі Байєса, попередні апостеріорні розподіли стають апріорними для наступної ітерації, а розрахунок нових апостеріорних розподілів повторюється відповідно аналогічним чином.

Висновки до розділу

В даному розділі розглянуто методику для виділення та моделювання лінійної складової процесу на базі регресійних моделей, описано підхід з використання отриманої моделі AR для формування оцінок прогнозів, розглянуто і вибрано критерії для оцінки адекватності отриманих моделей, такі як BIC.

Для опису нелінійної складової розглянуто авторегресивний підхід на базі ARIMA, з алгоритмом автоматичної побудови моделі, а також підходи на базі рекурентних (RNN) і згорткових (CNN) нейронних мереж, проаналізовано їх переваги та недоліки.

Регресивні моделі відносно просто розробляються та використовуються як з теоретичної так і з обчислювальної точки зору. Проте даний підхід має недоліки, через те, що він не може враховувати велику кількість зовнішніх факторів, через проблему мультиколінеарності, до того ж, якщо вони впливають нелінійно.

Рекурентні нейронні мережі типу LSTM також вирішують задачу моделювання послідовностей, до того ж враховують нелінійний або комбінований вплив зовнішніх факторів. Проте застосування даного підходу потребує великих обчислювальних витрат; крім того цей підхід не може бути

застосований для аналізу дуже довгих послідовностей, що є проблемою для сучасних задач з використанням великих даних.

Згорткові нейронні мережі CNN підходять для створення моделей комп’ютерного зору, оскільки вони здатні фіксувати найтонші деталі у зображеннях чи навіть 3D об'ємних даних. До того ж, вже існує велика кількість сучасних архітектур для згорткових нейронних мереж, такі як ResNet або DenseNet. Тому можна спробувати застосувати їх для ще більш простих 1D-даних, в яких ми можемо замінити 2D згортки на 1D. Вони демонструють велику ефективність, швидкі, можуть бути оптимізовані, добре функціонують як для розв’язання задач класифікації, так і для регресійного аналізу.

Для нейронних мереж проаналізовано підходи до оптимізації параметрів моделі, запропоновано алгоритм Adam, як найбільш ефективний.

Розглянуто різні підходи для багатокрокового прогнозування за допомогою нейронних мереж, їх переваги та недоліки. Вибрано підхід із застосуванням multi-output мереж як основний.

Розглянуто різні методи комбінування оцінок прогнозів різних моделей, їх переваги та недоліки. Вибрано метод бустінгу як основний для використання в роботі.

Розроблено критеріальну базу на основі RMSE, MAE, RMSLE для оцінки якості прогнозів.

Розглянуто необхідність адаптивної побудови моделей для прогнозування нелінійних нестаціонарних процесів, основні принципи адаптації таких моделей. Розглянуто відповідні підходи до адаптації як для лінійних моделей, так і для нейронних мереж.

Для адаптації лінійних моделей удосконалено метод оцінювання параметрів на основі методу Монте-Карло для марківських ланцюгів.

РОЗДІЛ 4
ЕКСПЕРИМЕНТИ З ПОБУДОВИ МОДЕЛЕЙ ТА ПРОГНОЗІВ ВИБРАНИХ ПРОЦЕСІВ

4.1. Прогнозування попиту на товари Walmart

Для першого експерименту використано вибірку з фактичних даних із продажів одного з 45-ти магазинів “Walmart”, розташованих у різних регіонах [178]. Задача полягає в прогнозуванні щотижневих продажів у різних магазинах і відділах роздрібної торгівлі. Обсяг вибірки – 138 тижнів, з 2010-02-05 по 2012-10-26 з яких для тестування й оцінювання якості прогнозів використано останні 28.

У таблиці 4.1 наведено приклад перших п’яти записів вибірки для першого магазину.

Таблиця 4.1

Перші п’ять записів вибірки даних продажів першого магазину “Walmart”

Date	Weekly_Sales
2010-02-05	24924,5
2010-02-12	46039,49
2010-02-19	41595,55
2010-02-26	19403,54
2010-03-05	21827,9

Побудовано графік часового ряду (рис. 4.1). З графіку можна припустити нестаціонарність ряду, є значні сезонні коливання та можна припустити наявність нелінійних елементів.

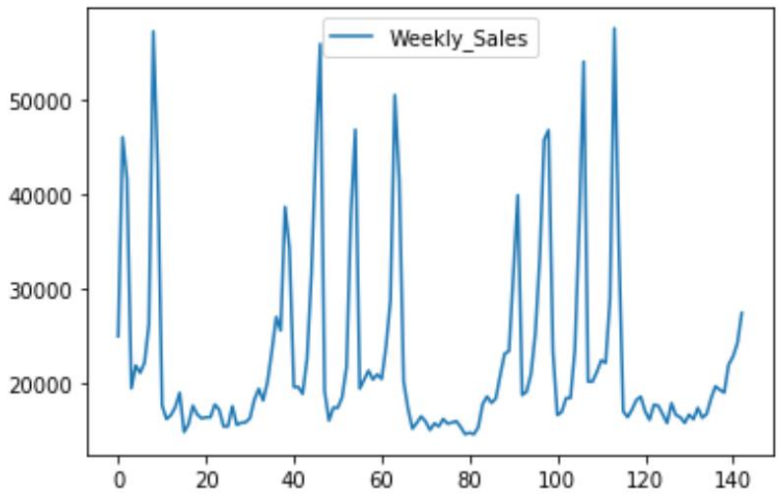


Рис. 4.1 Графік часового ряду з продажів Walmart

Побудувано також гістограму ряду (рис. 4.2), як було рекомендовано для візуального аналізу аномальних значень або викидів.

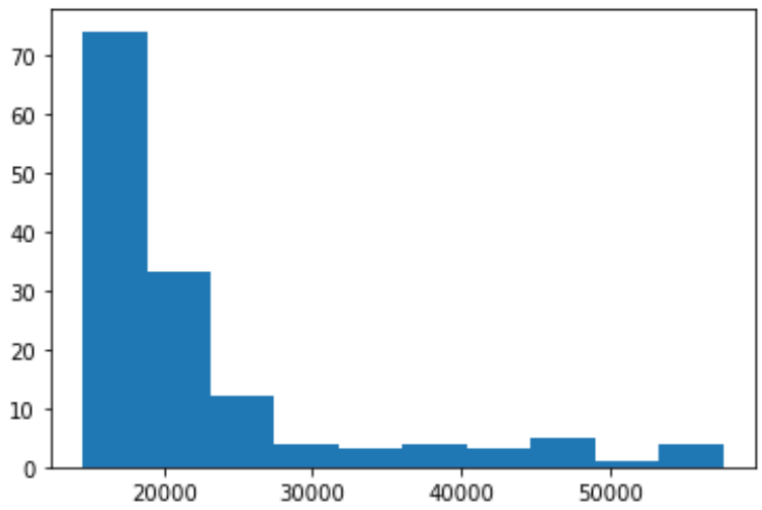


Рис. 4.2 Гістограма ряду з продажів Walmart

Гістограма не є «порожньою», великі значення мають сезонний вплив, що є доволі типовою поведінкою для даних, що описують процеси в роздрібній торгівлі, проте завжди важливо звертати на це увагу і з’ясовувати природу нетипових значень.

Тест KPSS (рис. 4.3) показує на нестаціонарність ряду – тестова статистика перевищує за критичне значення для 1% , що вказує на те, що нульова гіпотеза має бути відхилена. Тобто дані описують нестаціонарний процес.

```
#######################
# KPSS Unit Root Test #
#######################

Test is of type: mu with 4 lags.

Value of test-statistic is: 1.3305

Critical value for a significance level of:
                10pct  5pct 2.5pct  1pct
critical values 0.347 0.463  0.574 0.739
```

Рис. 4.3 Результати тесту KPSS

Однак, тест Уайта (рис. 4.4) показує, що дані однозначно описують нелінійний процес (p-value<<0.05).

```
        White Neural Network Test

data:  ts
X-squared = 48.343, df = 2, p-value = 3.18e-11
```

Рис. 4.4 Результати тесту Уайта

Отже, на етапі первинного аналізу даних можемо зробити висновок щодо нелінійності та нестаціонарності часового ряду, а отже застосовності запропонованої методики.

Для побудови лінійної авторегресійної моделі побудовано ЧАКФ ряду (рис. 4.5), з якої видно, що ряд не є білим шумом і присутні часові залежності в часовому ряді.

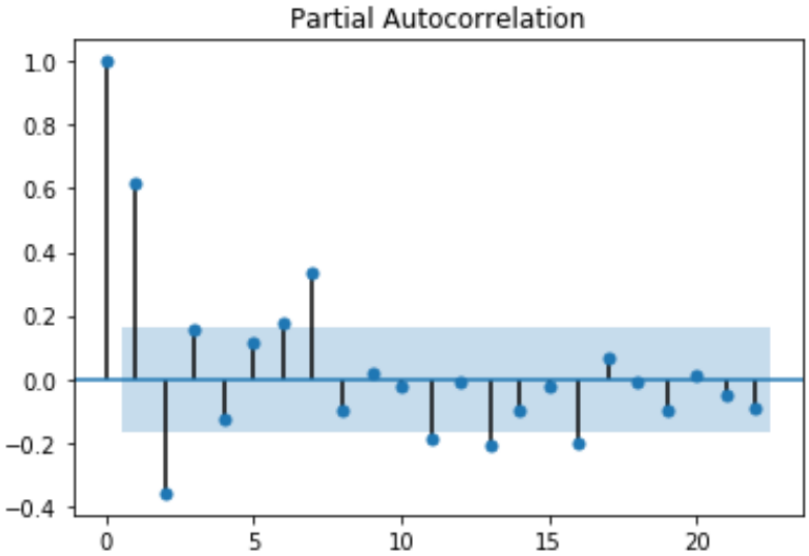


Рис. 4.5 ЧАКФ досліджуваного ряду

Побудуємо моделі АР(1), АР(2), АР(7) за допомогою МНК.

$$AR(1): y(k) = 9047.776613 + 0.611592 * y(k-1) \quad (4.1)$$

$$AR(2): y(k) = 12272.851511 + 0.822791 * y(k-1) - 0.361533 * y(k-2) \quad (4.2)$$

$$AR(7): y(k) = 5658.655374 + 0.824880 * y(k-1) - 0.469637 * y(k-2) + 0.247572 * y(k-3) - 0.166443 * y(k-4) + 0.075790 * y(k-5) - 0.099829 * y(k-6) + 0.339714 * y(k-7) \quad (4.3)$$

Виберемо найкращу модель за BIC (таблиця 4.2).

Таблиця 4.2

Модель за BIC

Модель	**BIC**
AR(1)	17.99218318591743
AR(2)	17.82889937686467
AR(7)	17.85820082699394

Отже, для моделювання лінійної складової процесу обрано модель AR(2).

Відповідно до запропонованої методики, необхідно взяти ряд залишків, в якому залишилася не описана нелінійна частина процесу і застосувати нейронні мережі для її моделювання.

Нейронні мережі вимагають попередньої стандартизації даних, що також була виконана на даному етапі (рис. 4.6). Видно, що дані змінили свою шкалу, середнє значення стало 0, а стандартне відхилення 1.

$$X = \frac{X-\mu}{\sigma} \tag{4.4}$$

```
mean = t_sales_train.mean()
std = t_sales_train.std()

t_sales_train=(t_sales_train-mean)/std
mean, std, t_sales_train.mean(), t_sales_train.std()
```

```
(tensor(23465.4453), tensor(10647.8506), tensor(6.3787e-08), tensor(1.))
```

```
t_sales_train
```

```
tensor([ 0.1370,  2.1201,  1.7027, -0.3815, -0.1538, -0.2275, -0.1248,  0.2596,
         3.1737,  1.8309, -0.5511, -0.6875, -0.6490, -0.5683, -0.4263, -0.8164,
        -0.7405, -0.5548, -0.6412, -0.6808, -0.6703, -0.6698, -0.5425, -0.5930,
        -0.7612, -0.7592, -0.5595, -0.7447, -0.7255, -0.7205, -0.6784, -0.4950,
        -0.3861, -0.5018, -0.3166, -0.0073,  0.3299,  0.1951,  1.4252,  1.0118,
        -0.3678, -0.3675, -0.4363, -0.0890,  0.7543,  2.0142,  3.0490, -0.4077,
        -0.7026, -0.5734, -0.5751, -0.4700, -0.1690,  1.3544,  2.1958, -0.3852,
        -0.2947, -0.2052, -0.2941, -0.2427, -0.2881,  0.0383,  0.4975,  2.5399,
         1.6949, -0.3125, -0.5851, -0.7822, -0.7254, -0.6603, -0.7121, -0.7971,
        -0.7309, -0.7609, -0.6871, -0.7335, -0.7230, -0.7084, -0.7673, -0.8383,
        -0.8242, -0.8385, -0.7690, -0.5371, -0.4630, -0.5265, -0.4816, -0.2506,
        -0.0364, -0.0107,  0.7621,  1.5422, -0.4485, -0.4146, -0.2399,  0.1717,
         0.9242,  2.0950,  2.1904, -0.0108, -0.6478, -0.6171, -0.4790, -0.4778,
         0.0042,  1.2700,  2.8733, -0.3138, -0.3148, -0.2184, -0.1032, -0.1275,
         0.5154,  3.2050])
```

Рис. 4.6 Результат попередньої стандартизації даних

Дані для обробки нейронною мережею перетворені за допомогою підходу з використанням ковзного вікна з кроком в 1 тиждень, горизонтом прогнозування на 1 тиждень вперед, і періодом для пошуку взаємозв’язків в минулому в 4 тижні (рис. 4.7)

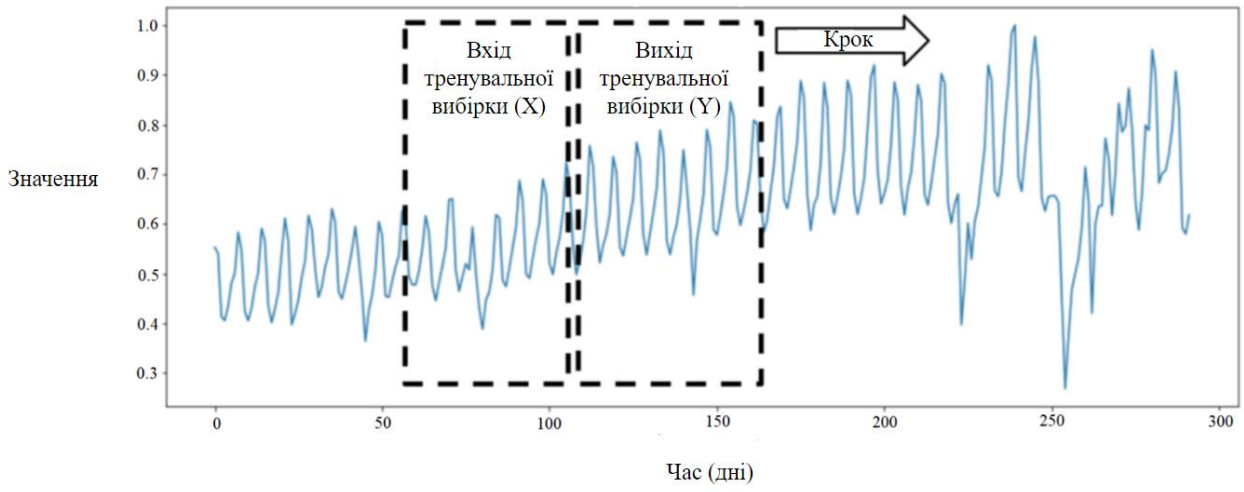


Рис. 4.7 Схема подачі дани для обробки нейронної мережі

Для побудови згорткової нейронної мережі використано одновимірні згортки з одним каналом, розміром ядра 2, і одиничним кроком. Для обрахування прогнозу використано лінійний прошарок (рис. 4.8)

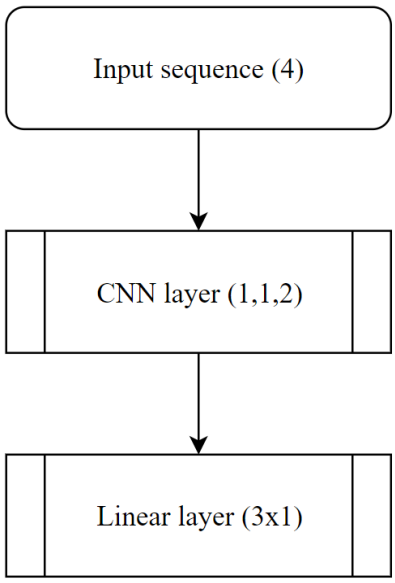


Рис. 4.8 Архітектура згорткової нейронної мережі

Параметри (ваги) моделі показано на рис. 4.9

```
list(learn.model.named_parameters())
```

```
[('cnn_layer.weight',
  Parameter containing:
  tensor([[[ 0.1099, -0.5702]]], requires_grad=True)),
 ('cnn_layer.bias',
  Parameter containing:
  tensor([-0.0252], requires_grad=True)),
 ('linear_layer.weight',
  Parameter containing:
  tensor([[-0.0975, -0.2001, -0.9083]], requires_grad=True)),
 ('linear_layer.bias',
  Parameter containing:
  tensor([-0.2846], requires_grad=True))]
```

Рис. 4.9 Параметри (ваги) моделі

Процес навчання моделі зображено на рис 4.10:

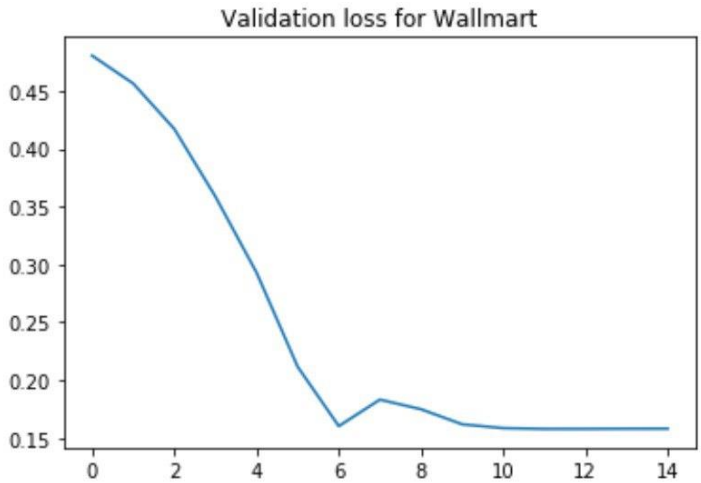


Рис. 4.10 Процес навчання нейронної мережі на даних Wallmart

Виконано прогноз на тестову вибірку, оцінено якість прогнозу за критерієм MAE.

Для порівняльного аналізу також побудований базовий наївний прогноз з використанням середнього значення ряду.

Побудована модель ARIMA за алгоритмом Хайдмана-Хандакара.

В загальному вигляді на основі отриманнях рівнянь повна ARIMA(0,0,2) модель для досліджуваного часового ряду записується як:

$$y(k) = 22070 + 0.8930 * ma(k-1) + 0.2029 * ma(k-2); \qquad (4.5)$$

Для рекурентної нейронної мережі використано 1 рекурентний прошарок розмірності 32 з активаційною функцією гіперболічного тангенсу, а також лінійний прошарок для обрахунку остаточного прогнозу (рис. 4.11)

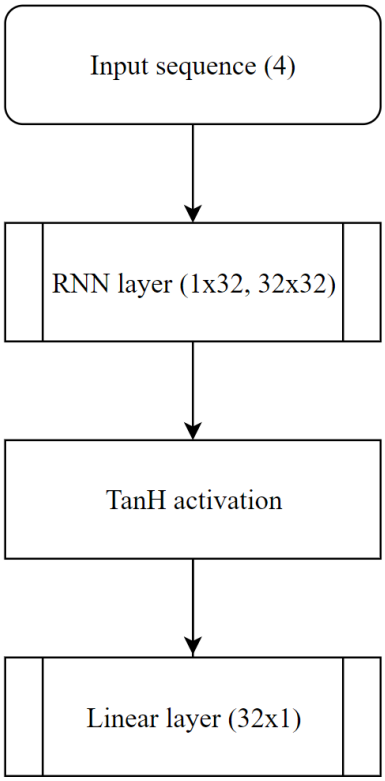


Рис. 4.11 Архітектура рекурентної нейронної мережі

Проведемо експеримент на всій вибірці даних, оптимізуючи параметри моделі так, щоб мінімізувати RMSE; виконаємо прогноз на тестовій вибірці та обрахуємо різні метрики (таблиця 4.3).

Таблиця 4.3

Значення різних метрик для оптимізованої за RMSE моделі

Метрика	Значення
MAE	2376,3
MAPE	24,196
MSE	1,3387e+07
R2	0,95816
RMSE	**3658,8**
RMSLE	3,56
SMAPE	17,048

Проведемо експеримент на всій вибірці даних, оптимізуючи параметри моделі так, щоб мінімізувати MAE; виконаємо прогноз на тестовій вибірці та обрахуємо різні метрики (табл. 4.4).

Таблиця 4.4

Значення різних метрик для оптимізованої за MAE моделі

Метрика	Значення
MAE	**2232**
MAPE	17,182
MSE	1,3847e+07
R2	0,95734
RMSE	3876.47
RMSLE	4,26
RMSPE	6875,9
SMAPE	13,851

MAE стала кращою (2232 < 2376), ніж тоді, коли ми оптимізували модель за RMSE, проте RMSE погіршилась (3876 > 3658). Отже, модель, краща за однією метрикою, не гарантує кращого результату за всіма іншими метриками

– саме тому доволі критичним є розуміння всіх основних критеріїв якості, їх переваг і недоліків для добору правильного для поставленої задачі критерію.

Для даної задачі було вибрано критерій MAE, оскільки вимагається якісне прогнозування в більшості днів, аномальні значення не мають великого значення за постановкою задачі а також попереднім аналізом даних.

Параметри рекурентної нейронної мережі показано на рис. 4.12

```
('rnn_layer.weight_hh_l0',
 Parameter containing:
 tensor([[ 0.1246, -0.1451,  0.0953,  ...,  0.1149,  0.0162, -0.0024],
         [ 0.1589, -0.0453, -0.0761,  ..., -0.0988,  0.1974,  0.0781],
         [-0.0318, -0.0390,  0.1839,  ...,  0.0896,  0.1012,  0.0943],
         ...,
         [ 0.0780, -0.1049, -0.1424,  ...,  0.0639, -0.1442, -0.0732],
         [-0.0655, -0.0085, -0.0206,  ...,  0.1070,  0.0851, -0.0247],
         [ 0.1219,  0.0007, -0.0239,  ...,  0.0412, -0.0308, -0.0788]],
        requires_grad=True)),
('rnn_layer.bias_ih_l0',
 Parameter containing:
 tensor([-0.0261,  0.1155,  0.0595, -0.0756, -0.0160, -0.1627,  0.0436, -0.1482,
          0.0435, -0.1528, -0.1608, -0.0984, -0.1023, -0.1357,  0.0421,  0.1676,
         -0.0754,  0.0190, -0.0008, -0.0369,  0.1281, -0.1210, -0.0115, -0.1283,
         -0.1553,  0.1528,  0.0603,  0.1807,  0.1765, -0.0405,  0.1722, -0.1070],
        requires_grad=True)),
('rnn_layer.bias_hh_l0',
 Parameter containing:
 tensor([-0.0080, -0.0542, -0.1705, -0.0498, -0.0601, -0.0659,  0.0066,  0.1162,
         -0.0072,  0.0766, -0.0541,  0.0640, -0.1046,  0.0750,  0.1635, -0.0731,
         -0.0245, -0.1129,  0.0220,  0.1106,  0.0339,  0.0409,  0.0991,  0.0886,
         -0.1766, -0.1697, -0.0294, -0.1345,  0.0598,  0.1075,  0.0405,  0.1341],
        requires_grad=True)),
('linear_layer.weight',
 Parameter containing:
 tensor([[ 0.1005, -0.1491,  0.0930,  0.1985, -0.1269, -0.0388,  0.1169, -0.1205,
          -0.1460,  0.0336,  0.1775, -0.0460,  0.0234,  0.1795, -0.1025, -0.0629,
           0.0507,  0.0267, -0.1564,  0.1415,  0.0723,  0.0471,  0.0283, -0.0130,
           0.1263,  0.1134, -0.1716,  0.0087,  0.0747,  0.0003, -0.1567, -0.0327]],
        requires_grad=True)),
('linear_layer.bias',
 Parameter containing:
 tensor([-0.0222], requires_grad=True))]
```

Рис. 4.12 Параметри рекурентної нейронної мережі

Бачимо, що рекурентна мережа є доволі важкою з обчислювальної точки зору для побудови і містить численну кількість параметрів.

Порівняльний аналіз якості всіх побудованих моделей і прогнозів показано в таблиці 4.5

Таблиця 4.5

Порівняльний аналіз якості

Модель	**MAE**	**RMSE**	**RMSLE**
Наївна модель	5855.576184210528	6245.05870092761	0.30317543751591
ARIMA(0,0,2)	5248.848827280204	5790.700797017442	0.27395252889559
RNN	3542.8014257812497	4296.246072428907	0.20658189972841
AR-CNN	2232.0122042410717	3876.471396400568	0.17304440444902

Як бачимо, найкращий результат за критерієм MAE показує розроблений підхід AR-CNN.

Побудуємо також графіки прогнозів різними методами для порівняльного аналізу (рис. 4.13):

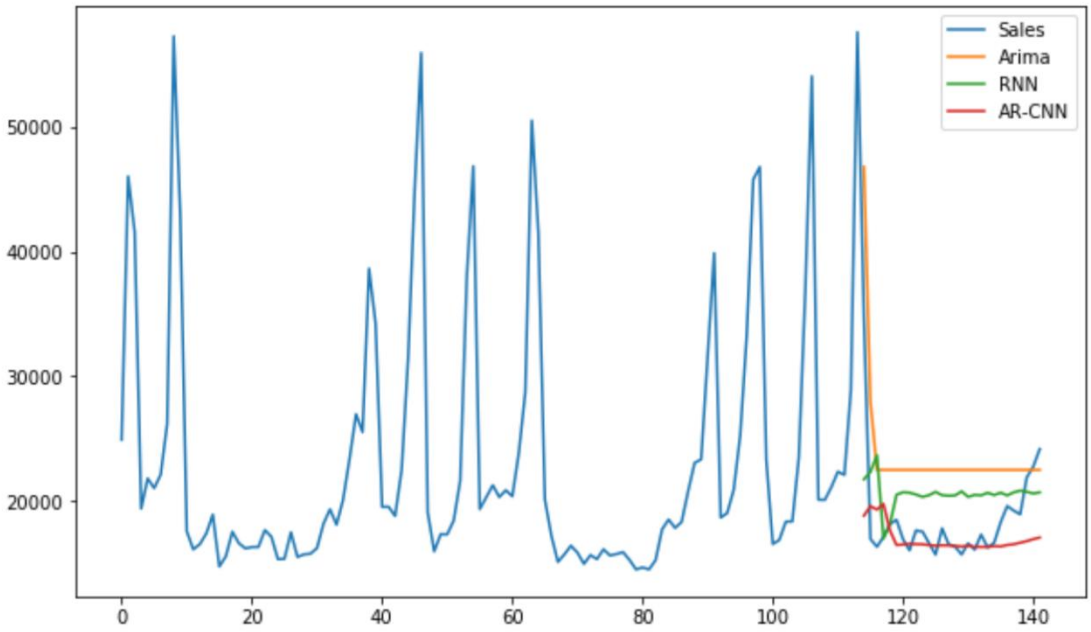


Рис. 4.13 Прогнозування попиту на товари Walmart: порівняльний аналіз

4.2. Побудова моделей з продажу ліків

Для другого експерименту використано фактичні дані з щомісячних продажів протидіабетичних препаратів в Австралії з 1991 по 2008 рік, які

необхідно прогнозувати [65]. Для тестування прогнозів взято останній рік.

У таблиці 4.6 наведено приклад перших п’яти записів вибірки.

Таблиця 4.6

Перші п’ять записів вибірки даних продажів протидіабетичних препаратів в Австралії

Date	Sales
1991-01	3.526591
1991-02	3.180891
1991-03	3.252221
1991-04	3.611003
1991-05	3.565869

Побудовано графік часового ряду (рис. 4.14). З графіку можна припустити нестаціонарність ряду, наявні сезонні коливання та можна припустити наявність нелінійних елементів.

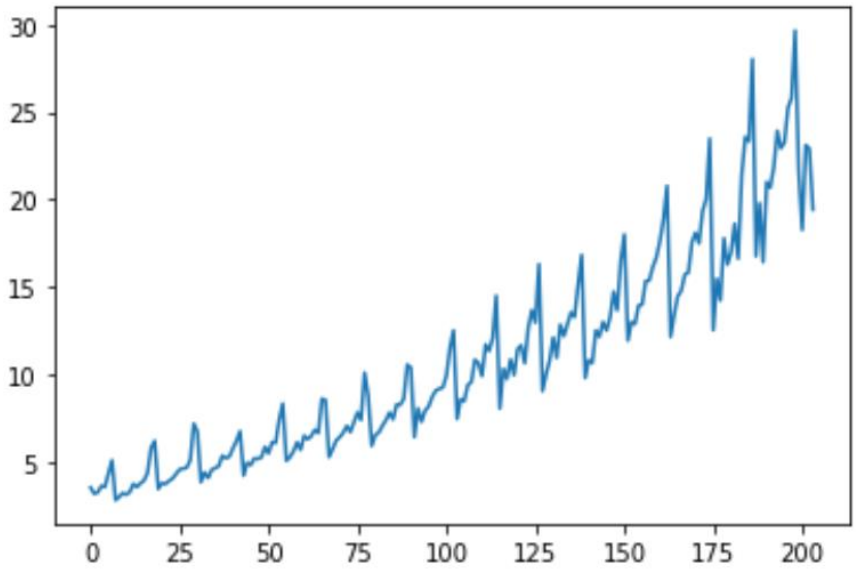


Рис. 4.14 Графік часового ряду з продажу ліків

Побудувано також гістограму ряду (рис. 4.15), як було рекомендовано для візуального аналізу аномальних значень або викидів.

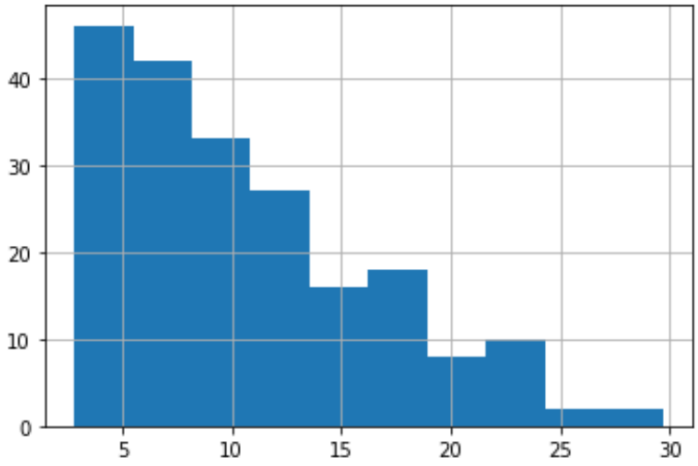


Рис. 4.15 Гістограма ряду з продажів ліків

Гістограма не є «порожньою», аномальних значень немає.

Тест KPSS (рис. 4.16) показує на нестаціонарність ряду – тестова статистика значно перевищує за критичне значення для 1% , що вказує на те, що нульова гіпотеза має бути відхилена. Тобто дані описують нестаціонарний процес.

```
#######################
# KPSS Unit Root Test #
#######################

Test is of type: mu with 4 lags.

Value of test-statistic is: 3.827

Critical value for a significance level of:
                10pct  5pct 2.5pct  1pct
critical values 0.347 0.463  0.574 0.739
```

Рис. 4.16 Результати тесту KPSS

Тест Уайта (рис. 4.17) показує, що дані однозначно описують нелінійний процес (p-value<<0.05).

```
White Neural Network Test

data:  ts
X-squared = 20.222, df = 2, p-value = 4.062e-05
```

Рис. 4.17 Результати тесту Уайта

Отже, на етапі первинного аналізу даних можемо зробити висновок щодо нелінійності та нестаціонарності часового ряду, а отже застосовності запропонованої методики.

Для побудови лінійної авторегресійної моделі побудовано ЧАКФ ряду (рис. 4.18), з якої видно, що ряд не є білим шумом і присутні часові залежності в часовому ряді.

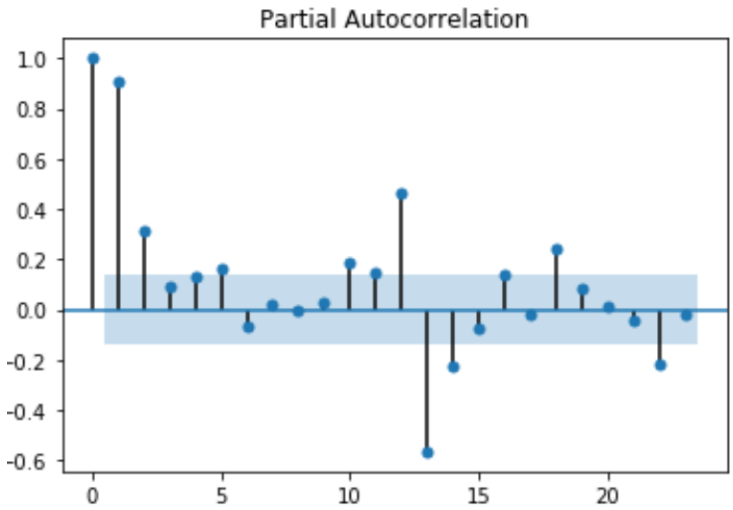


Рис. 4.18 ЧАКФ ряду

Побудуємо моделі АР(1), АР(2) за допомогою МНК.

$$AR(1): y(k) = 0.87986 + 0.91986 * y(k-1)$$
$$AR(2): y(k) = 0.568637 + 0.576595 * y(k-1) + 0.378751 * y(k-2) \quad (4.6)$$

Виберемо найкращу модель за BIC (таблиця 4.7).

Таблиця 4.7

Модель за BIC

Модель	**BIC**
AR(1)	1.570423252686786
AR(2)	1.4504462235108608

Отже, для моделювання лінійної складової процесу обрано модель AR(2).

Відповідно до запропонованої методики, необхідно взяти ряд залишків, в якому залишилася не описана нелінійна частина процесу і застосувати нейронні мережі для її моделювання.

Нейронні мережі вимагають попередньої стандартизації даних, що також була виконана на даному етапі (рис. 4.19). Видно, що дані змінили свою шкалу, середнє значення стало 0, а стандартне відхилення 1.

```
mean = t_sales_train.mean()
std = t_sales_train.std()

t_sales_train=(t_sales_train-mean)/std
mean, std, t_sales_train.mean(), t_sales_train.std()
```

```
(tensor(9.9146), tensor(5.1788), tensor(-3.2907e-08), tensor(1.))
```

```
t_sales_train
```

```
tensor([-1.2335e+00, -1.3003e+00, -1.2865e+00, -1.2172e+00, -1.2259e+00,
        -1.0829e+00, -9.3194e-01, -1.3710e+00, -1.3379e+00, -1.2956e+00,
        -1.3105e+00, -1.2829e+00, -1.1927e+00, -1.2273e+00, -1.1851e+00,
        -1.1567e+00, -1.0675e+00, -7.9248e-01, -7.1881e-01, -1.2481e+00,
        -1.1861e+00, -1.1934e+00, -1.1604e+00, -1.1325e+00, -1.0812e+00,
        -1.0335e+00, -1.0246e+00, -1.0131e+00, -9.3087e-01, -5.2805e-01,
        -6.1465e-01, -1.1727e+00, -1.0660e+00, -1.1275e+00, -1.0377e+00,
        -1.0174e+00, -9.9676e-01, -8.8129e-01, -9.0951e-01, -8.9075e-01,
        -7.9959e-01, -7.1639e-01, -6.1118e-01, -1.1004e+00, -9.5877e-01,
        -9.8316e-01, -9.1139e-01, -9.1601e-01, -8.9942e-01, -7.8384e-01,
        -8.5424e-01, -7.3364e-01, -7.3881e-01, -4.8236e-01, -3.0609e-01,
        -9.3552e-01, -8.9829e-01, -8.3369e-01, -7.3460e-01, -8.1592e-01,
        -6.6189e-01, -6.9786e-01, -6.6563e-01, -5.9589e-01, -6.3056e-01,
        -2.5251e-01, -2.6844e-01, -8.9533e-01, -8.1106e-01, -7.1447e-01,
        -6.7636e-01, -6.2697e-01, -5.5299e-01, -6.1978e-01, -5.1434e-01,
        -4.0452e-01, -4.8593e-01,  3.5063e-02, -2.1552e-01, -7.7168e-01,
        -6.5269e-01, -6.2542e-01, -5.5041e-01, -4.8877e-01, -4.0572e-01,
        -4.7941e-01, -3.1658e-01, -3.1942e-01, -2.5459e-01,  1.2441e-01,
```

Рис. 4.19 Стандартизація даних для нейронної мережі

Дані для обробки нейронною мережею перетворені за допомогою підходу з використанням ковзного вікна з кроком в 1 місяць, горизонтом прогнозування на 1 місяць вперед, і періодом для пошуку взаємозв'язків в минулому в 4 місяці.

Для побудови згорткової нейронної мережі використано одновимірні згортки з одним каналом, розміром ядра 2, і одиничним кроком. Для обрахування прогнозу використано лінійний прошарок (рис. 4.20)

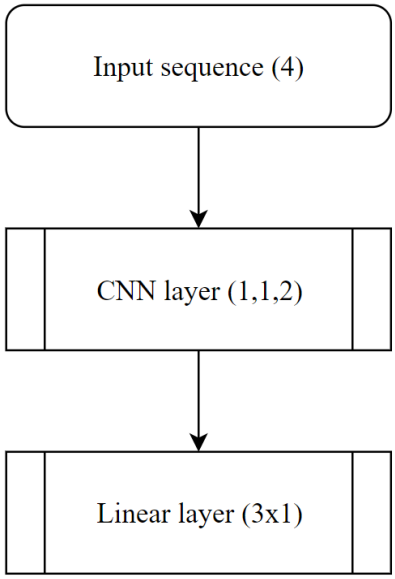


Рис. 4.20 Архітектура згорткової нейронної мережі

Параметри (ваги) моделі показано на рис 4.21

```
list(learn.model.named_parameters())
```

```
[('cnn_layer.weight',
  Parameter containing:
  tensor([[[-0.3440, -0.6689]]], requires_grad=True)),
 ('cnn_layer.bias',
  Parameter containing:
  tensor([-0.5679], requires_grad=True)),
 ('linear_layer.weight',
  Parameter containing:
  tensor([[-0.8722, -0.1501, -0.0680]], requires_grad=True)),
 ('linear_layer.bias',
  Parameter containing:
  tensor([-0.3849], requires_grad=True))]
```

Рис. 4.21 Параметри (ваги) моделі

Процес навчання моделі показано на рис. 4.22:

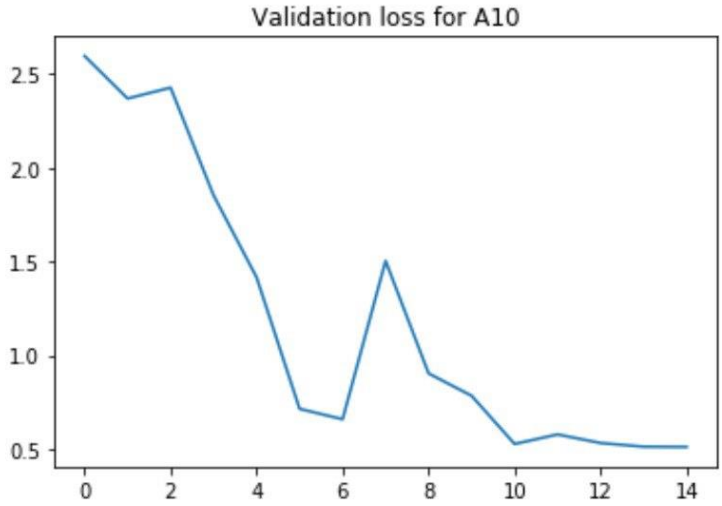


Рис. 4.22 Процес навчання нейронної мережі на даних з продажу ліків

Виконано прогноз на тестову вибірку, оцінено якість прогнозу за критерієм MAE.

Для порівняльного аналізу також побудований базовий наївний прогноз з використанням середнього значення навчальної вибірки.

Побудована модель ARIMA за алгоритмом Хайдмана-Хандакара.

В загальному вигляді на основі отриманнях рівнянь повна ARIMA(2,1,2) модель для досліджуваного часового ряду записується як:

$$y(k) = dy(k) + y(k-1), \tag{4.7}$$

де $dy(k) = 0.1139 - 0.6384 * dy(k-1) + 0.3582 * dy(k-2) + 0.0380 * ma(k-1) - 0.9094 * ma(k-2)$;

Для рекурентної нейронної мережі використано 1 рекурентний прошарок розмірності 32 з активаційною функцією гіперболічного тангенсу, а також лінійний прошарок для обрахунку остаточного прогнозу (рис. 4.23)

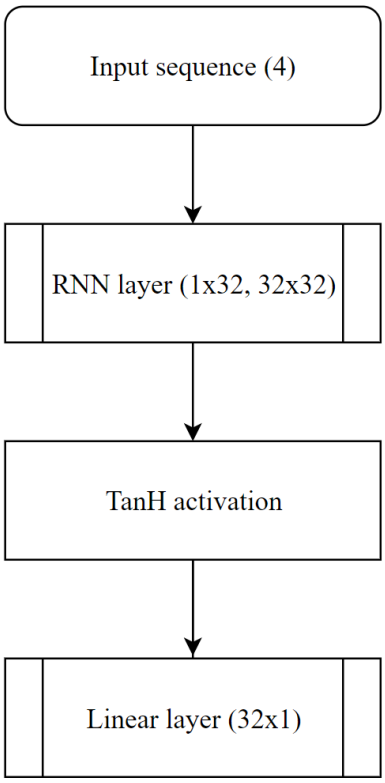


Рис 4.23 Архітектура рекурентної нейронної мережі

Параметри рекурентної нейронної мережі показано на рис. 4.24

```
('rnn_layer.weight_hh_l0',
 Parameter containing:
 tensor([[-0.0973, -0.0310,  0.0269,  ...,  0.1412,  0.0812, -0.0144],
         [-0.0243, -0.0341, -0.0746,  ...,  0.0779,  0.0834, -0.0317],
         [-0.0527, -0.0584,  0.0274,  ..., -0.2226, -0.1428,  0.0064],
         ...,
         [ 0.1087, -0.1207,  0.0119,  ...,  0.1166,  0.0322,  0.1237],
         [-0.0423,  0.0923, -0.0176,  ..., -0.1308,  0.0852, -0.1944],
         [ 0.0713, -0.0286,  0.0187,  ...,  0.0640,  0.0550, -0.1810]],
        requires_grad=True)),
('rnn_layer.bias_ih_l0',
 Parameter containing:
 tensor([-0.0349, -0.1516, -0.1036, -0.0203,  0.1335,  0.1067,  0.1627, -0.0379,
         -0.1667,  0.0374, -0.0506, -0.1180, -0.1238,  0.0544, -0.0591, -0.0894,
         -0.1346,  0.0274,  0.0099, -0.1647,  0.1045, -0.0866, -0.1003,  0.0516,
          0.0610, -0.0994,  0.1087,  0.0282, -0.0317, -0.1106,  0.0552,  0.0338],
        requires_grad=True)),
('rnn_layer.bias_hh_l0',
 Parameter containing:
 tensor([-0.0984,  0.0941, -0.1089, -0.1283, -0.0044, -0.1254, -0.1387,  0.0762,
          0.1527,  0.0929,  0.0157, -0.0092,  0.0746, -0.1476,  0.0085,  0.0002,
          0.0821,  0.1321, -0.1177,  0.0785,  0.1459,  0.0753,  0.0141, -0.0420,
         -0.0250,  0.0396,  0.0067, -0.0852, -0.0549,  0.1076,  0.0143, -0.0488],
        requires_grad=True)),
('linear_layer.weight',
 Parameter containing:
 tensor([[-0.1030, -0.0971,  0.1586, -0.2497,  0.0862,  0.1337,  0.0210,  0.0714,
          -0.1802,  0.0670, -0.1085,  0.0047, -0.0851,  0.0433,  0.0222, -0.1506,
          -0.0704, -0.0297, -0.0941, -0.0305, -0.1969, -0.2064, -0.1161,  0.0612,
           0.0651,  0.1085,  0.0881, -0.0010, -0.0522, -0.2215, -0.0825, -0.0293]],
        requires_grad=True)),
('linear_layer.bias',
 Parameter containing:
 tensor([-0.1369], requires_grad=True))]
```

Рис. 4.24 Параметри рекурентної нейронної мережі

Бачимо, що рекурентна мережа є доволі важкою з обчислювальної точки зору для побудови і містить численну кількість параметрів.

Для даної задачі було вибрано критерій MAE. Порівняльний аналіз якості всіх побудованих моделей і прогнозів показано в таблиці 4.8

Таблиця 4.8

Порівняльний аналіз

Модель	**MAE**	**RMSE**	**RMSLE**
Наївна модель	13.256305034427086	13.55559205561687	0.7967891968211
ARIMA(2,1,2)	3.3637884137976677	4.022812975400008	0.1676518268807
RNN	3.149934460786133	3.755676020440156	0.1551251746462
AR-CNN	2.529168635089925	3.533950120506629	0.1451300990214

Як бачимо, найкращий результат за критерієм MAE показує розроблений підхід AR-CNN.

Побудуємо також графіки прогнозів різними методами для порівняльного аналізу (рис. 4.25):

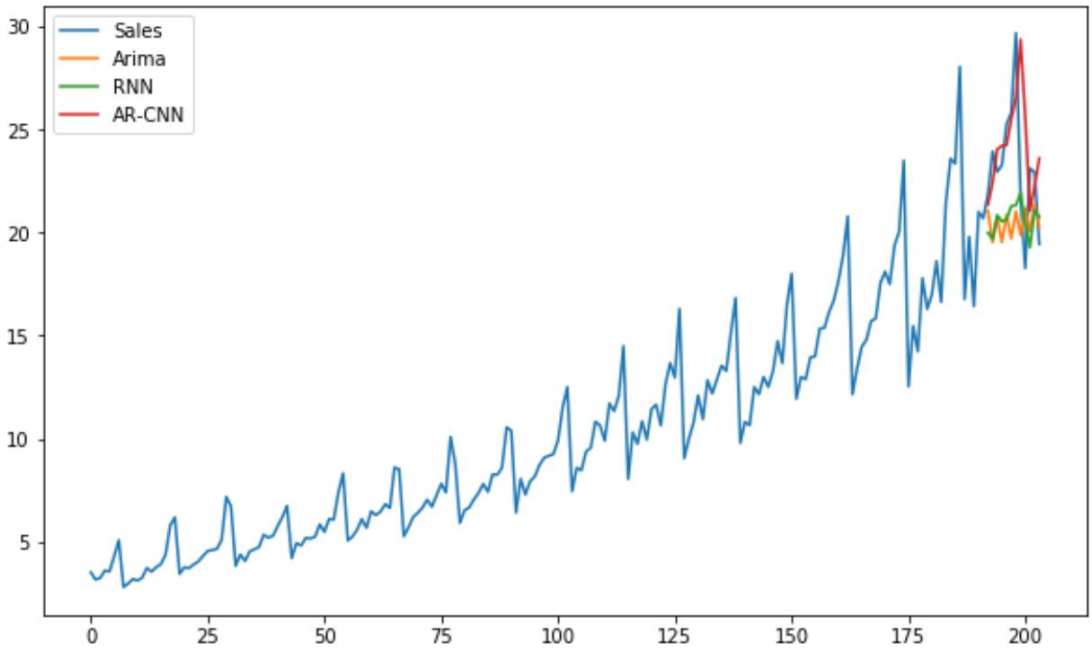


Рис. 4.25 Прогнозування продажів ліків: порівняльний аналіз

4.3. Побудова моделей процесу з продажу палива

Для третього експерименту використано фактичні дані з щотижневих продажів пального в США з 2014 по 2017 рік (в млн барелей), які необхідно прогнозувати [64]. Для тестування якості оцінок прогнозів взято останній рік.

У таблиці 4.9 наведено приклад перших п’яти записів вибірки.

Таблиця 4.9

Перші п’ять записів вибірки даних продажів палива в США

Week	Sales
01-2014	6,621
02-2014	6,433
03-2014	6,582
04-2014	7,224
05-2014	6,875

Побудовано графік часового ряду (рис. 4.26). З графіку можна припустити нелінійність та нестаціонарність ряду.

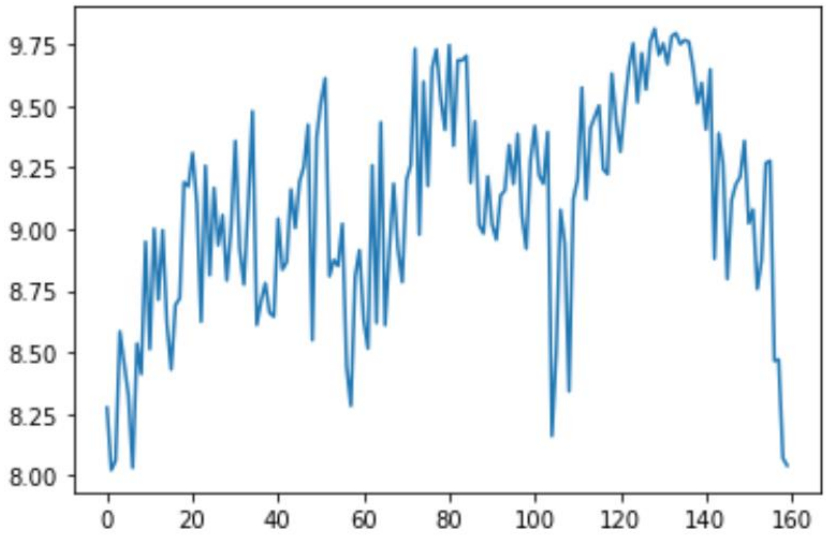


Рис. 4.26 Графік часового ряду з продажу палива

Побудувано також гістограму ряду (рис. 4.27), як було рекомендовано для візуального аналізу аномальних значень або викидів.

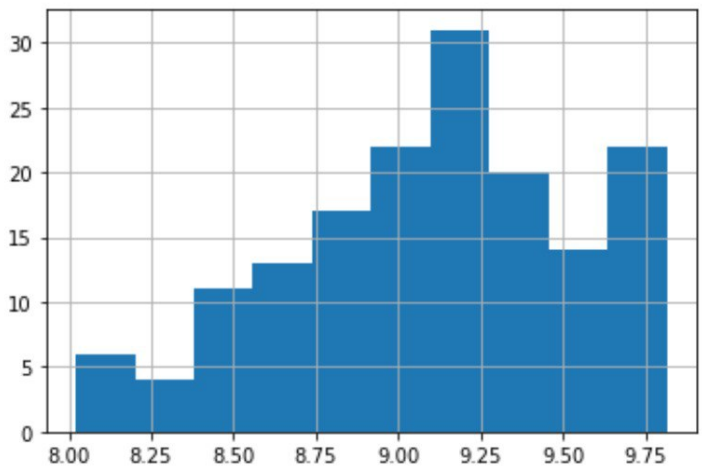


Рис. 4.27 Гістограма ряду з продажів ліків

Гістограма не є «порожньою», аномальних значень немає.

Тест KPSS (рис. 4.28) показує на нестаціонарність ряду – тестова статистика значно перевищує за критичне значення для 1% , що вказує на те, що нульова гіпотеза має бути відхилена. Тобто дані описують нестаціонарний процес.

```
########################
# KPSS Unit Root Test #
########################

Test is of type: mu with 4 lags.

Value of test-statistic is: 1.0845

Critical value for a significance level of:
                10pct  5pct 2.5pct  1pct
critical values 0.347 0.463  0.574 0.739
```

Рис. 4.28 Результати тесту KPSS

Тест Уайта (рис. 4.29) показує, що дані однозначно описують нелінійний процес (p-value<<0.05).

```
White Neural Network Test

data:  df
X-squared = 13.163, df = 2, p-value = 0.001386
```

Рис. 4.29 Результати тесту Уайта

Отже, на етапі первинного аналізу даних можемо зробити висновок щодо нелінійності та нестаціонарності часового ряду, а отже застосовності запропонованої методики.

Для побудови лінійної авторегресійної моделі побудовано ЧАКФ ряду (рис. 4.30), з якої видно, що ряд не є білим шумом і присутні часові залежності в часовому ряді.

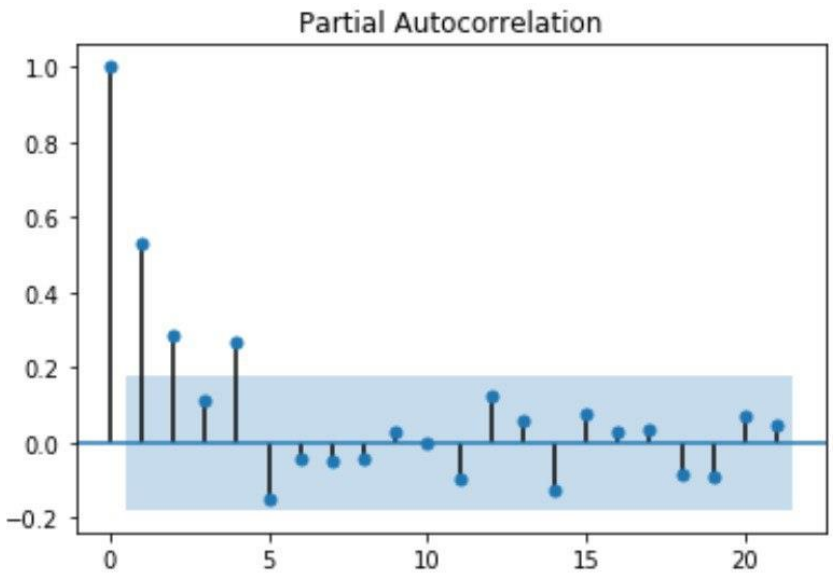


Рис. 4.30 ЧАКФ ряду

Побудуємо моделі АР(1), АР(2), АР(4) за допомогою МНК.

AR(1): $y(k)=4.228194+0.53215*y(k-1)$ (4.8)

AR(2): $y(k)=3.139370+0.339894*y(k-1)+0.313908*y(k-2)$ (4.9)

AR(4):

$y(k)=2.249794+0.232354*y(k-1)+0.178705*y(k-2)+0.058726*y(k-3)+0.283596$

*y(k−4)

(4.10)

Виберемо найкращу модель за BIC (таблиця 4.10).

Таблиця 4.10

Модель за BIC

Модель	**BIC**
AR(1)	-2.068302010479859
AR(2)	-2.156051551097187
AR(4)	-2.1851999756451357

Отже, для моделювання лінійної складової процесу обрано модель AR(2).

Відповідно до запропонованої методики, необхідно взяти ряд залишків, в якому залишилася не описана нелінійна частина процесу і застосувати нейронні мережі для її моделювання.

Нейронні мережі вимагають попередньої стандартизації даних, що також була виконана на даному етапі (рис. 4.31). Видно, що дані змінили свою шкалу, середнє значення стало 0, а стандартне відхилення 1.

```
mean = t_sales_train.mean()
std = t_sales_train.std()

t_sales_train=(t_sales_train-mean)/std
mean, std, t_sales_train.mean(), t_sales_train.std()
```

```
(tensor(9.9146), tensor(5.1788), tensor(-3.2907e-08), tensor(1.))
```

```
t_sales_train
```

```
tensor([-1.2335e+00, -1.3003e+00, -1.2865e+00, -1.2172e+00, -1.2259e+00,
        -1.0829e+00, -9.3194e-01, -1.3710e+00, -1.3379e+00, -1.2956e+00,
        -1.3105e+00, -1.2829e+00, -1.1927e+00, -1.2273e+00, -1.1851e+00,
        -1.1567e+00, -1.0675e+00, -7.9248e-01, -7.1881e-01, -1.2481e+00,
        -1.1861e+00, -1.1934e+00, -1.1604e+00, -1.1325e+00, -1.0812e+00,
        -1.0335e+00, -1.0246e+00, -1.0131e+00, -9.3087e-01, -5.2805e-01,
        -6.1465e-01, -1.1727e+00, -1.0660e+00, -1.1275e+00, -1.0377e+00,
        -1.0174e+00, -9.9676e-01, -8.8129e-01, -9.0951e-01, -8.9075e-01,
        -7.9959e-01, -7.1639e-01, -6.1118e-01, -1.1004e+00, -9.5877e-01,
        -9.8316e-01, -9.1139e-01, -9.1601e-01, -8.9942e-01, -7.8384e-01,
        -8.5424e-01, -7.3364e-01, -7.3881e-01, -4.8236e-01, -3.0609e-01,
        -9.3552e-01, -8.9829e-01, -8.3369e-01, -7.3460e-01, -8.1592e-01,
        -6.6189e-01, -6.9786e-01, -6.6563e-01, -5.9589e-01, -6.3056e-01,
        -2.5251e-01, -2.6844e-01, -8.9533e-01, -8.1106e-01, -7.1447e-01,
        -6.7636e-01, -6.2697e-01, -5.5299e-01, -6.1978e-01, -5.1434e-01,
        -4.0452e-01, -4.8593e-01,  3.5063e-02, -2.1552e-01, -7.7168e-01,
        -6.5269e-01, -6.2542e-01, -5.5041e-01, -4.8877e-01, -4.0572e-01,
        -4.7941e-01, -3.1658e-01, -3.1942e-01, -2.5459e-01,  1.2441e-01,
```

Рис. 4.31 Стандартизація даних для нейронної мережі

Дані для обробки нейронною мережею перетворені за допомогою підходу з використанням ковзного вікна з кроком в 1 тиждень, горизонтом прогнозування на 1 тиждень вперед, і періодом для пошуку взаємозв’язків в минулому в 4 тижні.

Для побудови згорткової нейронної мережі використано одновимірні згортки з одним каналом, розміром ядра 2, і одиничним кроком. Для обрахування прогнозу використано лінійний прошарок (рис. 4.32)

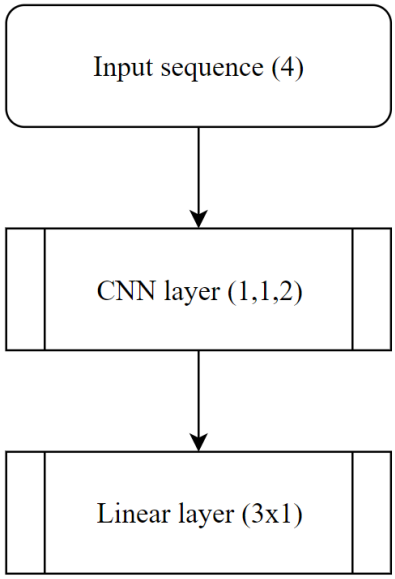


Рис. 4.32 Архітектура згорткової нейронної мережі

Параметри (ваги) моделі показано на рис 4.33

```
list(learn.model.named_parameters())
```

```
[('cnn_layer.weight',
  Parameter containing:
  tensor([[[0.3511, 0.3397]]], requires_grad=True)),
 ('cnn_layer.bias',
  Parameter containing:
  tensor([-0.0818], requires_grad=True)),
 ('linear_layer.weight',
  Parameter containing:
  tensor([[0.1504, 0.6694, 0.2094]], requires_grad=True)),
 ('linear_layer.bias',
  Parameter containing:
  tensor([0.1581], requires_grad=True))]
```

Рис. 4.33 Параметри (ваги) моделі

Процес навчання моделі показано на рис. 4.34:

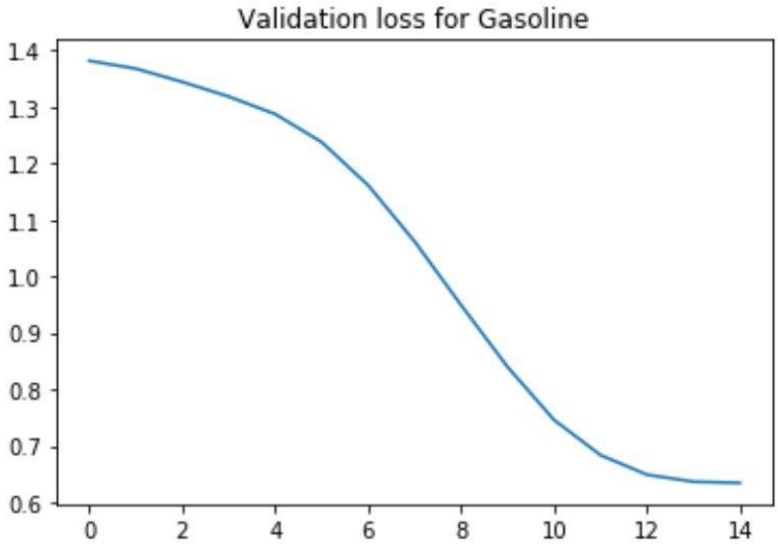


Рис. 4.34 Процес навчання нейронної мережі на даних з продажу палива

Виконано прогноз на тестову вибірку, оцінено якість прогнозу за критерієм MAE.

Для порівняльного аналізу також побудований базовий наївний прогноз з використанням середнього значення навчальної вибірки.

Побудована модель ARIMA за алгоритмом Хайдмана-Хандакара.

В загальному вигляді на основі отриманнях рівнянь повна ARIMA(0,1,1) модель для досліджуваного часового ряду записується як:

$$y(k) = dy(k) + y(k-1), \tag{4.11}$$

де $dy(k) = -0.6602 * ma(k-1)$;

Для рекурентної нейронної мережі використано 1 рекурентний прошарок розмірності 32 з активаційною функцією гіперболічного тангенсу, а також лінійний прошарок для обрахунку остаточного прогнозу (рис. 4.35)

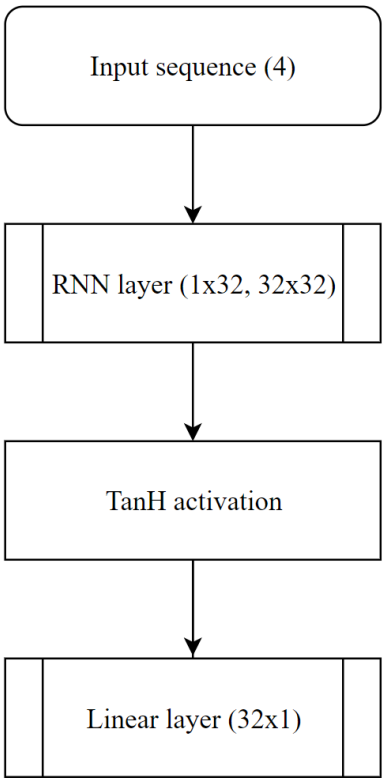


Рис 4.35 Архітектура рекурентної нейронної мережі

Параметри рекурентної нейронної мережі показано на рис. 4.36

```
('rnn_layer.weight_hh_l0',
 Parameter containing:
 tensor([[ 0.1367, -0.0605,  0.1183,  ..., -0.1438, -0.0887,  0.0774],
         [ 0.1262, -0.1001, -0.1254,  ..., -0.1207, -0.0749,  0.1042],
         [ 0.0734, -0.0022, -0.0264,  ...,  0.1864,  0.1040,  0.1707],
         ...,
         [ 0.0367,  0.0824,  0.1279,  ..., -0.1473, -0.1026, -0.1594],
         [ 0.0106, -0.0076,  0.1183,  ..., -0.0254, -0.0933, -0.0720],
         [-0.1145, -0.1915, -0.0995,  ...,  0.1193, -0.1055,  0.0252]],
        requires_grad=True)),
('rnn_layer.bias_ih_l0',
 Parameter containing:
 tensor([-0.0614, -0.1104,  0.0204, -0.0723,  0.0921, -0.0059,  0.0389, -0.0862,
         -0.0470,  0.1304,  0.0209, -0.1005,  0.1774, -0.0153,  0.0288, -0.1650,
         -0.1226, -0.1140,  0.1837, -0.1515,  0.0366,  0.1196,  0.1763,  0.0548,
         -0.1006,  0.0054, -0.0589,  0.0698, -0.0173, -0.0898, -0.0096, -0.1331],
        requires_grad=True)),
('rnn_layer.bias_hh_l0',
 Parameter containing:
 tensor([-0.0744,  0.0793,  0.1687, -0.0527, -0.1061,  0.0558,  0.0961,  0.1083,
         -0.0898, -0.1198, -0.0657, -0.1575,  0.1516, -0.0967,  0.0961, -0.1227,
          0.0745,  0.1418,  0.1131, -0.1859,  0.0386, -0.0880, -0.1230,  0.0763,
         -0.1641,  0.1149,  0.0363, -0.0106,  0.0788,  0.1370, -0.1404, -0.0097],
        requires_grad=True)),
('linear_layer.weight',
 Parameter containing:
 tensor([[-0.1025, -0.1818, -0.0665, -0.0234, -0.0360, -0.1545,  0.0698, -0.0234,
          -0.0988,  0.0523, -0.1290, -0.1467, -0.1157, -0.0996, -0.0542,  0.0974,
           0.1522,  0.1654,  0.0478, -0.0666, -0.0982, -0.0221,  0.0500, -0.0594,
          -0.0276, -0.0849,  0.1985,  0.0550,  0.0454,  0.1285, -0.0595,  0.0310]],
        requires_grad=True)),
```

Рис. 4.36 Параметри рекурентної нейронної мережі

Бачимо, що рекурентна мережа є доволі важкою з обчислювальної точки зору для побудови і містить численну кількість параметрів.

Для даної задачі було вибрано критерій RMSE. Порівняльний аналіз якості всіх побудованих моделей і прогнозів показано в таблиці 4.11

Таблиця 4.11

Порівняльний аналіз

Модель	**MAE**	**RMSE**	**RMSLE**
Наївна модель	0.4942800000000011	0.557488467992339	0.0547966646048
ARIMA(0,1,1)	0.3614000000000001	0.477592409728438	0.0477860973706
RNN	0.2814355491638184	0.335120477354268	0.0334710557494
AR-CNN	0.24118257274627686	0.309391989328102	0.0311970529181

Як бачимо, найкращий результат за критерієм RMSE показує розроблений підхід AR-CNN.

Побудуємо також графіки прогнозів різними методами для порівняльного аналізу (рис. 4.37):

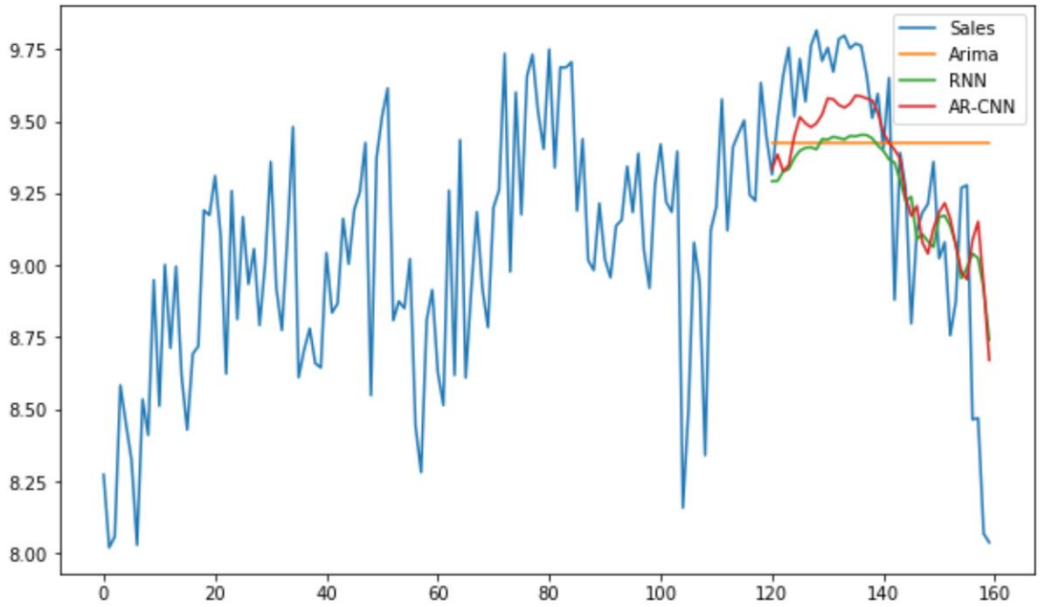


Рис. 4.37 Прогнозування продажів палива: порівняльний аналіз

Висновки до розділу

Розроблену загальну методику для моделювання і прогнозування неленійних нестаціонарних процесів застосовано на реальних статистичних даних. Зібрано статистичні дані та виконано моделювання та прогнозування попиту на споживчі товари магазину Walmart, а також продажів протидіабетичних препаратів в Австралії, продажів пального в США.

Проведений порівняльний аналіз отриманих прогнозів з результатами використання відомих підходів та методів.

В ході дослідження попиту на споживчі товари магазину Walmart, кращим за критерієм MAE виявився розроблений підхід AR-CNN (MAE=2232). Також високу точність показало застосування рекурентних нейронних мереж

(MAE=3542). Однак ця точність досягалася за рахунок великих обчислювальних витрат. Регресійні методи виявились менш точними, однак швидшими для побудови і використання. Серед регресійних моделей найкращою виявилась модель ARIMA (0,0,2) (MAE=5248).

В ході дослідження часового ряду з продажів протидіабетичних препаратів в Австралії, кращим за критерієм MAE також виявився розроблений підхід AR-CNN (MAE=2,53). Також високу точність показало застосування регресійних моделей ARIMA (2,1,2) (MAE=3,36). Рекурентні мережі виявились менш точними за AR-CNN (MAE=3,15), а також більш важкими для побудови і використання.

В ході дослідження даних з продажу палива в США, кращим за критерієм RMSE виявився розроблений підхід AR-CNN (RMSE=0.309). Також високу точність показало застосування рекурентних нейронних мереж (RMSE=0.335). Однак ця точність досягалася за рахунок великих обчислювальних витрат. Регресійні методи виявились менш точними, однак швидшими для побудови і використання. Серед регресійних моделей найкращою виявилась модель ARIMA (0,1,1) (RMSE=0.477).

Загалом, можна відзначити, що розроблений підхід AR-CNN дає високу якість прогнозів при відносній легкості побудови та невеликих обчислювальних втрат.

ВИСНОВКИ

У дисертаційному дослідженні розв’язана задача підвищення адекватності математичних моделей нелінійних нестаціонарних фінансово-економічних процесів та точності відповідних прогнозів, шляхом застосування сучасних методів інтелектуального аналізу даних до статистичних даних, представлених у вигляді часових рядів.

За результатами дисертаційної роботи можна зробити наступні висновки:

1. Сформульовано задачу прогнозування нелінійних нестаціонарних процесів, що є критично важливою для кожного підприємства, що займається торгівлею або інвестиційною діяльністю оскільки є невід’ємною частиною планування закупівлі, використання ресурсів, прийняття стратегічних рішень. Вибрано типи процесів для дослідження, такі як: інтегровані, гетероскедастичні процеси, процеси Леві, процеси зі стохастичним трендом, процеси логістичного типу. Зібрано статистичні дані для експериментів на базі продажів магазину Walmart та продажів протидіабетичних препаратів в Австралії, продажу пального в США.

2. Вперше запропоновано, сформульовано і описано загальну методику та її основні етапи для моделювання і прогнозування нелінійних нестаціонарних процесів на основі математичних моделей інтелектуального аналізу даних з використанням статистичних даних. Побудовано критеріальну базу на базі критеріїв MSE, MAE, RMSLE. Запропоновано методи обробки і підготовки даних до моделювання і прогнозування на базі графічного аналізу, статистичних тестів та процедур цифрової та ймовірнісної фільтраії. Запропонована методика побудови моделей нелінійних процесів, відрізняється застосуванням окремих процедур для оптимізації структури лінійної та нелінійної складових моделі з їх наступним адитивним об’єднанням в єдину

модель, що забезпечує підвищення адекватності моделі і точності прогнозів у цілому. Методика може стати основою для створення відповідної системи підтримки прийняття рішень для аналізу ННП.

3. Вперше запропоновано використання нових для вибраного типу процесів моделей та їх модифікацій на основі методів інтелектуального аналізу даних, а саме згорткових нейронних мереж, для формального опису і прогнозування ННП, побудовано відповідні нові моделі та прогнози.

4. Удосконалено існуючий підхід з використанням класичних статистичних моделей на основі авторегресійних підходів шляхом комбінування їх з методами інтелектуального аналізу даних. За результатами всіх практичних експериментів було показано, що використання розробленого підходу AR-CNN дозволяє отримувати адекватні моделі та точні прогнози при відносній простоті побудові та невеликих обчислювальних втратах.

5. Удосконалено метод оцінювання параметрів лінійних моделей для адаптації на основі методу Монте-Карло для марківських ланцюгів.

Результати роботи впроваджено у навчальний процес інституту прикладного системного аналізу НТУУ «КПІ імені Ігоря Сікорського».

Всі результати дисертаційного дослідження доведено до високого практичного інженерного рівня та впроваджено у АТ «ПУМБ» з метою покращення процесів прийняття рішень та побудови стратегій розвитку. Створено систему підтримки прийняття рішень, що побудована на основі оцінок прогнозів за допомогою розробленої загальної методики з використанням методів інтелектуального аналізу даних.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Белас О.М., Бідюк. П.І., Белас А.О. Порівняльний аналіз авторегресійних підходів та рекурентних нейронних мереж для моделювання і прогнозування нелінійних нестаціонарних процесів. *Information Technology and Security.* 2019 January-June.Vol. 7, No 1. P. 91-99.
2. Белас О.М., Белас А.О. Загальна методика прогнозування нелінійних нестаціонарних процесів на основі математичних моделей з використанням статистичних даних. *Системні дослідження та інформаційні технології.* 2021. № 1. С. 79-86.
3. Danilov V., Belas A., Bidyuk P., Gozhyj O., Kalinina I., Jirov O. Adaptive forecasting and risk estimation. *Системні дослідження та інформаційні технології.* 2020. № 1. С. 34-53.
4. Бідюк П.І., Белас А.О. Вибір критерію якості для оцінювання прогнозів нелінійних нестаціонарних процесів. *Наукові вісті КПІ.* 2021. № 2. С. 38-45.
5. Belas A., Bidyuk P. Convolutional neural networks for modeling and forecasting nonlinear nonstationary processes. *ScienceRise.* 2021. No 3. P. 12-20.
6. Belas A. O., Gavrylenko O. V., Bidyuk P. I. Bayesian modeling and forecasting for nonlinear nonstationary processes. *Проблеми інфокомунікацій*: матеріали Третьої всеукраїнської науково-технічної конференції, м.Полтава, Мінськ 12-2019.
7. Белас А.О., Бідюк П.І. Застосування фільтра Калмана для моделювання та прогнозування нелінійних нестаціонарних процесів. *Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів управління* : тези доп. 10-ї міжнар. наук.-техн. конф.

(Баку–Харків–Жиліна, 9-10 квітня 2020 р.). Т. 2, секції 3, 4. Військ. акад. збройних сил Азербайджанської Республіки (та ін.). Харків : Петров В. В., 2020. 92 с.

8. Белас А.О. Згорткові нейронні мережі для моделювання та прогнозування нелінійних нестаціонарних процесів. *Радіоелектроніка і молодь у XXI столітті*: матеріали 24-й Міжнародного молодіжного форуму, м.Харків, 7 – 9 квітня 2020 р., секція 6. Зб. матеріалів 24-го Міжнар. молодіж. форуму, Харків: ХНУРЕ, 2020. Т. 11. 68 с.
9. P. Ni, C. Zhang and Y. Ji A hybrid method for short-term sensor data forecasting in Internet of Things. 2014 11th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), 2014. P.369-373. doi: 10.1109/FSKD.2014.6980862.
10. Концеба С.М., Ліщук Р.І., Скуртол С.Д., Родащук Г.Ю., Васильченко І.П. Прогнозування обсягів виробництва сільськогосподарської продукції за допомогою методів інтелектуального аналізу даних. *Системні дослідження та інформаційні технології*. 2021. № 1. С. 87-97.
11. Dubas M. Expert systems in industrial practice: advantages and drawbacks. *Expert systems*. 1990. T. 7. № 3. C. 150-156.
12. Барсегян А.А. Анализ данных и процессов: учебное пособие. / Барсегян А.А., Куприянов М.С., Холод И.И., Тесс М.Д., Елизаров С.И. 3-е изд., перераб. и доп. СПб: БХВ-Петербург, 2009. 512 с.
13. Maslianko P., Sielskyi Y. Data Science - definition and structural representation. *System research and information technologies*. 2021. No 1. P. 61-78.
14. Объем данных/информации, созданных, собранных, скопированных и потребленных по всему миру с 2010 по 2025 год. URL: https://www.statista.com/statistics/871513/worldwide-data-created/.

15. Зайченко Ю.П. Основи проектування інтелектуальних систем. Київ: Видавничий дім “Слово”, 2004. 352с.
16. Samuel A. L. Artificial intelligence: a frontier of automation. *The Annals of the American Academy of Political and Social Science*. 1962. Vol. 340, No 1. P. 10-20.
17. Box J., Jenkins G. Time Series Analysis. Moscow, USSR: Mir, 1974.
18. Azevedo A., Santos M. KDD, SEMMA and CRISP-DM: a parallel overview. *in Proc. of the IADIS European Conference on Data Mining.* 2008. P. 182-185.
19. SEMMA in SAS Enterprise Miner. URL: https://web.archive.org/web/20120308165638/http://www.sas.com/offices/europe/uk/technologies/analytics/datamining/miner/semma.html/. Accessed on: Sep. 12, 2020.
20. Shumway R., Stoffer D. Time Series Analysis and Its Applications. New York, USA: Springer, 2011. doi: 10.1007/978-1-4757-3261-0.
21. Schapire R. E. The boosting approach to machine learning: An overview. Nonlinear estimation and classification, 2003. P. 149-171.
22. SAS: Analytics, Artificial Intelligence and Data Management | SAS. URL: https://www.sas.com/ru_ua/home.html.
23. SAS: Credit Scoring. URL: https://www.sas.com/en_us/software/credit-scoring.html.
24. SAS: Intelligent Planning Suite | SAS. URL: https://www.sas.com/ru_ua/software/intelligent-planning.html.
25. Data Mining Software, Model Development and Deployment, SAS Enterprise Miner. URL: https://www.sas.com/ru_ua/software/enterprise-miner.html.
26. The R Project for Statistical Computing. URL: https://www.r-project.org/about.html.

27. CRAN - Package Rdatasets. URL: https://vincentarelbundock.github.io/Rdatasets/datasets.html.
28. CRAN - Package readxl. URL: lhttps://cran.r-project.org/web/packages/readxl/index.html.
29. CRAN - Package DBI. URL: https://cran.r-project.org/web/packages/DBI/index.html.
30. Create Elegant Data Visualisations Using the Grammar of Graphics • ggplot2. URL: https://ggplot2.tidyverse.org/.
31. Tidyverse. URL: https://www.tidyverse.org/.
32. A Grammar of Data Manipulation • dplyr. URL: https://dplyr.tidyverse.org/.
33. CRAN - Package forecast. URL: https://cran.r-project.org/web/packages/forecast/index.html.
34. A Short Introduction to the caret Package. URL: https://cran.r-project.org/web/packages/caret/vignettes/caret.html.
35. PyTorch. URL: https://pytorch.org/.
36. TensorFlow. URL: https://www.tensorflow.org/.
37. CRAN - Package MLmetrics. URL: https://cran.r-project.org/web/packages/MLmetrics/index.html.
38. Shiny. URL: https://shiny.rstudio.com/.
39. About Python™ | Python.org. URL: https://www.python.org/about/.
40. Pandas documentation - pandas 1.4.0 documentation. URL: https://pandas.pydata.org/pandas-docs/stable/.
41. Matplotlib - Visualization with Python. URL: https://matplotlib.org/.
42. Seaborn: statistical data visualization - seaborn 0.11.2 documentation. URL: https://seaborn.pydata.org/.
43. SciPy. URL: https://www.scipy.org/.
44. Harris C. R. et al. Array programming with NumPy. *Nature*. 2020. Vol. 585, No 7825. P. 357-362.

45. Scikit-Learn: machine learning in Python - scikit-learn 1.0.2 documentation. URL: https://scikit-learn.org/stable/.
46. Introduction - statsmodels. URL: https://www.statsmodels.org/stable/index.html.
47. TIOBE - The Software Quality Company. URL: https://www.tiobe.com/tiobe-index/.
48. Бідюк П. І., Романенко В. Д., Тимощук О. Л. Аналіз часових рядів. Київ: Політехніка, 2010. 317 с.
49. Tsay R. S. Analysis of Financial Time Series. NJ.: Wiley, 2010. 716 p.
50. Наконечний С. І., Терещенко Т. О., Романюк Т. П. Економетрія: навч. посіб. 2-ге вид., перероб. та доп. Київ: КНЕУ, 2000. 296 с.
51. Bisaglia L., Gerolimetto M. Testing for (non) linearity in economic time series: a Monte Carlo comparison. *Working Paper Series, Department of Statistical Sciences, University of Padua*. 2014. No 3.
52. Granger C. W. J., and Andersen A. P. An Introduction to Bilinear Time Series Models. Vandenhoek and Ruprecht: Gottingen, 1978.
53. Tong H. Non-Linear Time Series: A Dynamical System Approach. Oxford University Press, 1990.
54. Priestley M. B. State-dependent models: a general approach to nonlinear time series analysis *Journal of Time Series Analysis.* 1980. No 1. P.47–71.
55. Hamilton J. D. A New Approach to the Economic Analysis of Nonstationary Time Series and the Business Cycle. *Econometrica: Journal of the econometric society*. 1989. P. 357-384.
56. Carlin B. P., Polson N. G., Stoffer D. S. A Monte Carlo Approach to Nonnormal and Nonlinear State Space Modeling. *Journal of the American Statistical Association*. 1992. Vol. 87, No 418. P. 493–500.
57. Chen R., Tsay R. S. Functional-coefficient autoregressive models. *Journal of the American Statistical Association*. 1993. No 88. P.298–308.

58. Chen R., Tsay R. S. Nonlinear additive ARX models. *Journal of the American Statistical Association*. 1993. No 88. P. 955–967.
59. Lewis P. A. W., Stevens J. G. Nonlinear Modeling of Time Series Using Multi-variate Adaptive Regression Spline (MARS). *Journal of the American Statistical Association*, 1991. No 86. P. 864–877.
60. Колмогоров А.Н. О представлении непрерывных функций нескольких переменных в виде суперпозиций непрерывных функций одной переменной и сложения. *Доклады Академии наук*. 1957. Т. 114, № 5. С. 953–956.
61. Hornik K., Stinchcombe M., White H. Multilayer Feedforward Networks are Universal Approximators. *Neural Networks*. 1989. Vol. 2, No 5. P. 359–366.
62. Ken-Iti S. Lévy processes and infinitely divisible distributions. Cambridge university press, 1999. P. 31–68.
63. Economy passenger load on Ansett Airlines URL: https://cran.r-project.org/package=fpp2.
64. Hyndman R., Athanasopoulos G. Forecasting Principles and Practice. OTexts, 2018. P. 292.
65. Monthly government expenditure (millions of dollars)for products falling under ATC code A10 as recorded by the Australian Health Insurance Commission. July 1991 - June 2008. URL: https://cran.r-project.org/web/packages/fpp2/index.htmlп
66. Keenan D.M. A Tukey nonadditivity type test for time series nonlinearity. *Biometrika*. 1985. Vol. 72, No 1. P. 39-44.
67. Tsay R.S. Nonlinearity tests for times series. *Biometrika*.1986. Vol. 73, No 2. P. 461-466.
68. Ramsey J.B. Tests for specification errors in classical linear least squares regression analysis. *Journal of the Royal Statistical Society: Series B (Methodological)*. 1969. Vol. 31, No 2. P. 350-371.

69. White H. Specification testing in dynamic models. *Advances in econometrics, Fifth world congress*. New York, NY, USA: Cambridge University Presse. 1987. Vol. 1. P. l-58.
70. McLeod A.I., Li W.K. Diagnostic checking ARMA time series models using squared residual autocorrelations. *Journal of Time Series Analysis*. 1983. Vol. 4, No 4. P. 169-176.
71. Broock W. A., Scheinkman J. A., Dechert W. D., LeBaron B. A test for independence based on the correlation dimension. *Econometric reviews*. 1996. Vol. 15, No 3. P. 197-235.
72. Hannan E.J. The estimation of the order of an ARMA process. *The Annals of Statistics*. 1980. P. 1071-1081.
73. Hinich M.J. Testing for gaussianity and linearity of a stationary time series. *Journal of Time Series Analysis*. 1982. Vol. 3, No 3. P. 169-l76.
74. Lee T. H., White H., Granger C. W. J. Testing for neglected nonlinearity in time series models: A comparison of neural network methods and alternative tests. *Journal of Econometrics*. 1993. Vol. 56, No 3. P. 269-290.
75. De R., Hinton G. E., Williams R. J. Learning internal representations by error propagation. *Parallel distributed processing: Explorations in the microstructure of cognition*. Vol. 1. Foundations. MIT Press, Cambridge, MA. 1986. P. 318-362.
76. White H. Some asymptotic results for learning in single hidden-layer feedforward network models. *Journal of the American Statistical Association*. 1989. Vol. 84, No 408. P. 1003-1013.
77. White H. Connectionist nonparametric regression: Multilayer feedforward networks can learn arbitrary mappings. *Neural networks*. 1990. Vol. 3, No 5. P. 535-549.

78. Hornik K., Stinchcombe M., White H. Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks. *Neural networks*. 1990. Vol. 3, No 5. P. 551-560.
79. Stinchombe M, White H. Universal approximation using feed-forward networks with nonsigmoid hidden layer activation functions. *Proc. IJCNN. Washington, DC,* 1989. P. 613-618.
80. Lapedes A., Farber R. Nonlinear Signal Processing Using Neural Networks. *IEEE conference on Neural Information Processing Systems-Natural and Synthetic,* 1987.
81. Gallant A. R., White H. On learning the derivatives of an unknown mapping with multilayer feedforward networks. *Neural Networks*. 1992. Vol. 5, No 1. P. 129-138.
82. Friedman J. H., Stuetzle W. Projection pursuit regression. *Journal of the American statistical Association.* 1981. Vol. 76, No 376. P. 817-823.
83. Diaconis P., Shahshahani M. On nonlinear functions of linear combinations. *SIAM Journal on Scientific and Statistical Computing*. 1984. Vol. 5, No 1. P. 175-191.
84. Huber P. J. Projection pursuit. *The annals of Statistics*. 1985. No 13. P. 435-475.
85. Jones L. K. A simple lemma on greedy approximation in Hilbert space and convergence rates for projection pursuit regression and neural network training. *The annals of Statistics*. 1992. P. 608-613.
86. Dickey D. A. Estimation and Hypothesis Testing in Nonstationary Time Series. Iowa State University, 1976.
87. Dickey D. A., Fuller W. A. Distribution of the estimators for autoregressive time series with a unit root. *Journal of the American statistical association.* 1979. Vol. 74, No 366a. P. 427-431.

88. Nelson C. R., Plosser C. R. Trends and random walks in macroeconmic time series: some evidence and implications. *Journal of monetary economics*. 1982. Vol 10, No 2. P. 139-162.
89. Said S. E., Dickey D. A. Testing for unit roots in autoregressive-moving average models of unknown order. *Biometrika*. 1984. Vol. 71, No 3. P. 599-607.
90. Phillips P. C. B. Time series regression with a unit root. *Econometrica: Journal of the Econometric Society*. 1987. No 55. P. 277-301.
91. Phillips P. C. B., Perron P. Testing for a unit root in time series regression. *Biometrika*. 1988. Vol. 75, No 2. P. 335-346.
92. Delong D. N., Nankervis J. C., Savin N. E., Whiteman C. H. Integrated versus trend stationarity in macroeconomic time series. Department of Economics, University of Iowa, Iowa City, IA. Working paper, 1989. No 89-99.
93. Diebold F.X., Rudebush G. D. On the power of Dickey-Fuller tests against fractional alternatives. *Economics letters*. 1991. Vol. 35. P. 155-160.
94. Kwiatkowski D., Phillips P. C., Schmidt P., Shin Y. Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root?.*Journal of econometrics.*1992. Vol. 54, No 1-3. P. 159-178.
95. Nabeya S., Tanaka K. Asymptotic theory of a test for the constancy of regression coefficients against the random walk alternative. *The Annals of Statistics*. 1988. No 16. P. 218-235.
96. Irwin J. O. On a criterion for the rejection of outlying observations. *Biometrika*. 1925,. Vol. 17. P. 238-250.
97. Grubbs F. E. Sample criteria for testing outlying observations. *The Annals of Mathematical Statistics.* 1950. Vol. 21. P. 27-58.

98. Grubbs F. E. Procedures for detecting outlying observations in samples. *Technometrics*. 1969. Vol. 11, No 1. P. 1-21. .

99. Rosner B. On the detection of many outliers. *Technometrics*. 1975. Vol. 17, No 2. P. 221-227.

100. Rosner, B. Percentage points for a generalized ESD many-outlier procedure. *Technometrics.* 1983. Vol. 25, No 2. P. 165-172.

101. Cleveland W. S. Robust locally weighted regression and smoothing scatterplots. *Journal of the American statistical association.* 1979. Vol. 74, No 368.P. 829-836.

102. Cleveland R. B., Cleveland W. S., McRae J. E., Terpenning I. STL: A seasonal-trend decomposition. *Journal of Official Statistics*. 1990. Vol. 6, No 1. P. 3-73.

103. Tsay R. S. Time series model specification in the presence of outliers. *Journal of the American Statistical Association*. 1986. Vol. 81, No 393. P. 132-141.

104. Chen C., Liu L. M. Joint estimation of model parameters and outlier effects in time series. *Journal of the American Statistical Association*. 1993. Vol. 88, No 421. P. 284-297.

105. Huber P. J., Ronchetti E. M. Robust Statistics. Hoboken, N.J.: Wiley, 1981. 384 p.

106. Hampel F. R., Ronchetti E. M., Rousseeuw P. J., Stahel W. A. Robust statistics: the approach based on influence functions. John Wiley & Sons. 2011. Vol. 196.

107. Tukey, J. W. Exploratory data analysis. Pearson, London, 1977, 712 p.

108. Hastings Jr. C., Mosteller F., Tukey J. W., Winsor C. P. Low moments for small samples: a comparative study of order statistics. The Annals of Mathematical Statistics. 1947. Vol. 18, No 3. P. 413-426. doi:10.1214/aoms/1177730388.

109. Dempster A. P., Laird N. M., Rubin D. B. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*. 1977. Vol. 39, No 1. P. 1-22.

110. Ford B.L. An overview of hot-deck procedures. In: Madow WG, Olkin I, Rubin DB, editors. *Incomplete Data in Sample Surveys*. Vol. 2. New York: Academic Press; 1983. P. 185–207.

111. Rubin D. B. Multiple imputation for nonresponse in surveys. John Wiley & Sons, 2004. Vol. 81.

112. Vacek P., Ashikaga T. An examination of the nearest neighbor rule for imputing missing values. Proceedings of the Statistical Computing Section, American Statistical Association. 1980. P. 326–331.

113. Kalman R. E. A new approach to linear filtering and prediction problems. Journal of Basic Engineering. 1960. No 82. P. 35–45. doi:10.1115/1.3662552. S2CID 1242324.

114. Згуровский М. З., Подладчиков В.Н. Аналитические методы калмановской фильтрации для систем с априорной неопределенностью. Київ: Наукова думка, 1995. 285с.

115. Humpherys J., Redd P., West J. A fresh look at the Kalman filter. *SIAM review*. 2012. Vol. 54, No 4. P. 801-823. doi:10.1137/100799666.

116. Holt C. E. Forecasting seasonals and trends by exponentially weighted averages (ONR Memorandum No. 52). *Carnegie Institute of Technology, Pittsburgh USA*. 1957. Vol. 10.

117. Федосеев В. В., Гармаш А. Н., Дайитбегов Д. М., Орлова И. В., Половников, В. А. Экономико-математические методы и прикладные модели: учебное издание / ред. Федосеев В. В. Москва: ЮНИТИ, 1999. 391 с.

118. Brown R. G. Statistical forecasting for inventory control. McGraw/Hill, 1959. 233 p.

119. Winters P. R. Forecasting sales by exponentially weighted moving averages. *Management science*. 1960. Vol. 6, No 3. P. 324-342.

120. Pegels C. C. Exponential forecasting: Some new variations. *Management Science*. 1969. Vol. 15, No 5. P. 311-315.

121. Gardner Jr E. S. Exponential smoothing: The state of the art. *Journal of forecasting*. 1985. Vol. 4, No 1. P. 1-28.

122. Taylor J. W. Exponential smoothing with a damped multiplicative trend. *International journal of Forecasting*. 2003. Vol. 19, No 4. P. 715-725.

123. James G., Witten D., Hastie T., Tibshirani R. An introduction to statistical learning. *New York : springer*, 2013. Vol. 112. P. 18.

124. Goldberger A. S. Classical Linear Regression. Econometric Theory. New York: John Wiley & Sons, 1964. 158 P.

125. Shumway R. H., Stoffer D. S., Stoffer D. S. Time series analysis and its applications. New York : springer, 2000. Vol. 3.

126. Гулфеллоу Я., Бенджио И., Курвилль А. Глубокое обучение .М.: ДМК Пресс, 2018. 652 с.

127. Box G. E. P., Pierce D. A. Distribution of Residual Autocorrelations in Autoregressive-Integrated Moving Average Time Series Models. *Journal of the American Statistical Association*. 1970. Vol.65, No 332, P. 1509-1526. doi: 10.1080/01621459.1970.10481180.

128. Durbin, J., Watson, G. S. Testing for Serial Correlation in Least Squares Regression, I. *Biometrika*. 1950. No 37 (3–4) .P. 409–428. doi:10.1093/biomet/37.3-4.409. JSTOR 2332391.

129. Durbin J., Watson G. S. Testing for Serial Correlation in Least Squares Regression, II. *Biometrika*. 1951. No 38 (1–2). P.159–179. doi:10.1093/biomet/38.1-2.159. JSTOR 2332325.

130. Walpole Ronald E. Probability & statistics for engineers & scientists. Myers, H. Raymond. (7th ed.). New Delhi: Pearson, 2006. ISBN 81-7758-404-9. OCLC 818811849.

131. Hyndman R. J., Khandakar Y. Automatic time series forecasting: the forecast package for R. *Journal of statistical software*. 2008. Vol. 27, No 3. P. 1-22.

132. McCulloch W. S., Pitts W. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*. 1943. Vol. 5, No 4. P. 115-133.

133. Rosenblatt, F. The Design of an Intelligent Automaton. *Res. Trends*. 1958, No VI, P. 1–7.

134. Perceptrons | The MIT Press. URL:https://mitpress.mit.edu/books/perceptrons.

135. Николенко С., Кадурин А., Архангельская Е. Глубокое обучение. СПб.: Питер, 2018. 480 с.

136. Glorot X., Bordes A., Bengio Y. Deep sparse rectifier neural networks. *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2011. P. 315-323.

137. Згуровский М. З., Зайченко Ю. П. Основы вычислительного интеллекта. К.: Наук. думка. 2013. 407 с.

138. Glorot X., Bengio Y. Understanding the Difficulty of Training Deep Feedforward Neural Networks. *International conference on artificial intelligence and statistics*. 2010. P. 249–256.

139. He K., Zhang X., Ren S., Sun J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *Proceedings of the IEEE international conference on computer vision.* 2015. P. 1026-1034.

140. Guenin B., Konemann J., Tunnel L. A Gentle Introduction to Optimization, *Cambridge University Press.* 2014.

141. Calafiore G. C., El Ghaoui L. Optimization Models. *Cambridge University Press.* 2014.

142. Polyak B. T. Some methods of speeding up the convergence of iteration methods. *Ussr computational mathematics and mathematical physics*. 1964. Vol. 4, No 5.P. 1-17.

143. Nesterov Y. E. A method for solving the convex programming problem with convergence rate O (1/k^ 2). *Dokl. akad. nauk SSSR.* 1983. Vol. 269. P. 543-547.

144. Nesterov Y. Introductory Lectures on Convex Optimization: A Basic Course. Applied optimization. Kluwer Academic Publ., Boston, Dordrecht, London, Springer. 2004.

145. Bengio Y., Boulanger-Lewandowski N., Pascanu R. Advances in optimizing recurrent networks. *IEEE international conference on acoustics, speech and signal processing*. IEEE, 2013. P. 8624-8628.

146. Duchi J., Hazan E., Singer Y. Adaptive subgradient methods for online learning and stochastic optimization. J*ournal of Machine Learning Research.* 2011. Vol. 12, No Jul. P. 2121–2159.

147. Hinton G. Neural networks for machine learning. Coursera, video lectures. URL: http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf.

148. Kingma D. P., Ba J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*. 2014.

149. Chollet F. Deep Learning with R. New York, USA: Manning, 2017.

150. Rumelhart D. E., Hinton G. E., Williams R. J. Learning representations by back-propagating errors. *Nature.* 1986. Vol. 323, No 6088. P. 533-536.

151. Hochreiter S. Untersuchungen zu dynamischen neuronalen Netzen. *Diploma, Technische Universität München*. . Vol. 91, No 1.

152. Bengio Y., Simard P., Frasconi P. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks.* 1994. Vol. 5, No 2. P. 157-166.

153. Hochreiter S., Schmidhuber J. Long short-term memory. *Neural computation.* 1997. Vol. 9, No 8. P. 1735-1780.

154. Gers F. A., Eck D., Schmidhuber J. Applying LSTM to time series predictable through time-window approaches. *Neural Nets WIRN Vietri-01. Springer, London.* 2002. P. 193-200.

155. Hochreiter S., Bengio, Y., Frasconi, P., Schmidhuber, J. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. 2001. URL: https://www.researchgate.net/profile/Y-Bengio/publication/2839938_Gradient_Flow_in_Recurrent_Nets_the_Difficulty_of_Learning_Long-Term_Dependencies/links/546cd26e0cf2193b94c577c2/Gradient-Flow-in-Recurrent-Nets-the-Difficulty-of-Learning-Long-Term-Dependencies.pdf.

156. Cho K., Van Merriënboer B., Bahdanau D., Bengio Y. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*. 2014.

157. LeCun Y. Boser B., Denker J., Henderson D., Howard R., Hubbard W., Jackel L. Backpropagation applied to handwritten zip code recognition. *Neural computation.* 1989. Vol. 1, No 4. P. 541-551.

158. Zeiler M. D., Fergus R. Visualizing and understanding convolutional networks. *European conference on computer vision*. Springer, Cham, 2014. P. 818-833.

159. LeCun Y., Bottou L., Bengio Y., Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*. 1998. Vol. 86, No 11. P. 2278-2324.

160. Krizhevsky A., Sutskever I., Hinton G. E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*. 2012. Vol. 25.

161. LeCun Y., Kavukcuoglu K., Farabet C. Convolutional networks and applications in vision. *Proceedings of 2010 IEEE international symposium on circuits and systems*. IEEE. 2010. P. 253-256.

162. He K., Zhang,X., Ren S., Sun J. Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2016. P. 770-778.

163. CNNs from different viewpoints. URL: https://medium.com/impactai/cnns-from-different-viewpoints-fab7f52d159c.

164. Kolassa S. Evaluating predictive count data distributions in retail sales forecasting. *Int. J. Forecasting.* 2016. Vol. 32, No 3. P. 788–803. doi: 10.1016/j.ijforecast.2015.12.004.

165. Krause E.F. Taxicab Geometry: An Adventure in Non-Euclidean Geometry. New York: Dover Publications, 1987. 96 p.

166. Armstrong J. S. Long-range Forecasting: From Crystal Ball to Computer. New York: John Wiley & Sons, 1978.

167. Makridakis S. Accuracy measures: theoretical and practical concerns. *Int. J. Forecasting.*1993. Vol. 9, No 4. P. 527–529. doi: 10.1016/0169-2070(93)90079-3.

168. Goodwin P., Lawton R. On the asymmetry of the symmetric MAPE. I*nt. J. Forecasting*. 1999. Vol. 15, No 4. P. 405–408. doi: 10.1016/S0169-2070(99)00007-2.

169. Hyndman R.J., Koehler A.B. Another look at measures of forecast accuracy. *Int. J. Forecasting.* 2006. Vol. 22, No 4. P. 679–688. doi: 10.1016/j.ijforecast.2006.03.001.

170. Petropoulos F., Kourentzes N. Forecast combinations for intermittent demand. *J. Operat. Res. Soc.* 2015. Vol. 66, No 6. P. 914–924. doi: 10.1057/jors.2014.62.

171. Davydenko A., Fildes R. Measuring forecasting accuracy: The case of judgmental adjustments to SKU-level demand forecasts. *Int. J. Forecasting.* 2013. Vol. 29, No 3. P. 510–522. doi: 10.1016/j.ijforecast.2012.09.002.

172. Jao C.S. Efficient decision support systems – practice and challenges from current to future. Rijeka, Croatia: Intech, 2011.

173. Dovgyj S.O., Bidyuk P.I., Trofymchuk O.M. DSS on the basis of statistical and probabilistic methods. Kyiv: Logos, 2014.

174. Zgurowskii M.Z., Pankratova N.D. System analysis: problems, methodology, applications. Kyiv: Naukova Dumka, 2005.

175. Gilks W.R., Richardson S., Spiegelhalter D.J. Markov Chain Monte Carlo in practice. New York: CRC Press LLC, 2000.

176. Bishop C.M. Pattern Recognition and Machine Learning. New York: Springer, 2006.

177. Hastings, W.K. Monte Carlo Sampling Methods Using Markov Chains and Their Applications. *Biometrika*. 1970. Vol. 57, No 1. P. 97–109. doi:10.1093/biomet/57.1.97.

178. Walmart Recruiting - Store Sales Forecasting | Kaggle. URL: https://www.kaggle.com/c/walmart-recruiting-store-sales-forecasting/data.