

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет комп'ютерної інженерії та управління  
(повна назва)

Кафедра електронних обчислювальних машин  
(повна назва)

**КВАЛІФІКАЦІЙНА РОБОТА**  
**Пояснювальна записка**

Рівень вищої освіти другий (магістерський)

Короткострокове прогнозування середньорічної  
температури повітря за допомогою штучних нейронних  
мереж  
(тема)

Виконав:

студент II курсу, групи СПМ-22-4  
Кононенко О.М.  
(прізвище, ініціали)

Спеціальність 123 «Комп'ютерна інженерія»  
(код і повна назва спеціальності)

Тип програми освітньо-наукова  
(освітньо-професійна або освітньо-наукова)

Освітня програма Системне програмування  
(повна назва освітньої програми)

Керівник: доц. Іващенко Г.С.  
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри ЕОМ

Коваленко А.А.  
(прізвище, ініціали)

2024 р.

Харківський національний університет радіоелектроніки

Факультет \_\_\_\_\_ комп'ютерної інженерії та управління \_\_\_\_\_

Кафедра \_\_\_\_\_ електронних обчислювальних машин \_\_\_\_\_

Рівень вищої освіти \_\_\_\_\_ другий (магістерський) \_\_\_\_\_

Спеціальність \_\_\_\_\_ 123 «Комп'ютерна інженерія» \_\_\_\_\_  
(код і повна назва)

Тип програми \_\_\_\_\_ освітньо-наукова \_\_\_\_\_  
(освітньо-професійна або освітньо-наукова)

Освітня програма \_\_\_\_\_ Системне програмування \_\_\_\_\_  
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_  
(підпис)

“ \_\_\_\_\_ ” \_\_\_\_\_ 20\_\_ р.

## ЗАВДАННЯ

### НА КВАЛІФІКАЦІЙНУ РОБОТУ

студенту \_\_\_\_\_ Кононенку Олександр Миколайовичу \_\_\_\_\_  
(прізвище, ім'я, по батькові)

1. Тема роботи Короткострокове прогнозування середньорічної температури повітря за допомогою штучних нейронних мереж

затверджена наказом по університету від “ 01 ” квітня 2024 р. № 257 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 15 червня 2024 р.

3. Вхідні дані до роботи \_\_\_\_\_

1) документація мови програмування Python;

2) інтегроване середовище: JetBrains PyCharm 2024;

3) набори даних про середньорічну температуру повітря.

4. Перелік питань, що потрібно опрацювати у роботі \_\_\_\_\_

1) аналіз предметної області;

2) огляд існуючих досліджень;

3) дослідження методів прогнозування часових рядів;

4) програмна реалізація обраних моделей прогнозування;

5) аналіз результатів дослідження;

6) висновки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) \_\_\_\_\_

Слайд-презентація – 14 слайдів \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1 )

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

### КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз проблеми та огляд існуючих рішень	02.04.24-08.04.24	
2	Вибір методики дослідження	09.04.24-16.04.24	
3	Вибір інструментальних засобів	17.04.24-22.04.24	
4	Розробка алгоритмічного забезпечення	23.04.24-06.05.24	
5	Проведення експериментів	07.05.24-23.05.24	
6	Оформлення матеріалів кваліфікаційної роботи	24.05.24-03.06.24	
7	Подання кваліфікаційної роботи керівникові та її попередній захист	04.06.24-07.06.24	
8	Подання кваліфікаційної роботи на рецензування	08.06.24-12.06.24	

Дата видачі завдання 01 квітня 2024 р.

Студент \_\_\_\_\_  
(підпис)

Керівник роботи \_\_\_\_\_  
(підпис)

доц. Іващенко Г.С.  
(посада, прізвище, ініціали)

## РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 78 с., 24 рис., 10 табл., 2 дод., 47 джерел.

ОБЧИСЛЮВАЛЬНИЙ ІНТЕЛЕКТ, ШТУЧНА НЕЙРОННА МЕРЕЖА, ЧАСОВІ РЯДИ, ПРОГНОЗУВАННЯ, MLP, LSTM, MAE, MARE, PYTHON.

Метою кваліфікаційної роботи є дослідження ефективності методів обчислювального інтелекту для короткострокового прогнозування середньорічної температури повітря за допомогою штучних нейронних мереж.

У ході виконання кваліфікаційної роботи було проведено аналіз існуючих методів обчислювального інтелекту для прогнозування часових рядів. Було оцінено їх переваги та недоліки з метою визначення більш ефективного підходу. Реалізовано декілька моделей короткострокового прогнозування на основі штучних нейронних мереж, зокрема, таких моделей, як багат шаровий персептрон та довга короткострокова пам'ять.

Для дослідження використовувалися дані щодо середньорічної температури повітря. Ці дані були зібрані та оброблені для тренування та тестування моделей, що дозволило здійснити об'єктивну оцінку їх ефективності.

Розроблений програмний продукт надає можливість прогнозувати часові ряди середньорічної температури повітря за допомогою моделей MLP, LSTM, а також включає інструменти для проведення порівняльного аналізу.

## ABSTRACT

Master's thesis: 78 pages, 24 figures, 10 tables, 2 appendices, 47 sources.

COMPUTATIONAL INTELLIGENCE, ARTIFICIAL NEURAL NETWORK, TIME SERIES, FORECASTING, MLP, LSTM, MAE, MAPE, PYTHON.

The major goal of this thesis is study of the effectiveness of computational intelligence methods for short-term forecasting of the average annual air temperature using artificial neural networks.

In the course of the qualification work, a thorough analysis of the existing methods of computational intelligence for forecasting time series was carried out. Their advantages and disadvantages were evaluated in order to determine a more effective approach. Several models of short-term prediction based on artificial neural networks have been implemented, including multilayer perceptron and long short-term memory.

Data on the average annual air temperature were used for the study. These data were collected and processed to train and test the models, which allowed for an objective evaluation of their performance.

The developed software product provides an opportunity to forecast time series of average annual air temperature using MLP, LSTM models, and also includes tools for comparative analysis.

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ .....	8
ВСТУП .....	9
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ .....	11
1.1 Актуальність завдання прогнозування .....	11
1.2 Часові ряди.....	12
1.3 Аналіз часових рядів.....	13
1.4 Прогнозування часових рядів .....	15
1.4.1 Класифікація методів та моделей прогнозування .....	15
1.4.2 Статистичні методи прогнозування .....	16
1.4.3 Критерії точності прогнозів .....	18
1.5 Аналіз існуючих досліджень.....	20
1.5.1 Порівняння ARIMA та LSTM у прогнозуванні часових рядах .....	20
1.5.2 Прогнозування часових рядів за допомогою CNN.....	21
1.6 Постановка задачі.....	22
2 ВИКОРИСТАННЯ ЗАСОБІВ ОБЧИСЛЮВАЛЬНОГО ІНТЕЛЕКТУ .....	24
2.1 Штучні нейронні мережі .....	24
2.2 Навчання та використання штучних нейронних мереж .....	25
2.3 Перенавчання нейронних мереж .....	27
2.4 Використані архітектури ШНМ .....	28
2.4.1 Багатошаровий перцептрон .....	29
2.4.2 Нейронні мережі з довгою короткостроковою пам'яттю (LSTM).....	31
3 ПРОГРАМНА РЕАЛІЗАЦІЯ.....	35
3.1 Середовище розробки .....	35
3.2 Використання програмних засобів машинного навчання .....	37
3.2.1 Клас Sequential.....	38

3.2.2 Клас Dense.....	39
3.2.3 Клас LSTM.....	39
3.3 Реалізація алгоритму прогнозування .....	40
3.4 Підготовка даних.....	40
3.5 Реалізація моделі довгої короткострокової пам'яті .....	41
3.6 Реалізація моделі багатошарового перцептронну .....	42
4 АНАЛІЗ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ .....	44
4.1 Досліджувані параметри моделей .....	44
4.2 Базові конфігурації моделей .....	46
4.3 Залежність результатів від параметрів прогнозування .....	47
4.4 Вплив гіперпараметрів на результати прогнозування .....	55
ВИСНОВКИ.....	62
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ .....	63
ДОДАТОК А Графічний матеріал кваліфікаційної роботи.....	68
ДОДАТОК Б Вихідний код розробленого програмного засобу .....	76

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ  
І ТЕРМІНІВ

ГА – генетичний алгоритм

ШІС – штучка імунна система

ШНМ – штучна нейронна мережа

ARIMA – авторегресійна інтегрована модель з ковзним середнім (англ., Autoregression integrated moving average)

LSTM – довга короткострокова пам'ять (англ., Long Short-Term Memory)

MAE – середня абсолютна помилка (англ., Mean absolute error)

MAPE – середня абсолютна похибка у відсотках (англ., Mean absolute percentage error)

MLP – багатошаровий перцептрон (англ., Multilayer Perceptron)

MRE – середня відносна помилка (англ., Mean relational error)

MSPE – середньоквадратична помилка у відсотках (англ., Mean squared percentage error)

SMAPE – симетрична середня абсолютна похибка у відсотках (англ., Symmetric Mean Absolute Percentage Error)

## ВСТУП

Прогнозування часових рядів залишається актуальною проблемою, і застосування різноманітних методів та моделей дозволяє покращити точність прогнозів та забезпечити можливість висновків щодо майбутнього стану систем. Прогнозування майбутніх значень часових рядів є актуальним завданням у багатьох наукових і технічних галузях, де важливо передбачати розвиток подій та тенденцій. Це дозволяє забезпечити не лише ефективне управління процесами, а й приймати обґрунтовані рішення на основі аналізу наявних даних. При цьому точність прогнозування залежить від ряду факторів, включаючи якість даних, тривалість періоду прогнозування [1-5], вплив зовнішніх факторів та можливі аномалії у даних [5-10].

Прогнозування часового ряду передбачає аналіз вихідних даних [11], у ході якого здійснюється виявлення трендів, сезонності та циклічності в даних. Для досягнення цих цілей використовуються різноманітні методи, такі як ARIMA (авторегресійна інтегрована модель з ковзним середнім), експоненційне згладжування, генетичні алгоритми (ГА). Ці методи дозволяють моделювати часові ряди та прогнозувати їх майбутні значення на основі історичних даних. Крім того, вони допомагають виявляти приховані закономірності та залежності в даних, що може бути корисно для прийняття рішень та оптимізації процесів [11].

Окрім широко відомих статистичних підходів, таких як експоненційне згладжування, регресійні та авторегресійні моделі [1, 2, 12], існують і нові, які дозволяють ефективно враховувати складні взаємозв'язки у даних. Зокрема, методи машинного навчання такі як штучні нейронні мережі [13-15] та штучні імунні системи [16-23], стають все більш популярними завдяки своїй здатності до адаптації та прогнозування складних динамічних процесів. Методи, що базуються на генетичних алгоритмах, можуть бути використані для прогнозування часових рядів шляхом створення та оптимізації

прогнозуючих моделей [24]. Поширеним підходом є методи штучних імунних систем [24] які використовуються для прогнозування часових рядів, моделюючи поведінку штучної імунної системи у відповідь на зміни самого середовища. Наприклад, алгоритми, що базуються на антигенних взаємодіях, можуть знаходити шаблони у часових рядах та прогнозувати майбутні значення на їх основі. Один з прикладів таких систем – алгоритм AIS-ARIMA [24], який поєднує методи ШІС та класичну модель ARIMA для прогнозування часових рядів.

Найпоширенішими серед засобів обчислювального інтелекту є штучні нейронні мережі, які надають інструмент для прогнозування часових рядів. Протягом останніх років розроблено багато методів та алгоритмів для покращення точності та ефективності прогнозування [11].

Для успішного прогнозування важливо мати необхідну кількість даних. Якщо даних недостатньо, можна застосовувати техніки ресемплінга [24], щоб збільшити їх кількість. Також важливо проводити аналіз якості даних, виявляти та коригувати аномальні значення, щоб покращити точність моделі.

При прогнозуванні важливо не тільки використовувати ефективні моделі, а й враховувати стабільність вихідних даних. Стабільність вхідних даних означає, що ці дані не демонструють значних змін у часі або не мають великих коливань, що може суттєво вплинути на результати аналізу або моделювання. Аномалії та помилки у даних можуть суттєво вплинути на точність прогнозування, тому їхнє виявлення та корекція є одним з головних етапів аналізу часових рядів.

У роботі прогнозування часових рядів розглянуто в контексті прогнозування середньорічної температури повітря за допомогою штучних нейронних мереж (ШНМ).

## 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

### 1.1 Актуальність завдання прогнозування

Аналіз часових рядів є складним та міжгалузевим процесом, який включає в себе різноманітні методи та підходи для прогнозування динаміки даних у часі. Він є необхідним інструментом для прийняття обґрунтованих рішень у різних галузях, наприклад: фінансових, медичних, економічних, екологічних [11, 24, 44-46].

У фінансовій сфері – допомагає інвесторам та трейдерам розуміти ринкові тенденції, виявляти можливості для прибуткових інвестицій та управляти ризиками. Прийняття рішення базується на моделях прогнозування цін активів, валютних курсів та товарів, що дозволяє приймати обґрунтовані рішення щодо торгівлі на фондовому ринку [11].

У медицині аналіз часових рядів використовується для прогнозування захворювань, виявлення тенденцій у ефективності лікування та планування медичних ресурсів. Це дозволяє організаціям охорони здоров'я приймати найефективніші рішення щодо розподілу персоналу та обладнання для забезпечення потреб пацієнтів [44].

У економіці аналіз часових рядів є необхідним для прогнозування економічних показників, таких як інфляція та безробіття, що допомагає урядам та підприємствам розробляти та реалізовувати стратегії розвитку та бюджетування [24].

У кліматології прогнозування часових рядів використовується для отримання інформації щодо майбутніх погодних умов та виявлення змін в кліматі, що дозволяє урядам та організаціям ефективно реагувати на негативні впливи природних явищ [45].

У екології прогнозування часових рядів використовується для вивчення забруднення навколишнього середовища, змін клімату та екологічних

параметрів. Воно дозволяє приймати рішення щодо зменшення викидів забруднюючих речовин та підтримувати стабільність екосистем [46].

У транспорті прогнозування часових рядів використовується для планування та оптимізації розкладів та маршрутів транспортних систем, що сприяє підвищенню ефективності перевезень та зменшенню заторів [47].

З розвитком технологій та зростанням обсягу даних прогнозування часових рядів стає все більш точним та ефективним інструментом для прийняття обґрунтованих рішень у різних галузях.

## 1.2 Часові ряди

Часовий ряд представляє собою послідовність даних, де кожне значення вимірюється і фіксується впродовж певного періоду часу з відомим часовим інтервалом або порядковим номером [25]. Ці послідовності даних використовуються для аналізу та прогнозування, дозволяючи встановити вірогідні зміни використаних у часовому ряді показників у майбутньому. Наприклад, такі дані можуть вказувати на кількість завантажень мобільного застосунку за день.

Показники, які утворюють часові ряди, можуть бути різноманітними і охоплювати не лише технічні дані, а й економічні, соціальні або природні аспекти [25].

Основні види часових рядів:

- стаціонарні ряди;
- нестаціонарні ряди.

Стаціонарні часові ряди характеризуються стабільними статистичними властивостями протягом часу. Це означає, що середнє значення, дисперсія та автокореляційна функція не змінюються з часом. Стаціонарність є важливою властивістю для багатьох методів аналізу часових рядів, зокрема для класичних моделей ARIMA (AutoRegressive Integrated Moving Average). Основні характеристики стаціонарних рядів можна визначити такі:

- постійне середнє: середнє значення ряду залишається постійним з часом;
- постійна дисперсія: дисперсія даних залишається сталою;
- постійна автокореляція: кореляція між значеннями ряду залежить тільки від лагу (відстані між значеннями), а не від часу, в який ці значення взяті.

Приклади стаціонарних часових рядів:

- білий шум (послідовність випадкових значень з постійним середнім і дисперсією);
- фінансові індекси в коротких проміжках часу, якщо вони не демонструють трендів або сезонності.

Нестаціонарні часові ряди мають змінні статистичні властивості. Це означає, що середнє значення, дисперсія або автокореляція змінюються з часом. Нестаціонарність у часових рядах може проявлятися через тренди, сезонні компоненти або структурні зміни.

Основні характеристики нестаціонарних рядів:

- змінне середнє: середнє значення ряду змінюється з часом;
- змінна дисперсія: дисперсія даних не залишається сталою;
- змінна автокореляція: кореляція між значеннями ряду змінюється залежно від часу.

Приклади нестаціонарних часових рядів:

- температурні ряди, що демонструють сезонність;
- економічні показники, такі як ВВП або інфляція, які мають довгострокові тренди.

### 1.3 Аналіз часових рядів

У дослідженні часових рядів значна увага приділяється виявленню закономірностей їх змін протягом тривалого часу. Зазвичай, типовий часовий ряд складається з детермінованої та випадкової складових.

Тренд, або основна тенденція – стійка систематично змінна довготривала складова. Визначається як характеристика, що описує зміни певного явища протягом тривалого часу, без врахування випадкових коливань, які створюються іншими факторами. Тенденція присутня, коли спостерігається тривале збільшення або зменшення даних, і вона не обов'язково повинна бути лінійною. Наприклад, ряди, що відображають приріст населення або кількість будинків у місті, демонструють зростаючу тенденцію, тоді як тенденцію зниження можна побачити в часових послідовностях, що стосуються рівня смертності або епідемій.

Циклічна складова [36] виявляється протягом тривалого часу і є результатом дії факторів з післядією або тих, що циклічно змінюються з часом. Цикли виникають, коли дані демонструють підйоми та спади без фіксованої частоти. Характерними прикладами є цикли ділової активності, демографічні та астрофізичні цикли. Цикли зазвичай тривають два або більше років. Більшість економічних та фінансових часових рядів мають певні циклічні коливання. Наприклад, діловий цикл включає чотири фази: процвітання, занепад, депресія та відновлення [35].

Сезонна складова зумовлена дією механізмів, що періодично повторюються в певні часи року. Важливими факторами сезонних коливань є кліматичні умови, традиційні звичаї тощо. Наприклад, продажі морозива зростають влітку, а вовняних тканин – взимку.

Випадкова або нерегулярна складова не піддається обліку та реєстрації, оскільки формується в результаті дії багатьох зовнішніх факторів, які не беруть участь у формуванні детермінованої складової. Такі зміни можуть бути викликані подіями, як-от війна, страйк, землетрус, повінь, революція тощо. Не існує визначеної статистичної методики вимірювання випадкових коливань у часових рядах [35, 37].

Для опису часових рядів, використовуються такі характеристики, як дисперсія, середнє значення, середні показники. Середні показники часових рядів включають середній рівень ряду, середнє прискорення та зміну, а також

середні темпи росту. Ці показники характеризують тенденції і є необхідними для узагальнення показників тенденцій за тривалий період, для різних періодів; вони потрібні для порівняння розвитку за неоднакові за тривалістю відрізки часу та при виборі аналітичного вираження тренду. За наявності значних коливань рівнів у динамічному ряді, визначення середніх показників тенденцій вимагає застосування спеціальних статистичних методів.

Таким чином, часовий ряд розглядають як суму чотирьох основних компонентів, а саме випадкового, трендового, сезонного, циклічного.

#### 1.4 Прогнозування часових рядів

Прогнозування є одним із основних етапом у вивченні даних, де використовуються наявні дані та їх прогноз для оцінки можливих майбутніх подій та умов. Сучасні методи аналізу даних та машинного навчання забезпечують ефективність та точність в прогнозуванні різних параметрів та подій.

##### 1.4.1 Класифікація методів та моделей прогнозування

При прогнозуванні часових рядів за допомогою обчислювального інтелекту розглядається кілька факторів. Серед них важливість вибору відповідних методів, наявність достатньої кількості даних, відбір ознак та доступність потрібних обчислювальних ресурсів [37].

Методи прогнозування можна розділити на дві основні групи: інтуїтивні та формалізовані [39].

Інтуїтивний підхід використовується там, де об'єкт прогнозування занадто простий або, навпаки, дуже складний для математичного опису. Ці методи не базуються на розробці формальних моделей, але відображають індивідуальні думки та досвід експертів. Формалізовані методи прогнозування ґрунтуються на використанні статистичних моделей, які

аналітично визначають залежність між майбутніми та фактичними значеннями часових рядів. Такі методи включають регресійні моделі, авторегресійні моделі та моделі експоненційного згладжування. Серед формалізованих підходів окремо слід зазначити структурні моделі, які визначають функціональну залежність між фактичними та майбутніми значеннями шляхом аналізу структури даних [39]. Це можуть бути нейромережеві моделі, моделі ланцюгів Маркова чи моделі на базі регресійних дерев.

Використання різних методів прогнозування сприяє отриманню більш точних та надійних результатів, що є важливим у різних галузях та дослідженнях [40].

#### 1.4.2 Статистичні методи прогнозування

Розглянуто наступні статистичні методи прогнозування:

- найвний метод (рисунок 1.1);
- метод ковзних середніх
- метод експоненційного згладжування;
- моделі ARMA та ARIMA.

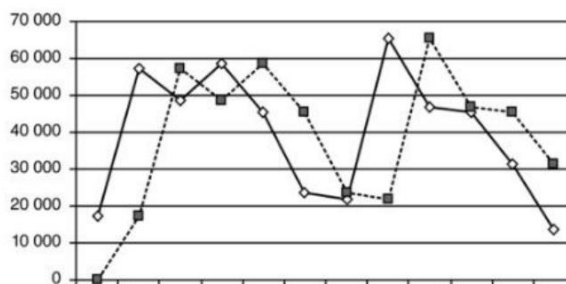


Рисунок 1.1 – Приклад найвного прогнозу часового ряду

Метод ковзних середніх використовується для усереднення значень в межах певного вікна, яке переміщується по часовому ряду. Ширина цього

вікна визначає кількість попередніх періодів, які беруться до уваги при прогнозуванні (рисунок 1.2).

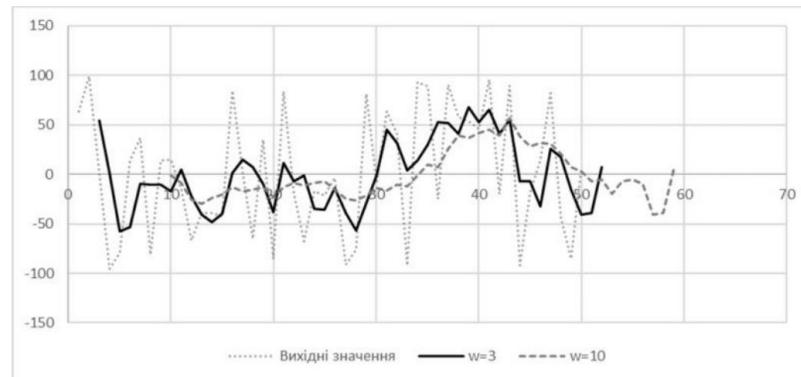


Рисунок 1.2 – Приклад прогнозування методом ковзних середніх

Експоненціальне згладжування демонструє найкращу ефективність у випадку, коли значення часового ряду слідує за плавним трендом та відображають сезонну динаміку, де значення змінюються за певною циклічною схемою через задану кількість часових інтервалів (рисунок 1.3).

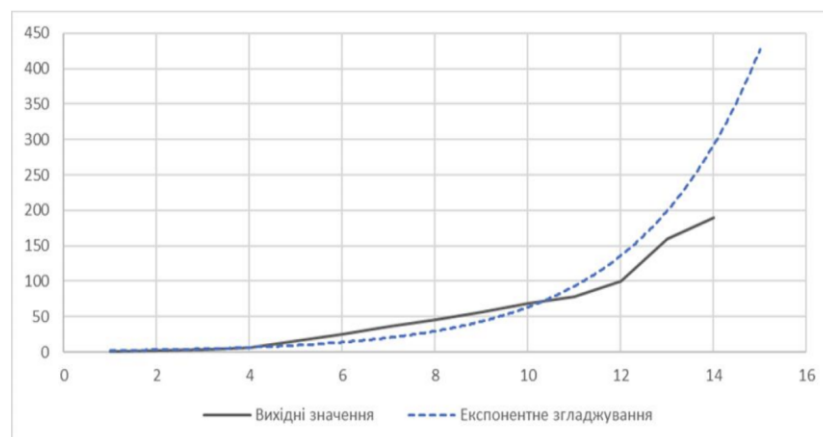


Рисунок 1.3 – Прогнозування часового ряду методом експоненційного згладжування

Метод експоненційного згладжування має досить високу точність для прогнозів на короткі строки, проте ця точність поступово зменшується зі

збільшенням горизонту прогнозування. На відміну від методу ковзних середніх, де всі дані мають однакову вагу, у методі експоненціального згладжування найбільший вплив надається найновішому спостереженню, що дає йому перевагу (рисунок 1.3).

ARMA (autoregressive moving-average model) – це модель, яка базується на припущенні, що поточні значення прогнозованого часового ряду залежать лише від лінійної комбінації попередніх значень цього ряду та випадкових помилок.

Недоліком моделі ARMA є її обмеження застосування лише до стаціонарних часових рядів. У разі присутності тренду у досліджуваному ряді чи його нестационарності ARMA (модель без модифікацій) не може бути використана. У таких випадках може бути застосована модель ARIMA (autoregressive integrated moving average), яка враховує різниці між спостереженнями (наприклад, віднімання спостереження на кроці  $n$  з спостереження на кроці  $n+1$ ) для того, щоб зробити часовий ряд стаціонарним [26].

#### 1.4.3 Критерії точності прогнозів

Показники точності моделі прогнозування засновані на даних часових рядів, що використовуються для генерування моделі. Усі показники точності ґрунтуються на хронологічних даних. Показники точності можуть також використовуватися як індикатор точності прогнозу, але на майбутні значення вони не переносяться.

Верифікація та оцінка якості побудованої моделі часових рядів у системній методології статистичного підходу базується на проведенні експериментів. Вони використовуються для оцінки поведінки моделі при різних вхідних даних та умовах. Це дозволяє перевірити, як модель реагує на зміну параметрів і чи може вона відтворювати реальні процеси. У цьому процесі здійснюється багатокритеріальне оцінювання якості моделі.

Визначаються внутрішні та зовнішні параметри якості моделі, використовуючи стандартизовані критерії, головними серед яких є критерії точності моделювання, що дозволяють порівнювати конкуруючі моделі часових рядів [40].

Мірою точності або адекватності прогнозу є величина різниці між фактичними та прогнозованими значеннями досліджуваних даних. Існує багато критеріїв, які допомагають визначити точність прогнозу конкретної моделі, такі як:

- коефіцієнт Тейла;
- коефіцієнт детермінації.

Одним із важливих критеріїв точності моделі та її сумісності є коефіцієнт Тейла, який обчислюється через середньоквадратичне значення похибки прогнозу. Значення  $U$  може дорівнювати числу в діапазоні від 0 до 1: якщо  $U$  дорівнює 1, то це свідчить, що модель не може бути використана для прогнозу, а якщо коефіцієнт  $U$  рівний 0, у такому випадку модель може бути використана для даної задачі прогнозування. Коефіцієнт детермінації  $R^2$  – це квадрат множинного коефіцієнта кореляції, який чисельно показує частку варіації залежної змінної, яку можна прогнозувати на основі незалежної змінної [41].

Коефіцієнт детермінації застосовується в контексті статистичних моделей для прогнозування за допомогою часових рядів. Основна мета полягає у прогнозуванні майбутніх результатів або перевірці гіпотез на основі іншої пов'язаної інформації. Він визначає, як добре обрана модель відтворює спостережувані результати, оцінюючи частку загальної варіації результатів, що пояснюється цією моделлю. Можна зробити висновок, що модель прогнозування є адекватною, якщо коефіцієнт детермінації наближається до одиниці. Цей показник також часто використовується для порівняння кількох моделей та вибору найкращої [41].

У даній роботі використовувалися найбільш поширені абсолютні критерії адекватності моделей прогнозів на основі часових рядів.

## 1.5 Аналіз існуючих досліджень

Для формування задачі та вибору методів обчислювального інтелекту для прогнозування часових рядів проведено аналіз існуючих досліджень у цій області. Дослідження охопили використання широко відомих моделей та методів прогнозування, а також спеціалізовані моделі, призначені для вирішення конкретних завдань [13, 15, 16, 21].

Один з напрямків досліджень стосувався застосування стандартних моделей та методів прогнозування для різних типів часових рядів. Це включало в себе використання авторегресійних моделей, експоненційного згладжування, а також методів машинного навчання, таких як нейронні мережі. Ці дослідження дозволили отримати значні відомості про ефективність та обмеження кожного з цих методів у різних ситуаціях [13].

Крім того, проведено аналіз спеціалізованих моделей прогнозування, які були розроблені для рішення конкретних завдань або врахування особливостей певних типів даних. Ці моделі можуть базуватися на комбінації різних методів та підходів, включаючи адаптивні алгоритми прогнозування, обробку сезонності та врахування зовнішніх факторів, що можуть впливати на часовий ряд [15].

Результати аналізу надали базу для вибору методу прогнозування, який буде відповідати потребам конкретної задачі та забезпечить найкращі результати прогнозування. Крім того, ці дослідження дозволили ідентифікувати можливі напрямки подальших вдосконалень у цій області, такі як розробка нових алгоритмів прогнозування, використання інтегрованих моделей та поєднання різних методів машинного навчання.

### 1.5.1 Порівняння ARIMA та LSTM у прогнозуванні часових рядах

У дослідженні [42] розглянуто алгоритм глибокого навчання для прогнозування часових рядів, зокрема модель довгої короткострокової

пам'яті (LSTM), та проведено порівняння з традиційними методами, такими як авторегресійне інтегроване ковзне середнє (ARIMA).

Експериментальні дослідження проводились на фінансових часових рядах великого обсягу. Результати емпіричних досліджень показали, що алгоритми на основі глибокого навчання, зокрема LSTM, перевершують традиційні методи, такі як ARIMA. Зокрема, середнє зниження помилок, виміряне за допомогою метрики MAPE, становило близько 85% при використанні LSTM у порівнянні з ARIMA, що демонструє перевагу LSTM. Також було встановлено, що кількість періодів навчання не мала значного впливу на ефективність навченої моделі прогнозування.

### 1.5.2 Прогнозування часових рядів за допомогою CNN

У дослідженні, запропонованому у [43], розглянуто використання згорткової нейронної мережі, яка автоматично навчається для прогнозування часових рядів. Модель CNN була навчена на різноманітних часових рядах з різних джерел. Після закінчення обробки даних CNN знову навчалася, використовуючи правильні дані, отримані раніше. Потім ця модель застосовувалася для прогнозування часових рядів.

У цьому дослідженні фінансові часові ряди з фондових ринків Шанхаю та Шеньчженя, які вважаються найбільшими в Китаї, були використані як приклади. Дані про ціни закриття, обсягу та ринкові ціни акцій з 2000 по 2009 рік були використані для навчання згорткової нейронної мережі. Для тестування моделі були зібрані дані з 2010 по 2015 рік у формі трьох різних тестових наборів. Прогнозування було здійснене з використанням моделі, навченої на даних з 2009 року. Всі дані часових рядів були нормалізовані для відповідності вимогам моделі згорткової нейронної мережі. Оцінка результатів прогнозування за допомогою CNN на фінансових часових рядах показала, що запропонований метод працює ефективніше, ніж традиційний метод, такий як смуги Боллінджера.

## 1.6 Постановка задачі

Короткострокове прогнозування середньорічної температури повітря є важливим завданням у сфері кліматології та екології. Точні прогнози температури мають значний вплив на сільське господарство, енергетичний сектор, планування міської інфраструктури та інші галузі. З розвитком технологій машинного навчання, зокрема штучних нейронних мереж (ШНМ), з'явилася можливість підвищити точність таких прогнозів.

Метою даної роботи є розробка моделі для короткострокового прогнозування середньорічної температури повітря за допомогою штучних нейронних мереж. Це передбачає аналіз існуючих методів прогнозування, вибір оптимальної архітектури нейронної мережі та її навчання на основі історичних даних про температуру.

Для досягнення мети необхідно виконання відповідних завдань, а саме:

- дослідити сучасні методи прогнозування температури, зокрема ті, що використовують штучні нейронні мережі. Визначити переваги та недоліки різних підходів;
- розглянути різні типи нейронних мереж, такі як багат шарові перцептрони (MLP), рекурентні нейронні мережі (RNN) та довготривалу короткочасну пам'ять (LSTM);
- зібрати історичні дані про середньорічну температуру повітря з надійних джерел. Виконати попередню обробку даних, включаючи нормалізацію та вирішення проблем з пропущеними значеннями;
- вибрати оптимальну архітектуру штучної нейронної мережі для задачі прогнозування;
- реалізувати модель на обраному програмному забезпеченні та протестувати її на тестовому наборі даних для перевірки ефективності у реальних умовах;
- навчити обрану нейронну мережу на підготовлених даних. Виконати валідацію моделі для оцінки її точності та здатності до узагальнення;

- визначити міри точності прогнозу, такі як середньоквадратична похибка (MSE), середня абсолютна похибка (MAE) та коефіцієнт детермінації ( $R^2$ ). Порівняти результати з іншими методами прогнозування;

- на основі отриманих результатів зробити висновки про ефективність використання штучних нейронних мереж для короткострокового прогнозування середньорічної температури. Надати рекомендації щодо подальшого вдосконалення моделі та можливих напрямів майбутніх досліджень.

Очікується, що розроблена модель з використанням штучних нейронних мереж покаже високу точність прогнозування середньорічної температури повітря у порівнянні з традиційними методами. Це дозволить покращити планування в різних секторах, залежних від кліматичних умов, та сприятиме подальшому розвитку досліджень у галузі кліматичного моделювання.

## 2 ВИКОРИСТАННЯ ЗАСОБІВ ОБЧИСЛЮВАЛЬНОГО ІНТЕЛЕКТУ

### 2.1 Штучні нейронні мережі

Штучні нейронні мережі – це моделі, які використовуються у машинному навчанні для вирішення різноманітних завдань, таких як класифікація, розпізнавання, регресія, кластеризація та прогнозування. Ці моделі є математичними аналогами нервової системи живих організмів, де велика кількість нейронів пов'язані між собою [27].

Штучні нейронні мережі складаються з шарів нейронів, які обробляють вхідні дані та передають результати на наступний шар штучних нейронів. Цей процес продовжується до досягнення останнього шару, що генерує кінцевий результат (рисунок 2.1). Кожен нейрон у штучній нейронній мережі має свої ваги та зміщення, які налаштовуються під час навчання, для оптимальної адаптації мережі до конкретних завдань [27].

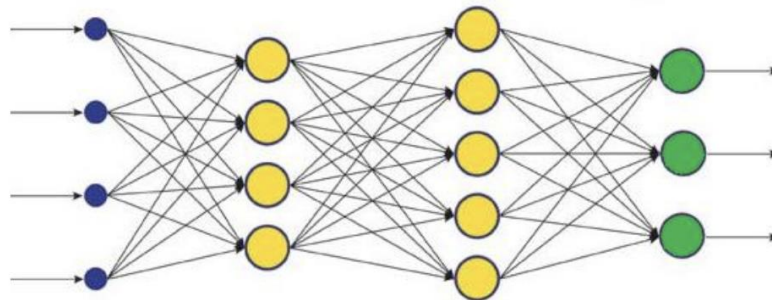


Рисунок 2.1 – Штучна нейронна мережа

У штучному нейроні  $w_1..w_n$  є вхідними сигналами,  $w_0$  – відповідає за модифікацію зсуву, символ  $\sum$  – позначає суматор вхідних сигналів,  $y$  – вказує на результуючий сигнал, а символ  $\emptyset$  – позначає функцію активації (рисунок 2.2).

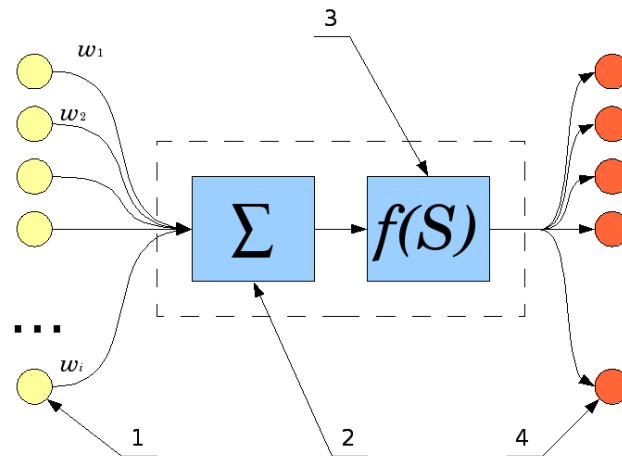


Рисунок 2.2 – Схематичне зображення штучного нейрона

Проблеми інтерпретації результатів виникають зазвичай через непрозорість або складність моделей штучних нейронних мереж (ШНМ). Однією з проблем, проблема «чорного ящика». Вона не є критичною, але ускладнює процес аналізу вже навчених мереж. Сутність проблема «чорного ящика» полягає в тому, що із навченої нейронної мережі не можна видобути ніяку корисну інформацію, крім тієї, яка подається на вихід. Через це неможливо встановити, на підставі чого нейронна мережа приймає рішення [27]. Для вирішення цього питання розробляються алгоритми витягування правил, які повинні отримувати інформацію з нейронних мереж у формі дерев рішень, математичних або логічних виразів [27].

## 2.2 Навчання та використання штучних нейронних мереж

Одним з алгоритмів навчання штучних нейронних мереж є алгоритм зворотного поширення помилки (Backpropagation) [29], який вважається одним з найпоширеніших. Його основні етапи:

- пряме поширення: вхідні дані проходять через всі шари нейронів, щоб отримати вихідні значення;
- обчислення помилки: визначається різниця між отриманим результатом та фактичним значенням за допомогою функції втрат;

- зворотне поширення помилки: помилка розповсюджується від останнього до першого шару, обчислюючи внесок кожного нейрона у загальну помилку;
- оновлення ваг: ваги зв'язків між нейронами оновлюються відповідно до обчисленої помилки та градієнту функції активації;
- повторення етапів навчання: описані кроки повторюються для кожного спостереження в навчальній вибірці до досягнення бажаної точності;
- регуляризація: застосовується для запобігання перенавчанню моделі, яке виникає, коли модель надмірно пристосовується до навчальних даних і втрачає здатність узагальнювати нові дані;
- налаштування параметрів: вибір оптимальних параметрів, таких як кількість шарів та нейронів у кожному шарі, та тип функції активації;
- перевірка моделі: оцінюється якість моделі на тестовій вибірці, яка не використовувалася під час навчання;
- донавчання: використовується для покращення точності моделі на тестовій вибірці. Воно дозволяє адаптувати модель до специфічних даних, зберігаючи загальні знання, отримані під час початкового навчання;
- застосування моделі: модель використовується для розв'язання конкретних задач;
- оцінка продуктивності: оцінюється продуктивність моделі у реальних умовах та порівняння з іншими методами;
- оновлення моделі: модель оновлюється при отриманні нових даних або зміні умов завдання;
- розгортання моделі: модель розгортається у продакшн-середовище для практичного застосування;
- моніторинг моделі: модель періодично відстежується та навчається на нових даних для забезпечення тривалої ефективності;
- оптимізація моделі: застосовуються методи оптимізації для покращення продуктивності та ефективності моделі.

Після завершення навчання модель перевіряється на тестовій вибірці, а також може піддаватися регуляризації та налаштуванню параметрів. Потім модель може бути використана для розв'язання задачі, та перевірена на продуктивність у реальних умовах. При необхідності модель може бути оновлена або оптимізована для підвищення її ефективності.

### 2.3 Перенавчання нейронних мереж

Однією з основних проблем, що виникають у процесі навчання нейронних мереж, є явище перенавчання. Це відбувається, коли нейронна мережа надмірно адаптується до вхідних даних з навчальної вибірки, замість того, щоб виявляти в них закономірності. Як результат, мережа втрачає гнучкість і здатність до узагальнення [29].

Для подолання проблеми перенавчання було розроблено кілька методів, серед яких найпоширенішим є «Метод випадкового відключення» або Dropout [29]. Основна ідея цього методу полягає в тому, щоб навчити кілька незалежних нейронних мереж і потім усереднити їх ваги. Для застосування цього методу необхідно випадково відключати частину нейронів у різних шарах мережі під час тренування (рисунок 2.3).

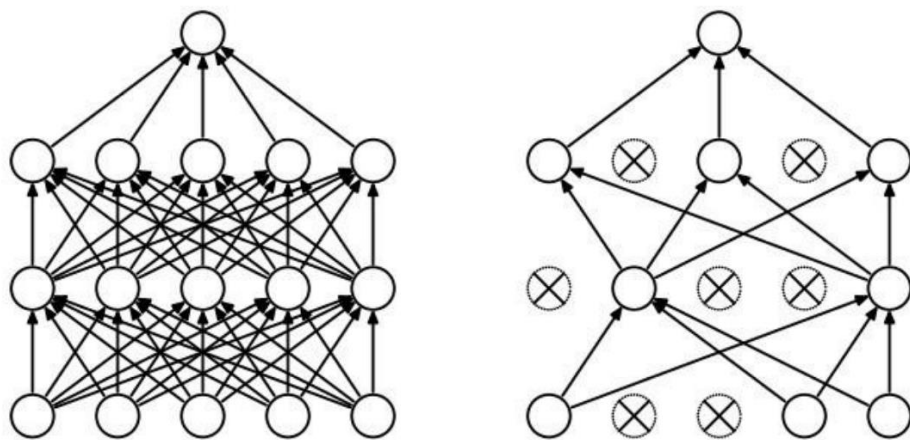


Рисунок 2.3 – Графічне представлення нейронної мережі до і після застосування методу Dropout

Вимкнення нейрона означає, що при будь-яких вхідних сигналах він буде видавати вихідне значення 0. Оскільки вимкнуті нейрони не беруть участі в процесі навчання і зворотному розповсюдженні помилки, їх вимкнення еквівалентне навчанню нової нейронної мережі [29].

## 2.4 Використані архітектури ШНМ

Порівняння моделей та аналіз їхнього використання для вирішення проблеми прогнозування часових рядів є важливим етапом для визначення їхньої ефективності та придатності для конкретних завдань [13-16]. Штучні нейронні мережі, зокрема моделі перцептронів з багатьма шарами (MLP) та довга короткострокова пам'ять (LSTM), є важливими інструментами у сучасних системах машинного навчання і глибокого навчання.

Модель багатошарового перцептронів є однією з найпоширеніших форм штучних нейронних мереж. Вперше вона була запропонована в середині 20-го століття та з тих пір активно використовується у багатьох областях, включаючи комп'ютерне зорове розпізнавання, обробку природних мов, прогнозування та інше. MLP складається з нейронів, організованих у шари: вхідний шар, приховані шари та вихідний шар. Кожен нейрон у вхідному шарі пов'язаний з кожним нейроном у прихованому шарі, а кожен нейрон у прихованому шарі – з кожним нейроном у вихідному шарі. MLP використовується для вирішення різноманітних завдань, таких як класифікація, регресія та прогнозування [30-32].

Довга короткострокова пам'ять (LSTM) є спеціальною формою рекурентних нейронних мереж, яка була розроблена для роботи з послідовними даними, такими як тексти, часові ряди або аудіозаписи. Вона має здатність запам'ятовувати довгострокові залежності в даних та ефективно працювати з послідовними вхідними даними. LSTM складається з комірок пам'яті, які можуть зберігати інформацію протягом тривалого часу, і використовує спеціальні ворота для регулювання потоку інформації через

мережу. Це дозволяє LSTM ефективно моделювати складні залежності в послідовних даних та вирішувати завдання, пов'язані з прогнозуванням, класифікацією та іншими областями [33].

#### 2.4.1 Багатошаровий перцептрон

Багатошаровий перцептрон (MLP) – це вид нейронної мережі, яка включає декілька шарів нейронів, включаючи вхідний, прихований та вихідний шари, які взаємодіють між собою через зв'язки з певною вагою (рисунок 2.4) [30].

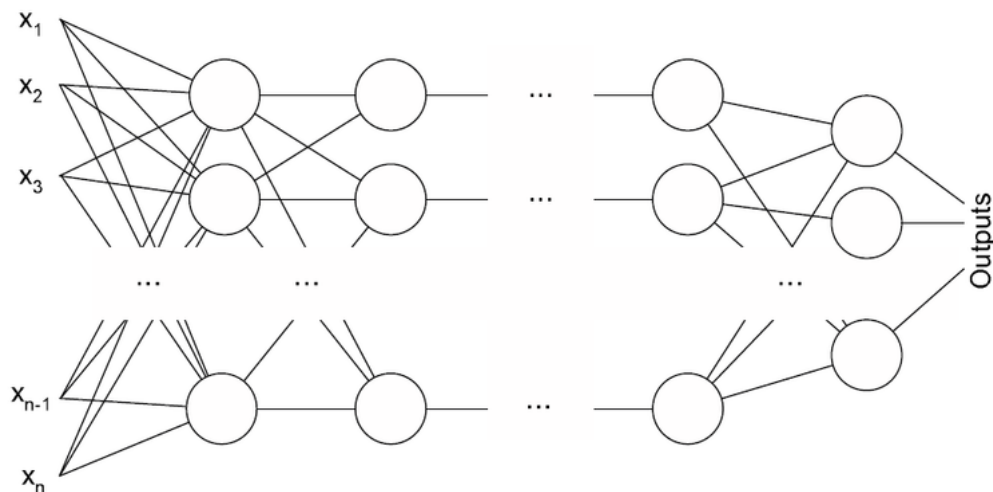


Рисунок 2.4 – Багатошаровий перцептрон

На вхідний шар (рисунок 2.5) надходять дані у вигляді вектора значень, де кожне значення відповідає конкретній ознаці об'єкта, який потрібно класифікувати або прогнозувати. Нейрони у вхідному шарі не проводять жодних обчислень, а лише передають ці значення на наступний шар – прихований.

Прихований шар виконує послідовність обчислень, які дозволяють нейронній мережі встановлювати зв'язки між вхідними даними та очікуваними вихідними значеннями. Кожен нейрон у прихованому шарі

отримує вхідні дані від усіх нейронів попереднього шару і виконує обчислення з використанням ваг та функції активації.

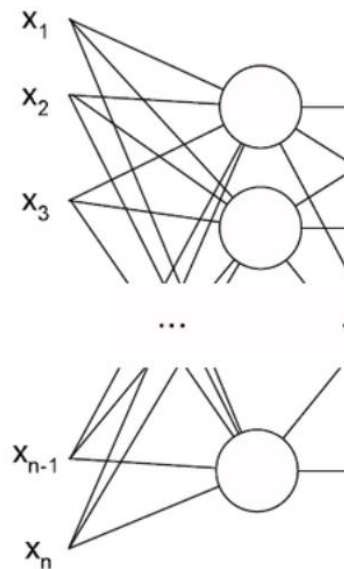


Рисунок 2.5 – Вхідний шар

Функція активації отримує зважену суму вхідних даних та їх ваг і повертає результат. Ця функція може бути реалізована, наприклад, за допомогою сигмоїди, гіперболічного тангенсу або ReLU [30].

Вихідний шар отримує дані від прихованого шару та проводить останні обчислення. Тип функції активації у вихідному шарі залежить від конкретного завдання, що потрібно вирішити.

Під час навчання багатозарового персептрону (MLP) за допомогою алгоритму зворотного поширення помилки, значення помилки на вихідному шарі порівнюються з фактичними значеннями, і ця помилка передається назад через нейронну мережу. Потім ваги нейронної мережі оновлюються таким чином, щоб мінімізувати цю помилку. Цей процес повторюється для кожного об'єкта в навчальному наборі даних до тих пір, поки ваги нейронної мережі не будуть оптимізовані [31].

Однією з особливостей багатозарового персептрону є його здатність автоматично знаходити з вхідних даних ознаки, що є значущими, це дозволяє

мережі враховувати складні залежності між вхідними ознаками та цільовими змінними. Проте для успішного використання MLP необхідно правильно вибрати структурні параметри моделі, такі як кількість прихованих шарів, кількість нейронів у кожному шарі та типи функцій активації [32].

Для покращення продуктивності та ефективності роботи багатошарового перцептронного застосовуються різні методи оптимізації, такі як проріджування та квантизація. Проріджування дозволяє зменшити розмір MLP шляхом видалення нейронів, які не впливають на результати обчислень або дають малий внесок у загальну роботу мережі. Це може значно зменшити розмір моделі та прискорити її роботу без втрати точності прогнозу [32].

Квантизація дозволяє зменшити кількість біт, що використовуються для зберігання ваг та активацій нейронів. Наприклад, використання 8-бітових чисел замість 32-бітових може значно знизити вимоги до пам'яті та прискорити обчислення на сучасних процесорах та графічних прискорювачах.

#### 2.4.2 Нейронні мережі з довгою короткостроковою пам'яттю (LSTM)

Довга короткострокова пам'ять (Long short-term memory, LSTM) представляє собою архітектуру рекурентних нейронних мереж, спеціально розроблену для вирішення проблеми довгострокових залежностей, що притаманні часовим рядам (рисунок 2.6).

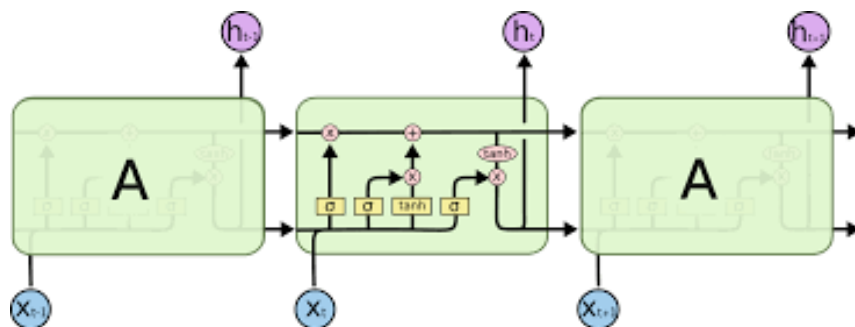


Рисунок 2.6 – Структура розгортки LSTM мережі

Особливістю LSTM є її здатність зберігати інформацію протягом тривалих періодів часу. Її структура, подібно до звичайних рекурентних нейронних мереж, має форму ланцюга, але є більш складною, оскільки включає чотири шари нейронів, що взаємодіють специфічним чином.

Одним з основних елементів LSTM є стан комірки (cell state), який можна уявити як горизонтальну лінію, що проходить через усю схему зверху вниз (рисунок 2.7).

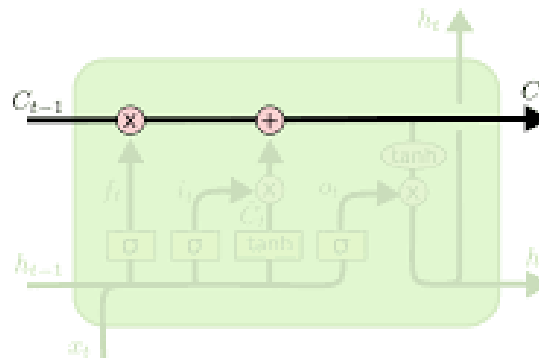


Рисунок 2.7 – Стан комірки на схемі розгортки LSTM мережі

Механізм LSTM має здатність видаляти інформацію зі стану комірки за допомогою фільтрів, які контролюють цей процес. Ці фільтри регулюють, яка інформація має передаватися в залежності від певних умов. Вони складаються зі шару сигмоїдальних нейронів та операції множення.

Сигмоїдальний шар повертає значення від нуля до одиниці, що вказує на те, яку частину кожного блоку інформації слід передати далі по мережі. Нуль вказує на те, що інформація не повинна бути передана, а одиниця – на те, що вона повинна бути передана в повному обсязі. У LSTM використовується три таких фільтри, які призначені для контролю і управління станом комірки [33].

Перший етап у роботі LSTM полягає у визначенні того, яка інформація може бути виключена із стану комірки (рисунок 2.8). Щоб визначити, яку інформацію можна виключити зі стану комірки, застосовується

сигмоїдальний шар, відомий як «шар фільтра забування» (forget gate layer). Цей шар використовує вихідні дані з попередньої ітерації, а також вхідні дані поточної ітерації, і повертає значення від 0 до 1, де 1 вказує на «повне збереження», а 0 – на «повне ігнорування».

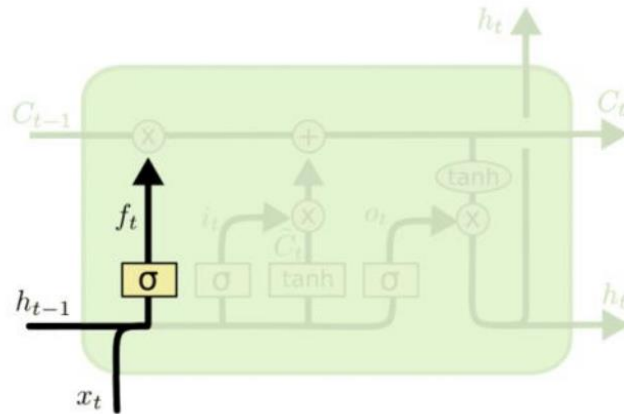


Рисунок 2.8 – Схематичне зображення першого етапу LSTM

Наступний крок – вирішити, яка нова інформація буде збережена в стані комірки (рисунок 2.9). Ця операція складається з двох частин. Спочатку сигмоїдальний шар, відомий як «шар вхідного фільтру» (input layer gate), визначає, які значення слід оновити. Потім tanh шар генерує нові можливі значення на основі поточного входу і попереднього прихованого стану, які можуть бути інтегровані у довгострокову пам'ять мережі (стан комірки).

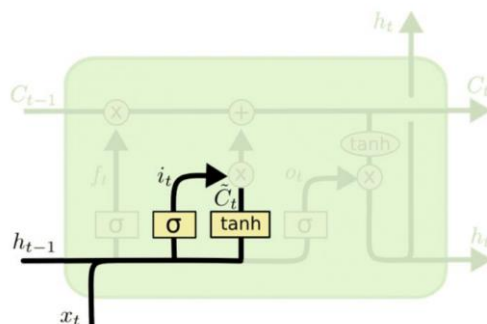


Рисунок 2.9 – Схематичне зображення другого етапу LSTM

На третьому етапі (рисунок 2.10) проводиться операція заміни стану. Під час цього етапу виконується множення значення попереднього стану на значення, що отримані з «шару фільтра забування», та їх підсумовування з отриманими значеннями з «шару вхідного фільтра».

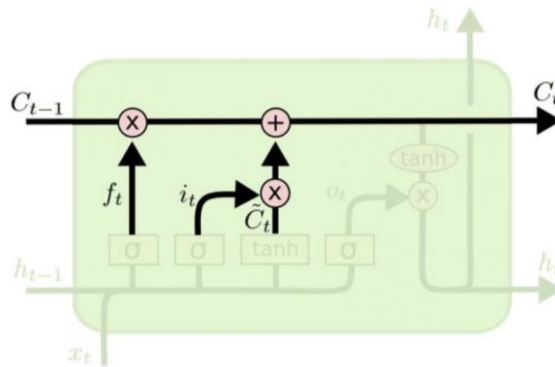


Рисунок 2.10 – Схематичне зображення третього етапу LSTM

На завершальному етапі визначається, яка інформація потрібна для виведення (рисунок 2.11).

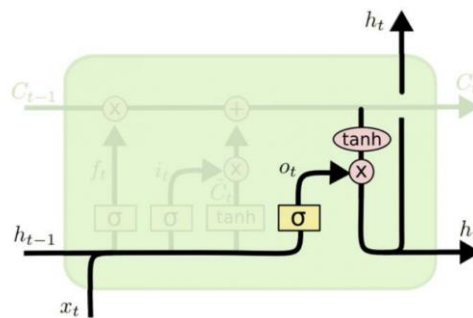


Рисунок 2.11 – Схематичне зображення останнього етапу LSTM мережі

Для визначення, яка інформація потрібна для виведення, значення стану комірки проходять через шар  $\tanh$ , що повертає значення від  $-1$  до  $1$ . Потім ці значення множаться на значення сигмоїдального шару, що дозволяє виводити лише необхідну інформацію [33].

## 3 ПРОГРАМНА РЕАЛІЗАЦІЯ

В даному розділі розглядаються програмні засоби, які були використані для реалізації поставленої задачі щодо прогнозування середньорічної температури повітря за допомогою штучних нейронних мереж (ШНМ). Розглянуті інструментальні засоби, такі як середовище розробки, мова програмування, основні бібліотеки та інші допоміжні інструменти для роботи з підходами, що відносяться до машинного навчання, зокрема, штучними нейронними мережами. Також розглянуті застосовані в роботі реалізації моделей прогнозування часових рядів на основі ШНМ.

### 3.1 Середовище розробки

Інтегроване середовище розробки (IDE) є комплексним програмним інструментом, що дозволяє розробникам здійснювати весь цикл розробки програмного забезпечення: написання коду, його виконання, компіляцію, налагодження та тестування. IDE забезпечують зручність роботи завдяки наявності численних функцій, які полегшують процес створення програмного забезпечення. Серед таких функцій – автодоповнення коду, інтеграція з системами контролю версій, інструменти для налагодження та інші.

Існує багато IDE, які підтримують різні мови програмування. Деякі з IDE, включаючи IntelliJ IDEA, Eclipse, MyEclipse та NetBeans, дозволяють працювати з кількома мовами програмування, надаючи розробникам гнучкість у виборі технологій для проектів.

Для вирішення задач прогнозування та роботи з нейронними мережами була обрана мова програмування Python. Ця мова надає інструменти для машинного навчання та аналізу даних, включаючи бібліотеки, такі як TensorFlow, Keras, Scikit-learn, NumPy та Pandas. Python також відзначається

своєю простотою, що робить його доступним для швидкого освоєння та використання в дослідницьких проектах.

Для розробки програм на Python можна використовувати різноманітні інтегровані середовища розробки, зокрема:

- pyCharm: професійне IDE для розробки на Python, яке надає достатню кількість інструментів для рефакторингу, аналізу коду та налагодження;

- spyder: IDE, орієнтоване на наукові обчислення та аналіз даних, яке інтегрується з бібліотеками для машинного навчання та статистики;

- pyDev: плагін для Eclipse, який дозволяє розробляти проекти на Python, забезпечуючи можливість роботи з великим набором функцій цього IDE;

- visual Studio Code: легкий та розширюваний редактор коду з підтримкою Python через відповідні розширення;

- jupyter Notebook: веб-застосунок для створення та обміну документами, що містять код, візуалізації та текст, що є необхідним для аналізу даних та експериментів з моделями машинного навчання.

Кожне з описаних середовищ має свої особливості та переваги. Вибір конкретного IDE залежить від уподобань розробника та специфіки проекту. Розробник обирає те середовище, яке буде найбільш зручним та ефективним для його роботи.

Завдяки використанню мови програмування Python, наявних бібліотек та засобів для роботи з моделями машинного навчання, відповідних інтегрованих середовищ розробки можна ефективно вирішувати задачі прогнозування середньорічної температури повітря за допомогою штучних нейронних мереж. Саме тому у даній кваліфікаційній роботі було вирішено використовувати PyCharm, тому що вона має широкий набір інструментів для всіх етапів розробки програмного забезпечення, від написання коду до його тестування та розгортання.

### 3.2 Використання програмних засобів машинного навчання

Python є високоефективною мовою програмування, яка підтримує різноманітні парадигми програмування, що дозволяє досягти високої продуктивності розробників та якості програмного забезпечення. Особливістю Python є забезпечення дотримання правил та норм написання кодової бази, завдяки чому синтаксис Python дозволяє описувати структуру програм з меншим обсягом коду. Це робить процес розробки більш зручним та ефективним. Крім того, Python підтримує повторне використання коду завдяки наявності модулів та пакетів, що полегшує управління та підтримку програмних проектів. Основні переваги Python:

- читабельність коду: синтаксис Python орієнтований на те, щоб бути зрозумілим та легким для читання. Це знижує витрати на підтримку коду та дозволяє програмістам ефективніше взаємодіяти в команді;

- лаконічний синтаксис: Python дозволяє описувати логіку програм меншим обсягом коду в порівнянні з багатьма іншими мовами програмування. Це досягається завдяки використанню високорівневих конструкцій та вбудованих функцій;

- повторне використання коду та модульність: Python підтримує модульність через використання модулів та пакетів. Це дозволяє програмістам організовувати код в окремі логічні блоки, що полегшує його повторне використання та обслуговування;

- наявна екосистема бібліотек: Python має багато сторонніх бібліотек та модулів, які розширюють його функціональні можливості. Наприклад, бібліотеки для роботи з даними (Pandas, NumPy), машинного навчання (TensorFlow, Keras, Scikit-learn), веб-розробки (Django, Flask) та багато іншого;

- платформонезалежність: програми, написані на Python, можуть виконуватися на різних платформах без змін в коді. Це забезпечує високу гнучкість при розробці кросплатформених застосунків;

- велика та активна спільнота: Python має велику та активну спільноту користувачів та розробників, що забезпечує доступ до численних ресурсів, документації, форумів та підтримки.

Для реалізації поставленої задачі прогнозування середньорічної температури повітря була обрана мова програмування Python завдяки її простоті, гнучкості та потужним інструментам для машинного навчання та роботи з даними. Зокрема, було використано такі бібліотеки:

- numPy та Pandas для обробки та аналізу даних;
- matplotlib для візуалізації результатів;
- scikit-learn для нормалізації даних;
- tensorflow та Keras для побудови та тренування моделей нейронних мереж.

Завдяки цьому інструментарію можливе ефективне виконання всіх етапів аналізу та прогнозування часових рядів, від підготовки даних до побудови моделей та оцінки їхньої точності.

### 3.2.1 Клас Sequential

Клас Sequential забезпечує реалізацію лінійної структури в моделях глибокого навчання, де шари розміщені послідовно один за одним. У моделі Sequential кожен шар має один вхід і один вихід, передаючи інформацію від одного шару до наступного в послідовному порядку.

Основні методи класу Sequential включають:

- add(), який використовується для додавання шарів до моделі. Шари додаються послідовно, і порядок їх додавання визначає порядок обробки даних. Метод приймає як аргумент об'єкт шару, який додається до моделі;
- метод compile(), який використовується для конфігурації моделі з функцією втрат (loss), оптимізатором (optimizer) та метриками (metrics) для навчання. Приймає як аргументи оптимізатор (наприклад, 'adam'), функцію втрат (наприклад, 'mse') та список метрик для оцінки продуктивності моделі;

- метод `fit()`, який використовується для навчання моделі на основі наданих даних. Приймає як аргументи вхідні дані (`x`), цільові значення (`y`), а також інші параметри, такі як кількість епох (`epochs`), розмір пакету (`batch_size`) та частку даних для валідації (`validation_split`);

- `predict()` використовується для отримання прогнозів на основі нових вхідних даних. Приймає як аргумент вхідні дані і повертає прогнозовані значення.

### 3.2.2 Клас Dense

Клас Dense використовується для реалізації базових шарів у штучних нейронних мережах. Ці базові шари є повнозв'язковими (`fully connected layers`), де кожен нейрон вхідного шару з'єднаний з усіма нейронами вихідного шару.

Об'єкти класу Dense мають такі параметри:

- `units` – ціле число, яке є обов'язковим параметром і визначає кількість нейронів у шарі. Наприклад, при `units=64` буде створений шар з 64 нейронами;

- `activation` – функція активації, що застосовується до виходів нейронів. Це може бути одна з відомих функцій активації, таких як `relu`, `sigmoid`, `tanh`, `softmax` або інша функція активації. За замовчуванням цей параметр може бути відсутнім (`None`).

### 3.2.3 Клас LSTM

Клас LSTM (`Long Short-Term Memory`) реалізує модель LSTM, яка є різновидом рекурентних нейронних мереж (`RNNs`). Модель LSTM здатна ефективніше обробляти послідовні дані, зберігаючи інформацію про довгострокові залежності в послідовностях (наприклад, часові ряди). Об'єкти класу LSTM мають такі параметри:

- `units` – ціле число, що визначає кількість блоків пам'яті LSTM (або нейронів) у шарі;
- `activation` – функція активації, яка використовується для активації блоків пам'яті. За замовчуванням встановлена функція `tanh`, але можна використовувати й інші функції активації, такі як `sigmoid` або `relu`.
- `input_shape` – це форма вхідних даних до шару таких як довжина послідовності, кількість характеристик.

### 3.3 Реалізація алгоритму прогнозування

Для короткострокового прогнозування часових рядів використано зібрані впродовж певного часу дані середньорічної температури повітря. Дані завантажуються в розроблене програмне забезпечення вручну і використовуються як для навчання моделей, так і для прогнозування майбутніх значень. Для навчання моделей дані проходять підготовку, розбиваються на окремі вибірки вхідних та вихідних значень залежно від величини горизонту прогнозування.

Після завантаження даних необхідно визначити горизонт прогнозування та параметри всіх моделей штучних нейронних мереж, такі як кількість шарів та кількість нейронів у кожному шарі.

Дані для навчання та прогнозування надходять одночасно до моделей багат шарового перцептронів та довготривалої короткострокової пам'яті, що дозволяє отримати результати обчислювальних експериментів для всіх реалізованих моделей прогнозування одразу.

### 3.4 Підготовка даних

Для навчання штучної нейронної мережі необхідно поділити вхідну послідовність даних на кілька шаблонів вхідних і вихідних значень, щоб створити навчальну вибірку (лістинг 3.1).

### Лістинг 3.1 – Підготовка даних

```
def create_sequences(data, seq_length, horizon):
    sequences = []
    labels = []
    for i in range(len(data) - seq_length - horizon + 1):
        seq = data[i:i + seq_length]
        label = data[i + seq_length:i + seq_length + horizon]
        sequences.append(seq)
        labels.append(label)
    return np.array(sequences), np.array(labels)
seq_length = 5 # Довжина послідовності
horizon = 5 # Горизонт прогнозу
X_train_lstm, y_train_lstm = create_sequences(train_normalized,
seq_length, horizon)
X_test_lstm, y_test_lstm = create_sequences(test_normalized,
seq_length, horizon)
X_train_lstm = X_train_lstm.reshape((X_train_lstm.shape[0],
seq_length, 1))
X_test_lstm = X_test_lstm.reshape((X_test_lstm.shape[0],
seq_length, 1))
X_train_mlp, y_train_mlp = create_sequences(train_normalized,
seq_length, horizon)
X_test_mlp, y_test_mlp = create_sequences(test_normalized,
seq_length, horizon)
```

Перші елементи часового ряду використовуються як вхідні дані, а наступні елементи, або група елементів залежно від горизонту прогнозування, використовуються як вихідні дані для прогнозу.

### 3.5 Реалізація моделі довгої короткострокової пам'яті

Для реалізації моделі довгої короткострокової пам'яті (LSTM) використовувалася бібліотека Keras у Python, де модель створювалася за допомогою класу Sequential (лістинг 3.2).

### Лістинг 3.2 – Модель довгої короткострокової пам'яті

```
model_lstm = Sequential()
model_lstm.add(LSTM(50, activation='relu',
input_shape=(seq_length, 1)))
```

```

model_lstm.add(Dense(horizon))
model_lstm.compile(optimizer='adam', loss='mse')
model_lstm.fit(X_train_lstm, y_train_lstm, epochs=50,
batch_size=32, validation_data=(X_test_lstm, y_test_lstm),
verbose=2)
predictions_lstm = model_lstm.predict(X_test_lstm)
predictions_lstm = scaler.inverse_transform(predictions_lstm)

```

Спочатку до моделі додається шар LSTM за допомогою функції `model_lstm.add(LSTM())`, вказуючи кількість нейронів (50), функцію активації (`relu`) та форму вхідних даних (довжина послідовності та розмірність вхідних даних). Після цього додається повнозв'язний шар через функцію `model_lstm.add(Dense())`, де вказується кількість вихідних нейронів, що відповідає горизонту прогнозу.

Далі модель компілюється за допомогою методу `compile()`, де задається оптимізатор (`adam`) та функцію втрат (`mse`). Після цього модель навчається, для чого передавалися навчальні дані та мітки, вказуючи кількість епох (50) та розмір пакету (32). Навчальні дані (`X_train_lstm`) та мітки (`y_train_lstm`) використовувалися для тренування моделі.

Після навчання модель застосовується для прогнозування, передаючи нові дані через функцію `model_lstm.predict()`. Прогнозовані значення потім інвертувалися назад до початкового масштабу за допомогою `scaler.inverse_transform()`.

### 3.6 Реалізація моделі багатошарового перцептронну

Для реалізації моделі багатошарового перцептрона (MLP) використовується бібліотека Keras у Python. Моделі створюються за допомогою класу `Sequential` (лістинг 3.3).

#### Лістинг 3.3 – Модель багатошарового перцептронну

```

# Створення та компіляція моделі MLP
model_mlp = Sequential()

```

```

model_mlp.add(Dense(50, activation='relu',
input_dim=seq_length))
model_mlp.add(Dense(horizon))
model_mlp.compile(optimizer='adam', loss='mse')
# Тренування моделі MLP
model_mlp.fit(X_train_mlp, y_train_mlp, epochs=50,
batch_size=32, validation_data=(X_test_mlp, y_test_mlp),
verbose=2)
# Прогнозування на тестовому наборі для MLP
predictions_mlp = model_mlp.predict(X_test_mlp)
predictions_mlp = scaler.inverse_transform(predictions_mlp)

```

Спочатку створюється модель, до якої додається щільний шар через функцію `model_mlp.add(Dense())`, вказуючи кількість нейронів (50) та функцію активації (`relu`). Потім додається вихідний шар, де кількість нейронів відповідає горизонту прогнозу.

Модель MLP компілюється за допомогою методу `compile()`, де задаються оптимізатор (`adam`) та функція втрат (`mse`). Навчання моделі проводиться з використанням навчальних даних та міток, вказуючи кількість епох (50) та розмір пакету (32). Після навчання модель MLP використовувалася для прогнозування, де нові дані передавалися через функцію `model_mlp.predict()`, а прогнозовані значення інвертувалися до початкового масштабу за допомогою `scaler.inverse_transform()`.

## 4 АНАЛІЗ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

### 4.1 Досліджувані параметри моделей

У дипломній роботі, присвяченій короткостроковому прогнозуванню середньорічної температури повітря за допомогою штучних нейронних мереж, детально досліджено вплив ряду параметрів моделей, що використовувались. Основними моделями, які були розглянуті, є багатошаровий перцептрон (MLP), довга короткострокова пам'ять (LSTM).

Для багатошарового перцептронну (MLP) досліджувалися такі параметри, як кількість прихованих шарів та кількість нейронів у кожному шарі. Кількість шарів змінювалась від одного до кількох, а кількість нейронів у кожному шарі – від 10 до 100. Було обрано активаційну функцію ReLU (Rectified Linear Unit) для всіх прихованих шарів, оскільки вона сприяє кращій збіжності та ефективності моделі.

Для моделі довгої короткострокової пам'яті (LSTM) досліджувалися кількість LSTM-шарів та кількість нейронів у кожному шарі. Кількість LSTM-шарів варіювалася від одного до двох, а кількість нейронів – від 50 до 200. Функція активації tanh використовувалася для блоків пам'яті, тоді як для гейтів обрана функція активації sigmoid. Ця комбінація функцій дозволяє моделі ефективно знаходити довгострокові залежності в даних.

Для всіх моделей було також досліджено вплив гіперпараметрів, таких як розмір мініпакетів (batch size) та кількість епох навчання. Ці параметри були оптимізовані для досягнення найкращої продуктивності моделей на тестових даних. Розмір мініпакетів змінювався від 16 до 64, а кількість епох – від 50 до 200, залежно від конкретної моделі та результатів валідації.

Параметр seq\_length визначає розмір вибірки даних часового ряду, що використовується для навчання моделей штучних нейронних мереж. Збільшення значення параметру seq\_length робить навчання моделі більш

складним, але водночас може зменшити ризик перенавчання. Проте, якщо `seq_length` занадто великий, це може призвести до негативного впливу на точність прогнозування через перенасичення інформацією або втрату релевантності даних.

Параметр `train_size` визначає розмір вибірки даних, що використовується для виконання прогнозування часового ряду.

Параметр `horizont` визначає розмір горизонту прогнозування, тобто на скільки значень наперед відбувається прогнозування.

Параметр `epoch` визначає кількість ітерацій навчання. Більше значення цього параметру зазвичай призводить до підвищення точності моделі, але занадто велика кількість епох може викликати ефект перенавчання, коли модель надмірно адаптується до навчальних даних і втрачає здатність до узагальнення.

Параметр `units` визначає кількість нейронів у шарі. Більша кількість нейронів дозволяє моделі вловлювати більш складні закономірності, але при цьому збільшує обсяг пам'яті, необхідний для зберігання моделі.

Метрики MAE та MAPE використовуються для оцінки точності моделей. MAE обчислюється шляхом підсумовування абсолютних значень різниць між прогнозованими значеннями та фактичними значеннями цільової змінної, а потім поділу цієї суми на кількість спостережень. Це дозволяє оцінити середнє абсолютне відхилення прогнозів від фактичних значень. MAPE обчислюється шляхом знаходження абсолютних значень відсоткових різниць між прогнозами моделі та фактичними значеннями, а потім поділу цієї суми на кількість спостережень. Це дозволяє оцінити середнє відсоткове відхилення прогнозів від фактичних значень.

Ці параметри та їх значення були проаналізовані для визначення найефективнішої конфігурації моделей для короткострокового прогнозування середньорічної температури повітря. Результати показали, що кожна з моделей має свої переваги в залежності від характеристик вхідних даних та мети прогнозування.

## 4.2 Базові конфігурації моделей

Для моделей штучних нейронних мереж були визначені наступні базові гіперпараметри.

Багатошаровий перцептрон (MLP):

- кількість одиниць (units): 50;
- функція активації (activation): reLU;
- оптимізатор (optimizer): rdam;
- функція втрат (loss): mse.

Довготривала короткострокова пам'ять (LSTM):

- кількість одиниць (units): 50;
- функція активації (activation): reLU;
- оптимізатор (optimizer): adam;
- функція втрат (loss): mse.

Загальні гіперпараметри:

- епохи (epochs): 50;
- горизонт (horizon): 8.

Функція активації ReLU є простою в обчисленні та має ряд переваг. Вона ефективно усуває проблему згасання градієнта, яка може виникати під час навчання глибоких нейронних мереж. ReLU моделює нелінійні залежності та створює розріджені активації, коли багато нейронів неактивні (мають нульові значення). Функція універсальна і придатна для будь-яких завдань, не вимагаючи великих обчислювальних потужностей. Усі негативні значення функція замінює на нуль і передає позитивні значення без змін. Функція ReLU активує лише окремі нейрони, залишаючи інші неактивними. Завдяки цьому мережа може навчатися на більш розріджених уявленнях даних і знижується ризик перенавчання.

Використовується оптимізатор Adam (Adaptive Moment Estimation) який є методом стохастичної оптимізації, який оновлює ваги моделі з урахуванням як поточного градієнта, так і історії попередніх градієнтів. Це

дозволяє прискорити сходимість до оптимального рішення. Adam автоматично адаптує швидкість навчання для кожного параметра на основі спостережуваних градієнтів, що дозволяє більш ефективно навчати модель на різних типах даних і з різною архітектурою нейронної мережі. Основні переваги Adam включають простоту використання, адаптивну швидкість навчання та врахування історії градієнтів.

Функція середньоквадратичної похибки (MSE) використовується для оцінки помилки моделі під час навчання. MSE вимірює середнє значення квадратів різниць між прогнозами моделі та фактичними значеннями цільової змінної. Середньоквадратична помилка обчислюється шляхом знаходження квадратів різниць між прогнозами моделі та фактичними значеннями, а потім поділу цієї суми на кількість спостережень. Мінімізація функції втрат MSE у процесі навчання допомагає моделі знаходити оптимальні значення параметрів для досягнення відповідності між прогнозами та реальними значеннями цільових змінних.

### 4.3 Залежність результатів від параметрів прогнозування

Для дослідження впливу різних параметрів на ефективність моделей штучних нейронних мереж були використані базові конфігурації, вказані у розділі 4.2. Моделі були навчені з різними значеннями параметрів, такими як розмір вибірки даних для навчання (`train_size`), розмір вибірки даних для прогнозування (`seq_length`) та горизонт прогнозування (`horizon`). Це дозволило досліджувати, як зміна цих параметрів впливає на точність прогнозів моделей.

Такий підхід дозволяє отримати більше інформації про те, які із параметрів, а саме розмір вибірки даних для навчання (`train_size`), розмір вибірки даних для прогнозування (`seq_length`) та горизонт прогнозування (`horizon`), є найбільш важливими для покращення результатів прогнозування.

Таблиця 4.1 – Залежність результатів від розміру вибірки даних для навчання моделі довгої короткострокової пам'яті

№	Параметри			Помилка	
	train_size	seq_length	horizon	MAE	MAPE
1	20%	10	1	1,00	1,90
2	30%	10	1	1,04	1,98
3	40%	10	1	0,91	1,74
4	50%	10	1	1,24	2,32
5	60%	10	1	1,31	2,44
6	70%	10	1	1,29	2,38
7	80%	10	1	1,11	2,07
8	90%	10	1	1,47	2,71

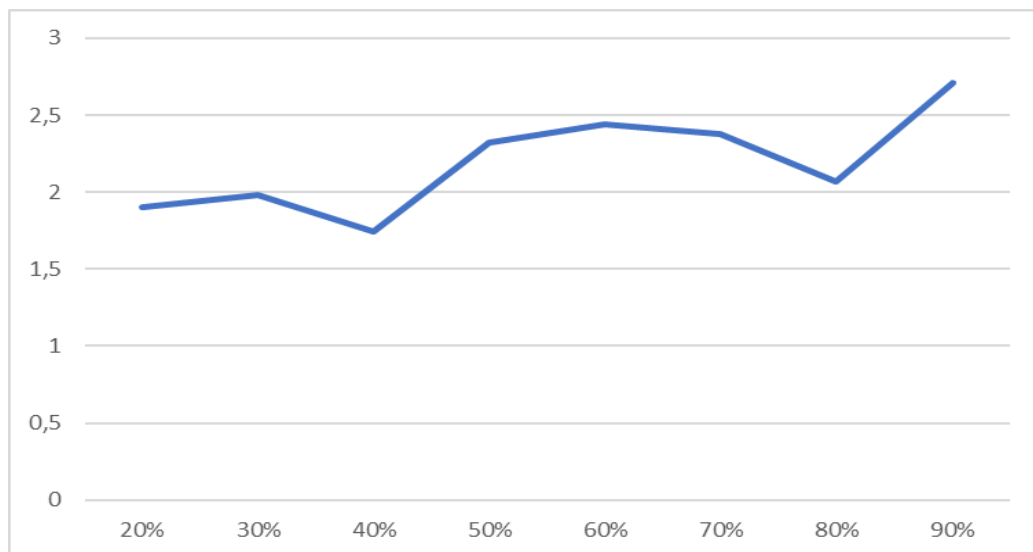


Рисунок 4.1 – MAPE в залежності від розміру вибірки даних для навчання моделі довгої короткострокової пам'яті

Збільшення розміру вибірки сприяє зниженню MAPE (таблиця 4.1). Найкращий результат MAPE (найнижче значення) досягається при тренуванні моделі з розміром вибірки приблизно 40% від загального обсягу даних (рисунок 4.1).

Таблиця 4.2 – Залежність результатів від розміру вибірки даних для навчання моделі багатошарового перцептронну

№	Параметри			Помилка	
	train_size	seq_length	horizon	MAE	MAPE
1	20%	10	1	0,74	1,41
2	30%	10	1	0,84	1,61
3	40%	10	1	0,84	1,59
4	50%	10	1	0,81	1,53
5	60%	10	1	0,85	1,60
6	70%	10	1	0,83	1,54
7	80%	10	1	0,88	1,65
8	90%	10	1	0,29	0,54

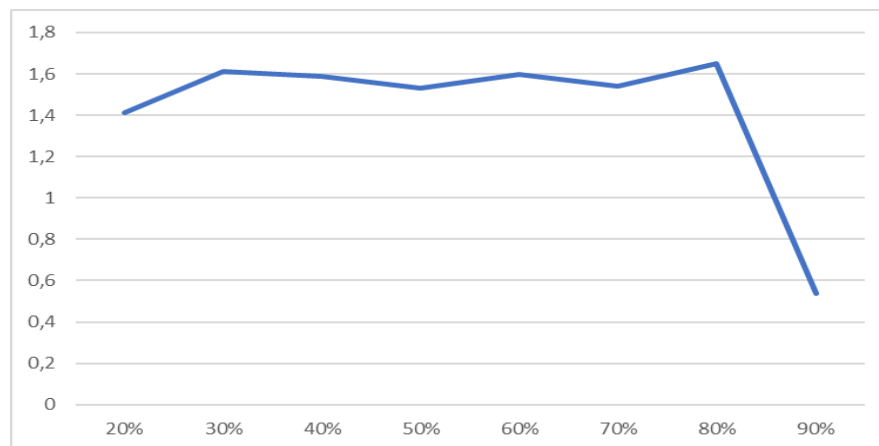


Рисунок 4.2 – MAPE в залежності від розміру вибірки даних для навчання багатошарового перцептронну

З ростом розміру тренувальної вибірки MAPE схильно зменшується до певної межі, після чого може стабілізуватись або навіть зростати.

Найнижче значення MAPE, а отже і найкраща точність прогнозів, зафіксована при використанні 90% тренувальної вибірки, що становило 0,54% (таблиця 4.2).

Це свідчить про те, що збільшення тренувальної вибірки допомагає покращити точність моделі, проте існує певний момент насичення, коли додаткове збільшення тренувальної вибірки може не призвести до значного подальшого покращення точності (рисунок 4.2).

Таблиця 4.3 – Залежність результатів від розміру вибірки даних для прогнозування моделі довгої короткострокової пам'яті

№	Параметри			Помилки	
	train_size	seq_length	horizon	MAE	MAPE
1	80%	3	1	1,77	3,26
2	80%	6	1	1,39	2,58
3	80%	9	1	1,29	2,39
4	80%	12	1	1,36	2,52
5	80%	15	1	1,26	2,33
6	80%	18	1	1,36	2,50
7	80%	21	1	1,18	2,17

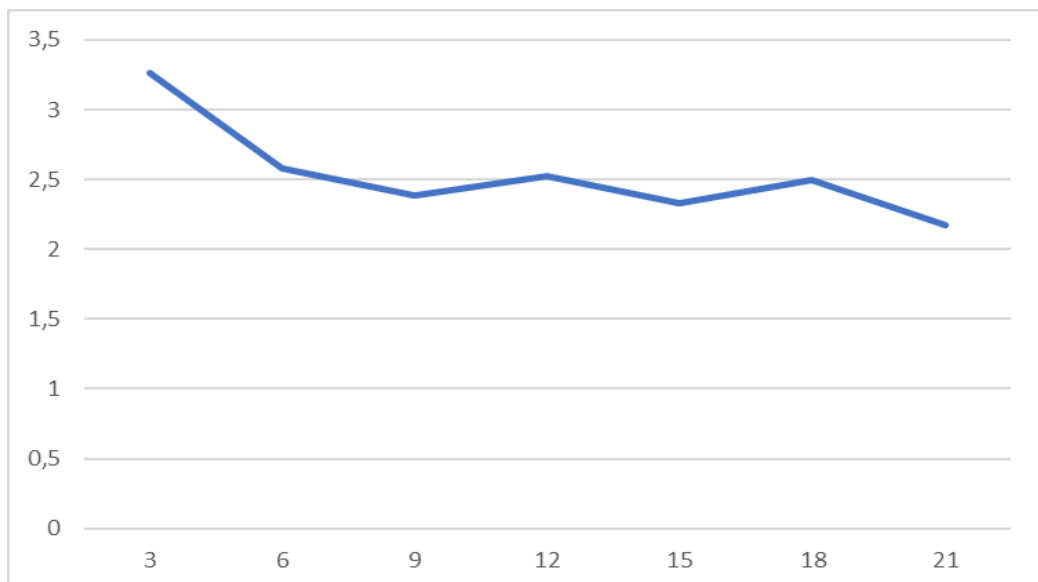


Рисунок 4.3 – Залежність MAPE від розміру вибірки для прогнозування при використанні LSTM

Аналіз залежності середньої абсолютної помилки у відсотках (MAPE) від довжини послідовностей (`seq_length`) при використанні моделі LSTM показує, що збільшення довжини послідовностей, загалом, сприяє зниженню помилки прогнозування.

Найгірший результат, з MAPE рівним 3,26%, був отриманий при використанні послідовностей довжиною 3 (таблиця 4.3).

По мірі збільшення довжини послідовностей до 9, MAPE поступово знижувалась, досягнувши 2,39%. Подальше збільшення довжини до 21 призвело до найбільш точних прогнозів з MAPE рівним 2,17%.

Це свідчить про те, що використання більш довгих послідовностей даних для навчання LSTM моделі дозволяє покращити якість прогнозування (рисунок 4.3).

Залежність середньої абсолютної помилки у відсотках (MAPE) від довжини послідовностей (`seq_length`) при використанні моделі багатосарового перцептронну (MLP) (таблиця 4.4) показує, що якість прогнозування покращується зі збільшенням довжини послідовностей. При `seq_length` рівному 3, MAPE становила 1,46%. Збільшення `seq_length` до 6 знизило помилку до 1,40%. Однак подальше збільшення до 9 і 12 призвело до зростання MAPE до 1,58% і 1,61% відповідно.

Найбільш значне покращення прогнозування спостерігалось при збільшенні `seq_length` до 15, 18 і 21, де MAPE становила 0,88%, 0,61% і 0,76% відповідно. Особливо виділяється значення `seq_length` 18, яке показало найнижчу помилку 0,61%. Це свідчить про те, що для моделі MLP оптимально використовувати довші послідовності, що дозволяє суттєво підвищити точність прогнозування (рисунок 4.4).

MAPE не демонструє постійного зростання чи зменшення зі збільшенням горизонту прогнозування. Спостерігається коливання, що може вказувати на те, що має різну ефективність прогнозування залежно від горизонту прогнозу.

Таблиця 4.4 – Залежність результатів від розміру вибірки даних для прогнозування моделі багат шарового перцептрону

№	Параметри			Показники	
	train_size	seq_length	horizon	MAE	MAPE
1	80%	3	1	0,78	1,46
2	80%	6	1	0,75	1,40
3	80%	9	1	0,84	1,58
4	80%	12	1	0,86	1,61
5	80%	15	1	0,48	0,88
6	80%	18	1	0,33	0,61
7	80%	21	1	0,41	0,76

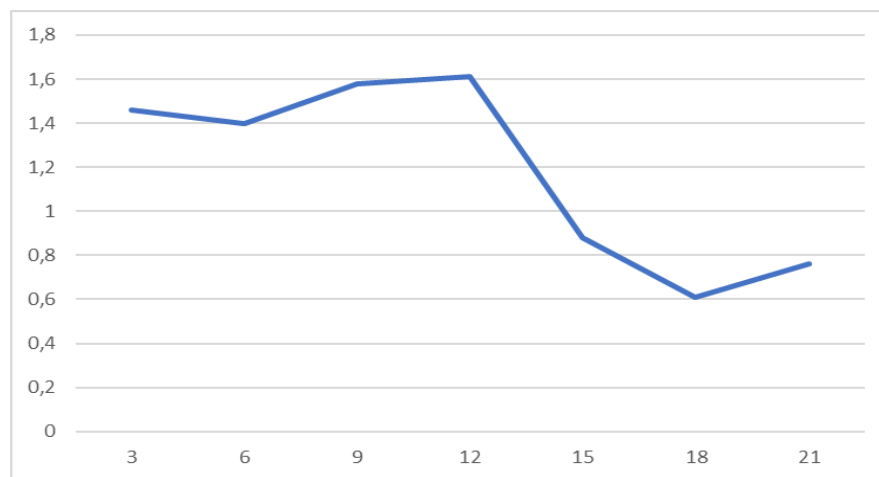


Рисунок 4.4 – Залежність MAPE від розміру вибірки для прогнозування при використанні MLP

Найнижча помилка спостерігається при горизонті прогнозування 12, де MAPE становить 1,83%. Це свідчить про те, що ефективність прогнозування моделі висока на цьому горизонті.

Найвища помилка спостерігається при горизонті прогнозування 8, де MAPE становить 3,45%. Це вказує на те, що ефективність прогнозування моделі низька на цьому горизонті (таблиця 4.5).

Для горизонтів прогнозування 1, 2, 4, 6, 10, і 14 MAPE знаходиться в діапазоні від 2,05% до 2,77%, що є помірними значеннями і вказує на середню точність прогнозування (рисунок 4.5).

Таблиця 4.5 – Залежність результатів від горизонту прогнозування для моделі довгої короткострокової пам'яті

№	Параметри			Помилка	
	train_size	seq_length	horizon	MAE	MAPE
1	80%	10	1	1,11	2,05
2	80%	10	2	1,49	2,77
3	80%	10	4	1,22	2,27
4	80%	10	6	1,47	2,72
5	80%	10	8	1,86	3,45
6	80%	10	10	1,26	2,34
7	80%	10	12	0,99	1,83
8	80%	10	14	1,05	1,95

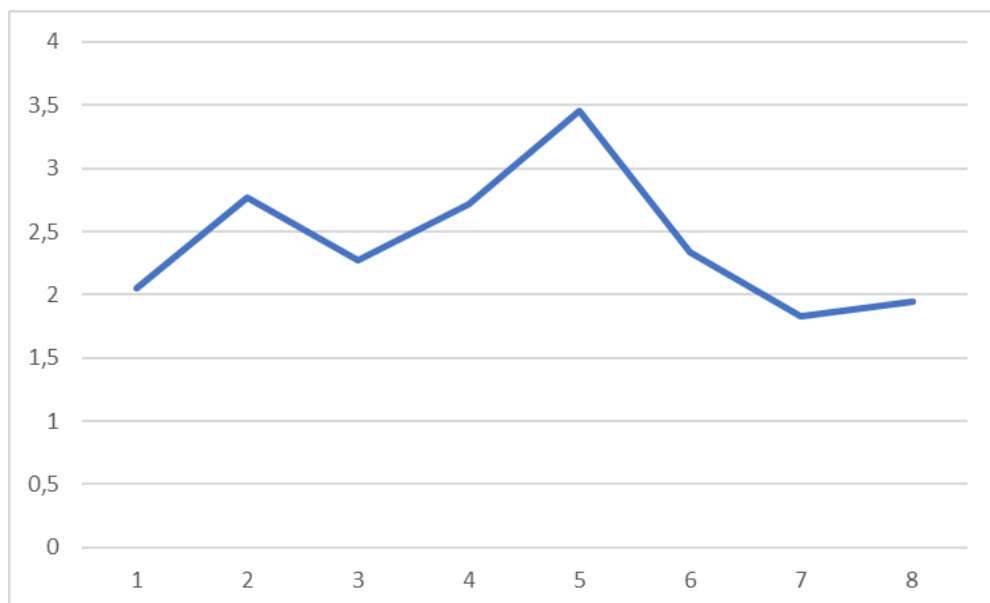


Рисунок 4.5 – Залежність MAPE від горизонту прогнозування при використанні LSTM

Зі збільшенням горизонту прогнозування MAPE спочатку зменшується, а потім стабілізується на певному рівні. Це спостереження може вказувати на те, що при більшому горизонті модель здатна зробити більш точний прогноз, оскільки враховується більше інформації про майбутні значення.

Таблиця 4.6 – Залежність результатів від горизонту прогнозування для моделі багатошарового перцептронну

№	Параметри			Помилка	
	train_size	seq_length	horizon	MAE	MAPE
1	80%	10	1	0,92	1,72
2	80%	10	2	0,87	1,64
3	80%	10	4	1,01	1,88
4	80%	10	6	0,89	1,66
5	80%	10	8	0,84	1,57
6	80%	10	10	0,81	1,51
7	80%	10	12	0,70	1,30
8	80%	10	14	0,72	1,36

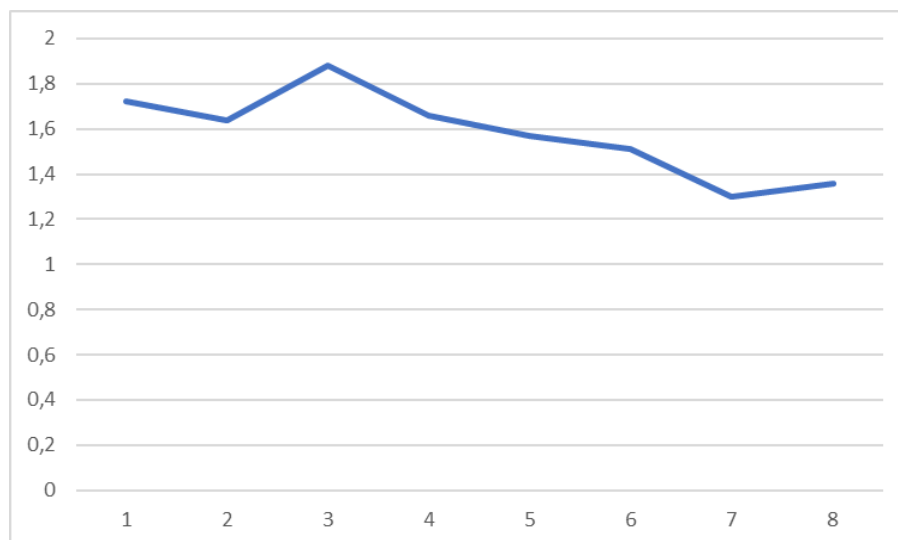


Рисунок 4.6 – Залежність MAPE від горизонту прогнозування при використанні MLP

Оптимальний горизонт прогнозування з точки зору MAPE знаходиться на рівні приблизно 12. Для цього значення горизонту MAPE становить 1,30%, що є одним з найнижчих значень серед вказаних (таблиця 4.6).

Зміна точності моделі при різних значеннях горизонту може відрізнятися залежно від конкретного набору даних і самої моделі. Із збільшенням горизонту до 12 спостерігається значне покращення точності, що може бути важливим аспектом при виборі оптимального значення для конкретного застосування (рисунок 4.6).

#### 4.4 Вплив гіперпараметрів на результати прогнозування

Для дослідження впливу гіперпараметрів (параметрів, які визначають структуру моделей) на результати прогнозування використовувалися часові ряди, де 80% даних було виділено для навчання, розмір вибірки даних для прогнозування становив 10, а горизонт прогнозування складав 1 значення.

Значення MAPE не монотонно змінюється зі збільшенням кількості нейронів у шарі LSTM. Спостерігається як зменшення, так і збільшення значення MAPE при різній кількості нейронів.

Найменше значення MAPE (2,22%) спостерігається при 140 нейронах у шарі LSTM. Це вказує на те, що для цього набору даних і моделі найкраща точність досягається при використанні 140 нейронів (таблиця 4.7).

Зі збільшенням кількості нейронів від 20 до 60, значення MAPE знижується з 3,29% до 2,29%, що вказує на покращення точності прогнозу. Однак, подальше збільшення кількості нейронів не завжди призводить до зменшення MAPE. Наприклад, при 80 нейронах MAPE збільшується до 2,73%. Зміна кількості нейронів від 60 до 160 показує, що точність прогнозу коливається, причому значення MAPE змінюються в межах від 2,22% до 2,94%. Це може вказувати на те, що після певного порогу збільшення кількості нейронів не обов'язково покращує точність, а іноді навіть погіршує її.

Таблиця 4.7 – Вплив кількості нейронів у шарі на результати прогнозування для моделі довгої короткострокової пам'яті

№	Параметри		Помилка	
	units	epochs	MAE	MAPE
1	20	50	1,77	3,29
2	40	50	1,32	2,45
3	60	50	1,24	2,29
4	80	50	1,47	2,73
5	100	50	1,28	2,37
6	120	50	1,35	2,51
7	140	50	1,19	2,22
8	160	50	1,59	2,94

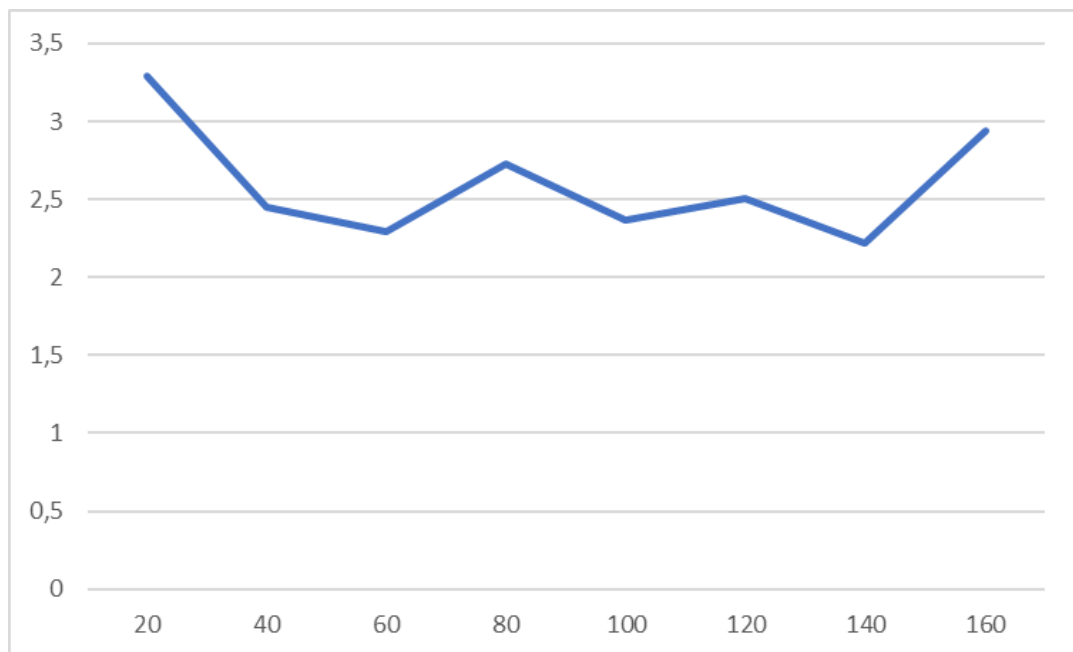


Рисунок 4.7 – Значення MAPE в залежності від кількості нейронів у шарі у моделі довгої короткострокової пам'яті

Значення MAPE, що збільшується при великій кількості нейронів (наприклад, 80 і 160 нейронів), може вказувати на перенавчання моделі, коли

модель стає надто складною і починає підлаштовуватися під шум у навчальних даних (рисунок 4.7).

Таблиця 4.8 – Вплив кількості нейронів у шарі на результати прогнозування для моделі багатошарового перцептронну

№	Параметри		Помилка	
	units	epochs	MAE	MAPE
1	20	50	0,77	1,45
2	40	50	0,87	1,62
3	60	50	0,92	1,72
4	80	50	0,86	1,61
5	100	50	0,77	1,43
6	120	50	0,97	1,81
7	140	50	0,74	1,38
8	160	50	0,96	1,79

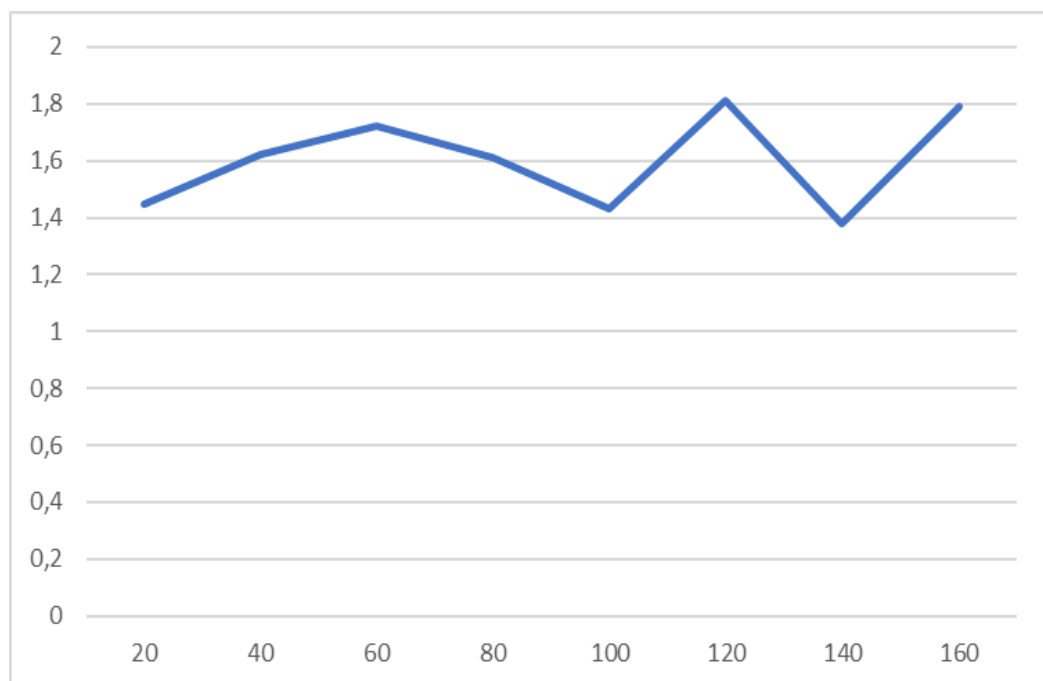


Рисунок 4.8 – Значення MAPE в залежності від кількості нейронів у моделі багатошарового перцептронну

Значення MAPE не змінюється монотонно зі збільшенням кількості нейронів. Спостерігаються як зменшення, так і збільшення значень MAPE при різній кількості нейронів.

Найменше значення MAPE (1,38%) спостерігається при 140 нейронах у шарі MLP. Це вказує на те, що для цього набору даних і моделі MLP найкраща точність досягається при використанні 140 нейронів (таблиця 4.8).

При збільшенні кількості нейронів від 20 до 100, значення MAPE змінюється в межах від 1,45% до 1,43%. Це показує, що до 100 нейронів точність моделі залишається приблизно на одному рівні з невеликими коливаннями.

Зміна кількості нейронів до 120 і більше показує більші коливання значення MAPE. Наприклад, при 120 нейронах значення MAPE підвищується до 1,81%, а при 140 нейронах знижується до 1,38%.

Збільшення кількості нейронів після 140 не завжди призводить до поліпшення точності. Наприклад, при 160 нейронах значення MAPE підвищується до 1,79%, що може вказувати на можливе перенавчання моделі (рисунок 4.8).

Зі збільшенням кількості епох навчання значення MAPE спочатку зменшується, а потім значно зростає, після чого знову знижується. Найменше значення MAPE (2,91%) досягається при 100 епохах навчання. Після цього значення MAPE починає збільшуватися (таблиця 4.9).

Починаючи з 500 епох, значення MAPE зростає до 4,58%, а потім при 900 епохах (9,71%) і досягає максимуму при 2500 епохах (31,94%).

Значне збільшення MAPE зі збільшенням кількості епох може свідчити про перенавчання моделі. Модель може запам'ятовувати дані тренувального набору, втрачаючи здатність узагальнювати на нових даних.

При 2900 епохах значення MAPE знову зменшується до 4,69%, що вказує на можливе поліпшення моделі після тривалого навчання. Проте це значення все ще більше, ніж при 100 епохах (рисунок 4.9).

Таблиця 4.9 – Вплив кількості ітерацій навчання на помилки прогнозування для моделі довгої короткострокової пам'яті

№	Гіперпараметри		Помилка	
	units	epochs	MAE	MAPE
1	50	100	1,57	2,91
2	50	500	2,47	4,58
3	50	900	5,20	9,71
4	50	1300	6,86	12,78
5	50	1700	8,11	15,11
6	50	2100	10,44	19,49
7	50	2500	17,09	31,94
8	50	2900	2,51	4,69

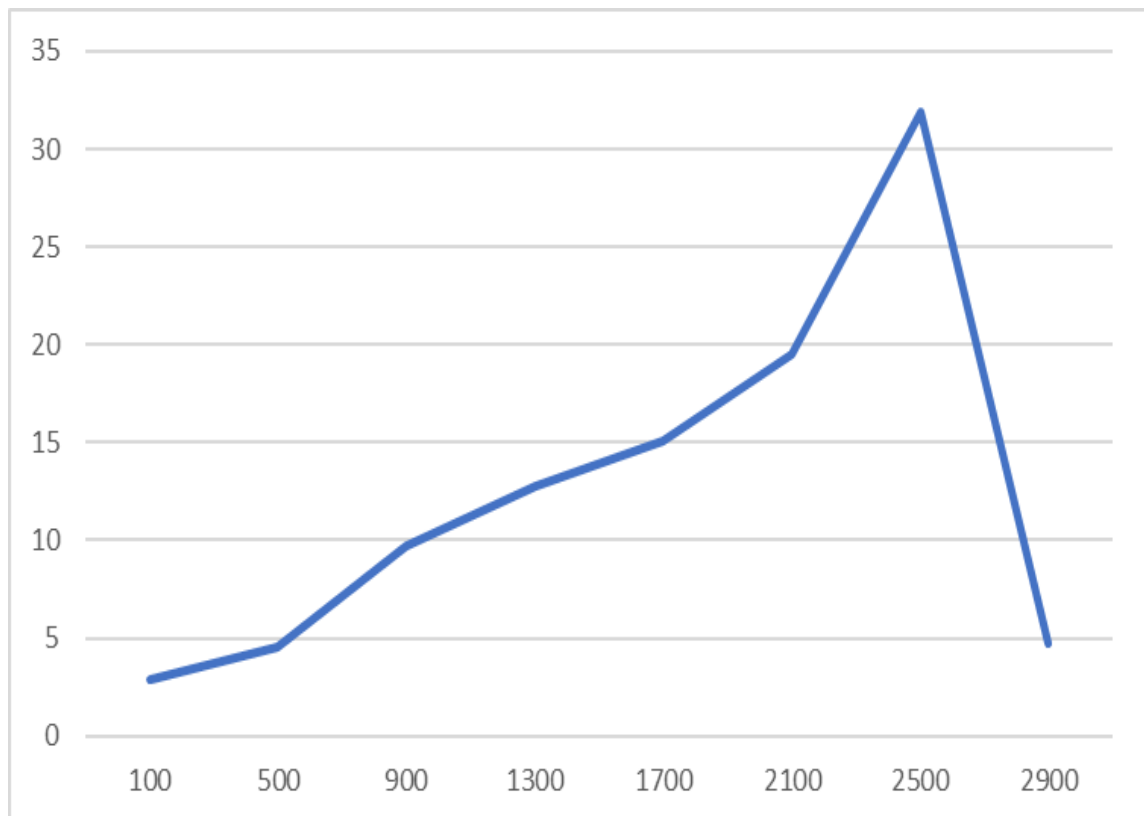


Рисунок 4.9 – Значення MAPE в залежності від ітерацій навчання для моделі довгої короткострокової пам'яті

Таблиця 4.10 – Вплив кількості ітерацій навчання на помилку прогнозування для моделі багатошарового перцептронну

№	Гіперпараметри		Помилка	
	units	epochs	MAE	MAPE
1	50	100	0,76	1,41
2	50	500	0,69	1,28
3	50	900	0,97	1,80
4	50	1300	0,51	0,95
5	50	1700	0,47	0,87
6	50	2100	0,63	1,17
7	50	2500	0,58	1,07
8	50	2900	0,27	0,50

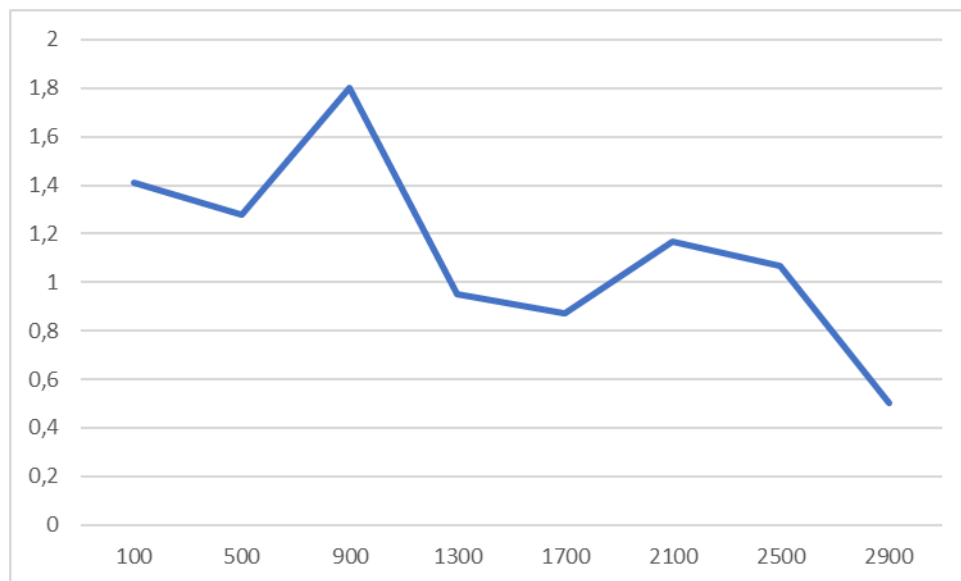


Рисунок 4.10 – Значення MAPE в залежності від кількості ітерацій навчання для моделі багатошарового перцептронну

Спостерігається значне зменшення значення MAPE зі збільшенням кількості епох навчання. Це вказує на те, що модель покращує свою здатність прогнозувати дані зі збільшенням часу навчання.

Найменше значення MAPE (0,50%) досягається при 2900 епохах навчання, що вказує на те, що модель продовжує вчитися та покращувати свої прогнози навіть після тривалого навчання (таблиця 4.10).

Значення MAPE зменшується від 1,41% при 100 епохах до 0,87% при 1700 епохах, а потім зростає до 1,17% при 2100 епохах. Після цього значення MAPE знову зменшується до 0,50% при 2900 епохах.

Існують певні коливання у значеннях MAPE (наприклад, при 900 епохах воно становить 1,80%, а при 1300 – 0,95%), що може бути спричинено періодичними уповільненнями в оптимізації моделі або локальними мінімумами.

На відміну від моделі LSTM, у даному випадку не спостерігається явного перенавчання. Модель MLP покращується навіть при великій кількості епох (рисунок 4.10).

## ВИСНОВКИ

У роботі досліджено використання засобів обчислювального інтелекту для прогнозування часових рядів температури. За результатами проведених досліджень моделі багатошарового персептрону показали кращі залежності між значенням середньої абсолютної помилки у відсотках (МАРЕ) та різними гіперпараметрами ніж модель довгої короткострокової пам'яті.

При зміні кількості нейронів у моделі багатошарового персептрону, середнє значення МАРЕ варіювалося наступним чином: 20 нейронів – 1,45%, 40 нейронів – 1,62%, 60 нейронів – 1,72%, 80 нейронів – 1,61%, 100 нейронів – 1,43%, 120 нейронів – 1,81%, 140 нейронів – 1,38%, 160 нейронів – 1,79%. Таким чином, найкраще значення МАРЕ було досягнуто при 140 нейронах у шарі, що становить 1,38%.

При зміні кількості ітерацій навчання для моделі багатошарового персептрону, значення МАРЕ також демонстрували значну варіативність: 100 ітерацій – 1,41%, 500 ітерацій – 1,28%, 900 ітерацій – 1,80%, 1300 ітерацій – 0,95%, 1700 ітерацій – 0,87%, 2100 ітерацій – 1,17%, 2500 ітерацій – 1,07%, 2900 ітерацій – 0,50%. Найкраще значення МАРЕ було досягнуто при 2900 ітераціях навчання, що становить 0,50%.

Моделі багатошарового персептрону показала значну залежність точності прогнозування від кількості нейронів у шарі та кількості ітерацій навчання. При більшій кількості ітерацій навчання модель демонструвала кращі результати, що свідчить про важливість тривалого навчання для досягнення високої точності прогнозування.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Канторович Г. Г. Аналіз часових рядів [Текст] / Г. Г. Канторович // Экономический журнал Высшей школы экономики. – 2002. – Т. 6, №4. – С. 498–523.
2. Лукашин, Ю. П. Adaptive methods for short-term time series forecasting [Текст] / Ю. П. Лукашин. – Финанси та статистика, 2003. – 416 с.
3. Вох, Г. Е. Р. Time series analysis: forecasting and control [Текст] / G. E. P. Voh, G. M. Jenkins, G. C. Reinsel. – Hoboken, N. J: John Wiley, 2008. – 746 p.
4. Parzen, E. Long memory of statistical time series modeling [Текст] / E. Parzen // NBER/NSF Time Series Conference. – Texas A&M University, 2004. – Pp. 10–11.
5. Трофименко, С.В. Modification of Irwin's method for identifying anomalous levels of time series: methodology and numerical experiments [Текст] / С. В. Трофименко, Я. И. Маршалов, Н. Н. Гриб, И. И. Колодезников // Modern problems of science and education. – №5. – С. 204–208.
6. Литтл, Р. Дж. А. Statistical analysis of missing data [Текст] / Р. Дж. А. Литтл, Д. Б. Рубин. – Финанси та статистика, 1991. – 430 с.
7. Кузовлев, В. И. A method for identifying anomalies in source data when constructing a predictive decision tree model in decision support systems [Текст] / В. И. Кузовлев, А. О. Орлов // Наука и образование. – 2012. – №9. – С. 209–218.
8. Кузовлев, В. И. Predictive data analysis method ID3O [Текст] / В. И. Кузовлев, А.О. Орлов // Наука та освіта. – 2012. – №10. – С. 305–316.
9. Бучацька, В. В. Processing of anomalous values of time series levels as a stage of comprehensive information assessment in the forecasting subsystem for the situation center [Текст] / В. В. Бучацька // Вісник адигейського державного університету. – 2013. – №3. – С. 98–102.

10. Орлов, А. О. Проблема пошуку відстаней між значеннями категоріальних атрибутів при виявленні викидів даних [Текст] / А. О. Орлов // У світі наукових відкриттів. – 2012. – №8(32). – С. 142–155.
11. Palit, A. Computational Intelligence in Time Series Forecasting [Текст] / A. Palit, D. Popovic. – 2006. – 372 p.
12. Prajakta, S. K. Time series Forecasting using Holt-Winters Exponential Smoothing. – 2015 [Електронний ресурс]. – Режим доступа: [www/ URL. http://www.it.iitb.ac.in/~praj/acads/seminar/04329008\\_ExponentialSmoothing.pdf](http://www.it.iitb.ac.in/~praj/acads/seminar/04329008_ExponentialSmoothing.pdf). – 27.03.2024 p. – Загол. з екрану.
13. Бодянский, Е. В. Искусственные нейронные сети: архитектуры, обучение, применения [Текст] / Е. В. Бодянский, О.Г. Руденко. – Харьков: Телетех, 2004. – 372 с.
14. Хайкин, С. Нейронные сети: полный курс [Текст] / С. Хайкин. – Москва: ООО «И. Д. Вильямс», 2006. – 1104 с.
15. Рутковская, Д. Нейронные сети, генетические алгоритмы и нечеткие системы [Текст] / Д. Рутковская, М. Пилиньский, Л. Рутковский. – Москва: Горячая линия – Телеком, 2006. – 452 с.
16. Дасгупта, Д. Искусственные иммунные системы и их применение [Текст] / Д. Дасгупта. – Москва: ФИЗМАТЛИТ, 2006. – 344 с.
17. Timmis, J. I. An Overview of Artificial Immune Systems [Текст] / J.I. Timmis, T. Knight, L.N. De Castro, E.H. Art // Computation in Cells and Tissues. – Springer. – 2004. – P. 51–86.
18. Tarakanov, A.O. Immunocomputing. Principles and application [Текст] / A. O. Tarakanov, V. A. Skormin, S. P. Sokolova. – Springer, 2003. – 193 p.
19. Cutello, V. Exploring the capability of immune algorithms: a characterisation of hypermutation operators [Текст] / V. Cutello, G. Nicosia, M. Parvone // Springer Lecture Notes in Computer Science. – 2004. – №3239. – Pp. 263–276.
20. Литвиненко, В. І. Побудова штучних імунних систем [Текст] / В. І. Литвиненко // Наукові праці. Комп'ютерні технології. – 2010. – Т. 134. –

№121. – С. 166–178.

21. Литвиненко, В. И. Штучні імунні системи як засіб індуктивної побудови оптимальних моделей [Текст] / В.И. Литвиненко // Проблеми управління та інформатики. – 2008. – №3. – С. 30–42.

22. Dasgupta, D. Advances in Artificial Immune Systems [Текст] / D. Dasgupta // IEEE Computational Intelligence Magazine. – 2006. – №11. – Pp. 40–49.

23. Kephart, J. O. A biologically inspired immune system for computer [Текст] / J. O. Kephart // Artificial Life: Conf., July 6-8. – 1994. – Pp. 143–168.

24. Chen, S. Computational Intelligence in Economics and Finance [Текст] / S. Chen, P. Wang. – 2007. – 53 p.

25. Бриллинджер, Д. Р. Временные ряды. Обработка данных и теория [Текст] / Д. Р. Бриллинджер. – М.: Мир, 1980. – 536 с.

26. Chatfield, C. Time-Series Forecasting [Текст] / C. Chatfield. – Chapman and Hall/CRC, 2000. – 280 с.

27. Gurney, K. An Introduction to Neural Networks [Текст] // K. Gurney, 1997. – 234 p.

28. Lachtermacher, G. Back propagation in time-series forecasting [Текст] / G. Lachtermacher, D. Fuller // Journal of forecasting. – 1995. – P. 381–393.

29. Srivastava, N. Dropout: A Simple Way to Prevent Neural Networks from Overfitting [Текст] / N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever // Journal of Machine Learning Research. – 2014. – №15. – P. 1929–1958.

30. Ahmed, N. An Empirical Comparison of Machine Learning Models for Time Series Forecasting [Текст] / N. Ahmed, A. Atiya, N. Gayar // Econometric Reviews. – 2010. – Vol. 29. – P. 594–621.

31. Voyant, C. Uncertainties in global radiation time series forecasting using machine learning: The multilayer perceptron case [Текст] / C. Voyant, G. Notton, C. Darras // Energy. – 2017. – Vol. 125. – P. 248–257.

32. Ebrahimpour, R. Mixture of MLP-experts for trend forecasting of time series: A case study of the Tehran stock exchange [Текст] / R. Ebrahimpour,

H. Nikoo, S. Masoudnia // *International Journal of Forecasting*. – 2011. – Vol. 27. – P. 804–816.

33. Understanding LSTM Networks [Електронний ресурс] – Режим доступу: [www/ URL: http://colah.github.io/posts/2015-08-Understanding-LSTMs/](http://colah.github.io/posts/2015-08-Understanding-LSTMs/) – 27.03.2024 р. – Загол. з екрану.

34. Іващенко, Г. С. Моделі глибокого навчання для прогнозування часових рядів [Текст] / Г. С. Іващенко, Д. О. Тимошенко, О. В. Близнюк, О. М. Кононенко // *Системи управління, навігації та зв'язку*. – 2024. – №1. – С. 82–87.

35. Ratnadip Adhikari An Introductory Study on Time Series Modeling and Forecasting [Текст] / Agrawal R. K., Adhikari Ratnadip // LAP Lambert Academic Publishing. – 2013. – 76 p.

36. Афанасьев В. Н. Аналіз часових рядів та прогнозування [Текст] / М. М. Юзбашев, В. Н. Афанасьев . – 2001. – С.228.

37. Korablev, N. M. Parallel immune algorithm of short-term forecasting based on model of clonal selection [Текст] / N. M. Korablev, G. S. Ivaschenko // *Radio Electronics, Computer Science, Control*. – 2014. – № 2. – P. 73–78.

38. Руденко, О. Г. Штучні нейронні мережі [Текст] / О. Г. Руденко, Є. В. Бодянський. – Харків: ТОВ «Компанія СМІТ», 2006. – С. 404.

39. Armstrong, J. S. Forecasting for Marketing [Текст] / J. S. Armstrong // *Quantitative Methods in Marketing*. – 1999. – P. 92–119.

40. Pedro, L-B. Temporal Convolutional Networks Applied to Energy-related Time Series Forecasting [Текст] / José María Luna-Romera, José María Luna-Romera // *Applied Sciences*. – 2020. – Vol. 10. – №7. – P. 23–22.

41. Müller, C. Introduction to machine learning with Python: a guide for data scientists [Текст] / C. Müller , S. Guido // 1st ed. Sebastopol: CA: O'Reilly Media. – Inc. – 2016. – 398 p.

42. Namini, S. A Comparison of ARIMA and LSTM in Forecasting Time Series [Текст] / S. Namini, N. Tavakoli, A. Namin // *IEEE International Conference on Machine Learning and Applications*. – 2018. – Vol. 17.

43. Zeng, Z. Self CNN-based time series stream forecasting [Текст] / Z. Zeng, H. Xiao, X. Zhang // *Electronics Letters*. – 2016. – Vol. 52. – P. 1857–1858.
44. Piccialli, F. Artificial intelligence and healthcare: Forecasting of medical bookings through multi-source time-series fusion [Текст] / F. Piccialli, F. Giampaolo, P. Edoardo // *Information*. – 2021. – Vol. 74. – P. 1–16.
45. Kousari, M. Introducing an operational method to forecast long-term regional drought based on the application of artificial intelligence capabilities [Текст] / M. Kousari, M. Hosseini, H. Ahani // *Theoretical and Applied Climatology*. – 2015. – Vol. 127. – P. 361–380.
46. Mostafa, M. A neuro-computational intelligence analysis of the ecological footprint of nations [Текст] / M. Mostafa, R. Natarajan // *Computational Statistics & Data Analysis*. – 2009. – Vol. 53. – P. 3516–3531.
47. Afan, H. Past, present and prospect of an Artificial Intelligence (AI) based model for sediment transport prediction [Текст] / H. Afan, A. Shafie, W. Mohtar // *Journal of Hydrology*. – 2016. – Vol. 541. – P. 902–913.