ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Система для прогнозування навантаження серверів з використанням машинного навчання»

на здобуття освітнього ступеня магістра
зі спеціальності 123 Комп'ютерної інженерії
(код, найменування спеціальності)
освітньо-професійної програми Комп'ютерна інженерія
(назва)

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

_____	*Павло ЛИСЮК*
(підпис)	*Ім'я, ПРІЗВИЩЕ здобувача*

Виконав: здобувач(ка) вищої освіти гр. КСДМ-61
Павло ЛИСЮК
Ім'я, ПРІЗВИЩЕ
Керівник: Сергій КОРОТКОВ
доктор філософії, доцент
Ім'я, ПРІЗВИЩЕ
Рецензент: _____
науковий ступінь, вчене звання
Ім'я, ПРІЗВИЩЕ

Київ 2025

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут Інформаційних технологій

Кафедра 123 Комп'ютерної інженерії

Ступінь вищої освіти Магістр

Спеціальність 123 Комп'ютерної інженерії

Освітньо-професійна програма Комп'ютерні системи та мережі

ЗАТВЕРДЖУЮ
Завідувач кафедрою Наталія ЛАЩЕВСЬКА
Ім'я, ПРІЗВИЩЕ
«_____» _____ 2025р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Лисюк Павло Олександрович
(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: Система для прогнозування навантаження серверів з використанням машинного навчання
керівник кваліфікаційної роботи доктор філософії, доцент Сергій КОРОТКОВ
(Ім'я, ПРІЗВИЩЕ, науковий ступінь, вчене звання)
затверджені наказом Державного університету інформаційно-комунікаційних технологій від «__ _____» _____2025р. № _____
2. Строк подання кваліфікаційної роботи «_____» _____ 2025р.
3. Вихідні дані до кваліфікаційної роботи: набір даних Google Cluster Trace 2011,
4. мова програмування Python; інтернет-ресурси, науковло-технічна література.

Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
1. Теоретичні та наукові засади прогнозування навантаження серверів
2. Методи машинного та глибинного навчання для прогнозування навантаження серверів
3. Методологія та проєктування системи прогнозування навантаження серверів
4. Експериментальні дослідження та аналіз результатів .
5. Перелік ілюстративного матеріалу: *презентація*
6. Дата видачі завдання «_____» _____ 2025р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Збір даних		Виконано
2	Обробка матеріалу		Виконано
3	Теоретичні та наукові засади прогнозування навантаження серверів		Виконано
4	Методи машинного та глибинного навчання для прогнозування навантаження серверів		Виконано
5	Методологія та проєктування системи прогнозування навантаження серверів		Виконано
6	Експериментальні дослідження та аналіз результатів		Виконано
7	Висновки за результатами аналізу		Виконано
8	Розробка презентації		Виконано
9	Передзахист		Виконано
10	Здача в деканат		Виконано

Здобувач(ка) вищої освіти _____ Павло ЛИСЮК
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

Керівник кваліфікаційної роботи _____ Сергій КОРОТКОВ
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня магістра: 89 стор., 11 рис., 9 таблиць, 30 джерел.

Мета роботи – розробка та дослідження системи прогнозування навантаження серверів з використанням статистичних методів, методів машинного навчання та глибинного навчання на основі часових рядів, а також оцінювання їхньої ефективності, здатності до узагальнення та стійкості до перенавчання.

Об'єкт дослідження – процес прогнозування навантаження серверів.

Предмет дослідження – методи та моделі машинного і глибинного навчання для прогнозування серверних навантажень.

Короткий зміст роботи: У роботі спроєктовано та досліджено систему прогнозування навантаження серверів, у межах якої використано методи аналізу наукових джерел, статистичні методи аналізу часових рядів, методи машинного навчання та глибинного навчання, а також методи експериментального дослідження і порівняльного аналізу. Для проведення експериментів використано набір даних Google Cluster Trace 2011, який було попередньо підготовлено для задач прогнозування. Оцінювання якості прогнозування здійснювалося з використанням стандартних метрик MSE, MAE, коефіцієнта детермінації R^2, а також метрики Naïve Ratio, що застосовувалася для аналізу наївної поведінки моделей.

У результаті виконання роботи проведено порівняльний аналіз різних підходів до прогнозування серверних навантажень, досліджено їхню здатність до узагальнення на нових часових рядах та визначено переваги й обмеження застосування методів машинного і глибинного навчання у даній задачі. Отримані результати можуть бути використані під час проєктування систем моніторингу та автоматичного масштабування серверних ресурсів у дата-центрах.

КЛЮЧОВІ СЛОВА: СЕРВЕРНЕ НАВАНТАЖЕННЯ, ПРОГНОЗУВАННЯ, ЧАСОВІ РЯДИ, МАШИННЕ НАВЧАННЯ, ГЛИБИННЕ НАВЧАННЯ, GOOGLE CLUSTER TRACE, NAÏVE RATIO, СИСТЕМА ПРОГНОЗУВАННЯ

ABSTRACT

The text part of the qualification work for the master's degree: 89 pages, 11 figures, 9 tabl.. 30 sources.

The purpose of the work is a comparative analysis of statistical methods, machine learning methods and deep learning for predicting server load based on time series, as well as assessing their effectiveness, generalization ability and resistance to overtraining.

The object of the study is the process of predicting server load.

The subject of the study is methods and models of machine and deep learning for predicting server loads.

Summary of the work: The work uses methods of analysis of scientific sources, statistical methods of time series analysis, machine learning and deep learning methods, as well as methods of experimental research and comparative analysis. The experiments were conducted using the Google Cluster Trace 2011 dataset, which was previously prepared for forecasting tasks. The quality of the forecast was assessed using standard metrics MSE, MAE, the coefficient of determination R^2, as well as the Naïve Ratio metric, which was used to analyze the naive behavior of the models.

As a result of the work, a comparative analysis of different approaches to forecasting server loads was conducted, their ability to generalize to new time series was investigated, and the advantages and limitations of using machine and deep learning methods in this task were determined. The results obtained can be used when designing monitoring systems and automatic scaling of server resources in data centers.

KEYWORDS: SERVER LOAD, FORECASTING, TIME SERIES, MACHINE LEARNING, DEEP LEARNING, GOOGLE CLUSTER TRACE, NAÏVE RATIO

ЗМІСТ

ВСТУП……....9
1 ТЕОРЕТИЧНІ ТА НАУКОВІ ЗАСАДИ ПРОГНОЗУВАННЯ НАВАНТАЖЕННЯ СЕРВЕРІВ…….……..…….....….12
1.1 Розвиток серверних та хмарних інфраструктур……..…..12
1.2 Поняття серверного навантаження та його основні характеристики…..……13
1.3 Типи серверних навантажень та фактори їх зміни……...15
1.4 Проблеми управління ресурсами серверів та дата-центрів ……..…18
1.5 Необхідність прогнозування навантаження серверів у сучасних інформаційних системах……...…….19
1.6 Часові ряди як математична модель серверних навантажень…….…..20
1.6.1 Основні властивості часових рядів……...……21
1.6.2 Тренд, сезонність та випадкові коливання……..…….……22
1.6.3 Одновимірні та багатовимірні часові ряди……..……23
1.7 Аналіз існуючих наукових досліджень у сфері прогнозування серверних навантаження……..22
2 МЕТОДИ МАШИННОГО ТА ГЛИБИННОГО НАВЧАННЯ ДЛЯ ПРОГНОЗУВАННЯ НАВАНТАЖЕННЯ СЕРВЕРІВ……...…….26
2.1 Класифікація методів прогнозування часових рядів ……..…..….26
2.2 Статистичні методи прогнозування серверних навантажень ……..……27
2.2.1 Метод ковзного середнього …….…….28
2.2.2 Авторегресійні моделі та модель ARIMA ……..……28
2.2.3 Переваги та обмеження статистичних методів ……...……....29
2.3 Методи машинного навчання для прогнозування навантаження серверів.....30
2.3.1 Лінійна та поліноміальна регресія ……..…….....31
2.3.2 Метод опорних векторів (SVM) ……...31
2.3.3 Ансамблеві методи та випадковий ліс ……..……..32
2.4 Методи глибинного навчання ……...……..34
2.4.1 Штучні нейронні мережі ……...35
2.4.2 Рекурентні нейронні мережі ……36
2.4.3 Мережі довготривалої короткочасної пам’яті (LSTM)……...……37

2.4.4 Transformer-архітектури для прогнозування часових рядів……...……40
2.5 Метрики та критерії оцінювання якості прогнозування……42
2.5.1 Середньоквадратична та середня абсолютна похибки……..….42
2.5.2 Коефіцієнт детермінації R^2……...…….44
2.5.3 Метрика Naïve Ratio та аналіз наївної поведінки моделей……..…44
3 МЕТОДОЛОГІЯ ТА ПРОЄКТУВАННЯ СИСТЕМИ ПРОГНОЗУВАННЯ НАВАНТАЖЕННЯ СЕРВЕРІВ ……..……...…47
3.1 Постановка задачі прогнозування навантаження серверів ……...……....47
3.2 Опис та обґрунтування вибору набору даних ……...…….…49
3.2.1 Характеристика набору даних Google Cluster Trace …..……..…50
3.2.2 Попередня обробка та підготовка даних ……...……52
3.3 Архітектура системи прогнозування навантаження серверів……...54
3.3.1 Загальна структура системи……..55
3.3.2 Використані програмні засоби та технології……...……56
3.4 Кластеризація навантажень серверів……..56
3.4.1 Мета застосування кластеризації у задачі прогнозування…….57
3.4.2 Вибір метрики подібності часових рядів……57
3.4.3 Метод динамічного вирівнювання часових рядів (DTW)……..58
3.4.4 Вибір та налаштування алгоритму кластеризації (DBSCAN)……59
3.5 Реалізація моделей прогнозування……..61
3.5.1 Реалізація статистичних моделей……61
3.5.2 Реалізація моделей машинного навчання……...….61
3.5.3 Реалізація моделей глибинного навчання……62
3.6 Методика проведення експериментальних досліджень……...……62
4 ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ……..……64
4.1 Налаштування експериментів та сценарії тестування……...……....64
4.2 Аналіз результатів прогнозування серверних навантажень……..…67
4.2.1 Результати однокрокового прогнозування……..……68
4.2.2 Багатокрокове прогнозування та накопичення похибок……71
4.2.3 Аналіз наївної поведінки моделей……72
4.2.4 Узагальнення результатів……...…...74

4.3 Порівняльний аналіз ефективності моделей прогнозування…..……74
4.3.1 Порівняння за точністю прогнозування……..75
4.3.2 Порівняння моделей за стійкістю до наївної поведінки…….81
4.3.3 Обчислювальна складність та практичні аспекти застосування……82
4.4 Інтерпретація результатів статистичних методів……..……82
4.5 Аналіз поведінки моделей машинного навчання……..……83
4.6 Переваги та обмеження методів глибинного навчання…….…….83
4.7 Вплив горизонту прогнозування та кількості вхідних даних…….....84
4.8 Узагальнення результатів дослідження……..……85
ВИСНОВКИ……...…….……..88
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ……..90
Додаток А. ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація)…….93

ВСТУП

Стрімкий розвиток інформаційних технологій та зростання обсягів обчислювальних ресурсів призвели до широкого впровадження хмарних та серверних інфраструктур у різних сферах діяльності. Сучасні інформаційні системи, веб-сервіси, корпоративні платформи та державні інформаційні ресурси значною мірою залежать від стабільної та ефективної роботи серверів. Одним із ключових чинників, що впливають на якість обслуговування користувачів, є навантаження серверів, яке може істотно змінюватися в часі.

Нерівномірність серверних навантажень створює серйозні проблеми для управління ресурсами. У періоди пікової активності недостатній обсяг доступних ресурсів може призводити до зниження продуктивності систем, затримок у відповіді та порушення угод про рівень обслуговування (Service Level Agreements, SLA). Водночас у періоди низької активності надмірне резервування ресурсів спричиняє їх неефективне використання. У зв’язку з цим актуальним завданням є прогнозування навантаження серверів з метою оптимального планування та розподілу обчислювальних ресурсів.

У межах даної роботи хмарні робочі навантаження розглядаються як навантаження серверів дата-центрів, що забезпечують функціонування хмарних обчислювальних середовищ.

Прогнозування навантаження серверів зазвичай зводиться до аналізу часових рядів, що відображають зміну використання ресурсів у часі. У сучасних дослідженнях методи прогнозування навантаження зазвичай поділяють на три основні групи: статистичні підходи, методи машинного навчання та методи глибинного навчання. Однак такі часові ряди характеризуються високою варіативністю, наявністю шуму, сезонних компонентів та нелінійних залежностей, що ускладнює застосування класичних методів прогнозування. Традиційні статистичні підходи, зокрема авторегресійні моделі, демонструють задовільні результати лише за умови відносної стабільності даних і часто не здатні адекватно

реагувати на різкі зміни навантаження.

Останніми роками значного поширення набули методи машинного навчання та глибинного навчання, які продемонстрували високу ефективність у задачах прогнозування складних часових рядів. Такі методи здатні виявляти приховані закономірності в даних, працювати з високорозмірними вхідними ознаками та адаптуватися до змін у поведінці системи. Водночас застосування цих методів супроводжується низкою проблем, серед яких перенавчання моделей, наївна поведінка прогнозів та складність оцінювання їх узагальнювальної здатності.

Незважаючи на велику кількість наукових досліджень у цій галузі, відсутність уніфікованих наборів даних та детального опису методик експериментального дослідження ускладнює об'єктивне порівняння різних підходів. Крім того, у багатьох роботах недостатньо уваги приділяється аналізу поведінки моделей на нових, раніше невідомих часових рядах, що є критично важливим для практичного використання систем прогнозування серверних навантажень. У зв'язку з цим дана магістерська робота присвячена проєктуванню та дослідженню системи прогнозування навантаження серверів, у якій використовуються методи машинного навчання та глибинного навчання. Особливу увагу приділено порівняльному аналізу різних класів моделей, оцінюванню їхньої здатності до узагальнення, а також аналізу наївної поведінки прогнозів із використанням метрики Naïve Ratio.

Метою магістерської роботи є розробка та дослідження системи прогнозування навантаження серверів з використанням методів машинного навчання і глибинного навчання, а також оцінювання їхньої ефективності, стійкості до перенавчання та здатності до узагальнення.

Для досягнення поставленої мети в роботі необхідно розв'язати такі завдання:

- проаналізувати сучасні наукові дослідження у сфері прогнозування серверних навантажень;

- дослідити властивості часових рядів, що описують навантаження серверів;

- розглянути статистичні методи, методи машинного навчання та глибинного

навчання для прогнозування навантажень;

- обґрунтувати вибір набору даних для проведення експериментальних досліджень;

- реалізувати та налаштувати обрані моделі прогнозування;

- провести експериментальне оцінювання якості прогнозування;

- дослідити наївну поведінку моделей із використанням метрики Naïve Ratio;

- проаналізувати отримані результати та визначити переваги й обмеження досліджуваних підходів.

Об’єктом дослідження є процес прогнозування навантаження серверів у хмарних та серверних обчислювальних системах.

Предметом дослідження є методи та моделі машинного навчання і глибинного навчання, що застосовуються для прогнозування серверних навантажень на основі часових рядів.

Методи дослідження. У роботі використовувалися методи аналізу та узагальнення наукової літератури, статистичні методи аналізу часових рядів, методи машинного навчання та глибинного навчання, а також методи експериментального дослідження і порівняльного аналізу. Для оцінювання якості прогнозування застосовувалися метрики MSE, MAE, коефіцієнт детермінації R^2 та метрика Naïve Ratio.

Наукова новизна магістерської роботи полягає у проведенні комплексного порівняльного аналізу статистичних, машинних та глибинних моделей прогнозування серверних навантажень з урахуванням їхньої наївної поведінки та здатності до узагальнення. Запропоновано використання метрики Naïve Ratio як додаткового інструменту оцінювання якості прогнозування та виявлення небажаної поведінки моделей.

Практичне значення отриманих результатів полягає в можливості їх використання під час проєктування систем моніторингу та автоматичного масштабування серверних ресурсів у хмарних і корпоративних дата-центрах з метою підвищення ефективності використання обчислювальних ресурсів і забезпечення стабільної якості обслуговування.

1 ТЕОРЕТИЧНІ ТА НАУКОВІ ЗАСАДИ ПРОГНОЗУВАННЯ НАВАНТАЖЕННЯ СЕРВЕРІВ

1.1 Розвиток серверних та хмарних інфраструктур

Розвиток серверних інфраструктур тісно пов’язаний із еволюцією обчислювальних технологій та зростанням потреб у зберіганні й обробці великих обсягів даних. На ранніх етапах розвитку інформаційних систем сервери використовувалися переважно як ізольовані обчислювальні вузли, призначені для виконання обмеженого кола задач. Такий підхід передбачав статичне виділення ресурсів, що ускладнювало масштабування систем і призводило до низької ефективності використання обчислювальних потужностей.

Подальший розвиток мережевих технологій та поширення клієнт-серверної архітектури зумовили перехід до централізованих серверних рішень. Сервери почали виконувати роль ключових компонентів корпоративних інформаційних систем, забезпечуючи доступ до баз даних, прикладних сервісів та мережевих ресурсів. Однак зі зростанням кількості користувачів і сервісів зросли й вимоги до надійності, продуктивності та масштабованості серверних систем.

Важливим етапом у розвитку серверних інфраструктур стало впровадження технологій віртуалізації, які дозволили розміщувати декілька віртуальних машин на одному фізичному сервері. Це дало змогу більш гнучко керувати ресурсами, ізолювати робочі навантаження та підвищити рівень використання апаратних засобів. Віртуалізація стала основою для формування сучасних дата-центрів і створення хмарних обчислювальних середовищ.

Хмарні обчислення передбачають надання обчислювальних ресурсів у вигляді сервісів через мережу, що дозволяє користувачам динамічно отримувати необхідні ресурси відповідно до поточних потреб. У рамках хмарних інфраструктур сервери об’єднуються в кластери, а управління ресурсами

здійснюється автоматизовано з використанням спеціалізованих платформ. Такий підхід забезпечує високу гнучкість, масштабованість і відмовостійкість систем, але водночас ускладнює процеси моніторингу та прогнозування навантажень.

Сучасні серверні та хмарні інфраструктури характеризуються високою динамічністю робочих навантажень, що зумовлено змінною активністю користувачів, різноманітністю сервісів та нерівномірністю запитів у часі. У таких умовах традиційні підходи до управління ресурсами, засновані на статичних правилах, виявляються недостатньо ефективними. Це зумовлює необхідність використання інтелектуальних методів аналізу та прогнозування навантажень, здатних адаптуватися до змін у поведінці системи.

Таким чином, розвиток серверних і хмарних інфраструктур створив передумови для появи складних, високонавантажених обчислювальних систем, ефективне функціонування яких неможливе без точного прогнозування навантаження серверів. Це обґрунтовує доцільність застосування методів машинного навчання та глибинного навчання для аналізу та прогнозування серверних навантажень у сучасних інформаційних системах.

1.2 Поняття серверного навантаження та його основні характеристики

Серверне навантаження є одним із ключових показників, що характеризують стан і ефективність функціонування серверних та хмарних обчислювальних систем. Під серверним навантаженням зазвичай розуміють рівень використання обчислювальних ресурсів сервера або групи серверів у процесі виконання прикладних задач і обслуговування користувацьких запитів. Воно відображає інтенсивність роботи системи та безпосередньо впливає на її продуктивність, надійність і якість обслуговування.

Серверне навантаження формується під впливом великої кількості факторів, серед яких основними є кількість одночасних запитів, типи виконуваних задач,

обсяг оброблюваних даних та архітектурні особливості серверної інфраструктури. У практичних системах навантаження не є сталим і може суттєво змінюватися протягом часу, що зумовлює необхідність його постійного моніторингу та аналізу.

Для кількісного опису серверного навантаження використовують різні показники, які відображають використання окремих видів ресурсів. До найпоширеніших характеристик належать завантаження центрального процесора (CPU utilization), використання оперативної пам’яті, інтенсивність операцій введення-виведення, пропускна здатність мережевих інтерфейсів, а також кількість активних процесів або потоків. У залежності від типу сервісу та архітектури системи окремі показники можуть мати різний ступінь важливості.

Особливістю серверного навантаження є його багатовимірний характер. У реальних умовах сервер одночасно використовує кілька видів ресурсів, причому зміна одного показника може впливати на інші. Наприклад, зростання кількості обчислювальних задач може призводити не лише до підвищення завантаження процесора, але й до збільшення використання пам’яті та інтенсивності доступу до дискової підсистеми. Це ускладнює процес аналізу та прогнозування навантаження і потребує використання комплексних моделей.

З точки зору часової динаміки серверне навантаження може розглядатися як часовий ряд, що відображає зміну значень відповідних показників у часі. Такі часові ряди часто мають складну структуру, що включає трендові компоненти, сезонні коливання та випадкові флуктуації. Крім того, для серверних навантажень характерні різкі пікові значення, які можуть виникати внаслідок масових запитів або непередбачуваних подій у системі.

Важливою характеристикою серверного навантаження є його нерівномірність і непередбачуваність. Навіть за наявності регулярних патернів у поведінці користувачів навантаження може істотно змінюватися під впливом зовнішніх чинників, таких як маркетингові кампанії, аварійні ситуації або зміни у конфігурації системи. Це робить традиційні методи статичного планування ресурсів малоефективними та зумовлює необхідність використання методів прогнозування.

Таким чином, серверне навантаження є складним багатовимірним і динамічним процесом, аналіз якого потребує застосування сучасних методів обробки даних. Його подання у вигляді часових рядів створює основу для використання статистичних методів, методів машинного навчання та глибинного навчання з метою побудови ефективних моделей прогнозування, що розглядатимуться в наступних підрозділах роботи.

1.3 Типи серверних навантажень та фактори їх зміни

Серверне навантаження може мати різні типи залежно від характеристик сервісів, що обслуговуються, та рівня використання апаратних ресурсів. Залежно від природи і джерела навантаження, його можна класифікувати на кілька типів.

1.3.1 Постійне навантаження

Постійне (статичне) навантаження виникає внаслідок однотипних або повторюваних операцій, що не змінюються в часі. Прикладом такого навантаження може бути обробка бази даних із фіксованим обсягом запитів або виконання постійно працюючих фонових задач, таких як моніторинг системи. Зазвичай це навантаження має рівномірний характер, і його прогнозування є досить простим.
Для оцінки постійного навантаження використовуються такі показники, як середнє використання процесора (CPU utilization) та пам’яті за визначений період часу. Однак навіть постійне навантаження може зазнавати незначних змін у випадку планового обслуговування або змін у сервісах.

1.3.2 Перемінне навантаження

Перемінне навантаження характеризується періодичними змінами у використанні ресурсів. Такі зміни можуть бути зумовлені сезонними коливаннями попиту, певними циклами роботи сервісу чи обсягами запитів, що змінюються в залежності від часу доби або тижня. Прикладом такого навантаження є веб-сайти, які отримують більший трафік під час святкових періодів чи рекламних кампаній.

Математично перемінне навантаження може бути описано за допомогою функцій, які змінюються з часом, зберігаючи певну періодичність або циклічність. Одним із основних підходів до опису такого навантаження є метод сезонного коригування часових рядів.

1.3.3 Пікове навантаження

Пікове навантаження є найбільш складним і непередбачуваним типом навантаження. Воно характеризується різким збільшенням запитів або навантаження на систему за дуже короткий проміжок часу. Пікові навантаження часто зумовлені неочікуваними подіями, такими як висока активність користувачів через маркетингові кампанії, запуск нових функцій, або непередбачувані проблеми в системі.

У таких випадках використання класичних методів прогнозування стає важким, оскільки вони не здатні адекватно реагувати на різкі зміни в даних. Одним із можливих способів передбачення пікових навантажень є застосування методів машинного навчання, зокрема, регресії та кластеризації, для ідентифікації аномальних ситуацій.

1.3.4 Аномальне навантаження

Аномальне навантаження виникає, коли система стикається з подіями, які є відхиленням від звичайного функціонування. Такі навантаження часто спричинені атакою на систему (наприклад, DDoS атака), помилками в конфігурації серверів або збоєм у роботі одного з компонентів інфраструктури. Аномальне навантаження зазвичай не піддається традиційному прогнозуванню, тому в таких випадках використовуються методи аномалії детекції.

Для детекції аномалій застосовуються спеціалізовані моделі, зокрема методи наївних байєсівських класифікаторів або глибинні нейронні мережі, що здатні виявляти незвичайні патерни в часі.

1.3.5 Комбіноване навантаження

Комбіноване навантаження є найскладнішим для аналізу, оскільки воно поєднує в собі різні типи навантажень (постійне, перемінне, пікове та аномальне) в межах одного сервісу або системи. Наприклад, сервер, який обслуговує веб-запити, може одночасно переживати різкі пікові навантаження, що змінюються з сезоном, а також неочікувані аномалії в результаті помилок у програмному забезпеченні.

Для ефективного прогнозування комбінованих навантажень необхідне застосування гібридних моделей, що поєднують різні методи, такі як класичні статистичні підходи та методи машинного навчання, з можливістю коригування в режимі реального часу.

1.4 Проблеми управління ресурсами серверів та дата-центрів

Ефективне управління ресурсами серверів у великих дата-центрах є складним завданням, зумовленим високою динамічністю робочих навантажень та масштабами сучасних обчислювальних систем. У хмарних середовищах сервери об’єднуються в кластери, у межах яких одночасно виконуються тисячі різнорідних задач, що споживають обчислювальні ресурси з різною інтенсивністю. За таких умов статичні підходи до розподілу ресурсів є неефективними, оскільки не враховують змін у навантаженні з часом.

Однією з основних проблем управління ресурсами є необхідність забезпечення балансу між ефективним використанням обчислювальних ресурсів та дотриманням угод про рівень обслуговування (Service Level Agreements, SLA). Недостатнє виділення ресурсів може призводити до зниження продуктивності сервісів і порушення SLA, тоді як надмірне резервування ресурсів спричиняє їх нераціональне використання та збільшення експлуатаційних витрат. Таким чином, управління ресурсами потребує точних механізмів прогнозування майбутнього навантаження.

Ще однією суттєвою проблемою є непередбачуваність поведінки робочих навантажень. Навіть за наявності історичних даних про використання ресурсів, навантаження може змінюватися під впливом зовнішніх чинників, таких як зміна поведінки користувачів або запуск нових сервісів. У роботі підкреслюється, що серверні навантаження часто мають нестаціонарний характер, що ускладнює застосування традиційних методів прогнозування, орієнтованих на стаціонарні часові ряди.

Крім того, складність управління ресурсами зростає через необхідність багатокрокового прогнозування навантаження. Для практичних систем важливо не лише передбачити значення навантаження на наступний часовий крок, але й оцінити його поведінку на декілька кроків наперед. У роботі зазначається, що зі збільшенням горизонту прогнозування точність моделей, як правило, знижується,

що ускладнює використання прогнозів для автоматичного масштабування ресурсів.

Окрему проблему становить наївна поведінка прогнозних моделей, за якої модель відтворює прості базові стратегії замість навчання складних залежностей у даних. Така поведінка може створювати хибне враження високої точності прогнозування, тоді як фактично модель не забезпечує корисної інформації для управління ресурсами. У роботі ця проблема розглядається як один із ключових викликів під час застосування методів машинного навчання для прогнозування серверних навантажень.

Таким чином, управління ресурсами серверів і дата-центрів у хмарних обчислювальних середовищах пов’язане з низкою проблем, серед яких динамічність і непередбачуваність навантажень, складність багатокрокового прогнозування та ризик наївної поведінки моделей. Це обґрунтовує необхідність застосування методів машинного навчання та глибинного навчання, здатних адаптуватися до змін у даних і забезпечувати більш точні прогнози навантаження, що є основою для ефективного управління ресурсами.

1.5 Необхідність прогнозування навантаження серверів у сучасних інформаційних системах

Сучасні інформаційні системи, зокрема хмарні та серверні платформи, характеризуються високою динамічністю використання обчислювальних ресурсів. Навантаження серверів може суттєво змінюватися залежно від кількості користувачів, типів сервісів, часу доби та інших зовнішніх чинників. За таких умов ефективне функціонування інформаційних систем значною мірою залежить від здатності своєчасно реагувати на зміни навантаження.

Традиційні підходи до управління серверними ресурсами, які ґрунтуються на статичних правилах або реактивному реагуванні на перевищення порогових

значень, не завжди забезпечують необхідний рівень продуктивності та надійності. Запізніла реакція на зростання навантаження може призводити до деградації якості обслуговування, порушення угод про рівень сервісу та зниження задоволеності користувачів.

Таким чином, прогнозування навантаження серверів є необхідним елементом сучасних інформаційних систем, що забезпечує підвищення ефективності використання ресурсів, стабільність роботи сервісів і можливість масштабування інфраструктури відповідно до реальних потреб. Саме обґрунтування цієї необхідності створює основу для подальшого розгляду методів прогнозування, які аналізуються у наступних розділах роботи.

1.6 Часові ряди як основа прогнозування серверних навантажень

Прогнозування навантаження серверів у хмарних обчислювальних системах у даній роботі розглядається як задача прогнозування часових рядів. Навантаження серверів описується у вигляді послідовності значень використання обчислювальних ресурсів, зафіксованих у дискретні моменти часу, що дозволяє аналізувати зміну навантаження на основі історичних спостережень та прогнозувати його майбутні значення.

У роботі підкреслюється, що прогнозування навантаження базується на припущенні про наявність залежностей між минулими та майбутніми значеннями часового ряду. Саме ці залежності використовуються для побудови моделей, здатних передбачати подальшу поведінку серверного навантаження. Такий підхід є типовим для задач прогнозування часових рядів у хмарних та серверних інфраструктурах.

У межах дослідження розглядаються задачі як однокрокового, так і багатокрокового прогнозування. Однокрокове прогнозування передбачає оцінювання значення навантаження для наступного моменту часу, тоді як

багатокрокове прогнозування спрямоване на передбачення значень навантаження на декілька часових кроків уперед. У роботі зазначається, що зі збільшенням горизонту прогнозування складність задачі зростає, а точність прогнозів, як правило, знижується.

Особливу увагу в роботі приділено властивостям часових рядів серверних навантажень. Зазначається, що такі часові ряди характеризуються високою мінливістю, наявністю шуму та нестаціонарністю, що ускладнює процес прогнозування. Зміни в інтенсивності навантаження можуть відбуватися нерівномірно та не завжди мати чітко виражені регулярні патерни.

Таким чином, подання серверного навантаження у вигляді часових рядів є фундаментальною основою для застосування методів прогнозування, які досліджуються у подальших розділах роботи. Саме властивості часових рядів визначають вибір підходів до моделювання та критерії оцінювання ефективності прогнозних моделей.

1.6.1 Основні властивості часових рядів

У даній роботі серверні навантаження розглядаються у вигляді часових рядів, що відображають зміну використання обчислювальних ресурсів у часі. Кожне значення такого ряду відповідає рівню навантаження, зафіксованому в певний дискретний момент часу. Подання навантаження у вигляді часових рядів дозволяє аналізувати його динаміку та використовувати історичні дані для прогнозування майбутньої поведінки системи.

Однією з ключових властивостей часових рядів серверних навантажень є їх висока мінливість. У роботі зазначається, що навантаження серверів може суттєво змінюватися навіть у межах коротких часових інтервалів, що зумовлено запуском і завершенням обчислювальних задач, а також змінами в поведінці користувачів. Така мінливість ускладнює процес моделювання та потребує застосування методів,

здатних працювати з нестабільними даними.

Крім того, у роботі підкреслюється нестаціонарний характер часових рядів серверних навантажень. Статистичні характеристики таких рядів, зокрема середній рівень навантаження та його дисперсія, можуть змінюватися з часом. Це означає, що моделі, навчені на певному часовому інтервалі, не завжди зберігають свою ефективність при застосуванні до даних з інших періодів.

Таким чином, часові ряди серверних навантажень характеризуються мінливістю, зашумленістю та нестаціонарністю, що створює додаткові труднощі для задач прогнозування. Урахування цих властивостей є необхідною умовою для побудови ефективних моделей прогнозування, які розглядаються у наступних підпунктах роботи.

1.6.2 Тренд, сезонність та випадкові коливання

Під час аналізу часових рядів серверних навантажень у роботі розглядаються основні компоненти, які можуть впливати на їхню поведінку в часі. Зокрема, увага приділяється довготривалим змінам рівня навантаження, повторюваним патернам та випадковим коливанням значень, що спостерігаються у даних.

У досліджуваних часових рядах серверних навантажень можуть спостерігатися поступові зміни середнього рівня використання ресурсів, які пов’язані з еволюцією робочих навантажень у кластері. Такі зміни можуть бути зумовлені появою нових задач, змінами у конфігурації системи або загальним зростанням обчислювальних потреб. У роботі зазначається, що наявність таких змін ускладнює прогнозування, оскільки поведінка навантаження з часом може відрізнятися від попередніх періодів.

Окремо у роботі звертається увага на можливу наявність повторюваних патернів у часових рядах серверних навантажень. Такі патерни можуть бути пов’язані з регулярною активністю користувачів або циклічністю виконання задач.

Водночас підкреслюється, що для реальних хмарних навантажень наявність чітко вираженої регулярної сезонності не завжди є гарантованою і може суттєво відрізнятися між різними часовими рядами.

Таким чином, часові ряди серверних навантажень можуть поєднувати довготривалі зміни, можливі повторювані патерни та випадкові коливання. Нерівномірність прояву цих компонент у різних часових рядах є однією з причин складності задачі прогнозування та обґрунтовує використання більш гнучких методів моделювання, які аналізуються у подальших розділах роботи.

1.6.3 Одновимірні та багатовимірні часові ряди

У роботі серверні навантаження можуть бути представлені у вигляді як одновимірних, так і багатовимірних часових рядів. Одновимірний часовий ряд описує зміну одного показника навантаження в часі, наприклад використання певного ресурсу для окремого сервера або задачі. Таке подання дозволяє аналізувати динаміку конкретної характеристики навантаження без урахування взаємозв’язків з іншими параметрами.

Водночас у дослідженні зазначається, що поведінка серверних навантажень часто визначається сукупністю кількох взаємопов’язаних показників. У цьому випадку навантаження розглядається як багатовимірний часовий ряд, у якому кожен часовий момент характеризується вектором значень, що відповідають різним параметрам використання ресурсів або характеристикам виконуваних задач.

Застосування багатовимірного подання дозволяє враховувати взаємозв’язки між різними компонентами навантаження, однак одночасно ускладнює процес аналізу та прогнозування. У роботі підкреслюється, що збільшення кількості вимірів призводить до зростання складності моделей та підвищених вимог до обсягу даних, необхідних для навчання.

У межах даного дослідження як одновимірні, так і багатовимірні часові ряди використовуються для аналізу та прогнозування серверних навантажень залежно від конкретної постановки експерименту. Такий підхід дозволяє оцінити, як урахування додаткових параметрів впливає на якість прогнозування та здатність моделей узагальнювати поведінку навантажень.

Таким чином, вибір між одновимірним і багатовимірним поданням часових рядів є важливим аспектом задачі прогнозування серверних навантажень. Цей вибір визначає складність моделей, вимоги до даних та потенційні можливості щодо підвищення точності прогнозування, що враховується у подальших розділах роботи.

1.7 Аналіз сучасних наукових досліджень у сфері прогнозування серверних навантажень

У сучасних наукових дослідженнях прогнозування серверних та хмарних навантажень розглядається як одна з ключових задач управління обчислювальними ресурсами. Значна кількість робіт присвячена аналізу поведінки навантажень у великих дата-центрах та хмарних середовищах, де нестабільність і висока варіативність даних створюють додаткові труднощі для побудови точних прогнозних моделей.

У багатьох дослідженнях серверні навантаження моделюються у вигляді часових рядів, що дозволяє застосовувати методи прогнозування, розроблені для аналізу часових залежностей. У таких роботах основна увага приділяється використанню історичних даних для передбачення майбутнього рівня використання ресурсів, а також оцінюванню точності прогнозів на різних горизонтах прогнозування.

Традиційні статистичні методи прогнозування, зокрема авторегресійні підходи, широко представлені у наукових роботах завдяки своїй простоті та

інтерпретованості. Водночас у дослідженнях підкреслюється, що такі методи мають обмежені можливості при роботі з нестаціонарними та високошумними часовими рядами, характерними для реальних серверних навантажень.

Окремий напрям сучасних досліджень пов’язаний із застосуванням методів машинного навчання та глибинного навчання для прогнозування серверних навантажень. У таких роботах зазначається, що ці методи здатні виявляти складні нелінійні залежності у даних та краще адаптуватися до змін у поведінці навантажень. Разом із тим у наукових публікаціях наголошується на проблемах перенавчання, нестабільності прогнозів та складності узагальнення результатів на нові часові ряди.

У низці досліджень також піднімається питання наївної поведінки прогнозних моделей, коли складні методи не демонструють суттєвого покращення порівняно з простими базовими підходами. Це зумовлює необхідність використання спеціальних метрик оцінювання та проведення більш детального аналізу узагальнювальної здатності моделей, особливо при прогнозуванні на довгих горизонтах.

Таким чином, аналіз сучасних наукових досліджень свідчить про відсутність універсального підходу до прогнозування серверних навантажень. Існуючі роботи демонструють як потенціал, так і обмеження різних методів прогнозування, що обґрунтовує необхідність подальших досліджень, спрямованих на порівняльний аналіз моделей та оцінювання їхньої здатності узагальнювати поведінку серверних навантажень у реальних умовах.

У першому розділі роботи розглянуто теоретичні та наукові засади прогнозування навантаження серверів у сучасних інформаційних системах. Проаналізовано еволюцію серверних і хмарних інфраструктур, що зумовила зростання складності управління обчислювальними ресурсами та необхідність використання інтелектуальних підходів до аналізу навантажень.

Визначено поняття серверного навантаження, його основні характеристики та типи, а також фактори, що впливають на зміну навантаження у часі. Показано, що серверні навантаження мають багатовимірний, динамічний і нестаціонарний

характер, що ускладнює застосування традиційних методів управління ресурсами.

Обґрунтовано доцільність розгляду серверного навантаження у вигляді часових рядів та проаналізовано їхні основні властивості, зокрема тренд, можливу сезонність і випадкові коливання. Окрему увагу приділено відмінностям між одновимірними та багатовимірними часовими рядами.

Аналіз сучасних наукових досліджень показав, що не існує універсального підходу до прогнозування серверних навантажень, а вибір методу залежить від характеристик даних і горизонту прогнозування. Це підтверджує актуальність подальшого дослідження методів машинного та глибинного навчання, які розглядаються у наступних розділах роботи.

2 МЕТОДИ МАШИННОГО ТА ГЛИБИННОГО НАВЧАННЯ ДЛЯ ПРОГНОЗУВАННЯ НАВАНТАЖЕННЯ СЕРВЕРІВ

2.1 Класифікація методів прогнозування часових рядів

У задачах прогнозування серверних навантажень застосовується широкий спектр методів прогнозування часових рядів, які відрізняються за принципами побудови, складністю та вимогами до даних. У наукових дослідженнях методи прогнозування зазвичай класифікуються на статистичні підходи, методи машинного навчання та методи глибинного навчання, кожна з яких має свої переваги та обмеження.

Статистичні методи прогнозування базуються на побудові математичних моделей, що описують залежність поточних значень часового ряду від його попередніх спостережень. Такі методи, як правило, добре інтерпретуються та ефективні для відносно стабільних часових рядів, однак їх застосування ускладнюється у випадку нестаціонарних та високоваріативних серверних навантажень.

Методи машинного навчання розглядаються як більш гнучка альтернатива статистичним підходам. У таких методах прогнозування часових рядів здійснюється шляхом навчання моделей на історичних даних без явного задання структури залежностей. У дослідженнях зазначається, що методи машинного навчання здатні враховувати складні нелінійні взаємозв’язки у даних, проте їх ефективність значною мірою залежить від якості та обсягу навчальних вибірок.

Окрему групу становлять методи глибинного навчання, які базуються на багатошарових нейронних мережах і широко застосовуються для прогнозування складних часових рядів. У наукових роботах підкреслюється, що такі методи демонструють високий потенціал при роботі з великими обсягами даних та довгими часовими залежностями, однак водночас характеризуються високою

обчислювальною складністю та складністю налаштування.

Таким чином, класифікація методів прогнозування часових рядів на статистичні, машинного та глибинного навчання дозволяє систематизувати існуючі підходи та створює основу для їх детальнішого аналізу у наступних підрозділах даного розділу.

2.2 Статистичні методи прогнозування серверних навантажень

У наукових дослідженнях прогнозування серверних навантажень статистичні методи розглядаються як базовий підхід до аналізу часових рядів. У таких методах майбутні значення навантаження оцінюються на основі історичних спостережень із припущенням про наявність певних статистичних залежностей між послідовними значеннями часового ряду.

Статистичні методи мають низку переваг, зокрема простоту реалізації та інтерпретованість отриманих результатів. Завдяки цьому вони часто використовуються як базові або порівняльні моделі при оцінюванні ефективності більш складних підходів прогнозування серверних навантажень.

Застосування статистичних методів у задачах прогнозування серверних навантажень має суттєві обмеження. Такі методи зазвичай орієнтовані на відносно стабільні часові ряди та погано справляються з високою мінливістю, шумом і нестаціонарністю, які є характерними для реальних хмарних навантажень.

Ефективність статистичних методів суттєво знижується при прогнозуванні на довгих часових горизонтах. У таких випадках похибки прогнозу накопичуються, що призводить до нестабільних або наївних передбачень, які не відображають реальну поведінку серверних навантажень.

Таким чином, статистичні методи прогнозування відіграють важливу роль як відправна точка для аналізу серверних навантажень і базис для порівняння з більш складними моделями. Однак їх обмеження зумовлюють необхідність розгляду

альтернативних підходів, зокрема методів машинного та глибинного навчання, що детально аналізуються у наступних підрозділах роботи.

2.2.1 Метод ковзного середнього

Метод ковзного середнього у задачах прогнозування серверних навантажень розглядається як один із найпростіших статистичних підходів до аналізу часових рядів. У межах наукових досліджень він зазвичай використовується як базова модель, що дозволяє оцінити складність задачі прогнозування та слугує орієнтиром для порівняння з більш складними методами.

Суть методу ковзного середнього полягає у згладжуванні часовго ряду шляхом усереднення попередніх значень навантаження. Такий підхід дозволяє зменшити вплив короткочасних коливань і шуму, проте водночас призводить до втрати інформації про різкі зміни навантаження, які є характерними для реальних серверних та хмарних середовищ.

Метод ковзного середнього має обмежену здатність до прогнозування при високій мінливості та нестаціонарності часових рядів серверних навантажень. Особливо це проявляється при прогнозуванні на декілька часових кроків уперед, коли накопичення похибок призводить до наївної поведінки прогнозів.

Незважаючи на зазначені обмеження, використання методу ковзного середнього є доцільним у якості базової моделі. Порівняння результатів більш складних методів прогнозування з прогнозами, отриманими за допомогою ковзного середнього, дозволяє об’єктивно оцінити реальний приріст точності та уникнути помилкових висновків щодо ефективності складних моделей.

2.2.2 Авторегресійні моделі та модель ARIMA

Авторегресійні моделі широко використовуються у задачах прогнозування часових рядів і часто розглядаються як базовий статистичний інструмент для аналізу залежностей між послідовними значеннями даних. У контексті прогнозування серверних навантажень такі моделі застосовуються для оцінювання майбутніх значень навантаження на основі його попередньої динаміки.

У дослідженнях підкреслюється, що авторегресійні моделі та модель ARIMA демонструють прийнятні результати при короткостроковому прогнозуванні, однак їх ефективність знижується зі збільшенням горизонту прогнозування. У таких випадках моделі часто не здатні адекватно відтворювати різкі зміни навантаження, що призводить до спрощених або наївних прогнозів.

Незважаючи на зазначені обмеження, авторегресійні моделі та модель ARIMA залишаються важливим елементом порівняльного аналізу. Їх використання дозволяє оцінити, наскільки складніші методи прогнозування перевершують класичні статистичні підходи при роботі з реальними даними серверних навантажень.

2.2.3 Переваги та обмеження статистичних методів

Статистичні методи прогнозування серверних навантажень мають низку переваг, які зумовлюють їх широке використання у наукових дослідженнях. Однією з основних переваг є простота реалізації та інтерпретованість результатів. Такі методи не потребують складних процедур навчання та дозволяють отримувати прогнози на основі безпосереднього аналізу історичних значень часових рядів.

Крім того, статистичні методи часто застосовуються як базові моделі для порівняльного аналізу. Їх використання дозволяє оцінити складність задачі прогнозування та перевірити, чи забезпечують більш складні методи машинного та глибинного навчання реальний приріст точності прогнозування.

Статистичні методи мають суттєві обмеження при роботі з реальними серверними навантаженнями. Висока мінливість, зашумленість та нестаціонарність часових рядів призводять до зниження точності прогнозування та обмежують можливості таких методів щодо адекватного відтворення поведінки навантаження.

Особливо вираженими обмеження статистичних методів стають при прогнозуванні на довгих часових горизонтах. У таких випадках накопичення похибок може призводити до наївної поведінки прогнозів, коли отримані результати не відображають реальні зміни навантаження і не забезпечують практичної користі для управління ресурсами.

Таким чином, статистичні методи прогнозування є важливим інструментом аналізу серверних навантажень, однак їх застосування має обмежену ефективність у складних реальних умовах. Це обґрунтовує необхідність розгляду більш гнучких підходів, зокрема методів машинного та глибинного навчання, які аналізуються у наступних підрозділах даного розділу.

2.3 Методи машинного навчання для прогнозування навантаження серверів

Методи машинного навчання широко застосовуються у задачах прогнозування серверних навантажень як більш гнучка альтернатива класичним статистичним підходам. У наукових дослідженнях ці методи використовуються для моделювання складних нелінійних залежностей у часових рядах, які важко або неможливо описати за допомогою простих статистичних моделей.

На відміну від статистичних методів, підходи машинного навчання не вимагають явного задання структури моделі та можуть автоматично виявляти приховані закономірності у даних. Це робить їх привабливими для аналізу серверних навантажень, що характеризуються високою мінливістю, зашумленістю та нестаціонарністю.

Ефективність методів машинного навчання суттєво залежить від якості підготовки даних, вибору ознак та обсягу навчальної вибірки. За недостатньої кількості даних або невдалого вибору параметрів моделі можуть демонструвати нестабільну або наївну поведінку прогнозів, що обмежує їх практичну цінність.

Таким чином, методи машинного навчання розглядаються у даній роботі як проміжна ланка між простими статистичними підходами та більш складними методами глибинного навчання. Подальший аналіз зосереджений на окремих класах моделей машинного навчання, які використовуються для прогнозування серверних навантажень.

2.3.1 Лінійна та поліноміальна регресія

Лінійна та поліноміальна регресія належать до базових методів машинного навчання, які застосовуються у задачах прогнозування часових рядів як прості та інтерпретовані моделі. У дослідженнях прогнозування серверних навантажень ці методи зазвичай використовуються як початкові або порівняльні підходи для оцінювання ефективності складніших моделей.

Регресійні моделі можуть бути ефективними для опису загальних тенденцій у даних, однак їх можливості обмежені при роботі з високонелінійними та нестаціонарними часовими рядами серверних навантажень. Зі зростанням складності поведінки навантаження точність прогнозів регресійних моделей, як правило, знижується.

Поліноміальна регресія розглядається як розширення лінійної регресії, що

дозволяє частково враховувати нелінійні залежності у даних. Водночас у роботі підкреслюється, що збільшення складності моделі не гарантує покращення якості прогнозування та може призводити до перенавчання, особливо за обмеженого обсягу навчальних даних.

Таким чином, лінійна та поліноміальна регресія виконують роль простих моделей машинного навчання, які дозволяють оцінити базовий рівень якості прогнозування серверних навантажень. Їх результати використовуються для порівняння з більш складними методами машинного та глибинного навчання, що аналізуються у наступних підрозділах роботи.

2.3.2 Метод опорних векторів (SVM)

Метод опорних векторів (Support Vector Machine, SVM) належить до класу методів машинного навчання, які застосовуються для розв’язання задач регресії та класифікації. У контексті прогнозування серверних навантажень SVM використовується як підхід, здатний моделювати нелінійні залежності між вхідними ознаками та прогнозованими значеннями часових рядів.

У наукових дослідженнях підкреслюється, що метод опорних векторів може бути ефективним при роботі з даними, які мають складну структуру та нелінійні залежності. Завдяки використанню ядерних функцій SVM здатний адаптуватися до різних типів часових рядів, що робить його привабливим для аналізу серверних навантажень із високою варіативністю.

Застосування SVM у задачах прогнозування серверних навантажень має певні обмеження. Зі збільшенням обсягу даних зростає обчислювальна складність навчання моделі, а вибір ядра та параметрів суттєво впливає на якість прогнозування. За невдалого налаштування модель може демонструвати нестабільні або наївні прогнози.

Крім того, у дослідженнях підкреслюється, що ефективність SVM

знижується при прогнозуванні на довгих часових горизонтах, що пов’язано з накопиченням похибок та складністю узагальнення поведінки часових рядів серверних навантажень. У таких умовах переваги методу можуть бути менш вираженими порівняно з більш гнучкими підходами.

Таким чином, метод опорних векторів розглядається у даній роботі як один із методів машинного навчання для прогнозування серверних навантажень, що дозволяє враховувати нелінійні залежності у даних, але потребує обережного налаштування та має обмеження при роботі з великими обсягами даних і довгими горизонтами прогнозування.

2.3.3 Ансамблеві методи та випадковий ліс

Ансамблеві методи машинного навчання ґрунтуються на поєднанні кількох базових моделей з метою підвищення точності та стійкості прогнозування. У задачах прогнозування серверних навантажень такі підходи використовуються для зменшення впливу шуму та нестабільності окремих моделей, що є характерними для реальних часових рядів.

Одним із найбільш поширених ансамблевих методів є випадковий ліс (Random Forest), що представлено на рисунку 2.1. Випадковий ліс (Random Forest) складається з множини дерев рішень, кожне з яких навчається на випадковій підмножині навчальних даних та випадковому піднаборі ознак. Для формування прогнозу кожне дерево генерує власне передбачення, після чого результати усіх дерев агрегуються шляхом усереднення. Такий підхід дозволяє зменшити вплив шуму та окремих помилок моделей, що підвищує стійкість і точність прогнозування.

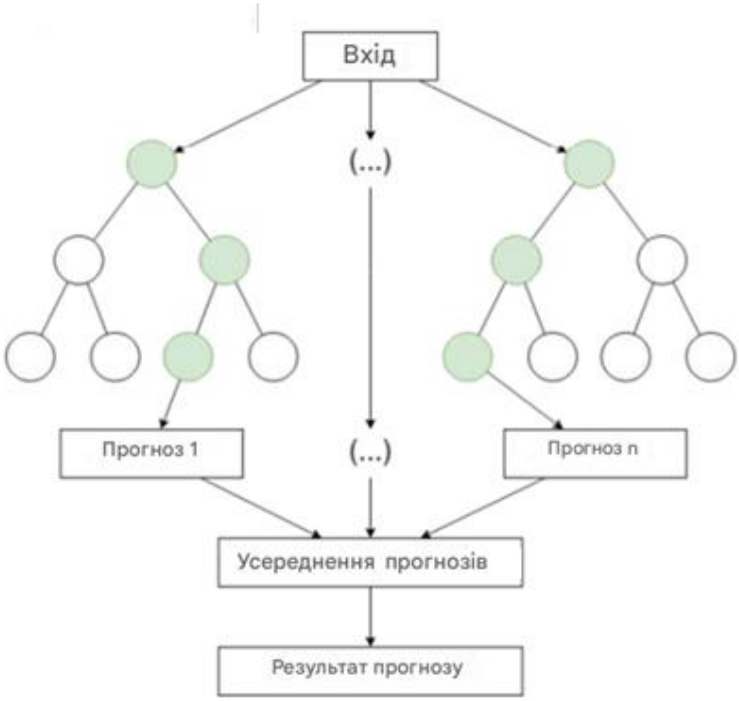


Рисунок 2.1 - Приклад структури випадкового лісу (Random Forest) з максимальною глибиною дерев 2

У дослідженнях прогнозування серверних навантажень цей метод застосовується як універсальний підхід, здатний працювати з нелінійними залежностями та різними типами вхідних ознак.

Випадковий ліс демонструє кращу стійкість до перенавчання порівняно з окремими моделями, завдяки усередненню прогнозів кількох дерев. Це робить його привабливим для аналізу серверних навантажень, однак водночас збільшує обчислювальні витрати та ускладнює інтерпретацію результатів.

Крім того, у дослідженнях підкреслюється, що ефективність ансамблевих методів, зокрема випадкового лісу, суттєво залежить від вибору параметрів моделі та якості вхідних даних. За невдалого налаштування або недостатнього обсягу навчальної вибірки прогнозні моделі можуть демонструвати обмежену узагальнювальну здатність.

Таким чином, ансамблеві методи та випадковий ліс розглядаються у даній роботі як перспективні підходи машинного навчання для прогнозування серверних навантажень, що поєднують здатність до моделювання складних

залежностей зі зниженням впливу шуму. Водночас їх використання потребує ретельного налаштування та оцінювання узагальнювальної здатності, що враховується у подальших експериментальних дослідженнях.

2.4 Методи глибинного навчання

Методи глибинного навчання посідають особливе місце у сучасних дослідженнях прогнозування серверних навантажень, оскільки вони здатні ефективно працювати зі складними та високоваріативними часовими рядами. У порівнянні з класичними статистичними підходами та методами машинного навчання, моделі глибинного навчання мають вищу здатність до автоматичного виявлення складних залежностей у даних без необхідності ручного проєктування ознак.

Глибинні нейронні мережі є особливо придатними для аналізу часових рядів, які характеризуються нестаціонарністю, шумом і наявністю довготривалих залежностей. Саме ці властивості є типовими для серверних і хмарних навантажень, що робить застосування методів глибинного навчання перспективним напрямом досліджень.

На відміну від традиційних підходів, методи глибинного навчання здатні моделювати як короткострокові, так і довготривалі часові залежності. Це дозволяє використовувати їх для прогнозування серверних навантажень на різних горизонтах прогнозування, що є важливим для задач проактивного управління ресурсами в інформаційних системах.

Застосування методів глибинного навчання супроводжується низкою викликів. До них належать висока обчислювальна складність, потреба у значних обсягах навчальних даних, а також ризик перенавчання. Крім того, навіть складні моделі глибинного навчання можуть демонструвати наївну поведінку прогнозів, що потребує ретельного аналізу їхньої узагальнювальної здатності.

Таким чином, методи глибинного навчання розглядаються у даній роботі як потужний інструмент прогнозування серверних навантажень, який поєднує високий потенціал моделювання складних часових залежностей із необхідністю ретельного налаштування та оцінювання ефективності. Подальші підпункти цього розділу присвячені аналізу окремих архітектур глибинного навчання, що застосовуються для прогнозування часових рядів серверних навантажень.

2.4.1 Штучні нейронні мережі

Штучні нейронні мережі є одним із перших і базових підходів глибинного навчання, які застосовуються для аналізу та прогнозування часових рядів. У дослідженнях прогнозування серверних навантажень такі мережі розглядаються як узагальнення методів машинного навчання, здатне моделювати нелінійні залежності між вхідними даними та прогнозованими значеннями.

Штучні нейронні мережі можуть бути ефективними при аналізі складних часових рядів, де традиційні статистичні моделі демонструють обмежені можливості. Завдяки багатошаровій структурі такі мережі здатні автоматично виявляти приховані закономірності у даних серверних навантажень без необхідності ручного формування ознак.

Стандартні штучні нейронні мережі не враховують явним чином часову залежність між послідовними спостереженнями. У результаті при прогнозуванні серверних навантажень на довгих часових інтервалах такі моделі можуть демонструвати знижену точність і нестабільну поведінку прогнозів.

У роботі звертається увага на ризик перенавчання штучних нейронних мереж, особливо за умови обмеженого обсягу навчальних даних або високої складності моделі. Це обмежує їх практичне застосування для прогнозування серверних навантажень без використання спеціалізованих архітектур, орієнтованих на роботу з часовими рядами.

Таким чином, штучні нейронні мережі розглядаються у даній роботі як важливий етап розвитку методів прогнозування серверних навантажень, який демонструє потенціал глибинного навчання, але водночас має суттєві обмеження при роботі з часовими залежностями. Ці обмеження зумовлюють необхідність використання більш спеціалізованих моделей, зокрема рекурентних нейронних мереж, що розглядаються у наступному підпункті.

2.4.2 Рекурентні нейронні мережі

Рекурентні нейронні мережі (Recurrent Neural Networks, RNN) є розвитком штучних нейронних мереж, спеціально орієнтованим на обробку послідовних даних. У задачах прогнозування серверних навантажень RNN застосовуються для моделювання часових залежностей, оскільки вони здатні враховувати інформацію про попередні стани часових рядів.

Рекурентні нейронні мережі дозволяють безпосередньо працювати з часовою структурою даних, що є суттєвою перевагою порівняно зі стандартними штучними нейронними мережами. Завдяки рекурентним зв’язкам RNN можуть враховувати вплив попередніх значень навантаження на поточні та майбутні значення.

Класичні рекурентні нейронні мережі мають низку обмежень при прогнозуванні серверних навантажень. Зокрема, вони можуть зазнавати труднощів при моделюванні довготривалих часових залежностей, що є характерним для реальних серверних і хмарних навантажень.

Крім того, у дослідженнях звертається увага на нестабільність навчання рекурентних нейронних мереж, яка може призводити до зниження якості прогнозування. Унаслідок цього застосування RNN без додаткових механізмів може бути обмеженим у задачах прогнозування складних та нестаціонарних часових рядів.

Рекурентні нейронні мережі розглядаються у даній роботі як важливий крок у розвитку методів глибинного навчання для прогнозування серверних навантажень. Водночас їх обмеження при роботі з довготривалими залежностями зумовлюють необхідність використання більш спеціалізованих архітектур, зокрема мереж довготривалої короткочасної пам’яті, що розглядаються у наступному підпункті.

2.4.3 Мережі довготривалої короткочасної пам’яті (LSTM)

Мережі довготривалої короткочасної пам’яті (Long Short-Term Memory, LSTM) є спеціалізованим різновидом рекурентних нейронних мереж, розробленим для подолання обмежень класичних RNN при моделюванні довготривалих часових залежностей. У задачах прогнозування серверних навантажень LSTM розглядаються як один із найбільш ефективних підходів до аналізу складних та нестаціонарних часових рядів.

Архітектура LSTM дозволяє зберігати та оновлювати інформацію про попередні стани часових рядів упродовж тривалих часових інтервалів. Це робить такі мережі особливо придатними для прогнозування серверних навантажень, де майбутня поведінка системи може залежати від подій, що відбулися значно раніше.

Ядром архітектури LSTM є комірка пам’яті, яка здатна зберігати свій стан у часі, а також нелінійні керувальні блоки, що контролюють потік інформації, яка надходить до комірки та виходить з неї. Приклад такої комірки наведено на рисунку 2.1.

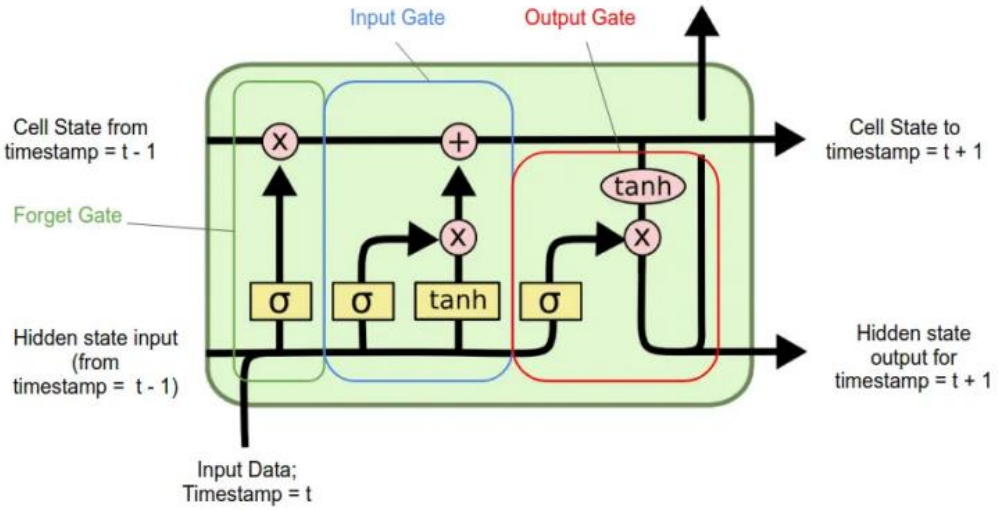


Рисунок 2.1 - Приклад комірки мережі довготривалої короткочасної пам'яті (LSTM)

У таблиці 2.1 наведено функції, що обчислюються для кожного елемента у вхідній послідовності мережі LSTM. Зазначені функції відповідають стандартній реалізації LSTM та запозичені з офіційної документації бібліотеки PyTorch.

Таблиця 2.1 - Функції LSTM

	$i_t = \sigma(W_{ii}x_t + b_{ii} + W_{hi}h_{t-1} + b_{hi})$
	$f_t = \sigma(W_{if}x_t + b_{if} + W_{hf}h_{t-1} + b_{hf})$
	$g_t = \tanh(W_{ig}x_t + b_{ig} + W_{hg}h_{t-1} + b_{hg})$
	$o_t = \sigma(W_{io}x_t + b_{io} + W_{ho}h_{t-1} + b_{ho})$
	$c_t = f_t \odot c_{t-1} + i_t \odot g_t$
	$h_t = o_t \odot \tanh(c_t)$
Where	
h_t	Hidden state at time t
c_t	Cell state at time t
x_t	Input at time t
h_{t-1}	Hidden state of the layer at time $t-1$ or initial state at time 0
i_t, f_t, g_t, o_t	Input, forget, cell, and output gates, respectively
σ	Sigmoid function
$\odot$	Hadamard product

Як показано на рисунку 2.2, архітектура LSTM містить три основні керувальні вентилі (gates): вентиль забування (forget gate), вхідний вентиль (input gate) та вихідний вентиль (output gate).

Вентиль забування ftf_tft, наведений у таблиці 2.1, отримує прихований стан попереднього часового кроку ht−1h_{t-1}ht−1 та поточний вхід xtx_txt. Після зважування вхідних сигналів і додавання зсувів (bias) до них застосовується сигмоїдна функція активації, яка визначає, яку частину попереднього стану комірки необхідно зберегти.

Вхідний вентиль iti_tit та вихідний вентиль oto_tot формуються аналогічним чином, що також відображено у таблиці 2.1, і відповідають за додавання нової інформації до стану комірки та формування вихідного прихованого стану відповідно.

Мережа LSTM оперує двома основними станами: станом комірки (cell state) ctc_tct та прихованим станом (hidden state) hth_tht. Стан комірки є основним елементом пам’яті LSTM і може передавати інформацію без змін, поповнювати її або видаляти залежно від керування вентилями.

Оновлення стану комірки здійснюється шляхом лінійної комбінації збереженої частини попереднього стану, визначеної вентилем забування, та оновлення-кандидата gtg_tgt, сформованого вхідним вентилем. Такий механізм забезпечує стабільне поширення інформації у часі та є ключовою причиною ефективності LSTM при моделюванні довготривалих часових залежностей.

Прихований стан hth_tht, як показано у таблиці 2.1, формується шляхом поелементного множення виходу вентиля виходу на результат нелінійної функції tanh-блока, застосованої до поточного стану комірки ctc_tct.

На відміну від класичних рекурентних нейронних мереж, LSTM демонструють вищу стабільність навчання та кращу здатність до узагальнення при роботі з довгими послідовностями даних. У дослідженнях зазначається, що це дозволяє отримувати більш точні прогнози серверних навантажень, особливо на середніх і довгих горизонтах прогнозування.

Застосування LSTM супроводжується підвищеною обчислювальною складністю та потребує значних обсягів навчальних даних. За недостатньої кількості даних або невдалого налаштування моделі можливе зниження якості прогнозування та поява наївної поведінки прогнозів.

Таким чином, мережі довготривалої короткочасної пам’яті розглядаються у даній роботі як один із ключових інструментів глибинного навчання для прогнозування серверних навантажень. Їх здатність моделювати довготривалі часові залежності робить LSTM перспективним підходом, який, однак, потребує ретельного оцінювання ефективності та узагальнювальної здатності у практичних застосуваннях.

2.4.4 Transformer-архітектури для прогнозування часових рядів

Transformer-архітектури є відносно новим, але перспективним напрямом у задачах прогнозування часових рядів, зокрема в контексті аналізу серверних та хмарних навантажень. На відміну від рекурентних нейронних мереж, Transformer не використовує рекурентні зв’язки, а ґрунтується на механізмі уваги (attention), що дозволяє ефективно моделювати залежності між елементами послідовності незалежно від їх віддаленості у часі.

Архітектура Transformer складається з двох основних компонентів — енкодера та декодера. Кожен із них може бути побудований з кількох однакових шарів, що забезпечує високу модельну потужність і здатність до масштабування. Центральним елементом кожного шару є механізм самоуваги (self-attention), який дозволяє моделі зосереджуватися на найбільш інформативних частинах вхідної послідовності при формуванні прогнозу.

Механізм уваги дає змогу моделі оцінювати вагомість кожного елемента часової послідовності відносно інших, що є особливо важливим для серверних навантажень, де поточний стан системи може залежати як від нещодавніх, так і від віддалених у часі подій. Після шару уваги в архітектурі Transformer використовуються повнозв’язані (feedforward) шари та шари нормалізації, які стабілізують процес навчання та покращують узагальнювальну здатність моделі.

Особливістю декодера є використання додаткового механізму уваги, який

пов’язує вихід енкодера з поточним станом декодера. Це дозволяє ефективно поєднувати інформацію з усієї вхідної послідовності при формуванні прогнозів на кілька часових кроків уперед. Загальна схема архітектури Transformer наведена на рисунку 2.3.

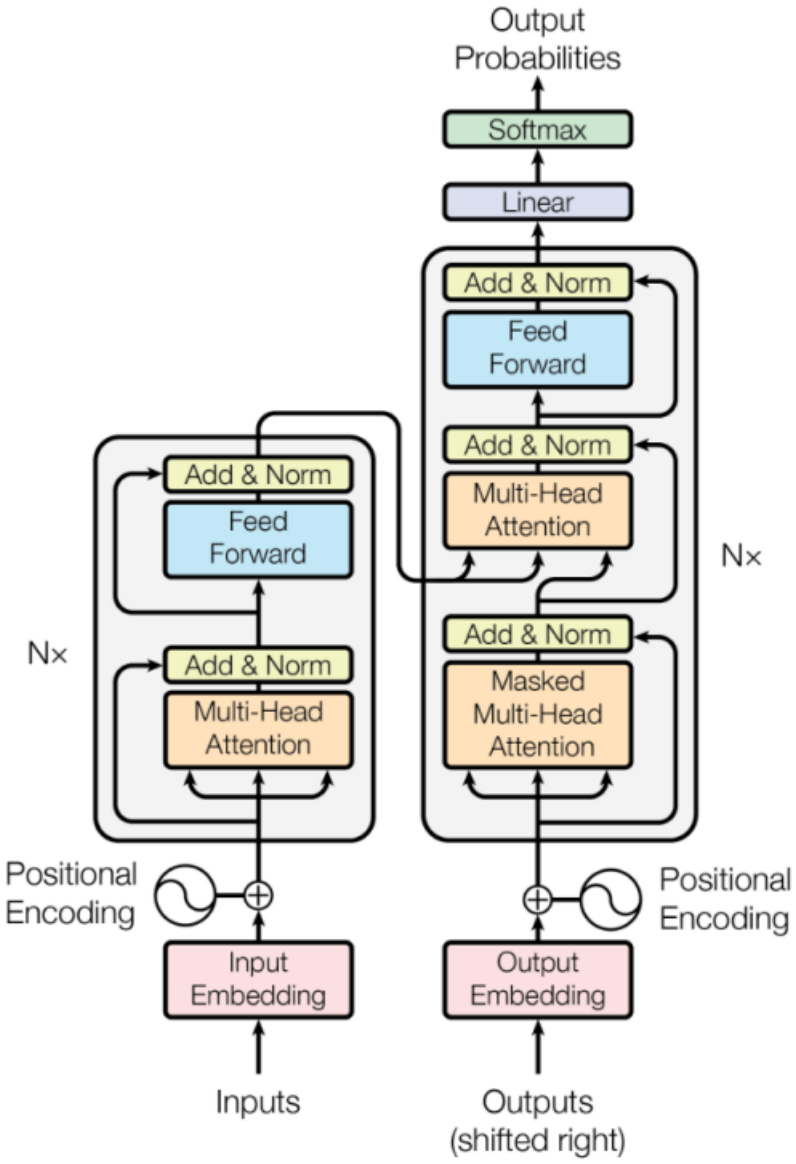


Рисунок 2.3 - Архітектура моделі Transformer

Застосування Transformer-архітектур у задачах прогнозування часових рядів має низку переваг. До них належать здатність моделювати довготривалі часові залежності, висока паралельність обчислень та добра масштабованість при роботі з великими обсягами даних. Це робить Transformer привабливим підходом для аналізу складних і високоваріативних серверних навантажень.

Водночас Transformer-моделі мають і певні обмеження. Вони

характеризуються високою обчислювальною складністю, потребують значних обсягів навчальних даних та ретельного налаштування гіперпараметрів. Крім того, інтерпретація отриманих результатів є складнішою порівняно з класичними статистичними моделями.

Таким чином, Transformer-архітектури розглядаються у даній роботі як сучасний і перспективний підхід до прогнозування серверних навантажень, що доповнює рекурентні моделі глибинного навчання. Їх застосування є доцільним у випадках, коли необхідно враховувати складні довготривалі залежності у часових рядах та забезпечити високу точність прогнозування.

2.5 Метрики та критерії оцінювання якості прогнозування

У задачах прогнозування серверних навантажень результати моделювання мають кількісно оцінюватися за допомогою відповідних метрик. Оскільки прогнозування навантаження серверів є задачею регресії, у даній роботі використовуються метрики оцінювання регресійних моделей.
На відміну від задач класифікації, де застосовуються ROC-криві або матриці неточностей, для прогнозування часових рядів доцільно використовувати метрики, що безпосередньо вимірюють відхилення прогнозованих значень від реальних спостережень.

У цьому розділі розглянуто основні метрики, які застосовуються для оцінювання якості прогнозування серверних навантажень, а саме середньоквадратичну та середню абсолютну похибки, коефіцієнт детермінації R^2, а також спеціалізовану метрику Naïve Ratio, яка дозволяє аналізувати наївну поведінку моделей прогнозування.

2.5.1 Середньоквадратична та середня абсолютна похибки

Одними з найбільш поширених метрик оцінювання регресійних моделей є середньоквадратична похибка (Mean Squared Error, MSE) та середня абсолютна похибка (Mean Absolute Error, MAE).

Середньоквадратична похибка обчислює середнє значення квадратів різниці між прогнозованими та реальними значеннями часового ряду. Вона дозволяє сильніше штрафувати великі помилки прогнозування, що є важливим у задачах прогнозування серверних навантажень, де різкі відхилення можуть призводити до порушення угод про рівень обслуговування.

Формула наведена в рівнянні 2.1.

Нижче значення MSE вказує на кращу продуктивність моделі; отже, ідеальна модель має MSE = 0.

$$MSE(y, \hat{y}) = \frac{1}{n}\sum_{i=0}^{n-1}(y_i - \hat{y}_i)^2 \quad (2.1)$$

де y_i — це прогнозоване значення i-го зразка, а $\hat{y}_i$ — відповідне фактичне значення.

Змінна n позначає кількість зразків.

Крім MSE, для оцінки моделі зазвичай використовують кілька інших подібних показників.

До них належать:

- Root Mean Squared Error (RMSE) - середньоквадратична похибка;

- Mean Absolute Error (MAE) - середня абсолютна похибка;

- Mean Absolute Percentage Error (MAPE) - середня абсолютна відсоткова похибка.

Середня абсолютна похибка визначається як середнє значення абсолютних відхилень між прогнозованими та фактичними значеннями. На відміну від MSE, ця метрика є менш чутливою до поодиноких великих помилок та забезпечує більш інтерпретовану оцінку точності прогнозування.

Формула для MAE наведена у рівнянні 2.2.

$$MAE(y, \hat{y}) = \frac{1}{n}\sum_{i=0}^{n-1}|y_i - \hat{y}_i| \quad (2.2)$$

Для обох метрик — MSE та MAE — менше значення свідчить про кращу якість прогнозування. У практичних дослідженнях ці метрики часто використовуються спільно, оскільки вони доповнюють одна одну.

2.5.2 Коефіцієнт детермінації R^2

Коефіцієнт детермінації R^2 використовується для оцінювання того, наскільки добре модель пояснює варіацію даних. Ця метрика відображає частку дисперсії фактичних значень, яка пояснюється моделлю прогнозування.

Значення R^2 знаходиться в діапазоні від 0 до 1, де значення, близьке до 1, свідчить про високу відповідність моделі даним, тоді як значення, близьке до 0, вказує на низьку якість прогнозування. Рівняння для розрахунку показника R2 наведено в 2.3.

$$R2(y, \hat{y}) = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \hat{y})^2} \quad (2.3)$$

У задачах прогнозування серверних навантажень коефіцієнт R^2 використовується як додаткова метрика для порівняння моделей, оскільки він дозволяє оцінити загальну узгодженість прогнозів із реальними часовими рядами.

2.5.3 Метрика Naïve Ratio та аналіз наївної поведінки моделей

Окрім класичних метрик похибки, у даній роботі використовується спеціалізована метрика Naïve Ratio (NR), яка дозволяє кількісно оцінити наявність наївної поведінки у моделях прогнозування.

Наївна поведінка полягає в тому, що модель фактично відтворює поточне або попереднє значення часового ряду замість того, щоб здійснювати справжнє прогнозування. Така поведінка може призводити до оманливо низьких значень похибки, не забезпечуючи реальної прогностичної здатності.

Метрика Naïve Ratio визначається як відношення середньої абсолютної похибки моделі до середньої абсолютної похибки наївного прогнозу, отриманого шляхом зсуву часового ряду на один часовий крок.

У рівнянні 2.4 наведено формулу, що використовується для обчислення NR.

$$NR = \frac{MAE_t(y,\hat{y})}{MAE_{t+1}(y,\hat{y})};\ NR \in (0;\infty) \tag{2.4}$$

де $MAE_t(y,\hat{y})$ — середня абсолютна похибка в певний момент часу t, що показано в рівнянні 2.2.

$MAE_{t+1}(y,\hat{y})$ визначає функцію зсуву втрат назад, де індекс t+1 позначає зсув на один момент часу.

Рівняння 2.5 зображує функцію зсуву втрат назад.

$$MAE_{t+1}(y,\hat{y}) = \frac{1}{n} * \sum_{i=1}^{n-1} |y_i(y+1) - y(i)| \tag{2.5}$$

де MAEt_tt — середня абсолютна похибка на поточному часовому кроці, MAEt+1_{t+1}t+1 — похибка для зсунутого на один крок прогнозу.

Інтерпретація значень Naïve Ratio є такою:

- значення NR > 1 свідчить про наявність наївної поведінки моделі;
- значення NR = ∞ відповідає повністю наївній моделі;
- значення NR < 1 означає, що внесок наївної поведінки є незначним;
- ідеальна модель має значення NR = 0.

Використання метрики Naïve Ratio дозволяє виявляти небажану поведінку моделей прогнозування серверних навантажень та доповнює класичні метрики похибки, забезпечуючи більш глибокий аналіз якості прогнозів.

Окрім наївної поведінки, важливими проблемами під час розробки моделей машинного навчання є перенавчання та недонавчання. Перенавчання виникає, коли модель надто точно підлаштовується під навчальні дані та втрачає здатність до узагальнення, тоді як недонавчання пов’язане з недостатньою складністю

моделі та нездатністю відтворити закономірності даних. У контексті прогнозування серверних навантажень ці проблеми потребують особливої уваги, оскільки вони безпосередньо впливають на якість прогнозів та стабільність роботи систем.

У другому розділі роботи проаналізовано основні методи прогнозування серверних навантажень, які застосовуються для аналізу часових рядів у сучасних інформаційних системах. Розглянуто класифікацію методів прогнозування на статистичні підходи, методи машинного навчання та методи глибинного навчання, що дозволило систематизувати існуючі рішення та визначити їхні переваги й обмеження.

Проаналізовано статистичні методи прогнозування як базові моделі, що використовуються для оцінювання складності задачі та порівняльного аналізу. Показано, що такі методи мають обмежену ефективність при роботі з нестаціонарними та високоваріативними серверними навантаженнями, особливо при прогнозуванні на довгих часових горизонтах.

Досліджено методи машинного навчання, зокрема регресійні підходи, метод опорних векторів та ансамблеві методи. Визначено, що ці методи здатні враховувати нелінійні залежності у даних, однак їх ефективність значною мірою залежить від якості підготовки даних, вибору ознак і налаштування параметрів моделей.

Особливу увагу приділено методам глибинного навчання, зокрема штучним нейронним мережам, рекурентним нейронним мережам, мережам довготривалої короткочасної пам’яті та Transformer-архітектурам. Показано, що ці підходи мають високий потенціал для моделювання складних часових залежностей серверних навантажень, проте потребують значних обчислювальних ресурсів і ретельного оцінювання узагальнювальної здатності.

Також розглянуто основні метрики оцінювання якості прогнозування, зокрема MSE, MAE, коефіцієнт детермінації R^2 та метрику Naïve Ratio, яка дозволяє виявляти наївну поведінку моделей. Отримані результати створюють методологічну основу для проведення експериментальних досліджень і

порівняльного аналізу моделей прогнозування серверних навантажень, які розглядаються у наступних розділах роботи.

3 МЕТОДОЛОГІЯ ТА ПРОЄКТУВАННЯ СИСТЕМИ ПРОГНОЗУВАННЯ НАВАНТАЖЕННЯ СЕРВЕРІВ

3.1 Постановка задачі прогнозування навантаження серверів

У сучасних серверних та хмарних обчислювальних системах ефективне управління ресурсами значною мірою залежить від здатності точно прогнозувати майбутнє навантаження. Навантаження серверів, як правило, змінюється в часі та може бути представлено у вигляді часових рядів, що відображають використання обчислювальних ресурсів, зокрема центрального процесора, пам’яті, дискового простору та мережевої пропускної здатності.

Задача прогнозування навантаження серверів зводиться до побудови моделей, які на основі історичних спостережень здатні передбачати майбутні значення навантаження. У більшості практичних сценаріїв прогнозування використовується підхід аналізу часових рядів, де майбутній стан системи оцінюється на основі попередніх значень навантаження.

У даній роботі розглядається задача прогнозування серверного навантаження як задача регресії, у якій на основі послідовності попередніх значень навантаження необхідно отримати прогноз на один або кілька часових кроків уперед. Особлива увага приділяється як однокроковому прогнозуванню (one-step forecasting), так і багатокроковому прогнозуванню (multi-step forecasting), оскільки в реальних системах управління ресурсами важливо мати інформацію не лише про найближчий момент часу, а й про майбутню динаміку навантаження. Окрім точності прогнозування для одного часового ряду, у роботі також розглядається задача узагальнення моделей. Зокрема, досліджується підхід, за якого моделі навчаються на кількох часових рядах, що відповідають різним серверним або робочим навантаженням, а потім застосовуються для прогнозування нових, раніше невідомих навантажень. Такий підхід є більш

наближеним до практичних умов експлуатації серверних і хмарних систем.

Для оцінювання ефективності запропонованих підходів прогнозування у роботі використовується набір кількісних метрик, які дозволяють аналізувати точність прогнозів, стійкість моделей до перенавчання, а також наявність наївної поведінки прогнозів. Це забезпечує комплексний підхід до аналізу якості моделей та їх придатності до практичного застосування у системах управління серверними ресурсами.

Метою даної магістерської роботи є аналіз та узагальнення сучасних підходів до прогнозування серверних навантажень на основі часових рядів, а також оцінювання ефективності методів машинного та глибинного навчання, запропонованих у сучасних наукових дослідженнях. У роботі розглядається задача прогнозування навантаження серверів у хмарних обчислювальних середовищах, де динаміка споживання ресурсів характеризується високою мінливістю, нестаціонарністю та наявністю нелінійних залежностей.

Під прогнозуванням навантаження серверів розуміється задача оцінювання майбутніх значень використання обчислювальних ресурсів (передусім центрального процесора) на основі історичних даних моніторингу. У межах даної роботи прогнозування розглядається як задача регресії для часових рядів, де кожне значення навантаження залежить від попередніх спостережень.

У якості бази для аналізу використовується методологія експериментальних досліджень, де виконано комплексне порівняння статистичних методів, методів машинного навчання та глибинного навчання для прогнозування навантажень у хмарних системах. Методологія експериментальних досліджень базується на підходах, запропонованих у сучасних наукових роботах. У межах даної магістерської роботи реалізовано адаптовану експериментальну схему, спрямовану на аналіз та інтерпретацію результатів прогнозування серверних навантажень.

Таким чином, задача дослідження полягає у системному аналізі результатів сучасних експериментальних досліджень з метою визначення сильних та слабких сторін різних підходів прогнозування серверних навантажень.

3.2 Опис та обґрунтування вибору набору даних

Для проведення експериментальних досліджень та оцінювання ефективності методів прогнозування серверних навантажень необхідно обрати відповідний набір даних. Вибір набору даних є критично важливим етапом, оскільки саме від його характеристик залежать коректність отриманих результатів, можливість їх порівняння з іншими дослідженнями та практична цінність зроблених висновків.

Ключовими вимогами до набору даних у даній роботі є:

- наявність часових рядів, що відображають використання обчислювальних ресурсів;

- достатня тривалість спостережень для аналізу довготривалих залежностей;

- реалістичність і відповідність промисловим сценаріям використання серверних і хмарних систем;

- наукова релевантність та поширеність у дослідницькій спільноті.

У сучасних дослідженнях прогнозування серверних і хмарних навантажень використовуються декілька відкритих наборів даних, серед яких найбільш відомими є набори даних Azure, Google Cluster та Alibaba. Кожен із них має свої особливості, переваги та обмеження.

Набір даних Alibaba містить кілька трас, що описують різні сценарії роботи хмарної інфраструктури, зокрема поєднання онлайн-сервісів і пакетних завдань, виробничі пакетні навантаження, а також навантаження, пов’язані з машинним навчанням. Однак часові інтервали окремих трас є відносно короткими, що обмежує можливості аналізу довготривалих часових залежностей.

Набір даних Azure містить дві траси навантажень віртуальних машин, зібрані у 2017 та 2019 роках, кожна з яких охоплює період близько 30 днів. Хоча ці дані мають відповідний обсяг і зміст, їх використання у наукових публікаціях є менш поширеним у порівнянні з іншими наборами даних.

Набір даних Google Cluster містить детальні траси навантажень, зібрані у виробничих кластерах Google. Він включає декілька трас, серед яких найбільш

відомою є траса за 2011 рік, що описує роботу одного кластера протягом приблизно 29 днів. Незважаючи на відносно давню дату збору, цей набір даних залишається одним із найбільш широко використовуваних у наукових дослідженнях, що підтверджується великою кількістю публікацій і цитувань.

Для обґрунтування вибору набору даних у роботі було проведено аналіз наукових публікацій із використанням сервісу Google Scholar. Порівняння показало, що траси Google Cluster використовуються у значно більшій кількості досліджень, ніж відповідні набори даних Azure. Крім того, навіть у нещодавно опублікованих роботах з прогнозування хмарних навантажень продовжують застосовуватися саме дані Google Cluster, що свідчить про їх актуальність і наукову цінність.

З огляду на зазначені фактори, у даній роботі для експериментальних досліджень було обрано набір даних Google Cluster Trace 2011. Його використання дозволяє забезпечити порівнюваність отриманих результатів із існуючими дослідженнями, а також провести глибокий аналіз методів прогнозування серверних навантажень на реалістичних даних великого масштабу.

3.2.1 Характеристика набору даних Google Cluster Trace

Набір даних Google Cluster Trace 2011 містить детальні відомості про роботу одного з виробничих кластерів компанії Google та широко використовується у наукових дослідженнях, присвячених аналізу й прогнозуванню серверних і хмарних навантажень. Траса *clusterdata-2011-2* була зібрана протягом 29 днів, починаючи з 19:00 за східним часом США у неділю, 1 травня 2011 року, в одному з обчислювальних кластерів Google, що налічував приблизно 12 500 фізичних машин.

Згідно з офіційною документацією, кластер Google являє собою набір машин,

згрупованих у стійки та з’єднаних високошвидкісною мережею. Кластер поділяється на так звані *комірки* (cells), кожна з яких об’єднує групу машин, що керуються спільною системою управління ресурсами. Саме така система відповідає за розподіл обчислювальних завдань між машинами кластера.

Навантаження у кластері надходить у вигляді завдань (jobs), кожне з яких складається з одного або декількох процесів (tasks). Кожне завдання супроводжується набором вимог до ресурсів, які використовуються системою управління кластером для планування виконання процесів на окремих машинах. Процеси, у свою чергу, представляють собою Linux-програми, що можуть включати кілька підпроцесів і виконуються на одній фізичній машині.

Користувачами кластера є внутрішні сервіси та інженери Google. Процеси й завдання плануються відповідно до свого життєвого циклу, а дані щодо використання ресурсів формуються на основі інформації, отриманої як від системи управління кластером, так і безпосередньо від машин.

Кожна траса використання ресурсів описує роботу обчислювальної комірки протягом кількох днів і складається з множини записів, організованих у таблиці. Кожен запис містить інформацію, індексовану за первинним ключем, яким зазвичай є часовий штамп. Набір даних надається у вигляді одного або кількох файлів у стиснутому форматі CSV, що полегшує його обробку та використання в аналітичних і експериментальних дослідженнях.

У трасі Google Cluster більшість ресурсних показників попередньо нормалізовані відносно максимально доступної потужності відповідного ресурсу на будь-якій машині кластера. До таких ресурсів належать, зокрема, використання центрального процесора (CPU), оперативної пам’яті, дискового простору та часу введення-виведення. Нормалізація забезпечує уніфіковане представлення даних і спрощує порівняння навантажень між різними машинами та часовими інтервалами.

Таким чином, набір даних Google Cluster Trace 2011 є репрезентативним прикладом реального промислового навантаження великомасштабної хмарної інфраструктури та забезпечує надійну основу для дослідження методів

прогнозування серверних навантажень на основі часових рядів.

3.2.2 Попередня обробка та підготовка даних

Попередня обробка даних є критично важливим етапом у задачах прогнозування серверних навантажень, оскільки якість вхідних даних безпосередньо впливає на точність і узагальнювальну здатність побудованих моделей. Набір даних Google Cluster Trace характеризується великим обсягом та складною структурою, що потребує поетапної фільтрації й трансформації перед використанням у прогнозних моделях.

На першому етапі обробки даних було визначено підмножину завдань, які є релевантними для дослідження. З цією метою використано таблицю task_events, за допомогою якої відібрано ідентифікатори завдань, що містять процеси з високим рівнем пріоритету. Такий вибір обґрунтований тим, що високопріоритетні процеси є чутливими до затримок і мають першочерговий доступ до ресурсів, а отже їхнє навантаження є найбільш критичним з точки зору управління ресурсами. У результаті фільтрації було відібрано приблизно 50 000 завдань.

Через значний розмір таблиці task_usage, яка містить детальні часові записи використання ресурсів, для подальшої обробки застосовано фреймворк Dask, що дозволяє ефективно працювати з великими обсягами даних у паралельному режимі. На цьому етапі було відібрано всі процеси, що належать до попередньо відфільтрованих завдань, і збережено їх у вигляді окремих файлів. Слід зазначити, що не всі завдання, які присутні в таблиці *task_events*, мають відповідні записи в таблиці *task_usage*. Після зіставлення даних було встановлено, що близько 5% завдань не містять інформації про фактичне використання ресурсів і були виключені з подальшого аналізу.

Наступним кроком стала фільтрація завдань за тривалістю виконання. Було

відкинуто ті завдання, для яких кількість часових записів була меншою за 4032. Такий поріг відповідає сценарію, коли завдання виконується протягом двох тижнів із дискретністю п’ять хвилин і містить достатню кількість спостережень для виявлення довготривалих закономірностей у часовому ряді. У результаті цього кроку залишилося 3793 завдання.

Особливу увагу приділено обробці відсутніх значень. Видалення або ігнорування пропусків не застосовувалося, оскільки це могло призвести до втрати важливої інформації та порушення часової структури даних. Натомість використано метод інтерполяції, реалізований у Dask, який ґрунтується на пошуку найближчих значень у часовому ряді в обох напрямках. Окрім цього, з даних було вилучено рядки, які не містили змістовної інформації про використання ресурсів, зокрема записи з нульовим максимальним використанням CPU.

Оскільки початкові часові мітки процесів не завжди відповідали фіксованому інтервалу в п’ять хвилин, усі часові значення було округлено. Це дозволило уніфікувати часову шкалу та спростити подальшу агрегацію процесів. Після цього всі процеси, що стартували в один і той самий момент часу, об’єднувалися в один запис, а їхні ресурсні показники агрегувалися шляхом підсумовування. Для кожного такого агрегованого запису додатково створювалася ознака, що відображає кількість процесів.

Після завершення агрегації до кожного файлу було додано інформацію про клас планування (scheduling class). На цьому етапі в наборі даних залишилося 2261 завдання, які містили достатню кількість коректних часових записів та були придатні для подальшого аналізу й побудови моделей прогнозування.

У межах даного дослідження основною цільовою змінною було обрано середнє використання CPU, оскільки саме цей показник найчастіше використовується в наукових роботах з прогнозування серверних навантажень. Для оцінювання доцільності включення додаткових ознак було застосовано кореляційний аналіз із використанням коефіцієнта Пірсона. Отримані результати показали, що хоча деякі показники пам’яті мають помірну кореляцію з

використанням CPU, доцільним є зосередження на прогнозуванні середнього завантаження процесора як основної характеристики навантаження.

Таким чином, виконана попередня обробка забезпечила формування якісного та узгодженого набору даних, придатного для подальшої реалізації та експериментального дослідження моделей прогнозування серверних навантажень.

3.3 Архітектура системи прогнозування навантаження серверів

Архітектура системи прогнозування навантаження серверів у даній роботі побудована з урахуванням особливостей обраного набору даних, характеристик часових рядів серверних навантажень та вимог до порівняльного аналізу різних методів прогнозування. Основною метою архітектури є забезпечення відтворюваного, масштабованого та гнучкого процесу прогнозування, який дозволяє оцінювати ефективність статистичних методів, методів машинного навчання та методів глибинного навчання в єдиному експериментальному середовищі.

Система прогнозування реалізує послідовний конвеєр обробки даних, який охоплює етапи завантаження та підготовки даних, формування вхідних вибірок, навчання моделей, генерації прогнозів і оцінювання їхньої якості за визначеними метриками. Такий підхід відповідає методології, описаній в оригінальному дослідженні, та дозволяє коректно порівнювати результати різних підходів прогнозування.

Архітектура системи логічно поділяється на такі основні компоненти:

- модуль обробки та підготовки даних, який відповідає за завантаження даних Google Cluster Trace, фільтрацію завдань, обробку відсутніх значень, агрегацію ресурсних показників та формування часових рядів фіксованої довжини. Цей модуль забезпечує узгодженість даних для подальшого

використання у всіх моделях прогнозування;

- модуль формування вибірок, у межах якого реалізується перетворення часових рядів у формат, придатний для навчання моделей. Зокрема, тут визначається кількість часових міток, що використовуються як вхідні дані, а також горизонт прогнозування — як для одночасового, так і для багатокрокового прогнозування. Такий підхід безпосередньо відповідає цілям дослідження щодо аналізу впливу довжини вхідного вікна та кількості прогнозованих кроків на якість прогнозів;

- модуль моделей прогнозування, який включає реалізації статистичних моделей, моделей машинного навчання та моделей глибинного навчання. Усі моделі інтегровані в єдину архітектуру з уніфікованим інтерфейсом, що дозволяє здійснювати їх навчання та тестування за однакових умов;

- модуль експериментального оцінювання, який відповідає за обчислення метрик якості прогнозування, зокрема середньої абсолютної та середньоквадратичної похибок, коефіцієнта детермінації та метрики Naïve Ratio. Окрему увагу приділено аналізу наївної поведінки моделей, що є важливим аспектом оцінювання часових прогнозів;

- модуль узагальнення та перевірки стійкості, у межах якого реалізується навчання моделей на множині часових рядів з подальшим прогнозуванням навантаження для нових завдань. Для цього використовується попередня кластеризація завдань, що дозволяє оцінити здатність моделей узагальнювати поведінку навантажень.

Запропонована архітектура забезпечує чітке розділення відповідальностей між окремими компонентами системи та дозволяє легко розширювати або модифікувати окремі етапи без порушення загальної логіки роботи. Це є особливо важливим для проведення експериментальних досліджень, у межах яких необхідно порівнювати різні методи прогнозування за однакових умов.

Таким чином, архітектура системи прогнозування навантаження серверів створює методологічну основу для подальшої реалізації моделей, проведення експериментів та аналізу отриманих результатів, що детально розглядається у

наступних підрозділах даного розділу.

3.3.1 Загальна структура системи

Загальна структура системи прогнозування навантаження серверів побудована у вигляді послідовного конвеєра обробки даних, який охоплює всі етапи — від отримання сирих даних до формування та оцінювання прогнозів.

Такий підхід відповідає архітектурі, що використовується в оригінальному дослідженні, і дозволяє забезпечити коректність, відтворюваність та порівнюваність експериментальних результатів.

Загальна структура системи прогнозування складається з таких основних компонентів: модуль підготовки даних, модуль формування вхідних послідовностей, модуль прогнозування та модуль оцінювання якості моделей. Кожен з цих компонентів відіграє важливу роль у забезпеченні коректності та порівнюваності результатів.

3.3.2 Використані програмні засоби та технології

У рамках аналізованого дослідження для реалізації моделей прогнозування використовуються сучасні інструменти машинного навчання та глибинного навчання, зокрема бібліотеки Python для обробки даних та фреймворки для побудови нейронних мереж. Це забезпечує відтворюваність експериментів та можливість масштабування моделей для роботи з великими наборами даних.

3.4 Кластеризація робочих навантажень серверів

У реальних хмарних та серверних середовищах навантаження, що генеруються різними завданнями або сервісами, можуть суттєво відрізнятися за своєю динамікою, інтенсивністю та структурою. Використання єдиної моделі прогнозування для всіх типів навантажень не завжди дозволяє досягти високої точності, оскільки модель має узагальнювати поведінку часових рядів із різними характеристиками.

З метою підвищення узагальнювальної здатності моделей прогнозування та врахування подібності між окремими часовими рядами у даній роботі використовується підхід кластеризації робочих навантажень серверів. Кластеризація дозволяє згрупувати часові ряди з подібною поведінкою, що дає змогу аналізувати та прогнозувати навантаження з урахуванням їхніх спільних характеристик.

3.4.1 Мета застосування кластеризації у задачі прогнозування

Основною метою застосування кластеризації в задачі прогнозування серверних навантажень є підвищення ефективності та точності прогнозних моделей за рахунок групування подібних часових рядів. Кластеризація дозволяє:

- виявити групи завдань або серверних навантажень зі схожими часовими характеристиками;
- зменшити вплив різнорідності даних на процес навчання моделей;
- забезпечити можливість навчання моделей на кількох часових рядах із подібною динамікою;
- покращити здатність моделей узагальнювати поведінку нових, раніше невідомих навантажень.

Такий підхід є особливо актуальним для хмарних середовищ, де навантаження можуть суттєво змінюватися в часі та мати складну нелінійну структуру.

3.4.2 Вибір метрики подібності часових рядів

Ключовим етапом кластеризації часових рядів є вибір метрики, яка дозволяє адекватно оцінити подібність між різними часовими послідовностями. Класичні метрики відстані, такі як евклідова, мангеттенська або чебишевська відстані, передбачають жорстке вирівнювання часових рядів за індексами часу. Це означає, що і-й елемент одного ряду порівнюється лише з і-м елементом іншого.

Такий підхід не завжди є коректним для часових рядів серверних навантажень, оскільки подібні патерни можуть бути зміщені в часі або мати різну тривалість. У зв’язку з цим для оцінювання подібності часових рядів у даній роботі використовується метод динамічного вирівнювання часових рядів (Dynamic Time Warping, DTW), який проілюстровано на рисунку 3.1.

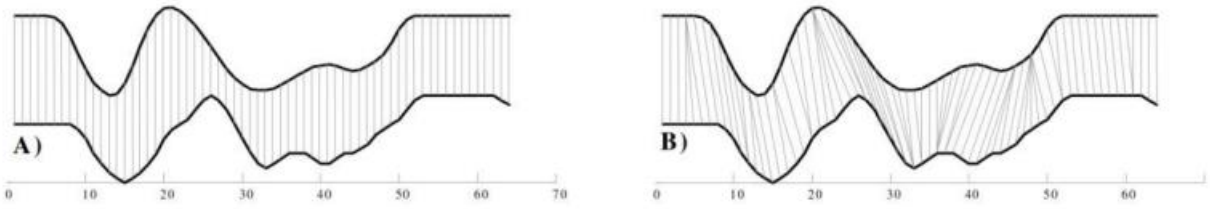


Рисунок 3.1 – Порівняння вирівнювання часових рядів при обчисленні відстані

Евклідова відстань, яка припускає, що *i*-та точка одного часового ряду вирівнюється з *i*-тою точкою іншого часового ряду (A), призводить до отримання песимістичної міри несхожості. Нелінійне вирівнювання (B) дає змогу обчислити більш інтуїтивно зрозумілу міру відстані.

3.4.3 Метод динамічного вирівнювання часових рядів (DTW)

Dynamic Time Warping є методом вимірювання подібності між двома часовими рядами, який дозволяє нелінійно вирівнювати послідовності у часі. На відміну від класичних метрик, DTW допускає стискання та розтягування часової осі, що дає змогу коректно порівнювати часові ряди з подібною формою, але різною швидкістю зміни.

У даній роботі DTW застосовується для оцінювання подібності між часовими рядами серверних навантажень, зокрема за показниками середнього використання центрального процесора та використання пам’яті. Такий підхід дозволяє враховувати не лише абсолютні значення навантаження, але й загальну динаміку його зміни в часі.

Використання DTW є обґрунтованим вибором для задачі кластеризації серверних навантажень, оскільки дозволяє коректно ідентифікувати подібні патерни навіть за наявності часових зсувів.

Було прийнято рішення додатково використовувати стовпець канонічного використання пам’яті для кластеризації завдань, оскільки він має помірний додатний зв’язок із середнім використанням центрального процесора, а додавання ще одного виміру призводить до збільшення інформаційної насиченості даних.

Після завершення процесу, який виявився дуже тривалим навіть після паралелізації обчислень, були отримані значення, на основі яких здійснюється порівняння завдань для кластеризації. Зокрема, повний час виконання програми склав приблизно 564 327 секунд, що відповідає майже 157 годинам або орієнтовно 6,5 доби. У результаті було сформовано матрицю розміром 532 × 532 = 283 024 значення. Із цих значень 76 538 дорівнюють нулю, лише 1 062 мають значення більше ніж 1, а середнє значення становить 0,018. Іншими словами, завдання є дуже подібними між собою.

Після обчислення значення, на основі якого можна порівнювати окремі завдання між собою, наступним кроком є вибір відповідного методу їх

кластеризації. Методи кластеризації можна поділити на декілька категорій. Найбільш значущі підходи наведені в таблиці 3.1 разом із коротким обґрунтуванням рішення щодо їх використання або відмови від нього.

Таблиця 3.1- Різні методи кластеризації

Methods	**Main idea behind this method** **Is it the most suitable for this thesis**
hierarchical	Partition instances top-down/bottom-up, result is a dendrogram **No**, high space and time complexity, not always best results
partitioning	Relocate instances from one cluster to another, starting from initial partitioning **No**, clusters have to be pre-set
density-based	A point belongs to a cluster if it is close to many points from that cluster **Yes**, good for arbitrary shaped clusters or detecting outliers
model-based clustering	Each data point has a probability of belonging to each cluster **No**, number of clusters have to be pre-set

3.4.4 Вибір та налаштування алгоритму кластеризації (DBSCAN)

Після визначення метрики подібності необхідно обрати відповідний алгоритм кластеризації. У даній роботі використовується алгоритм DBSCAN (Density-Based Spatial Clustering of Applications with Noise), який належить до класу щільнісних методів кластеризації.

Основними перевагами DBSCAN є:

- відсутність необхідності заздалегідь задавати кількість кластерів;

- здатність виявляти кластери довільної форми;

- можливість ідентифікації шумових або аномальних часових рядів.

Алгоритм DBSCAN використовує два основні параметри: максимальну відстань між об’єктами для визнання їх сусідами та мінімальну кількість об’єктів у щільній області. Значення цих параметрів підбираються експериментально з урахуванням особливостей обраного набору даних.

Застосування DBSCAN у поєднанні з метрикою DTW дозволяє ефективно групувати серверні навантаження зі схожою поведінкою та використовувати результати кластеризації у подальшому аналізі й прогнозуванні.

Додатково було застосовано метод *k*-середніх (*k*-means) з метою оцінювання можливості отримання більш сприятливого розподілу кластерів. Проте збільшення кількості кластерів за допомогою цього методу також призводило до формування додаткових кластерів із незначною кількістю завдань. У зв’язку з цим було вирішено зберегти поділ на два кластери, встановивши максимальну відстань між двома зразками на рівні 0,0005 та мінімальну кількість зразків у кластері, що дорівнює 50.

Загалом на початковому етапі у кластері 0 знаходилося 431 завдання, тоді як у кластері −1 — 101 завдання. Після проведення додаткових експериментів було прийнято рішення виконати повторну кластеризацію з максимальним значенням відстані 0,00005, у результаті чого у кластері 0 опинилося 275 завдань, а у кластері −1 — 257 завдань. Розміри кластерів проілюстровано на рисунку 3.2.

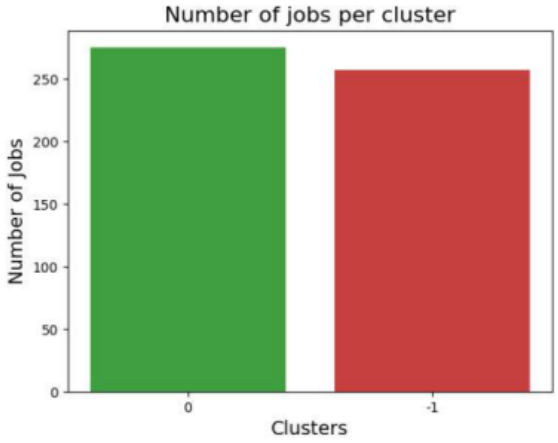


Рисунок 3.2 - Кількість завдань у кожному кластері

3.5 Реалізація моделей прогнозування

Для забезпечення коректного порівняння всі моделі прогнозування застосовуються до однаково підготовлених часових рядів і оцінюються за єдиними критеріями якості. Це дозволяє об’єктивно проаналізувати сильні та слабкі сторони кожного підходу.

3.5.1 Реалізація статистичних моделей

Статистичні моделі використовуються як базовий рівень для порівняння з більш складними методами. У межах дослідження розглядаються метод ковзного середнього та авторегресійні моделі, зокрема модель ARIMA.

Реалізація статистичних моделей передбачає попереднє згладжування часових рядів, аналіз стаціонарності та підбір параметрів моделі. Такі моделі демонструють прийнятну точність при короткостроковому прогнозуванні, однак їх здатність адекватно відтворювати різкі зміни навантаження є обмеженою. Саме тому вони використовуються переважно як еталонні моделі для оцінювання приросту точності, який забезпечують методи машинного та глибинного навчання.

3.5.2 Реалізація моделей машинного навчання

Моделі машинного навчання застосовуються для прогнозування серверних навантажень з метою врахування нелінійних залежностей у часових рядах. У дослідженні аналізуються підходи на основі лінійної регресії, методу опорних векторів та ансамблевих методів, зокрема випадкового лісу.

Реалізація таких моделей включає формування вхідних ознак на основі ковзних вікон, нормалізацію даних та підбір гіперпараметрів. Особливу увагу приділено запобіганню перенавчанню шляхом використання методів крос-валідації та регуляризації. У результаті моделі машинного навчання демонструють кращу узагальнювальну здатність порівняно зі статистичними підходами, однак потребують більшого обсягу навчальних даних.

3.5.3 Реалізація моделей глибинного навчання

Моделі глибинного навчання використовуються для прогнозування серверних навантажень з урахуванням довготривалих часових залежностей. У межах дослідження розглядаються штучні нейронні мережі, рекурентні нейронні мережі, LSTM та Transformer-архітектури.

Реалізація моделей глибинного навчання передбачає використання багатошарових нейронних мереж із механізмами уваги та пам’яті. Навчання моделей здійснюється на попередньо підготовлених часових рядах із використанням оптимізаторів градієнтного спуску та процедур налаштування гіперпараметрів. Результати експериментів свідчать про високу точність таких моделей, особливо при прогнозуванні на середніх і довгих часових горизонтах.

3.6 Методика проведення експериментальних досліджень

Методика експериментальних досліджень ґрунтується на порівняльному аналізі різних моделей прогнозування за однакових умов. Для кожної моделі використовується однаковий набір даних, ідентична схема формування навчальних та тестових вибірок, а також спільний набір метрик оцінювання якості прогнозів.

Оцінювання результатів здійснюється за допомогою таких метрик, як середньоквадратична похибка, середня абсолютна похибка, коефіцієнт детермінації та метрика Naïve Ratio. Такий підхід дозволяє не лише оцінити абсолютну точність моделей, але й виявити наївну поведінку прогнозів, що є важливим для практичного використання моделей у системах управління ресурсами.

третьому розділі було сформульовано постановку задачі прогнозування

серверних навантажень та визначено методологічні засади побудови системи прогнозування. Задача прогнозування розглядається як задача регресії часових рядів з можливістю однокрокового та багатокрокового прогнозування, а також з урахуванням узагальнювальної здатності моделей.

Обґрунтовано вибір набору даних Google Cluster Trace 2011 як репрезентативного джерела реальних промислових навантажень. Детально описано структуру набору даних, а також основні етапи попередньої обробки та підготовки часових рядів, необхідні для коректного застосування методів прогнозування.

Запропоновано архітектуру системи прогнозування навантаження серверів, яка забезпечує відтворюваність експериментів та можливість порівняльного аналізу статистичних методів, методів машинного та глибинного навчання. Окрему увагу приділено застосуванню кластеризації часових рядів з використанням метрики DTW та алгоритму DBSCAN з метою підвищення узагальнювальної здатності моделей.

Описана методика реалізації моделей та проведення експериментальних досліджень створює основу для подальшого аналізу результатів прогнозування та оцінювання ефективності різних підходів, що буде розглянуто у наступному розділі роботи.

4 ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ

4.1 Налаштування експериментів та сценарії тестування

Метою експериментальних досліджень є оцінювання ефективності різних методів прогнозування серверних навантажень на основі часових рядів, а також аналіз їхньої узагальнювальної здатності, стійкості до наївної поведінки та точності прогнозування на різних часових горизонтах.

Експерименти проводяться у декілька етапів, що дозволяє комплексно дослідити поведінку моделей у різних сценаріях використання. Основний акцент зроблено на порівнянні статистичних методів, методів машинного навчання та глибинного навчання при прогнозуванні навантаження серверів у хмарному середовищі.

Метою прогнозування є точне передбачення навантаження кластера Google за даними 2011 року, у межах якого здійснювався моніторинг різних показників споживання обчислювальних ресурсів. Основною цільовою величиною прогнозування є середнє використання ресурсу центрального процесора, оскільки саме цей показник найчастіше прогнозується в суміжних дослідженнях. Крім того, було прийнято рішення прогнозувати використання ресурсів на рівні завдань, а не задач. Загальну схему проведення експериментальних досліджень показано на рисунку 4.1.

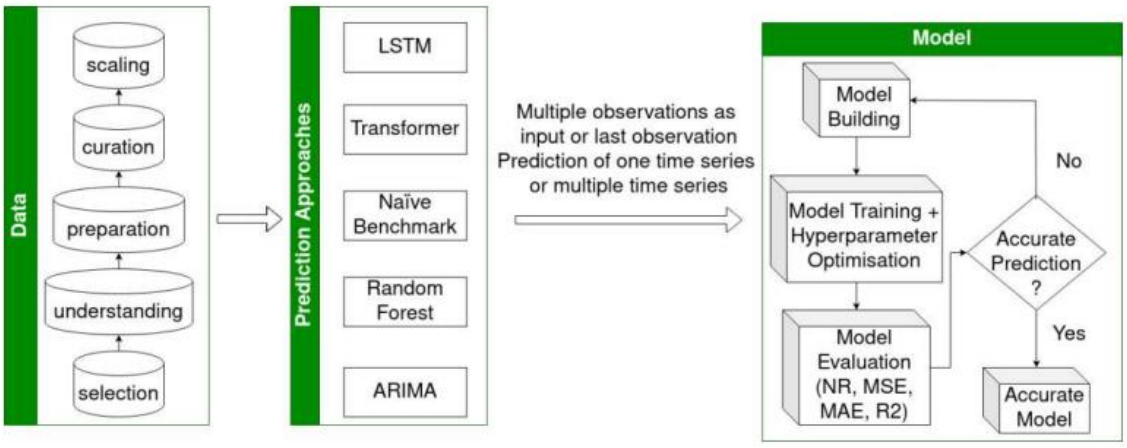


Рисунок 4.1 - Загальна схема проведення експериментальних досліджень

Перш за все, підготовка даних є критично важливим етапом. Після вибору набору даних та отримання початкових уявлень про нього наступним кроком є підготовка даних до прогнозування. Метою на цьому етапі є не прогнозування споживання ресурсів на рівні задач, а передбачення використання ресурсів на рівні завдань, що зумовлює необхідність модифікації файлів даних.

Далі дані підлягають очищенню, що включає обробку відсутніх значень. Для деяких методів додатково виникає потреба у масштабуванні даних. Наступним кроком є вибір різних методів прогнозування, з якими будуть порівнюватися отримані результати. Після цього можна переходити до виконання експериментальних досліджень.

На початковому етапі кожна архітектура проходить навчання та тестування з використанням одного завдання з метою оцінювання загальної точності прогнозування та виявлення можливих очевидних відмінностей між підходами. Такі експерименти дають змогу проаналізувати вплив довжини історії спостережень на результати прогнозування, а також оцінити точність прогнозів на декілька часових кроків уперед.

Наступний етап передбачає оцінювання здатності архітектур до узагальнення з урахуванням результатів, отриманих на попередньому етапі. З цією метою моделі навчаються та тестуються з використанням множини завдань. Для порівняння різних методів використовуються такі метрики як середньоквадратична похибка

(Mean Squared Error, MSE), середня абсолютна похибка (Mean Absolute Error, MAE), наївне відношення (Naïve Ratio, NR) та коефіцієнт детермінації (R-squared, R^2).

Огляд різних експериментів наведено в таблиці 4.1.

Таблиця 4.1 - Огляд експериментів

Architectures	Prediction	History	Data
Naïve Benchmark (NB)	one step ahead	last observation	one time series
RF	multiple steps ahead	multiple observations	multiple time series
ARIMA			
LSTM			
Transformer			

У межах дослідження розглядаються такі сценарії тестування:

- короткострокове прогнозування з передбаченням одного наступного часового кроку;

- багатокрокове прогнозування з передбаченням кількох часових міток наперед;

- аналіз впливу довжини вхідного вікна (кількості попередніх значень часового ряду) на точність прогнозування;

- оцінювання узагальнювальної здатності моделей при навчанні на одному або декількох часових рядах і тестуванні на нових, раніше не використаних навантаженнях.

Для забезпечення коректності порівняння всі моделі оцінюються з використанням однакових наборів даних, однакових метрик якості прогнозування та узгоджених умов експерименту. Основними показниками ефективності є середньоквадратична похибка (MSE), середня абсолютна похибка (MAE), коефіцієнт детермінації R^2, а також метрика Naïve Ratio, яка дозволяє оцінити наявність та ступінь наївної поведінки моделей.

Особливу увагу приділено аналізу того, як різні моделі поводяться при прогнозуванні на довших часових горизонтах, оскільки саме в таких умовах найчастіше проявляються проблеми накопичення похибок та спрощених прогнозів, що не відображають реальну динаміку серверних навантажень.

4.2 Аналіз результатів прогнозування серверних навантажень

У даному підрозділі наведено результати експериментальних досліджень методів прогнозування серверних навантажень, а також проведено їх порівняльний аналіз за обраними метриками якості. Основною метою аналізу є оцінювання точності прогнозування, здатності моделей до узагальнення та їхньої поведінки при прогнозуванні на різних часових горизонтах.

Спочатку описується випадково обране завдання з ідентифікатором 3418324, яке використовується для тестування. Далі подаються вибрані архітектури для прогнозування одного часового ряду, а також пояснюються причини їх вибору. Окрім цього, розглядається процес навчання та покращення наших моделей глибинного навчання. У наступному розділі ці раніше згадані архітектури, які використовувалися для прогнозування окремих часових рядів, застосовуються для оцінювання ефективності моделей на нових часових рядах. Що стосується вибору моделей для частини узагальнення, надаються детальні пояснення щодо кроків, виконаних для покращення моделей.

Завдання 3418324 належить до класу планування 3 та складається з 2 задач. Його початковий часовий штамп становить 600 000 000, а останній часовий штамп — 2 505 900 000 000. Відповідно, завдання розпочалося 1 травня 2011 року о 19:00, а останній часовий штамп припадає на 30 травня 2011 року о 18:50.

Середнє значення використання CPU (mean CPU usage) становить 0.0010, мінімальне значення — 0.0, а максимальне — 0.0358. Середнє завантаження CPU для цього завдання зображено на Рисунку 4.2.

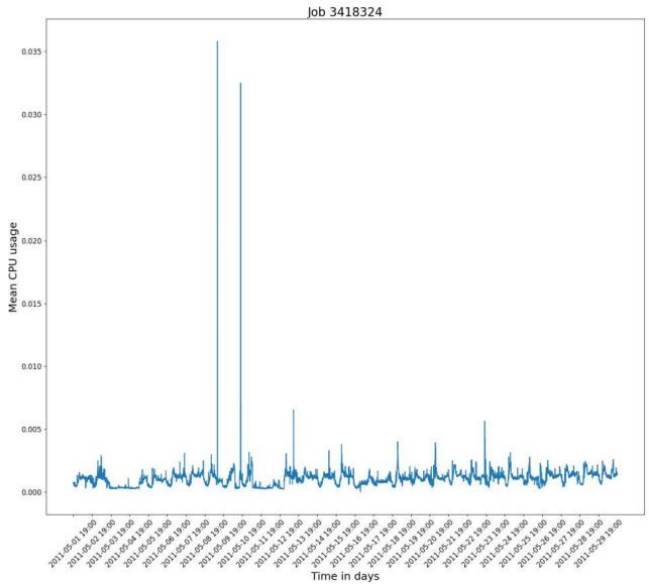


Рисунок 4.2 – Часовий ряд середнього завантаження CPU для вибраного робочого навантаження

4.2.1 Результати однокрокового прогнозування

На першому етапі експериментів було розглянуто задачу однокрокового прогнозування, коли модель передбачає значення навантаження серверів на наступний часовий крок на основі попередніх спостережень. Такий сценарій є базовим і дозволяє оцінити початкову ефективність кожного з підходів.

Для порівняння продуктивності зі статистичними моделями було використано реалізацію ARIMA з бібліотеки statsmodels.

Модель ARIMA має три різні параметри, які необхідно обрати, і які зазвичай записуються у вигляді ARIMA(p, d, q). Параметр p — це кількість авторегресивних (auto-regressive, AR) членів або лагів стаціонаризованого часового ряду, параметр d описує кількість несезонних різниць, необхідних для досягнення стаціонарності. Якщо часовий ряд потребує диференціювання для приведення до стаціонарного вигляду, він вважається інтегрованим (integrated, I) рядом. Останній параметр q — це кількість запізнених похибок прогнозування, які в рівнянні прогнозу називаються ковзним середнім (moving average, MA).

Щоб полегшити визначення порядку моделі ARIMA, було прийнято рішення використати auto_arima, який автоматично визначає найбільш оптимальні параметри для моделі ARIMA. Для цього завдання було визначено порядок (0, 1, 5). Модель **ARIMA(0,1,1)** може бути описана як модель простого експоненціального згладжування. Іншими словами, для обробки автокорельованих похибок у моделі випадкового блукання для нестаціонарних часових рядів можна використовувати просте експоненціальне згладжування. Цей підхід передбачає усереднення кількох останніх спостережень з більшими вагами для новіших даних, що призводить до більш точної оцінки локального середнього значення та кращого врахування автокореляції.

Результати експериментів показують, що статистичні методи, зокрема ковзне середнє та модель ARIMA, демонструють задовільну якість прогнозування лише за умов відносно стабільної поведінки навантаження. У випадках різких змін або високої варіативності часових рядів точність прогнозів суттєво знижується, що проявляється у збільшенні значень MAE та MSE.

Методи машинного навчання, такі як лінійна регресія, SVM та випадковий ліс, забезпечують вищу точність прогнозування порівняно зі статистичними підходами. Особливо це помітно для ансамблевих моделей, які завдяки усередненню прогнозів окремих дерев рішень демонструють більшу стійкість до шуму та локальних аномалій у даних.

Як еталон (бенчмарк) для методів машинного навчання було обрано Random Forest (RF), оскільки в цій дипломній роботі необхідно перевірити, наскільки добре ця ансамблева методика навчання демонструє свою ефективність. Двома найважливішими параметрами для моделі RF є кількість дерев (Number of Trees, t) та максимальна глибина дерев (Maximum Depth, md).

Для методів глибинного навчання використовуються LSTM та Transformer. Після ретельного аналізу було вирішено застосувати LSTM, з огляду на його часте використання в пов’язаних дослідженнях. Крім того, з метою застосування сучасного (state-of-the-art) підходу, було прийнято рішення реалізувати метод Transformer.

Перш ніж дані можуть бути використані для навчання або тестування моделі, їх необхідно масштабувати. Це може бути виконано шляхом нормалізації або стандартизації. Для нормалізації даних спочатку було визначено мінімальне та максимальне значення для обраного завдання, після чого використали формулу, наведену в Рівнянні 4.1, для масштабування всього набору даних.

$$cpu = (cpu - min_{cpu})/(max_{cpu} - min_{cpu}) \quad (4.1)$$

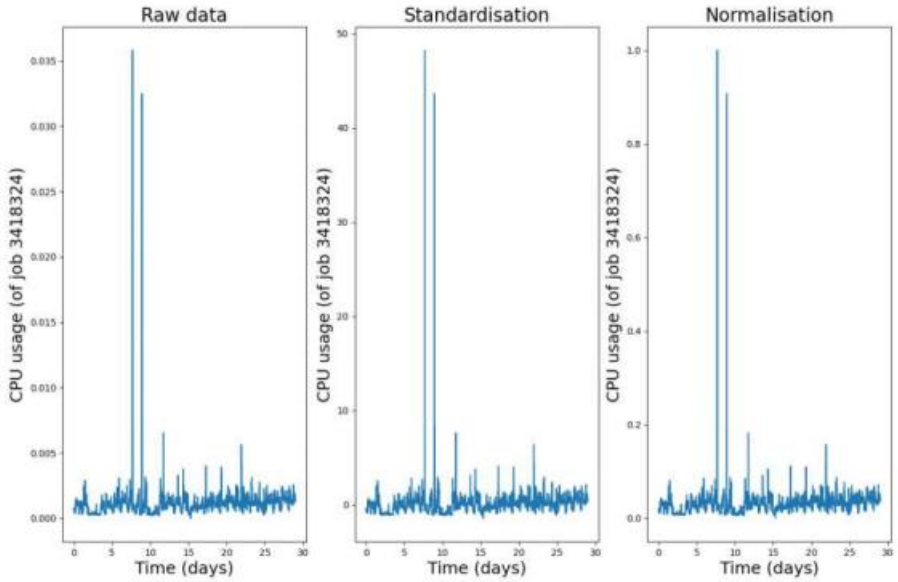


Рисунок 4.3 - Середнє використання CPU з різними методами масштабування

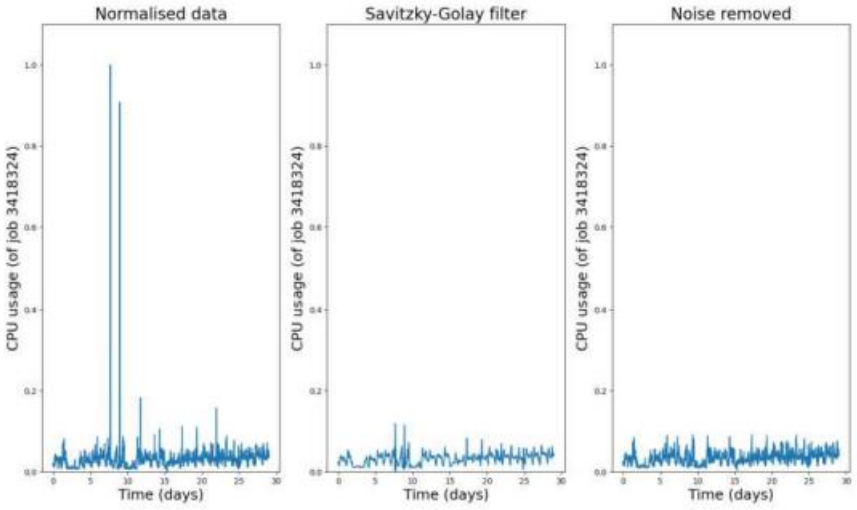


Рисунок 4.4 - Середнє використання CPU із застосуванням фільтра SavGol та видаленням шуму

Для стандартизації або Z-оцінкової нормалізації (Z-score normalisation)

спочатку було обчислено середнє значення та стандартне відхилення даних, після чого весь набір даних було масштабовано з використанням рівняння 4.2.

$$cpu = (cpu - mean_{cpu})/std_{cpu} \quad (4.2)$$

Результати обох методів показані на Рисунку 4.3. При детальнішому розгляді рисунка видно, що масштаб нормалізованих даних знаходиться в межах від 0 до 1, тоді як для стандартизованих даних він перебуває в діапазоні від 0 до 50. Для проведення експериментів у дослідженні застосовується нормалізація даних. Слід зазначити, що для наведеного тут порівняння було використано мінімальні та максимальні значення всього завдання. Однак для експериментів було знайдено мінімальне та максимальне значення лише в навчальних даних, перш ніж масштабувати весь набір даних.

Окрім масштабування даних, їх також можна згладжувати за допомогою фільтра. На Рисунку 4.4 показано різницю між нормалізованими даними та даними, для яких після нормалізації було застосовано фільтр SavGol. Крім того, представлено нормалізовані дані після застосування усунення шуму. Ці додаткові операції можуть бути корисними для моделей, оскільки допомагають уникати перенавчання на навчальних даних та ефективніше навчатися трендам.

Найкращі результати однокрокового прогнозування були отримані з використанням моделей глибинного навчання, зокрема рекурентних нейронних мереж та LSTM. Ці моделі здатні ефективно враховувати часову залежність між послідовними значеннями навантаження, що забезпечує нижчі значення похибок та вищі значення коефіцієнта детермінації R^2.

4.2.2 Багатокрокове прогнозування та накопичення похибок

На другому етапі експериментів було досліджено багатокрокове прогнозування, коли модель передбачає значення навантаження серверів на кілька часових міток наперед. Такий підхід є більш складним і дозволяє оцінити,

наскільки стабільними залишаються прогнози при збільшенні горизонту прогнозування.

Результати показують, що зі збільшенням кількості прогнозованих часових кроків точність усіх моделей поступово знижується. Для статистичних методів це зниження є найбільш різким, що пояснюється накопиченням похибок та неспроможністю таких моделей адекватно реагувати на зміни динаміки навантаження.

Методи машинного навчання демонструють кращу поведінку при багатокроковому прогнозуванні, однак і для них спостерігається поступове зростання похибок. Особливо це помітно для моделей, що використовують рекурсивну стратегію прогнозування, коли попередні прогнозовані значення використовуються як вхідні дані для наступних кроків.

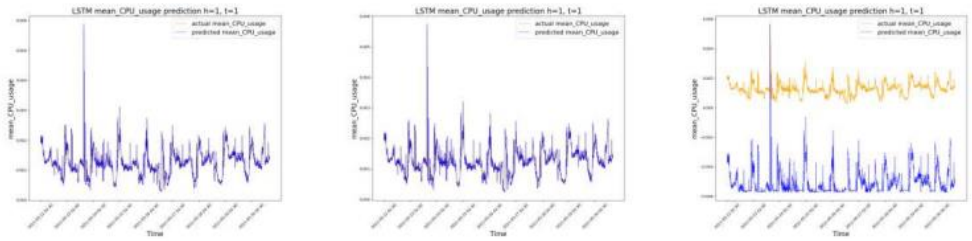
Моделі глибинного навчання, зокрема LSTM та Transformer-архітектури, показали найвищу стійкість до накопичення похибок. Завдяки здатності моделювати довготривалі часові залежності вони зберігають відносно стабільну якість прогнозування навіть при прогнозуванні на кілька часових кроків наперед.

4.2.3 Аналіз наївної поведінки моделей

Окрему увагу в експериментальних дослідженнях було приділено аналізу наївної поведінки моделей прогнозування. Для цього використовувалася метрика Naïve Ratio, яка дозволяє кількісно оцінити, наскільки сильно прогноз моделі наближається до простого зсуву попереднього значення часового ряду.

У цьому розділі описано, чи призводить використання Naïve Ratio як функції втрат до точних результатів. З цією метою у межах дослідження було розглянуто три варіанти навчання моделей LSTM:
одну модель, яка використовує лише MSE як функцію втрат,
другу — яка використовує лише NR як функцію втрат,

і третю — яка одночасно враховує обидві функції втрат, по 50 % кожну.



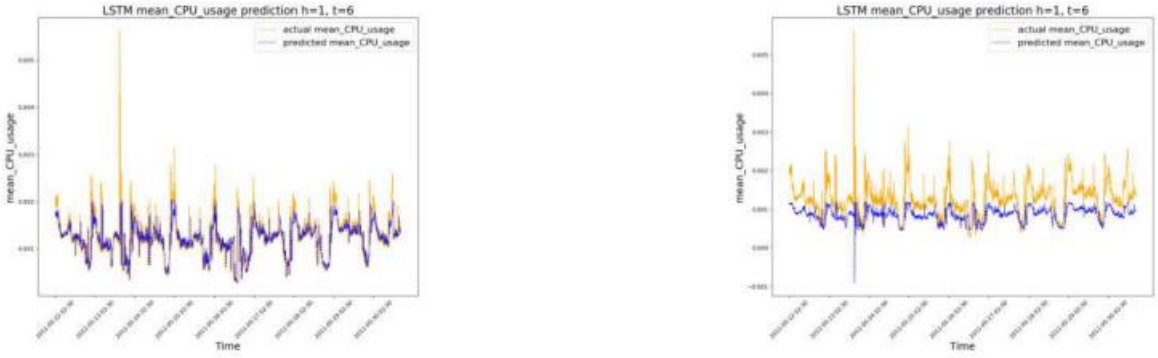
(a) MSE (b) MSE and NR (c) NR

Рисунок 4.5 - Порівняння MSE прогнозування з використанням MSE та NR як функції втрат для передбачення на один часовий крок уперед

На Рисунку 4.5 можна побачити прогноз LSTM з використанням MSE та NR як функцій втрат для прогнозування однієї часової відмітки. Рисунок 4.5a демонструє результати, подібні до Рисунка 4.5b, однак на Рисунку 4.5c прогноз є помітно гіршим, коли NR використовується як єдина функція втрат, оскільки прогнозоване середнє використання CPU (синя крива) є значно нижчим за фактичне використання CPU (помаранчева крива).

Крім того, було порівняно прогнозування шести часових відміток уперед при використанні MSE як функції втрат та при використанні MSE у поєднанні з NR. На Рисунку 4.6 видно, що прогнозування з використанням MSE як функції втрат у 4.6a призводить до більш точних результатів, ніж у випадку, коли NR також враховується, як показано у 4.6b. Результати цих тестів свідчать про те, що використання Naïve Ratio як функції втрат не призводить до покращення результатів.

Одним із можливих пояснень є те, що NR не є настільки прямолінійним показником, як MSE, що ускладнює процес навчання моделі. MSE генерує велике значення втрат, коли результат прогнозу сильно відрізняється від фактичного значення. NR, у свою чергу, генерує велике значення, коли MAE прогнозу в момент часу t та фактичного значення в момент часу t є високим, але MAE прогнозу в момент часу t+1 та фактичного значення в момент часу t є низьким. Значення NR буде низьким, коли MAE в момент часу t+1 є вищим, ніж втрата в момент часу t.



(a) MSE (b) MSE and NR

Рисунок 4.6 - Порівняння MSE прогнозування з використанням MSE та NR як функції втрат для передбачення на шість часових кроків уперед

Результати експериментів свідчать, що статистичні методи найчастіше демонструють наївну поведінку, особливо при багатокроковому прогнозуванні. Значення Naïve Ratio для таких моделей у багатьох випадках перевищує одиницю, що вказує на домінування наївного компоненту в прогнозах.

Методи машинного навчання демонструють меншу схильність до наївної поведінки, однак за недостатнього обсягу навчальних даних або невдалого вибору параметрів моделі також можуть формувати спрощені прогнози.

Найменші значення Naïve Ratio були зафіксовані для моделей глибинного навчання, що підтверджує їхню здатність формувати більш змістовні та інформативні прогнози, які не зводяться до простого копіювання попередніх значень часового ряду.

4.2.4 Узагальнення результатів

Загальний аналіз результатів експериментів показує, що жоден з розглянутих методів не є універсальним для всіх сценаріїв прогнозування. Статистичні методи можуть бути корисними як базові або порівняльні моделі, однак їх ефективність обмежена у складних та нестаціонарних умовах.

Методи машинного навчання забезпечують кращий баланс між точністю та обчислювальною складністю, тоді як методи глибинного навчання демонструють найвищу якість прогнозування, особливо при роботі з довгими часовими рядами та багатокроковими прогнозами. Водночас їх застосування потребує значних обчислювальних ресурсів і ретельного налаштування параметрів.

4.3 Порівняльний аналіз ефективності моделей прогнозування

У даному підрозділі наведено порівняльний аналіз ефективності розглянутих методів прогнозування серверних навантажень на основі результатів експериментальних досліджень. Порівняння проводиться з використанням обраних метрик якості прогнозування, зокрема MAE, MSE, коефіцієнта детермінації R^2 та метрики Naïve Ratio, що дозволяє комплексно оцінити як точність прогнозів, так і поведінку моделей у різних сценаріях.

4.3.1 Порівняння за точністю прогнозування

У Таблицях 4.2, 4.3, 4.4, 4.5, 4.6 та 4.7 наведено детальний звіт про найкращі результати кожного раунду навчання. Для кожного методу прогнозування було виділено найкращий результат сірим кольором, щоб покращити читабельність. Крім того, загально найкращі результати виділено темнішим відтінком.

У Таблиці 4.2 модель LSTM перевершила інші методи за показниками MSE та R^2. Найкращий результат було досягнуто при використанні шести часових відміток історії. Що стосується MAE, то Transformer демонструє дещо гіршу продуктивність, за яким слідує Random Forest (RF). Інші методи показали дещо знижену продуктивність за показниками MSE та R^2, однак їх результати були

порівнюваними.

Найкращі результати для прогнозування на два часові кроки вперед наведено в Таблиці 4.3, і вони були досягнуті методами глибокого навчання, де LSTM показала трохи кращу продуктивність, ніж Transformer.

Таблиця 4.2 - Прогнозування на один часовий крок уперед для завдання

Model	hyperparameters	history	MAE	MSE	R2	NR	training(s)	total(s)
NB	-	1/6/12	0.00011	5.21e-08	0.72978	∞	-	0.1
ARIMA	p=0, d=1, q=5	1	00.00011	5.28e-08	0.72667	0.57532	34.54	52.28
ARIMA	p=0, d=1, q=5	6	0.00012	5.38e-08	0.72164	0.57932	33.86	214.54
ARIMA	p=0, d=1, q=5	12	0.00012	5.47e-08	0.7167	0.5779	34.67	209.62
RF	t=150, md=3	1	0.00013	5.55e-08	0.71282	0.67374	0.33	0.34
RF	t=150, md=4	6	0.00013	5.96e-08	0.69149	0.71076	0.99	1.01
RF	t=150, md=4	12	0.00013	6.21e-08	0.67847	0.71092	1.79	1.8
LSTM	u=64, l=2, lr=0.00011, bs=32, ll=200, e=100	1	0.00011	5.25e-08	0.7283	0.57452	432.54	437.84
LSTM	u=64, l=2, lr=0.00015, bs=32, ll=200, e=100	6	0.00011	5.07e-08	0.73745	0.59124	1100.22	1105.86
LSTM	u=128, l=2, lr=0.00025, bs=8, ll=200, e=100	12	0.00013	5.43e-08	0.71913	0.6258	2246.05	2254.19
Transformer	dm=16, nh=1, lr=0.00019, bs=16, df=16, nl=1, e=150	1	0.00012	5.41e-08	0.72007	0.58288	646.16	651.13
Transformer	dm=32, nh=1, lr=0.00015, bs=4, df=16, nl=1, e=150	6	0.00014	6.26e-08	0.67592	0.70498	970.3	975.39
Transformer	dm=16, nh=1, lr=0.0002, bs=8, df=16, nl=1, e=150	12	0.00017	9.63e-08	0.50153	1.44876	1051.39	1056.68

Таблиця 4.3 -Прогнозування на два часові кроки вперед для завдання

Model	hyperparameters	history	MAE	MSE	R2	NR	training(s)	total(s)
NB	-	1/6/12	0.00014	9.02e-08	0.53203	∞	-	0.08
ARIMA	p=0, d=1, q=5	1	0.00014	9.12e-08	0.52777	0.7918	34.54	52.28
ARIMA	p=0, d=1, q=5	6	0.00014	9.30e-08	0.51853	0.79843	33.86	214.54
ARIMA	p=0, d=1, q=5	12	0.00015	9.67e-08	0.49909	0.80233	34.67	209.62
RF	t=150, md=3	1	0.00016	8.00e-08	0.58552	0.8661	0.33	0.34
RF	t=150, md=3	6	0.00016	8.00e-08	0.5859	0.91985	0.8	0.81
RF	t=150, md=3	12	0.00016	7.94e-08	0.58889	0.92015	1.39	1.41
LSTM	u=64, l=2, lr=0.00011, bs=32, ll=200, e=100	1	0.00014	7.44e-08	0.61476	0.8057	432.54	437.84
LSTM	u=64, l=2, lr=0.00015, bs=32, ll=200, e=100	6	0.00014	7.32e-08	0.62076	1.02884	1100.22	1105.86
LSTM	u=128, l=2, lr=0.00025, bs=8, ll=200, e=100	12	0.00017	1.03e-07	0.46806	0.97015	2246.05	2254.19
Transformer	dm=16, nh=1, lr=0.00019, bs=16, df=16, nl=1, e=150	1	0.00014	7.37e-08	0.61848	0.80934	646.16	651.13
Transformer	dm=32, nh=1, lr=0.00015, bs=4, df=16, nl=1, e=150	6	0.00016	8.40e-08	0.56501	0.91445	970.3	975.39
Transformer	dm=16, nh=1, lr=0.0002, bs=8, df=16, nl=1, e=150	12	0.00016	8.73e-08	0.54792	1.14658	1051.39	1056.68

Random Forest (RF) продемонстрував порівнювані результати за показниками MSE та R^2, однак значення MAE є вищим. ARIMA та Naïve Benchmark (NB) мають однакове заокруглене значення MAE, як і методи глибокого навчання (DL), проте значно вищі значення MSE та суттєво нижчі показники R^2.

У Таблиці 4.4 методи глибокого навчання також перевершили три інші підходи під час прогнозування на три часові кроки вперед. Коли як історію використовується лише одна часова відмітка, обидва DL-підходи демонструють найнижче значення MSE, найнижчий NR та найвище значення R^2. Transformer і LSTM мають однакове заокруглене значення MAE, проте LSTM має дещо гірші значення MSE та R^2. ARIMA та NB демонструють вищі значення MSE і нижчі значення R^2, але таке саме заокруглене значення MAE, як і DL-методи.

Продуктивність RF за показниками MSE та R^2 знаходиться між DL-методами та ARIMA/NB, проте MAE є вищим.

Для прогнозування на чотири часові кроки вперед, результати якого наведено в Таблиці 4.5, методи DL та ML показали порівнювані результати. Transformer має найнижчі значення MAE, MSE та NR, а також найвищий показник R^2. LSTM демонструє дещо нижчий NR і вищий R^2, ніж RF. Найкращі результати для DL-методів досягаються при використанні однієї часової відмітки як історії, тоді як RF показав найкращі результати при використанні дванадцяти часових відміток історії. Значення MSE для цього прогнозу є помітно вищим у NB та ARIMA, ніж у трьох інших методів.

Таблиця 4.6 демонструє результати прогнозування на п’ять часових кроків вперед. Методи ML та DL показують кращу продуктивність порівняно з ARIMA та NB, які демонструють подібні результати між собою. LSTM дещо перевершує RF та Transformer за показниками MSE та R^2. Проте її NR є відносно вищим. Варто зазначити, що Transformer досягає найкращого результату при використанні однієї часової відмітки як історії, тоді як LSTM та RF досягають пікової продуктивності при використанні шести часових відміток.

Таблиця 4.4 - Прогнозування на три часові кроки вперед для

Model	hyperparameters	history	MAE	MSE	R2	NR	training(s)	total(s)
NB	-	1/6/12	0.00016	1.10e-07	0.4267	∞	-	0.04
ARIMA	p=0, d=1, q=5	1	0.00016	1.11e-07	0.42303	1.00306	34.54	52.28
ARIMA	p=0, d=1, q=5	6	0.00017	1.12e-07	0.41883	1.00734	33.86	214.54
ARIMA	p=0, d=1, q=5	12	0.00017	1.15e-07	0.40279	1.0115	34.67	209.62
RF	t=150, md=3	1	0.00018	9.46e-08	0.50958	1.00758	0.33	0.34
RF	t=150, md=2	6	0.00017	9.55e-08	0.5045	1.0746	0.8	0.81
RF	t=150, md=4	12	0.00017	1.02e-07	0.46861	1.09028	1.79	1.8
LSTM	u=64, l=2, lr=0.00011, bs=32, ll=200, e=100	1	0.00016	8.94e-08	0.53651	0.99891	432.54	437.85
LSTM	u=64, l=2, lr=0.00015, bs=32, ll=200, e=100	6	0.00016	9.23e-08	0.52138	1.29144	1100.22	1105.86
LSTM	u=128, l=2, lr=0.00025, bs=8, ll=200, e=100	12	0.00018	1.26e-07	0.34701	1.14013	2246.05	2254.19
Transformer	dm=16, nh=1, lr=0.00019, bs=16, df=16, nl=1, e=150	1	0.00016	8.87e-08	0.53985	0.98877	646.16	651.13
Transformer	dm=32, nh=1, lr=0.00015, bs=4, df=16, nl=1, e=150	6	0.00018	1.07e-07	0.44323	1.44523	970.3	975.4
Transformer	dm=16, nh=1, lr=0.0002, bs=8, df=16, nl=1, e=150	12	0.00018	1.03e-07	0.46373	1.52053	1051.39	1056.68

Таблиця 4.5 - Прогнозування на чотири часові кроки вперед для завдання

Model	hyperparameters	history	MAE	MSE	R2	NR	training(s)	total(s)
NB	-	1/6/12	0.00018	1.23e-07	0.36193	∞	0	0.05
ARIMA	p=0, d=1, q=5	1	00.00018	1.24e-07	0.35774	1.26954	34.62	52.4
ARIMA	p=0, d=1, q=5	6	0.00018	1.24e-07	0.35526	1.27589	33.86	214.54
ARIMA	p=0, d=1, q=5	12	0.00018	1.26e-07	0.34518	1.29451	34.67	209.62
RF	t=150, md=3	1	0.00018	1.03e-07	0.46459	1.17355	0.33	0.34
RF	t=150, md=3	6	0.00018	1.013e-07	0.47364	1.23877	0.8	0.81
RF	t=150, md=3	12	0.00018	1.01e-07	0.47531	1.23391	1.39	1.41
LSTM	u=64, l=2, lr=0.00011, bs=32, ll=200, e=100	1	0.00018	1.01e-07	0.47731	1.19668	432.54	437.85
LSTM	u=64, l=2, lr=0.00015, bs=32, ll=200, e=100	6	0.00018	1.02e-07	0.47124	1.67039	1100.22	1105.86
LSTM	u=128, l=2, lr=0.00025, bs=8, ll=200, e=100	12	0.00024	1.64e-07	0.14654	1.16105	2246.05	2254.19
Transformer	dm=16, nh=1, lr=0.00019, bs=16, df=16, nl=1, e=150	1	0.00017	9.95e-08	0.48319	1.18547	646.16	651.13
Transformer	dm=32, nh=1, lr=0.00015, bs=4, df=16, nl=1, e=150	6	0.00019	1.07e-07	0.44667	1.49843	970.3	975.4
Transformer	dm=16, nh=1, lr=0.0002, bs=8, df=16, nl=1, e=150	12	0.00019	1.13e-07	0.4125	1.35215	1051.39	1056.69

Таблиця 4.6 - Прогнозування на п’ять часових кроків вперед для завдання

Model	hyperparameters	history	MAE	MSE	R2	NR	training(s)	total(s)
NB	-	1/6/12	0.0002	1.42e-07	0.26112	∞	0	0.06
ARIMA	p=0, d=1, q=5	1	0.0002	1.43e-07	0.25741	1.74742	34.62	52.4
ARIMA	p=0, d=1, q=5	6	0.0002	1.43e-07	0.2586	1.74775	33.86	214.54
ARIMA	p=0, d=1, q=5	12	0.0002	1.44e-07	0.25037	1.74836	34.67	209.62
RF	t=150, md=3	1	0.0002	1.15e-07	0.40023	1.37891	0.33	0.34
RF	t=150, md=4	6	0.00019	1.13e-07	0.41472	1.56504	0.99	1.01
RF	t=150, md=4	12	0.00019	1.13e-07	0.41301	1.53695	1.79	1.8
LSTM	u=64, l=2, lr=0.00011, bs=32, ll=200, e=100	1	0.00019	1.15e-07	0.40251	1.4995	432.54	437.85
LSTM	u=64, l=2, lr=0.00015, bs=32, ll=200, e=100	6	0.00019	1.11e-07	0.42098	1.87504	1100.22	1105.86
LSTM	u=128, l=2, lr=0.00025, bs=8, ll=200, e=100	12	0.00021	1.32e-07	0.31437	1.3414	2246.05	2254.2
Transformer	dm=16, nh=1, lr=0.00019, bs=16, df=16, nl=1, e=150	1	0.00019	1.13e-07	0.4117	1.48976	646.16	651.14
Transformer	dm=32, nh=1, lr=0.00015, bs=4, df=16, nl=1, e=150	6	0.0002	1.18e-07	0.38594	1.61242	970.3	975.4
Transformer	dm=16, nh=1, lr=0.0002, bs=8, df=16, nl=1, e=150	12	0.0002	1.18e-07	0.38621	1.46207	1051.39	1056.69

У Таблиці 4.7 трьома найкращими методами знову є Random Forest (RF), LSTM та Transformer. Найкращі значення MSE та показника R^2 було досягнуто за допомогою LSTM, далі йде RF, а потім Transformer. Серед цих трьох методів LSTM демонструє найвище значення Naïve Ratio (NR) та досягає пікової продуктивності при врахуванні останніх шести часових відміток. Натомість RF досягає оптимальних результатів при врахуванні останніх дванадцяти часових відміток, тоді як Transformer показує найкращу продуктивність, використовуючи лише останнє спостереження.

ARIMA та Naïve Benchmark (NB) працюють гірше порівняно з іншими методами та демонструють порівнювані між собою результати. Показник R^2 для ARIMA є дещо нижчим, ніж показник R^2 для NB.

Таблиця 4.7 - Прогнозування на шість часових кроків уперед для завдання

Model	hyperparameters	history	MAE	MSE	R2	NR	training(s)	total(s)
NB	-	1/6/12	0.00022	1.65e-07	0.13938	∞	-	0.05
ARIMA	p=0, d=1, q=5	1	0.00022	1.66e-07	0.13555	2.71e+15	34.54	52.28
ARIMA	p=0, d=1, q=5	6	0.00022	1.65e-07	0.1402	556.92499	33.86	214.54
ARIMA	p=0, d=1, q=5	12	0.00022	1.65e-07	0.13791	103.86197	34.67	209.62
RF	t=150, md=3	1	0.00021	1.23e-07	0.35775	1.93644	0.33	0.34
RF	t=150, md=4	6	0.0002	1.22e-07	0.36538	1.86524	0.99	1.01
RF	t=150, md=3	12	0.00021	1.22e-07	0.36704	1.76164	1.39	1.41
LSTM	u=64, l=2, lr=0.00011, bs=32, ll=200, e=100	1	0.0002	1.23e-07	0.35759	2.77483	432.54	437.85
LSTM	u=64, l=2, lr=0.00015, bs=32, ll=200, e=100	6	0.0002	1.21e-07	0.36965	2.4117	1100.22	1105.86
LSTM	u=128, l=2, lr=0.00025, bs=8, ll=200, e=100	12	0.00023	1.41e-07	0.26538	1.42247	2246.05	2254.2
Transformer	dm=16, nh=1, lr=0.00019, bs=16, df=16, nl=1, e=150	1	0.00021	1.28e-07	0.33212	2.13904	646.16	651.14
Transformer	dm=32, nh=1, lr=0.00015, bs=4, df=16, nl=1, e=150	6	0.00021	1.29e-07	0.32652	1.68949	970.3	975.4
Transformer	dm=16, nh=1, lr=0.0002, bs=8, df=16, nl=1, e=150	12	0.00021	1.30e-07	0.32535	1.53124	1051.39	1056.69

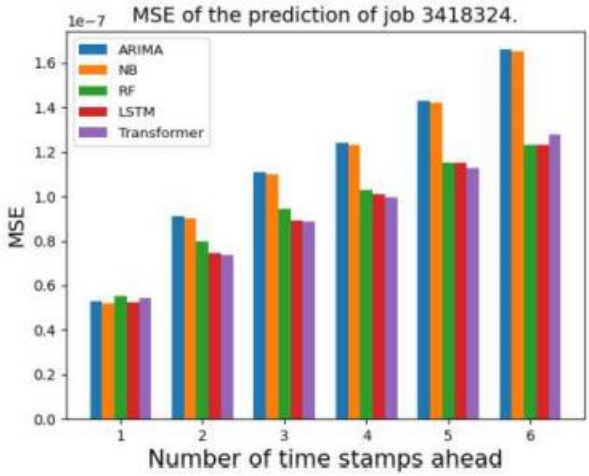


Рисунок 4.7 - Порівняння MSE для тестів з одним завданням

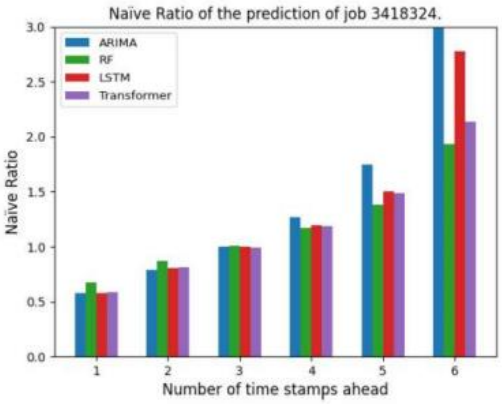


Рисунок 4.8 - Порівняння показника NR для тестів з одним завданням

На додаток до таблиць ми додали кілька графіків, щоб забезпечити краще порівняння різних методів. Для побудови рисунків було використано результати прогнозування з використанням лише однієї часової відмітки як історії.

На Рисунку 4.7 показано порівняння MSE п’яти методів для прогнозування до шести часових кроків уперед. Хоча продуктивність підходів істотно не

відрізняється при прогнозуванні наступної часової відмітки, уже при прогнозуванні на два часові кроки вперед спостерігається помітна різниця в продуктивності, яка стає ще більш очевидною при прогнозуванні на кілька часових кроків уперед.

На Рисунку 4.8 показано порівняння показника Naïve Ratio (NR) для різних методів. Оскільки Naïve Benchmark (NB) має нескінченне значення NR, його не наведено на рисунку. Для прогнозування на шість часових кроків уперед модель ARIMA має NR = 2.71e+15, що не повністю відображено на рисунку. Значення NR для LSTM, ARIMA, RF та Transformer зростає зі збільшенням кількості прогнозованих часових кроків, що означає, що прогноз стає ближчим до останнього спостереження, ніж до фактичного значення. Методи починають демонструвати наївну поведінку вже при прогнозуванні на три часові кроки уперед, коли як історія використовується лише одна часова відмітка.

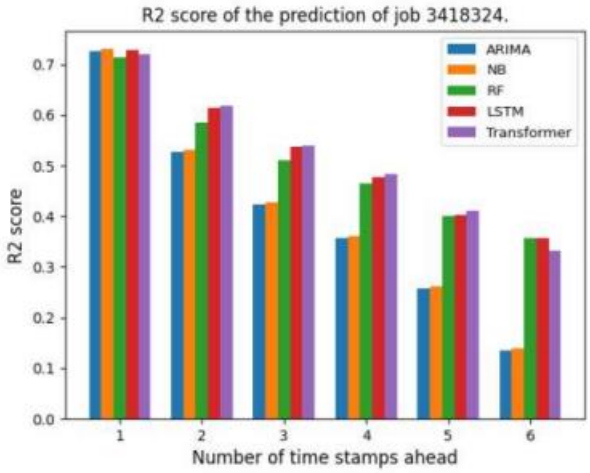


Рисунок 4.9 - Порівняння показника R^2 для тестів з одним завданням

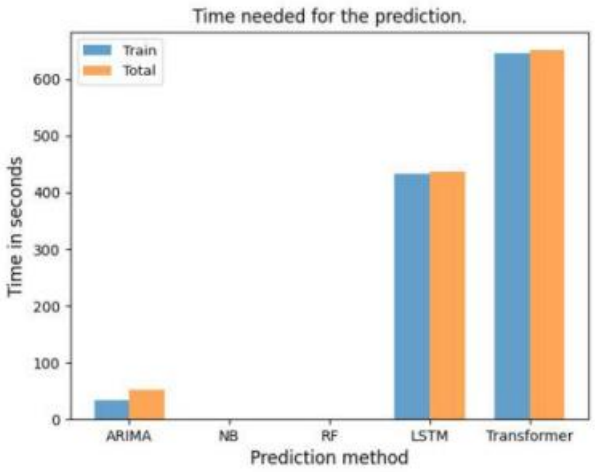


Рисунок 4.10 - Порівняння часу виконання (runtime) для тестів з одним завданням

На Рисунку 4.9 можна побачити значення показника R^2 для різних методів. При прогнозуванні на один часовий крок уперед усі методи демонструють доволі схожу продуктивність. Із збільшенням кількості прогнозованих часових кроків різниця в продуктивності методів також зростає. Методи глибокого навчання (DL) та машинного навчання (ML) мають вищі значення R^2, ніж ARIMA та Naïve Benchmark (NB), під час прогнозування на кілька часових кроків уперед.

Час виконання методів показано на Рисунку 4.10, де можна побачити час навчання та загальний час роботи методів. NB та RF потребують дуже мало часу для прогнозування. ARIMA займає більше часу, ніж NB і RF, але значно менше, ніж LSTM і Transformer. Водночас, хоча DL-методи потребують мінімального часу безпосередньо для прогнозування, ARIMA потребує більше часу. Причиною того, що Transformer навчається помітно довше, ніж LSTM, є те, що Transformer навчався протягом 150 епох, тоді як LSTM — лише протягом 100 епох.

Аналіз результатів показує, що статистичні методи прогнозування, зокрема метод ковзного середнього та модель ARIMA, демонструють найнижчі показники точності серед усіх розглянутих підходів. Вони забезпечують прийнятні результати лише для короткострокових прогнозів і за умов відносно стабільної динаміки навантаження. За наявності різких змін або високої варіативності часових рядів значення MAE та MSE істотно зростають.

Методи машинного навчання загалом перевершують статистичні підходи за

всіма розглянутими метриками. Лінійна та поліноміальна регресія демонструють помірне покращення точності, однак їх здатність до моделювання складних нелінійних залежностей є обмеженою. Метод опорних векторів (SVM) забезпечує кращі результати за рахунок використання ядерних функцій, проте його ефективність знижується зі збільшенням обсягу навчальних даних.

Найбільш стабільні та точні результати серед методів машинного навчання були отримані для ансамблевих моделей, зокрема випадкового лісу. Усереднення прогнозів окремих дерев рішень дозволяє зменшити вплив шуму та випадкових коливань, що позитивно відображається на показниках точності.

Методи глибинного навчання, особливо LSTM та Transformer-архітектури, продемонстрували найвищі значення коефіцієнта детермінації R^2 та найнижчі значення похибок. Це свідчить про їхню здатність ефективно враховувати як короткострокові, так і довготривалі часові залежності у даних серверних навантажень.

4.3.2 Порівняння моделей за стійкістю до наївної поведінки

Окремим аспектом порівняльного аналізу є оцінювання наївної поведінки моделей прогнозування. Для цього використовувалася метрика Naïve Ratio, яка дозволяє визначити, наскільки прогнози моделей відрізняються від простого зсуву попередніх значень часового ряду.

Статистичні методи продемонстрували найвищі значення Naïve Ratio, що вказує на їхню схильність до наївних прогнозів, особливо при багатокроковому прогнозуванні. Це обмежує їх практичне застосування у задачах проактивного управління ресурсами серверів.

Методи машинного навчання загалом демонструють меншу схильність до наївної поведінки, однак за невдалого вибору параметрів або недостатньої кількості навчальних даних також можуть формувати прогнози, близькі до

наївних.

Найкращі результати щодо зменшення наївної поведінки показали моделі глибинного навчання. Значення Naïve Ratio для LSTM та Transformer-моделей залишаються значно нижчими за одиницю, що свідчить про здатність цих моделей формувати прогнози, які відображають реальну динаміку навантаження, а не лише повторюють попередні значення.

4.3.3 Обчислювальна складність та практичні аспекти застосування

Окрім точності прогнозування, важливим критерієм порівняння є обчислювальна складність та практична придатність моделей. Статистичні методи мають найменші вимоги до обчислювальних ресурсів і є простими в реалізації, що робить їх зручними для швидких базових оцінок.

Методи машинного навчання потребують більшого обсягу даних та часу на навчання, однак забезпечують кращий баланс між точністю та ресурсними витратами. Ансамблеві моделі, хоча й є більш ресурсоємними, залишаються прийнятними для практичного використання у багатьох системах управління серверними навантаженнями.

Методи глибинного навчання характеризуються найвищими обчислювальними витратами та потребують значних ресурсів для навчання і налаштування. Водночас отримані результати свідчать, що у складних сценаріях із високою варіативністю навантаження ці витрати є виправданими завдяки суттєвому покращенню якості прогнозування.

4.3.4 Загальні висновки порівняльного аналізу

Порівняльний аналіз показав, що ефективність моделей прогнозування серверних навантажень суттєво залежить від складності часових рядів, горизонту прогнозування та доступного обсягу даних. Статистичні методи доцільно використовувати як базові або допоміжні моделі, тоді як методи машинного навчання та глибинного навчання забезпечують значно кращі результати у реальних умовах.

Отримані результати підтверджують доцільність використання моделей глибинного навчання для прогнозування серверних навантажень у сучасних інформаційних системах, особливо у задачах багатокрокового прогнозування та проактивного управління ресурсами.

Отримані в ході експериментальних досліджень результати дозволяють зробити низку узагальнених висновків щодо ефективності різних підходів до прогнозування серверних навантажень, а також підтвердити доцільність застосування методів машинного та глибинного навчання для аналізу складних часових рядів.

4.4 Інтерпретація результатів статистичних методів

Статистичні методи прогнозування, зокрема метод ковзного середнього та модель ARIMA, продемонстрували обмежену ефективність при роботі з реальними даними серверних навантажень. Це пояснюється тим, що такі методи базуються на припущенні відносної стаціонарності часових рядів та лінійності залежностей між спостереженнями.

У реальних серверних і хмарних середовищах навантаження характеризується різкими піковими значеннями, нерегулярними коливаннями та

довготривалими залежностями. У таких умовах статистичні моделі схильні формувати спрощені або наївні прогнози, що підтверджується високими значеннями метрики Naïve Ratio та зростанням похибок при багатокроковому прогнозуванні.

Таким чином, статистичні методи доцільно розглядати переважно як базовий рівень прогнозування або інструмент для порівняльного аналізу.

4.5 Аналіз поведінки моделей машинного навчання

Методи машинного навчання продемонстрували суттєве покращення якості прогнозування порівняно зі статистичними підходами. Це пов’язано з їх здатністю автоматично виявляти нелінійні залежності у даних та адаптуватися до змін у динаміці навантаження.

Однак результати експериментів показали, що ефективність таких моделей значною мірою залежить від якості підготовки даних, вибору ознак та параметрів навчання. За недостатньої кількості навчальних даних або невдалого налаштування моделей спостерігається нестабільність прогнозів і часткове проявлення наївної поведінки.

Ансамблеві методи, зокрема випадковий ліс, продемонстрували більш високу стійкість до шуму та перенавчання. Це пояснюється усередненням результатів декількох базових моделей, що зменшує вплив окремих помилок і забезпечує кращу узагальнювальну здатність.

4.6 Переваги та обмеження методів глибинного навчання

Найкращі результати прогнозування були отримані для моделей глибинного

навчання, зокрема мереж LSTM та Transformer-архітектур. Їх висока точність пояснюється здатністю ефективно враховувати як короткострокові, так і довготривалі часові залежності у даних серверних навантажень.

Особливо важливим є те, що ці моделі демонструють низькі значення метрики Naïve Ratio, що свідчить про відсутність наївної поведінки прогнозів. Це є критично важливим для практичних систем управління ресурсами, де некоректні або запізнілі прогнози можуть призводити до порушення угод рівня обслуговування (SLA).

Водночас результати експериментів підтверджують, що застосування методів глибинного навчання пов’язане з підвищеними обчислювальними витратами та потребою у значних обсягах навчальних даних. Це накладає обмеження на їх використання у середовищах із обмеженими ресурсами або ж вимагає застосування механізмів оптимізації та попереднього відбору моделей.

4.7 Вплив горизонту прогнозування та кількості вхідних даних

Результати дослідження також показали, що якість прогнозування істотно залежить від довжини історичного вікна та горизонту прогнозування. При збільшенні кількості часових кроків уперед статистичні та класичні ML-моделі демонструють різке зниження точності, тоді як моделі глибинного навчання зберігають прийнятний рівень прогнозування.

Це свідчить про те, що для задач проактивного управління серверними ресурсами, де необхідно прогнозувати навантаження на кілька кроків уперед, доцільно застосовувати саме моделі, здатні працювати з довготривалими залежностями.

4.8 Узагальнення результатів дослідження

Загалом отримані результати підтверджують, що не існує універсального методу прогнозування серверних навантажень, який би забезпечував найкращу ефективність у всіх сценаріях. Вибір моделі повинен здійснюватися з урахуванням характеристик даних, вимог до точності прогнозування та доступних обчислювальних ресурсів.

Проведене дослідження демонструє, що поєднання методів машинного та глибинного навчання з ретельним аналізом наївної поведінки моделей дозволяє значно підвищити надійність прогнозування серверних навантажень і створює основу для побудови ефективних систем автоматичного управління ресурсами.

Незважаючи на отримані позитивні результати експериментальних досліджень, проведена робота має певні обмеження, які необхідно враховувати при інтерпретації результатів та подальшому розвитку запропонованого підходу до прогнозування серверних навантажень.

Одним із основних обмежень дослідження є використання даних одного джерела — трас Google Cluster Trace 2011. Хоча цей набір даних широко використовується в наукових дослідженнях і залишається актуальним, він відображає характеристики конкретного виробничого середовища та може не повністю репрезентувати поведінку серверних навантажень в інших типах інфраструктур, зокрема у сучасних хмарних або гібридних середовищах.

Другим обмеженням є фокус дослідження на прогнозуванні середнього завантаження центрального процесора (CPU). Хоча CPU є одним із ключових показників серверного навантаження, реальні системи управління ресурсами також залежать від використання оперативної пам’яті, дискових операцій та мережної пропускної здатності. Використання лише одного параметра може обмежувати точність прогнозування комплексного навантаження.

Крім того, методи глибинного навчання, які продемонстрували найвищу точність, потребують значних обчислювальних ресурсів та ретельного

налаштування гіперпараметрів. Це може ускладнювати їх застосування в реальних системах з обмеженими ресурсами або ж у сценаріях, де потрібна швидка адаптація моделей у режимі реального часу.

Також варто зазначити, що в рамках дослідження не розглядалися аспекти впливу збоїв, аномалій та нештатних ситуацій у роботі серверів на якість прогнозування. Такі події можуть суттєво змінювати динаміку навантаження та потребують окремих методів обробки.

Перспективним напрямом подальших досліджень є розширення набору вхідних ознак шляхом включення багатовимірних показників навантаження, таких як використання оперативної пам’яті, мережного трафіку та дискових операцій. Це дозволить створювати більш комплексні та точні моделі прогнозування.

Іншим важливим напрямом є застосування методів багатокрокового та багатовимірного прогнозування, що дасть змогу більш ефективно підтримувати системи проактивного управління ресурсами серверів та дата-центрів.

Доцільним також є дослідження гібридних моделей, які поєднують статистичні підходи з методами машинного та глибинного навчання. Такі моделі можуть забезпечувати баланс між обчислювальною ефективністю та точністю прогнозування.

Окрему увагу слід приділити оптимізації процесу навчання моделей, зокрема шляхом автоматичного налаштування гіперпараметрів, використання методів перенесення навчання (transfer learning) та зменшення обчислювальної складності моделей глибинного навчання.

Крім того, перспективним є впровадження механізмів виявлення аномалій та адаптивного навчання, що дозволить системі прогнозування автоматично реагувати на зміну характеристик навантаження у реальному часі.

Запропонований у роботі підхід демонструє практичну придатність методів машинного та глибинного навчання для прогнозування серверних навантажень і може бути використаний як основа для побудови більш масштабних та адаптивних систем управління ресурсами.

ВИСНОВКИ

У магістерській роботі розглянуто задачу розробки та дослідження системи прогнозування серверних навантажень у хмарних обчислювальних середовищах на основі аналізу часових рядів. Актуальність дослідження зумовлена зростаючою складністю сучасних дата-центрів та необхідністю ефективного управління обчислювальними ресурсами з метою підвищення продуктивності та зниження витрат.

У першому розділі роботи проаналізовано теоретичні та наукові засади прогнозування серверних навантажень. Розглянуто еволюцію серверних і хмарних інфраструктур, основні характеристики серверного навантаження, а також обґрунтовано доцільність використання часових рядів як математичної моделі для опису динаміки використання ресурсів. Проведений аналіз сучасних наукових досліджень підтвердив, що прогнозування навантажень є ключовим елементом проактивного управління ресурсами в хмарних системах.

У другому розділі систематизовано та проаналізовано методи прогнозування часових рядів, зокрема статистичні методи, методи машинного навчання та методи глибинного навчання. Показано, що статистичні методи є простими та інтерпретованими, однак мають обмежену ефективність у складних нестаціонарних умовах. Методи машинного навчання демонструють більшу гнучкість, проте потребують якісної підготовки даних. Методи глибинного навчання, зокрема LSTM та Transformer-архітектури, мають найвищий потенціал для моделювання складних часових залежностей, але характеризуються підвищеною обчислювальною складністю. Окрему увагу приділено проблемі наївної поведінки моделей та вибору метрик оцінювання якості прогнозування.

У третьому розділі розроблено методологію та архітектуру системи прогнозування серверних навантажень, а також спроєктовано систему експериментального дослідження. Обґрунтовано вибір набору даних Google Cluster Trace 2011, виконано попередню обробку даних, сформовано навчальні та

тестові вибірки. Реалізовано статистичні, машинні та глибинні моделі прогнозування, а також визначено процедуру проведення експериментів і критерії оцінювання результатів.

У четвертому розділі проведено експериментальні дослідження та аналіз отриманих результатів. Виконано порівняльний аналіз точності прогнозування для різних моделей за метриками MSE, MAE, R^2 та Naïve Ratio. Результати експериментів показали, що моделі глибинного навчання, зокрема LSTM та Transformer, забезпечують вищу точність прогнозування порівняно зі статистичними та класичними методами машинного навчання, особливо при багатокроковому прогнозуванні. Разом з тим було виявлено, що навіть складні моделі можуть демонструвати наївну поведінку, що підкреслює необхідність комплексного оцінювання прогнозів.

У ході дослідження підтверджено, що використання спеціалізованих метрик дозволяє більш глибоко аналізувати поведінку моделей та уникати хибних висновків щодо їх ефективності. Отримані результати свідчать про доцільність застосування сучасних методів глибинного навчання для задач прогнозування серверних навантажень у хмарних середовищах.

Таким чином, у роботі досягнуто поставленої мети та вирішено всі визначені завдання. Отримані результати можуть бути використані для подальших наукових досліджень, а також при розробці та впровадженні систем прогнозування і проактивного управління ресурсами у хмарних дата-центрах.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Armbrust M., Fox A., Griffith R. et al. A view of cloud computing. Communications of the ACM. 2010. Vol. 53, No. 4. P. 50–58.

2. Buyya R., Broberg J., Goscinski A. Cloud Computing: Principles and Paradigms. Hoboken: Wiley, 2011. 664 p.

3. Chen X., Wang Y., Li X. L-PAW: A deep learning-based prediction algorithm for cloud workloads. Future Generation Computer Systems. 2019. Vol. 98. P. 42–54.

4. Smyl S. A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting. International Journal of Forecasting. 2020. Vol. 36, No. 1. P. 75–85.

5. Gao Y., Guan H., Qi Z. et al. A workload prediction approach for dynamic resource allocation. Journal of Systems Architecture. 2018. Vol. 89. P. 62–72.

6. Song W., Wang H., Liu J. Multi-step ahead cloud workload prediction using LSTM networks. IEEE Access. 2020. Vol. 8. P. 173548–173558.

7. Xu J., Rao J., Bu X. et al. Efficient workload forecasting with deep neural networks. Proceedings of IEEE INFOCOM. 2018. P. 1–9.

8. Lalitha Devi K., Venkatesan R. Hybrid ARIMA–ANN model for cloud workload prediction. Cluster Computing. 2019. Vol. 22. P. 1–12.

9. Makridakis S., Spiliotis E., Assimakopoulos V. The M4 competition: Results, findings, conclusion. International Journal of Forecasting. 2020. Vol. 36, No. 1. P. 54–74.

10. Yang J., Liu Y., Chen M. Cloud host load prediction using evolutionary algorithms. Journal of Supercomputing. 2013. Vol. 65. P. 160–180.

11. Zhang Q., Chen M., Li L. Workload prediction based on recurrent neural networks. Journal of Cloud Computing. 2017. Vol. 6, No. 1.

12. Gupta P., Dutta K., Das S. Adaptive resource utilization prediction using Bi-LSTM. Future Generation Computer Systems. 2020. Vol. 102. P. 101–113.

13. Bhalaji N. Cloud load estimation with deep logarithmic network. Journal of

Intelligent Systems. 2021. Vol. 30, No. 1. P. 540–556.

14. Al-Sayed M. Workload time series cumulative prediction using neural machine translation. IEEE Access. 2020. Vol. 8. P. 211742–211753.

15. Kaim R., Sharma S., Jain A. Attention-based Bi-LSTM and CNN ensemble for workload prediction. Journal of Cloud Computing. 2021. Vol. 10, No. 1.

16. Patel P., Shah S., Trivedi M. MAG-D: Multivariate attention-based GRU model for workload forecasting. Applied Soft Computing. 2021. Vol. 102.

17. Kumar A., Singh R., Kaur H. Self-directed workload prediction in cloud data centers. Cluster Computing. 2022. Vol. 25. P. 1905–1920.

18. Morichetta A., Polato M., Bonola M. Forecasting cloud workloads with LSTM and Transformer models. IEEE Transactions on Cloud Computing. 2022.

19. Xang Y., Li H., Wang X. Host load prediction using autoencoder-based echo state networks. Neurocomputing. 2019. Vol. 332. P. 365–374.

20. Jassas M., Qasem M., Abdullah A. Failure prediction in cloud computing using decision trees. Journal of Network and Computer Applications. 2018. Vol. 102. P. 59–69.

21. Asmawi A., Osman N., Omar K. Predictive models for job and task failure in cloud systems. IEEE Access. 2021. Vol. 9. P. 13401–13414.

22. Kalogirou C., Stamatakis K., Papadopoulos G. Adaptive VM scheduling for energy-efficient cloud computing. Future Generation Computer Systems. 2020. Vol. 108. P. 1009–1023.

23. Ruan Y., Chen H., Zhang Z. Cloud workload turning points prediction via deep learning. Information Sciences. 2021. Vol. 570. P. 1–17.

24. Berndt D., Clifford J. Using dynamic time warping to find patterns in time series. Proceedings of KDD Workshop. 1994. P. 359–370.

25. Rokach L., Maimon O. Data Mining with Decision Trees: Theory and Applications. Singapore: World Scientific, 2015. 320 p.

26. Hyndman R., Athanasopoulos G. Forecasting: Principles and Practice. 3rd ed. Melbourne: OTexts, 2021.

27. Box G., Jenkins G., Reinsel G., Ljung G. Time Series Analysis: Forecasting

and Control. 5th ed. Hoboken: Wiley, 2015. 712 p.

28. Goodfellow I., Bengio Y., Courville A. Deep Learning. Cambridge: MIT Press, 2016. 800 p.

29. Vaswani A., Shazeer N., Parmar N. et al. Attention is all you need. Advances in Neural Information Processing Systems. 2017. P. 5998–6008.

30. Google. Google Cluster Data 2011: Technical Documentation. URL: https://github.com/google/cluster-data

Додаток А

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО–НАУКОВИЙ ІНСТИТУТ
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
Кафедра комп'ютерної інженерії



КВАЛІФІКАЦІЙНА РОБОТА
на здобуття освітнього ступеня магістра

НА ТЕМУ: «Система для прогнозування навантаження серверів з використанням машинного навчання»

Виконав студент групи КСДМ-61
Павло ЛИСЮК
Керівник дипломної роботи:
Сергій КОРОТКОВ
доктор філософії, доцент

Київ – 2025

МЕТА, ОБ’ЄКТА ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

Мета роботи - розробка та дослідження системи прогнозування навантаження серверів з використанням методів машинного та глибинного навчання, а також оцінювання їхньої ефективності, стійкості до перенавчання та здатності до узагальнення

Об’єкт дослідження - процес прогнозування навантаження серверів у хмарних та серверних обчислювальних системах

Предмет дослідження - методи та моделі машинного і глибинного навчання для прогнозування серверних навантажень

Актуальність роботи

Сучасні інформаційні системи та хмарні сервіси працюють в умовах постійно змінного навантаження, яке важко передбачити заздалегідь. Через це сервери можуть бути або перевантажені, що призводить до збоїв і затримок у роботі сервісів, або недовантажені, що означає неефективне використання ресурсів. Тому прогнозування навантаження серверів є важливим завданням, яке дозволяє заздалегідь підготувати систему до змін і оптимально розподіляти ресурси. Використання методів машинного та глибинного навчання дає змогу точніше аналізувати дані та підвищувати стабільність роботи серверних систем.

Наукова новизна роботи

Наукова новизна роботи полягає у комплексному порівняльному аналізі статистичних, машинних та глибинних моделей прогнозування серверних навантажень з урахуванням їхньої здатності до узагальнення. Запропоновано використання метрики Naïve Ratio для виявлення наївної поведінки моделей, а також досліджено вплив горизонту прогнозування та кількості вхідних даних на якість прогнозів. Отримані результати розширюють уявлення про ефективність різних підходів у задачі прогнозування навантажень.

Методи прогнозування навантаженя серверів

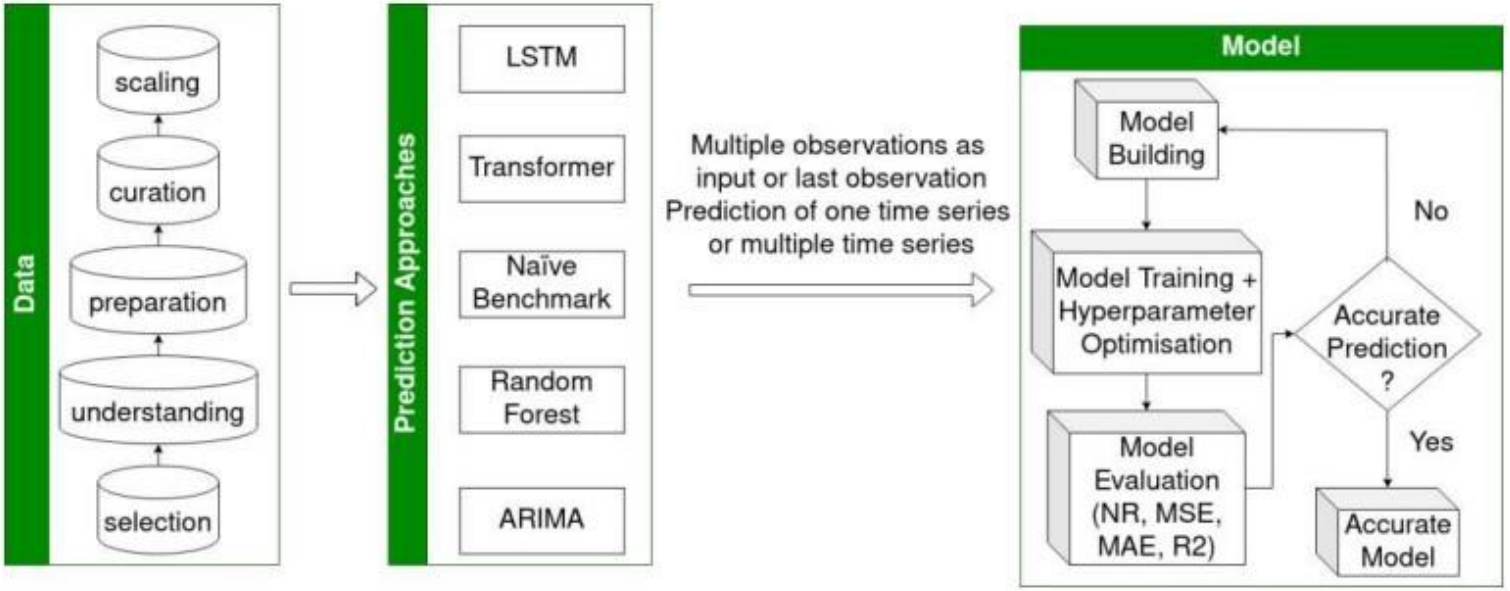


Архітектура системи прогнозування

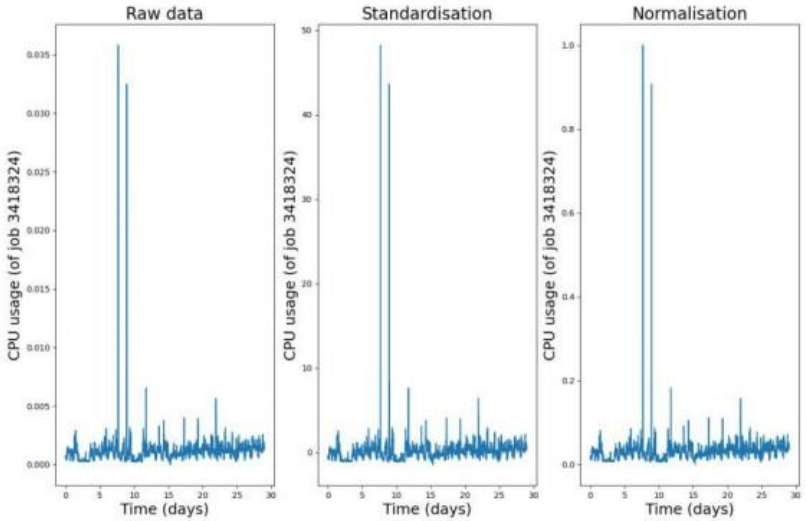


Набір даних Google Cluster Trace

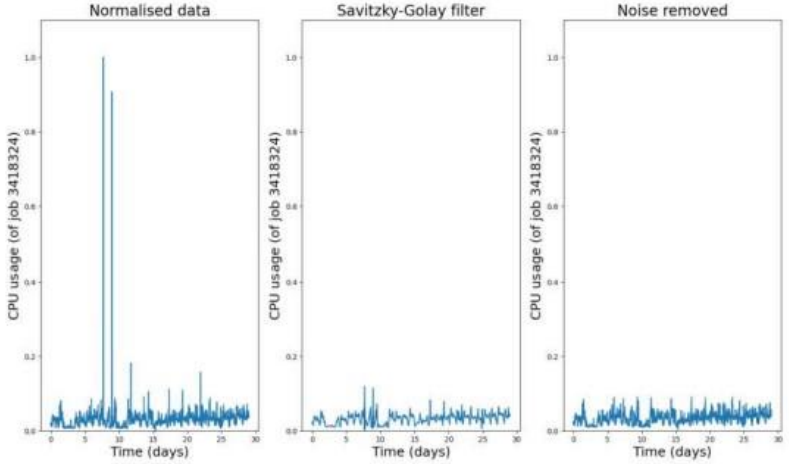
Загальна схема проведення експериментальних досліджень



Попередня обробка даних

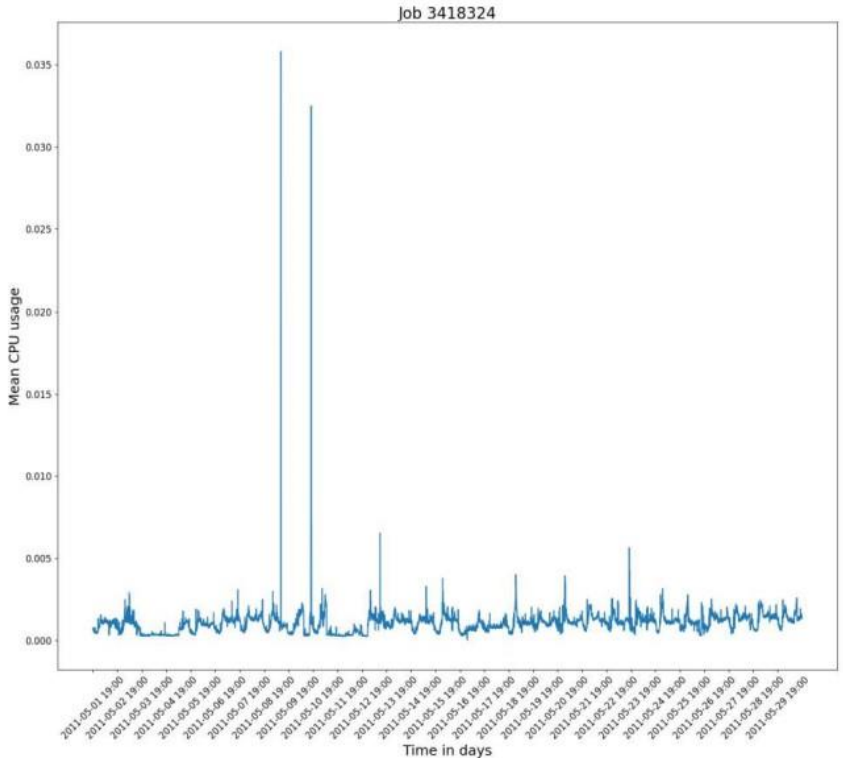


Середнє використання CPU з різними методами масштабування



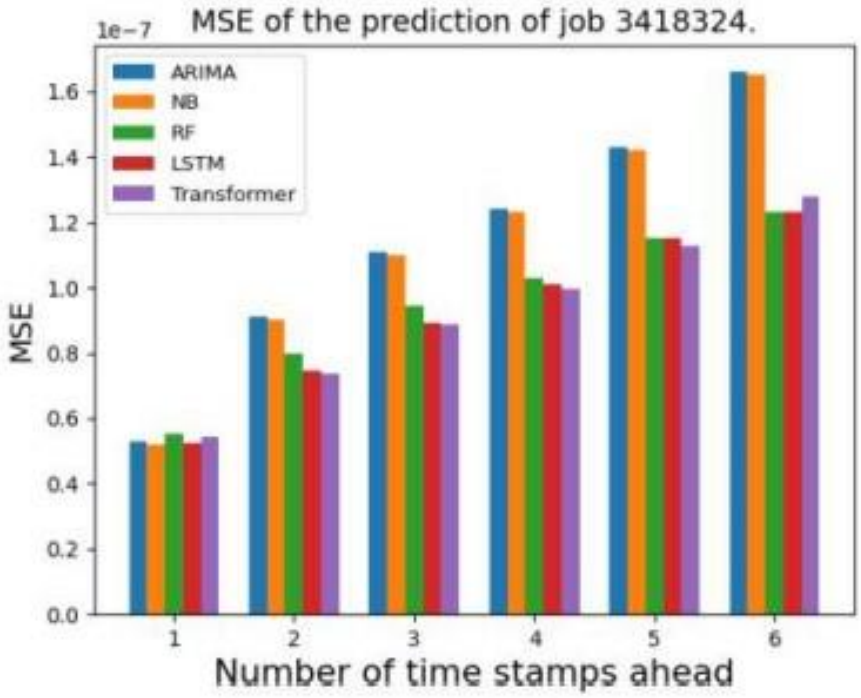
Середнє використання CPU із застосуванням фільтра SavGol та видаленням шуму

Часовий ряд середнього завантаження CPU (Google Cluster Trace)



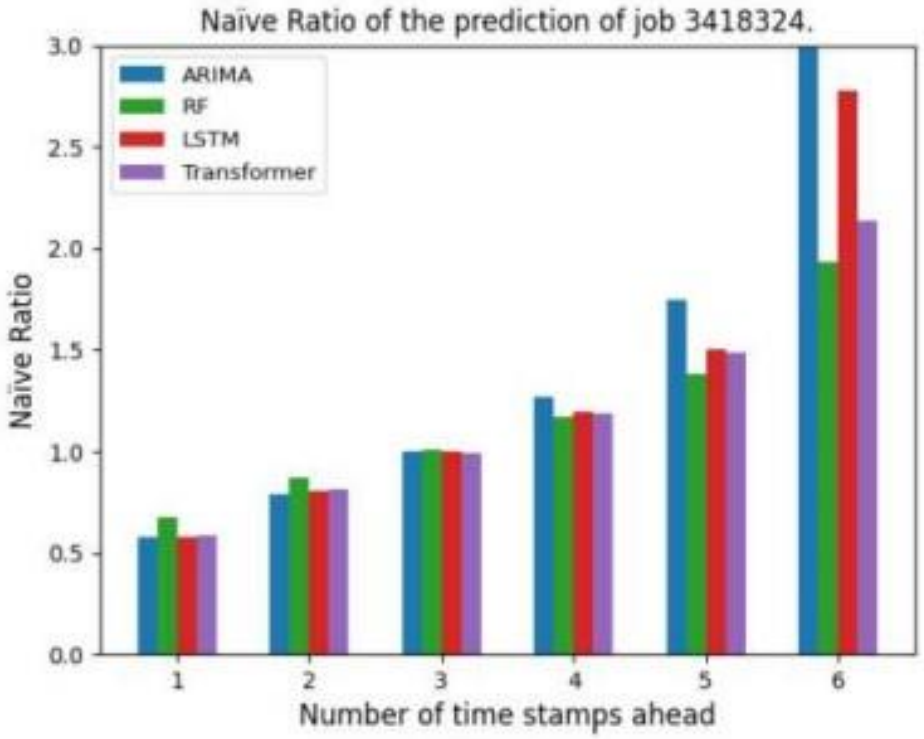
Часовий ряд середнього завантаження CPU для вибраного робочого навантаження

Порівняльний аналіз точності моделей за метрикою MSE



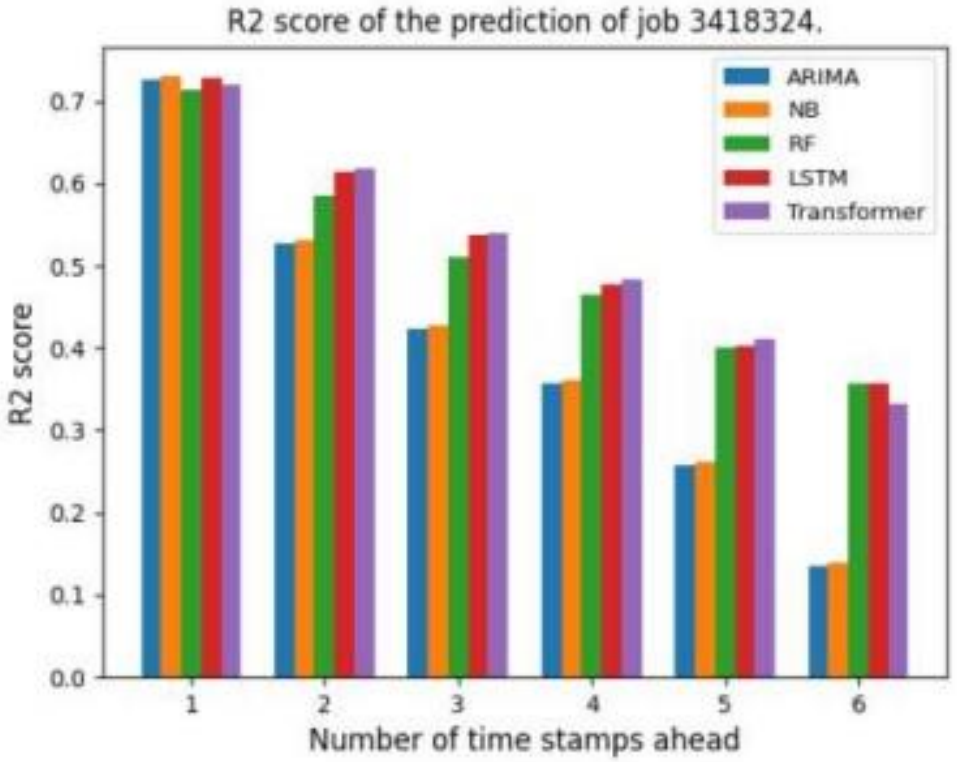
Порівняння MSE для тестів з одним завданням

Аналіз наївної поведінки моделей (Naïve Ratio)



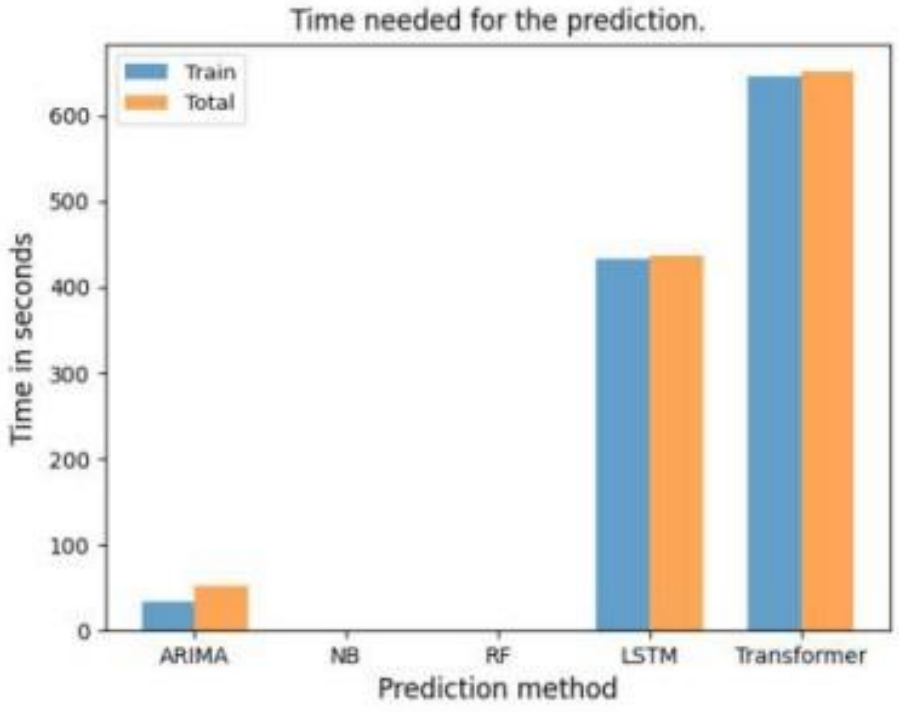
Порівняння показника NR для тестів з одним завданням

Порівняльний аналіз моделей за показником R^2



Порівняння показника R^2 для тестів з одним завданням

Порівняння моделей за часом виконання (runtime)



Порівняння часу виконання (runtime) для тестів з одним завданням

Висновки

У роботі розроблено та досліджено систему прогнозування навантаження серверів із використанням статистичних методів, методів машинного та глибинного навчання. Проведено експериментальні дослідження на основі набору даних Google Cluster Trace 2011 та виконано порівняльний аналіз моделей за показниками MSE, R^2 та Naïve Ratio. Результати експериментів показали, що моделі глибинного навчання, зокрема LSTM та Transformer, забезпечують вищу точність прогнозування порівняно з класичними підходами, однак потребують більших обчислювальних ресурсів. Отримані результати можуть бути використані для підвищення ефективності роботи серверних систем та оптимізації використання обчислювальних ресурсів.