

БАКАЛАВРСЬКА РОБОТА

БР. ІІ - 95.00.00.000 ІЗ

Група ІІ-22-4

Драбик Юрій

2026

Івано-Франківський національний технічний університет нафти і газу

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення

Драбик Юрій Романович

(прізвище, ім'я, по батькові)

УДК 004.942

(індекс)

БАКАЛАВРСЬКА РОБОТА

Прогнозування витрат на хмарні ресурси за допомогою машинного

навчання

(назва роботи)

Інженерія програмного забезпечення

(назва освітньої програми)

121 - Інженерія програмного забезпечення

(шифр і назва спеціальності)

Робота містить результати власних досліджень, використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело:

Здобувач освітнього ступеня Драбик Ю.Р.

(підпис, ініціали та прізвище здобувача)

Науковий керівник Корнута Володимир Андрійович, к.т.н., доцент

(підпис, прізвище, ім'я, по батькові, науковий ступінь, вчене звання керівника)

Допущено до захисту

Завідувач кафедри

доц. Бандура В.В.

(посада)

(підпис) (дата) (ініціали та прі-

звище)

Івано-Франківський національний технічний університет нафти і газу
Інститут, факультет інформаційних технологій
Кафедра інженерії програмного забезпечення
Ступінь вищої освіти бакалавр
Спеціальність 121 – Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ:

Зав. кафедрою

ІІЗ

доцент.

В.В. Бандура

“ ” 2026 р.

ЗАВДАННЯ
НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТОВІ

Драбик Юрій Романович

(прізвище, ім'я, по-батькові)

1. Тема проекту (роботи) "Прогнозування витрат на хмарні ресурси за допомогою машинного навчання"

керівник проекту (роботи) Корнута Володимир Андрійович, к.т.н., доцент

затверджені наказом вищого навчального закладу від “ 21 ” квітня 2026 р. № 179/7

2. Строк подання студентом проекту (роботи) 10 червня 2026 р.

3. Вихідні дані до проекту (роботи) Результати і матеріали отримані під час проходження переліченої практики

4. Зміст розрахунково - пояснювальної записки(перелік питань, які потрібно розробити)

1. Аналіз вимог до програмного забезпечення

2. Аналіз вимог і постановка задачі

3. Проектування системи

4. Реалізація проекту

5. Тестування та впровадження

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

1 Концептуальна основа управління ресурсами в хмарному середовищі (рис. 1.1, ст. 22)

2 Категоризація методів управління ресурсами (рис. 1.2, ст. 24)

4. Блок-схема методології дослідження (рис. 2.1, ст. 27)

4 Архітектура забезпечення ресурсами (рис. 2.2, ст. 29)

5 Структура управління еластичними ресурсами (рис. 3.2, ст. 55)

6. Консультанти розділів проекту (роботи)

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

2. Дата видачі завдання 2026 р.

Керівник

_____ (підпис)

Завдання прийняв до виконання

_____ (підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту	Примітка
1	Визначення та обґрунтування теми роботи	17.01.2026	виконано
2	Огляд існуючих концепцій, рішень та сервісів в даній області	21.02.2026	виконано
3	Побудова моделі або алгоритму власного рішення	01.03.2026	виконано
4	Документування реалізації власного оригінального рішення вибраними засобами	21.04.2026	виконано
5	Оформлення пояснювальної записки бакалаврської роботи	02.05.2026	виконано

Студент

_____ Драбик Ю.Р.
(підпис)

Керівник роботи

_____ Корнута В.А.
(підпис)

АНОТАЦІЯ

Бакалаврська робота містить 75 сторінок, 23 рисунки, 3 таблиці, список використаних джерел із 40 найменувань.

Метою роботи є дослідження методів прогнозування витрат на хмарні ресурси за допомогою технологій машинного навчання та аналіз підходів до підвищення ефективності управління хмарною інфраструктурою.

Об'єкт дослідження: процеси управління та прогнозування використання хмарних ресурсів у сучасних інформаційних системах.

Предмет дослідження: методи машинного навчання, прогнозування навантаження та інструменти аналізу витрат у хмарних середовищах.

Результати дослідження: проведено аналіз сучасних підходів до прогнозування витрат на хмарні ресурси, досліджено методи машинного навчання для оцінки навантаження та використання інфраструктури.

У першому розділі розглянуто теоретичні основи хмарних обчислень, особливості моделей надання хмарних сервісів, підходи до управління ресурсами та сучасні дослідження у сфері прогнозування витрат.

У другому розділі досліджено методи та інструменти прогнозування витрат, зокрема алгоритми машинного навчання, методології аналізу даних і моделі оцінювання споживання ресурсів у хмарному середовищі.

У третьому розділі розглянуто реалізацію системи прогнозування робочого навантаження, механізми управління еластичними ресурсами, а також оцінювання ефективності запропонованого підходу та аналіз результатів.

Висновок: використання методів машинного навчання для прогнозування витрат на хмарні ресурси дозволяє підвищити ефективність використання інфраструктури.

КЛЮЧОВІ СЛОВА: ХМАРНІ ОБЧИСЛЕННЯ, МАШИННЕ НАВЧАННЯ, ПРОГНОЗУВАННЯ ВИТРАТ, ХМАРНІ РЕСУРСИ, ЕЛАСТИЧНІСТЬ, АНАЛІЗ ДАНИХ, AWS, AZURE, GOOGLE CLOUD, ОПТИМІЗАЦІЯ РЕСУРСІВ.

ANNOTATION

The thesis consists of 75 pages, 23 figures, 3 tables, and a reference list containing 40 sources.

The aim of the work is to study methods for forecasting cloud resource costs using machine learning technologies and to analyze approaches to improving cloud infrastructure management efficiency.

Research object: processes of managing and forecasting cloud resource usage in modern information systems.

Research subject: machine learning methods, workload prediction models, and tools for cloud cost analysis in cloud environments.

Research results: modern approaches to forecasting cloud resource costs were analyzed, machine learning methods for workload and infrastructure usage estimation were studied, and mechanisms for managing elastic resources were reviewed.

The first section describes the theoretical foundations of cloud computing, the features of cloud service models, approaches to resource management, and current research in the field of cloud cost forecasting.

The second section analyzes methods and tools for cost forecasting, including machine learning algorithms, data analysis methodologies, and models for evaluating resource consumption in cloud environments.

The third section covers the implementation of a workload forecasting system, mechanisms for managing elastic resources, as well as evaluation of the effectiveness of the proposed approach and analysis of the obtained results.

Conclusion: the use of machine learning methods for forecasting cloud resource costs improves infrastructure utilization efficiency, reduces operational expenses, and enables more accurate workload planning in cloud environments.

KEY WORDS: CLOUD COMPUTING, MACHINE LEARNING, COST FORECASTING, CLOUD RESOURCES, ELASTICITY, DATA ANALYSIS, AWS, AZURE, GOOGLE CLOUD, RESOURCE OPTIMIZATION.

ЗМІСТ

РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ ПРОГНОЗУВАННЯ ВИТРАТ НА ХМАРНІ РЕСУРСИ	10
1.1 Основи хмарних обчислень та управління ресурсами	10
1.2 Аналіз існуючих досліджень і підходів до прогнозування витрат	15
1.3 Концепції та моделі управління хмарними ресурсами	20
1.4 Висновки по розділу	25
РОЗДІЛ 2. МЕТОДИ ТА ІНСТРУМЕНТИ ПРОГНОЗУВАННЯ ВИТРАТ ЗА ДОПОМОГОЮ МАШИННОГО НАВЧАННЯ	26
2.1 Методологія побудови систем прогнозування	26
2.2 Моделі та алгоритми машинного навчання для аналізу витрат	35
2.3 Проектування та специфікація системи прогнозування	41
2.4 Висновки по розділу	48
РОЗДІЛ 3. РОЗРОБКА СИСТЕМИ ТА ОЦІНКА ЕФЕКТИВНОСТІ	49
3.1 Прогнозування робочого навантаження хмарної інфраструктури	49
3.2 Управління еластичними ресурсами та оптимізація витрат	55
3.3 Оцінка ефективності та аналіз результатів прогнозування	65
3.4 Висновки по розділу	79
ВИСНОВКИ	80
СПИСОК ПОСИЛАНЬ НА ДЖЕРЕЛА	82

					БР.ІІ –95.00.00.000 ІІЗ			
Змн.	Арк.	№ докум.	Підпис	Дата	Прогнозування витрат на хмарні ресурси за допомогою машинного навчання Пояснювальна записка	Літ.	Арк.	Акрушіє
Розроб.		Драбик ю.Р.					7	
Перевір.		Корнута В.А.						
Реценз.		Саманів Л.В.						
Н. Контр.		Піх М.М.						
Затверд.		Бандура В. В.				ІФНТУНГ ІІ-22-4		

ВСТУП

У сучасному світі хмарні технології стали невід’ємною складовою розвитку інформаційних систем, забезпечуючи масштабованість, гнучкість та високу доступність обчислювальних ресурсів. Зростання обсягів даних, активне використання сервісів штучного інтелекту, веб-додатків та корпоративних платформ спричиняє значне збільшення навантаження на хмарну інфраструктуру. У зв’язку з цим особливої актуальності набуває проблема ефективного управління витратами на хмарні ресурси, оскільки неконтрольоване використання обчислювальних потужностей може призводити до суттєвих фінансових втрат для організацій. Сучасні виклики, пов’язані з динамічністю робочих навантажень, необхідністю швидкого масштабування сервісів та оптимізацією використання ресурсів, вказують на потребу у впровадженні інтелектуальних методів прогнозування та аналізу витрат у хмарних середовищах.

Актуальність теми зумовлена необхідністю створення ефективних підходів до прогнозування витрат на хмарні ресурси із застосуванням технологій машинного навчання. Існуючі методи моніторингу та керування витратами часто базуються на статичних моделях або ручному аналізі, що не дозволяє оперативно реагувати на зміни навантаження та особливості використання ресурсів у реальному часі. Використання алгоритмів машинного навчання дає можливість аналізувати великі обсяги даних, виявляти закономірності використання інфраструктури та прогнозувати майбутні витрати з високою точністю. У контексті активного розвитку хмарних платформ, таких як AWS, Microsoft Azure та Google Cloud Platform, розробка систем прогнозування витрат є важливим кроком для підвищення економічної ефективності та оптимізації використання ресурсів.

Метою роботи є дослідження та розробка підходів до прогнозування витрат на хмарні ресурси за допомогою методів машинного навчання для підвищення ефективності управління хмарною інфраструктурою та оптимізації використання ресурсів.

Завданнями дослідження є аналіз предметної області та існуючих рішень у сфері прогнозування витрат на хмарні ресурси, дослідження моделей хмарних

					БР.ІП - 95.00.00.000 ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

обчислень та методів управління ресурсами, визначення функціональних і нефункціональних вимог до системи прогнозування, аналіз алгоритмів машинного навчання для прогнозування навантаження та витрат, проектування архітектури системи, реалізація механізмів аналізу та прогнозування даних, тестування ефективності моделі та оцінка отриманих результатів.

Об'єктом дослідження є процеси використання та управління хмарними ресурсами в інформаційних системах, що включають аналіз навантаження, масштабування інфраструктури та контроль фінансових витрат.

Предметом дослідження є методи, моделі та інструменти прогнозування витрат на хмарні ресурси із застосуванням алгоритмів машинного навчання, а також підходи до аналізу та оптимізації використання обчислювальних ресурсів у хмарних середовищах.

Методи дослідження включають аналіз наукових джерел для вивчення теоретичних основ хмарних обчислень і машинного навчання, порівняльний аналіз алгоритмів прогнозування, моделювання процесів використання ресурсів, експериментальний метод для реалізації та тестування моделей прогнозування, а також статистичний аналіз отриманих результатів.

Розроблена система прогнозування передбачатиме можливість аналізу використання хмарних ресурсів, прогнозування витрат на основі історичних даних, управління еластичними ресурсами та оцінювання ефективності використання інфраструктури. Особлива увага приділятиметься точності прогнозування, масштабованості системи, автоматизації процесів аналізу даних та оптимізації фінансових витрат.

Бакалаврська робота містить 75 сторінок, 23 рисунки, 3 таблиці, 3 розділи, список використаних джерел із 40 найменувань.

					БР.ІІІ - 95.00.00.000 ПЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1 ТЕОРЕТИЧНІ ОСНОВИ ПРОГНОЗУВАННЯ ВИТРАТ НА ХМАРНІ РЕСУРСИ

1.1 Основи хмарних обчислень та управління ресурсами

Хмарні обчислення – це новий та перспективний метод, який поєднує групу сучасних технологій для управління бізнесом різними способами. Популярність хмарних обчислень зростає, оскільки вони мають перевагу в тому, що забезпечують клієнтам гнучкість та легкість розгортання своїх програм, що робить процес отримання ресурсів вражаюче простим. Крім того, користувачі платять лише на вимогу відповідно до фактичного використання ресурсів (плата за використання) [1]. Компанії можуть використовувати хмарні обчислення для різних цілей, починаючи від виконання пакетів завдань і закінчуючи у сферах освіти, державних досліджень та бізнесу, що робить його важливим інструментом для багатьох користувачів, які залежать від хмарних додатків у своєму особистому житті та повсякденній діяльності [2].

У найпростішому вираженні хмарні обчислення стосуються операції зберігання, обробки та аналізу програм і даних, що пропонуються як послуга клієнтам через Інтернет [3]. Еластичність та динамізм – це два центральні поняття в хмарних обчисленнях. У цьому випадку обчислювальні ресурси, такі як процесори, пам'ять та мережі, зазвичай надаються у формі віртуальних машин (ВМ).

Як згадувалося раніше, еластичність є основною характеристикою хмарних обчислень. Еластичність пов'язана зі здатністю системи адаптуватися до коливань робочого навантаження та запитів на ресурси. У хмарних обчисленнях ця властивість еластичності дозволяє програмам динамічно отримувати та вивільняти ресурси. Однак адаптація до коливань попиту та розподіл правильної кількості ресурсів, як-от у випадку прогнозування використання процесора, є непростю [4]. Варто зазначити, що це залишається складним завданням, і лише обмежена кількість досліджень у літературі розглядає цю проблему.

Для використання властивості еластичності хмари, а також для динаміч-

					БР.ІІІ - 95.00.00.000 ПЗ	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дата		

ного управління та оптимального використання обчислювальних ресурсів використовується метод автономного забезпечення ресурсами. Коли відбуваються зміни в навантаженні програми, доступні ресурси необхідно переналаштувати для забезпечення необхідних вимог та покращення продуктивності, одночасно забезпечуючи вимоги до якості обслуговування (QoS), уникаючи порушень угод про рівень обслуговування (SLA) та зменшуючи вартість оренди хмарних ресурсів у постачальників програмного забезпечення як послуги (SaaS) [5].

У цій роботі ми пропонуємо гібридний метод забезпечення ресурсами для хмарних обчислювальних програм, який базується на поєднанні концепції автономних обчислень та глибокого навчання. Тут ми зосереджуємося на прогнозуванні використання процесора віртуальними машинами в хмарних центрах обробки даних для майбутніх потреб робочого навантаження. У нашій реалізації було використано еталонну модель IBM для автономних обчислень, цикл керування MAPE [6]. Автономна модель має чотири компоненти: монітор (М), аналізатор (А), планувальник (Р) та виконавець (Е). Компонент монітора використовується для збору інформаційних метрик з датчиків щодо ресурсів та клієнтів. Метрики, зібрані на етапі моніторингу, обробляються компонентом аналізатора для прогнозування необхідної кількості запитів на хмарні сервіси в наступний період часу. Компонент планування є основним компонентом фреймворку; він визначає оптимальні дії, які слід реалізувати (наприклад, визначення кількості віртуальних машин, які потрібно додати або видалити для різних хмарних сервісів). Нарешті, компонент виконавця відповідає за виконання дій, таких як масштабування, які визначаються компонентом планувальника, для досягнення ефективного використання ресурсів.

Забезпечення Springer та досягнення компромісу між дотриманням угоди про рівень обслуговування та мінімізацією витрат [7] .

У нашому дослідженні ми пропонуємо ефективну модель глибокого навчання, засновану на дифузійній згортковій рекурентній нейронній мережі (DCRNN) [8]. Модель DCRNN базується на архітектурі нейронної мережі типу «кодер-декодер». Вона реалізує прогнозування за допомогою навчання

					БР.ІІІ - 95.00.00.000 ПЗ	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

«послідовність-до-послідовність». Архітектура «кодер-декодер» побудована з використанням простого варіанту нейронної мережі, відомого як стробовані рекурентні одиниці (GRU). У GRU для створення комірки DCRNN використовується операція дифузійної згортки замість множення матриць .

Ми застосовуємо DCRNN на етапі аналізу для прогнозування майбутнього попиту, таким чином враховуючи коливання робочого навантаження в наступному інтервалі. Наша робота має два основні внески:

- Ми перетворили дані процесора з робочого навантаження PlanetLab у форму мережевого графа.
- Ми використали підхід до прогнозування робочого навантаження на основі DCRNN, щоб покращити точність прогнозування та мінімізувати похибку між прогнозованим та фактичним робочим навантаженням; ми також включили обчислення похибки прогнозу, яка є різницею між фактичним та прогнозованим робочим навантаженням.

Ця модель може аналізувати поточне робоче навантаження та прогнозувати майбутній попит, який залежить від змін робочого навантаження, як автоматично, так і ефективно без втручання людини. Використовуючи цей метод, хмарні сервіси можуть надаватися за низькою ціною з оптимальним використанням процесора.

Нещодавно підходи до глибокого навчання перевершили інші сучасні методи в багатьох галузях завдяки своїм ієрархічним особливостям та представленням для навчання [9]. Наразі розроблено різні методи глибокого навчання, включаючи стекові автокодери, мережі глибоких довір (DBN) та згорткові нейронні мережі (CNN). Для навчання параметрів вилучення багаторівневих ознак у глибокому навчанні використовуються методи з учителем та без учителя [10]. Глибоке навчання є частиною машинного навчання (ML); це аспект сучасного покоління нейронних мереж (NN), і воно виникло як концепція з досліджень штучних нейронних мереж (ANN) [11].

В останні роки було розроблено різноманітні методи глибокого навчання для врахування змін робочого навантаження [12]. Однак точне прогнозування є

					БР.ІІІ - 95.00.00.000 ПЗ	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дата		

складним завданням, особливо за наявності непослідовних та нелінійних робочих навантажень [13]. Більше того, системи хмарних обчислень складаються з багатьох віртуальних машин, що збільшує складність проблеми прогнозування через кореляції між великими даними робочого навантаження в цих віртуальних машинах. Моделям прогнозування було важко інтегрувати точне прогнозування в режимі реального часу, оскільки це ще більше ускладнює прогнозування системи. Тому в цій роботі використовується підхід до прогнозування робочого навантаження DCRNN для підвищення точності прогнозування, який постачальники хмарних послуг можуть використовувати для покращення послуг. DCRNN застосовувалися в попередніх дослідженнях, включаючи прогнозування трафіку для дорожніх мереж, оцінку часу поїздки громадськими автобусами [14] та прогнозування навантаження трафіку на каналах телекомунікаційної мережі. Для всіх базових показників DCRNN продемонстрували чудову продуктивність.

Ми запропонували використовувати DCRNN як новий метод для обробки як просторової, так і часової поведінки розподілених віртуальних машин (ВМ). DCRNN забезпечує кращі результати прогнозування, оскільки він може краще обробляти довгострокову поведінку порівняно з іншими методами глибокого навчання, завдяки внутрішній природі рекурентних мереж, що є частиною DCRNN. Наскільки нам відомо, ми першими в цій галузі використовуємо DCRNN для прогнозування майбутнього використання процесора з метою правильного розподілу ресурсів для програм на основі їхніх потреб.

Хмарні обчислення надають настроювану платформу, яка дозволяє програмам отримувати відповідні ресурси перед виконанням необхідних програм на віртуальних машинах (ВМ) [15]. Постачальники хмарних послуг зазвичай використовують модель ціноутворення «оплата за використання», яка може призвести до зниження витрат та гнучкості для користувачів хмарних послуг. Широкий спектр розробок у сфері хмарних обчислень призвів до значного збільшення кількості користувачів хмарних послуг та розробки програм у хмарних середовищах для доступу до різних сервісів хмарних обчислень [16]. Кілька наукових застосувань використовують сервіси хмарних обчислень, що

					БР.ІІІ - 95.00.00.000 ПЗ	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дата		

призводить до різноманітного використання хмарних ресурсів [3, 4] . Отже, необхідно ефективно керувати ресурсами, щоб враховувати коливання попиту користувачів. Ефективне управління ресурсами в середовищі хмарних обчислень може сприяти оптимізації використання ресурсів, зниженню витрат та підвищенню продуктивності. Для досягнення ефективного управління ресурсами використовується прогнозування використання ресурсів.

Використання методів машинного навчання в дослідженнях отримало більше уваги в останні роки. Нейронні мережі є одним з найбільш широко використовуваних методів машинного навчання. Ключовою перевагою використання нейронних мереж є здатність приймати точні рішення. Багато застосувань, включаючи прогнозування електричного потоку, прогнозування річкового стоку тощо, використовують нейронні мережі [17]. З огляду на ці причини, у цій роботі нейронні мережі використовуються для прогнозування використання хмарних ресурсів. Ключовим питанням у впровадженні нейронних мереж є навчання вагових коефіцієнтів мережі. Навчання вагових коефіцієнтів мережі є складною задачею оптимізації. Для таких задач широко використовуються теплові та еволюційні алгоритми. Використання цих методів є кращим за традиційні математичні методи [5]. Для досягнення цієї мети запропонована модель використовує гібридний генетичний алгоритм (GA) та оптимізацію рою частинок (PSO) (GA-PSO), які використовують сильні сторони обох алгоритмів. Гібридні версії цих алгоритмів використовувалися в різних областях, і були досягнуті вражаючі результати. Застосування включають синтез діаграм спрямованості антенних решіток [6], правила асоціації видобутку ресурсів [7], прогнозування попиту на електроенергію [8], планування ресурсів у хмарних обчисленнях [9] та планування процесів [10]. Автори в [11] за допомогою моделювання довели, що гібридна версія долає недоліки окремих алгоритмів.

Прогнозування використання ресурсів широко досліджується, і з цього приводу доступна велика кількість літератури [18]. На моделі прогнозування ресурсів впливають деякі параметри, такі як точність, часова та пам'ятна складність моделі, обробка кількох ресурсів тощо. Точність прогнозування використання

					БР.ІІІ - 95.00.00.000 ПЗ	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		

ресурсів є проблемою через наявність кількох типів метрик, наприклад, використання процесора та пам'яті, дискового вводу/виводу, пропускну здатності мережі тощо. Можуть існувати деякі неявні зв'язки, такі як зв'язок між використанням процесора та пам'яті, а також зв'язок між дисковим вводом/виводом та пам'яттю. Важко знайти та передбачити зв'язок для кожного типу ресурсу. Таким чином, результати прогнозування не будуть придатними для практичного використання. Щоб вирішити цю проблему, автомасштабування на основі прогнозування хмарних ресурсів повинно мати можливість одночасно обробляти різні показники, щоб приймати точні рішення щодо масштабування.

1.2 Аналіз існуючих досліджень і підходів до прогнозування витрат

Прогнозування використання ресурсів стає дедалі популярнішим завдяки останнім досягненням у сфері управління ресурсами. У цьому підрозділі зібрано різні методи прогнозування, засновані на методологіях глибокого та машинного навчання.

Представлено алгоритм прогнозування споживання процесора на основі лінійної регресії для максимізації використання ресурсів хмарних обчислень. Щоб мінімізувати споживання енергії та порушення SLA, він включає міграцію в реальному часі для балансування активних хостів на основі потреб у ресурсах. Щоб уникнути перевантажень та перевести недовантажені хости в режими низького енергоспоживання, історичні дані процесора прогнозують використання хостів та запускають міграцію віртуальних машин. Його ефективність підтверджена дослідженнями відстеження реального робочого навантаження, які показують значне зниження споживання енергії та порушень SLA, а також вказують на кращу надійність та ефективність у хмарних середовищах.

Запропоновано моделі на основі машинного навчання для прогнозування використання ресурсів на рівні завдань та ресурсів, що дозволяє здійснювати розподіл хмарних ресурсів. Оцінки демонструють значне підвищення точності, зі зменшенням помилок прогнозування на 20% та покращенням до 89% за певних

					БР.ІІІ - 95.00.00.000 ПЗ	Арк.
						16
Змн.	Арк.	№ докум.	Підпис	Дата		

обставин. У хмарних обчисленнях цей метод пропонує оптимізований розподіл ресурсів та індивідуальне розподілення ресурсів.

Представлено унікальну систему прогнозування використання ресурсів для хмар IaaS на основі машинного навчання, яка використовує динамічне прогнозування ресурсів. Вона пропонує значні покращення в управлінні та оптимізації інфраструктури IaaS, поєднуючи історичні дані та моніторинг у режимі реального часу для оптимізації розподілу ресурсів, підвищення економічної ефективності та покращення загальної продуктивності IaaS .

Проемонстровано, наскільки важливо точно розподіляти ресурси в хмарних системах, щоб запобігти втратам або погіршенню якості обслуговування. Вони пропонують систему прогнозування робочого навантаження на основі ансамблю, яка створює точні прогностичні моделі за допомогою методів машинного навчання. Проект намагається підвищити точність прогнозування для ефективного використання ресурсів, залучаючи кількох учнів та узагальнення стека.

Досліджено роль машинного навчання (ML) у покращенні планування та розподілу ресурсів у великомасштабному виробництві. У дослідженні розглядаються складності виробничих операцій шляхом використання історичних даних для створення прогнозних моделей. Ці моделі, що охоплюють контрольовані та неконтрольовані алгоритми машинного навчання (ML), оптимізують рішення щодо планування, навчаючись на історичних закономірностях. Вони прогнозують графіки виробництва та ефективно розподіляють ресурси, підкреслюючи їхній потенціал для підвищення точності планування та ефективності використання ресурсів [19]. У дослідженні наголошується на здатності ML оптимізувати та оптимізувати виробничі процеси у великомасштабних виробничих системах, пропонуючи розуміння того, як підходи, засновані на даних, можуть підвищити операційну ефективність у складних виробничих середовищах.

Досліджено планування роботи центрів обробки даних, зосереджуючись на оптимізації використання процесора та виконанні вимог угоди про рівень обслуговування (SLA) за допомогою машинного навчання (ML). Досліджуючи проблеми ефективності використання процесора та виконання SLA, дослідження

					БР.ІІІ - 95.00.00.000 ПЗ	Арк.
						17
Змн.	Арк.	№ докум.	Підпис	Дата		

пропонує гібридний підхід ML, що інтегрує моделі кластеризації та регресії для планування. Використовуючи історичні дані про процесор, прогнознi моделі дозволяють проактивно керувати ресурсами, забезпечуючи ефективне використання процесора та дотримуючись SLA [20]. Дослідження демонструє ефективність планування на основі ML в оптимізації використання процесора, вдосконаленні стратегій розподілу ресурсів та підтримці зобов'язань щодо обслуговування в середовищах центрів обробки даних. Це підкреслює потенціал ML для покращення роботи центрів обробки даних та забезпечення дотримання SLA, одночасно оптимізуючи використання ресурсів.

Запропоновано ґрунтовний огляд застосувань машинного навчання (ML) в управлінні хмарними ресурсами. Охоплюючи еволюцію методів ML, вона досліджує їх різноманітне застосування в оптимізації, розподілі та прогнозному моделюванні в хмарних середовищах. Починаючи від традиційних алгоритмів до передового глибокого навчання, стаття демонструє ефективність ML у вирішенні проблем розподілу ресурсів. Вона також окреслює майбутні напрямки досліджень, підкреслюючи інтеграцію нових парадигм ML, таких як федеративне навчання та навчання з підкріпленням. Цей комплексний огляд виступає дорожньою картою, що прокладає шлях для інновацій в управлінні ресурсами, орієнтованому на ML, з метою підвищення адаптивності та ефективності інфраструктур хмарних обчислень.

Запропоновано новий підхід у хмарних обчисленнях, використовуючи алгоритм випадкового лісу для оптимізації планування ресурсів та зниження витрат. Алгоритм, відомий завдяки ансамблевому навчанню, будує дерева рішень для регресії та класифікації. У порівнянні з моделями XGBoost, Ridge та Lasso, він перевершує точність прогнозування використання процесора та пам'яті. Дослідження, проведене з використанням набору даних Materna в Google Colaboratory, демонструє ефективність алгоритму в покращенні розподілу ресурсів та мінімізації впливу на навколишнє середовище.

Проемонстровано новий метод прогнозування використання процесора у віртуалізованих середовищах. Використовуючи спрощений метод кластеризації

					БР.ІІІ - 95.00.00.000 ПЗ	Арк.
						18
Змн.	Арк.	№ докум.	Підпис	Дата		

віртуальних машин, він покращує точність прогнозування шляхом групування подібних віртуальних машин на основі моделей використання ресурсів. Використовуючи алгоритми машинного навчання, він аналізує історичні дані процесора для створення кластерів віртуальних машин з порівнянною поведінкою, що дозволяє робити точніші прогнози. У роботі детально описано методологію моделі, використані алгоритми, експериментальну установку та валідацію, демонструючи її потенціал для оптимізації розподілу ресурсів та підвищення продуктивності у віртуальних середовищах.

Зосереджено увагу на вдосконаленні моделей градієнтного підвищення потужності для прогнозування використання процесора в хмарних середовищах. Це, ймовірно, передбачає масштабну оптимізацію параметрів, включаючи точне налаштування гіперпараметрів, методи перехресної перевірки та інженерію функцій [21]. Метою дослідження є розробка вдосконаленого інструменту прогнозування, здатного точно оцінювати використання процесора в хмарних середовищах. Така точність є життєво важливою для оптимізації розподілу ресурсів та управління продуктивністю в хмарних системах, зрештою прагнучи економічної ефективності та ефективної роботи шляхом використання складного підходу градієнтного підвищення потужності, точно налаштованого на конкретні вимоги хмарних обчислювальних середовищ.

Представлено проактивний підхід до розгортання віртуальних машин у хмарних обчисленнях, використовуючи прогнозування використання процесора через нейронну мережу ARIMA-BP. Передбачаючи проблеми з продуктивністю, він революціонізує стратегії розгортання, забезпечуючи якість обслуговування та ефективність сервера. Метод, що складається з чотирьох ключових кроків, спрямований на превентивне управління ресурсами, оптимізацію використання та покращення загальної продуктивності в хмарних середовищах.

Досліджено використання рекурентних нейронних мереж (RNN) для прогнозування використання процесора в хмарних обчисленнях. Аналізуючи історичні дані про процесор та мережу, дослідження використовує RNN для фіксації часових залежностей та прогнозування моделей використання. Демонструючи

					БР.ІІІ - 95.00.00.000 ІІЗ	Арк.
						19
Змн.	Арк.	№ докум.	Підпис	Дата		

ефективність RNN у хмарних середовищах, дослідження встановлює їхню здатність точно прогнозувати використання процесора. Ці результати сприяють удосконаленню стратегій розподілу ресурсів, підкреслюючи потенціал для покращеної оптимізації в хмарних інфраструктурах [22].

Зосереджено увагу на управлінні коливаннями процесора в хмарних обчисленнях, впроваджуючи модель LSTM. Вона прогнозує використання процесора до 30 хвилин випередження, сприяючи динамічному масштабуванню потужності. За допомогою оцінки продуктивності за допомогою рекурентних нейронних мереж (RNN) та сучасних моделей оцінюється її точність прогнозування майбутнього використання. Підкреслюється потенціал LSTM для оптимізації розподілу ресурсів та мінімізації витрат у додатках, розміщених у публічній хмарі, що пропонує перспективні рішення для зменшення впливу на навколишнє середовище та витрат.

Запропоновано гібридну модель прогнозування використання хмарних ресурсів, що поєднує SARIMA для сезонних навантажень та LSTM/ARIMA для несезонних моделей. Вона підкреслює точність LSTM у нерегулярних моделях, ефективність SARIMA у прогнозуванні майбутнього використання та її значення для допомоги постачальникам уникненню надмірного або недостатнього виділення ресурсів.

Розглянуто проблему неефективного управління ресурсами, що впливає на продуктивність хмарних обчислень. Вони представляють метод RATS-HM, який поєднує в собі покращену оптимізацію Cat Swarm для планування завдань та групово-оптимізовану глибоку нейронну мережу (GO-DNN) для розподілу ресурсів. Для оцінки запропонованої моделі використовуються такі показники продуктивності, як споживання енергії, використання ресурсів, використання пропускної здатності, використання пам'яті та час відгуку. За допомогою моделювання в CloudSim вони демонструють, що стратегія RATS-HM оптимізує використання ресурсів пам'яті ефективніше порівняно з сучасними системами, такими як FCFS, ITSEPM та циклічний алгоритм.

					БР.ІІІ - 95.00.00.000 ПЗ	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дата		

1.3 Концепції та моделі управління хмарними ресурсами

Управління еластичними ресурсами – це процес розподілу обчислювальних, сховищних та мережевих ресурсів між набором програм для спільного досягнення цілей продуктивності програм, постачальників хмарних послуг (CSP) та користувачів хмарних ресурсів. Цілями CSP є ефективне та результативне використання ресурсів у межах обмежень Угод про рівень обслуговування (SLA) з користувачами хмари. Ефективне використання ресурсів зазвичай досягається за допомогою технологій віртуалізації, які сприяють статистичному мультиплексуванню ресурсів між клієнтами та програмами. На рис. 1.1 показано концептуальну основу для управління хмарними ресурсами, де m користувачів запросили виконання різних програм у центрі обробки даних. Кожна програма має певні вимоги до ресурсів, які мають бути виконані задовольняються центром обробки даних. Система керування ресурсами (RMS) розгорнута в центрі обробки даних для отримання запитів додатків від хмарних клієнтів та генерування задовільних відповідей на них шляхом розподілу необхідної ємності ресурсів (відповідно до потреб додатків) у вигляді віртуальних машин. RMS складається з блоку керування віртуальними машинами (VMU) та блоку керування завданнями (TMU). VMU відповідає за планування та розміщення віртуальних машин на фізичних машинах (PM), міграцію віртуальних машин у разі перевантаження/недовантаження на PM. TMU працює з додатками, отриманими від хмарних клієнтів/користувачів, розділяє їх на завдання, планує та призначає їх на вибрані віртуальні машини для виконання. Інформація про виконання робочого навантаження збирається у вигляді історичної бази даних робочого навантаження, яка використовується для навчання прогнозувальника робочого навантаження для оцінки майбутнього робочого навантаження та інформації про використання ресурсів [23]. Ця прогнозована інформація використовується для прийняття рішень щодо енергоефективного розподілу ресурсів та оптимізованого балансування навантаження.

					БР.ІІІ - 95.00.00.000 ПЗ	Арк.
						21
Змн.	Арк.	№ докум.	Підпис	Дата		

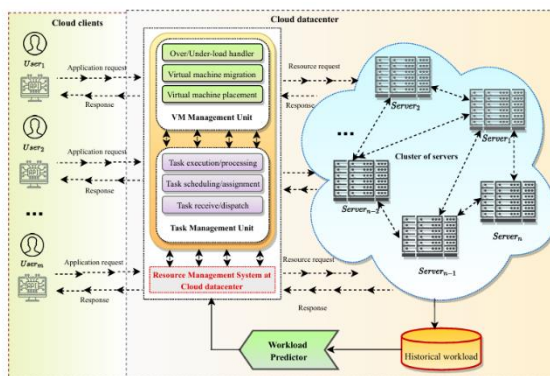


Рисунок 1.1 - Концептуальна основа управління ресурсами в хмарному середовищі

Методи управління віртуальними ресурсами в хмарному центрі обробки даних можна загалом розділити на управління робочим навантаженням на рівні застосунків, віртуальних машин та фізичних машин, як показано на рис. 1.2. На *рівні застосунків* управління навантаженням необхідне під час планування завдань та операцій призначення. Планування завдань означає порядок вибору завдання для виконання, тоді як призначення завдань стосується вибору відповідної віртуальної машини для виконання завдання. Планування та призначення завдань може виконуватися за допомогою різних методів, показаних на рис. 1.2. Традиційні методи планування завдань включають планування за принципом «перший прийшов - перший вийшов» (FIFO), «останній прийшов - перший вийшов» (LIFO), планування за найкоротшим терміном виконання (SJF), кругове планування (RR) тощо. Також завдання можуть плануватися на основі вартості виконання та терміну виконання або багатоцільової комбінації, що задовольняє кілька цілей одночасно з пріоритетом. Крім того, вибір віртуальної машини для виконання завдання на основі кількох обмежень виконується за допомогою традиційних підходів, таких як FIFO, LIFO, сортування віртуальних машин за ресурсною потужністю, ймовірнісний метод або метод максимальної ймовірності, евристичні та еволюційні алгоритми оптимізації.

Керування робочим навантаженням на *рівні віртуальних машин* можна розділити на призначення або розміщення віртуальних машин, міграцію віртуальних машин та консолідацію віртуальних машин [24]. *Призначення віртуальних*

					БР.ІП - 95.00.00.000 ПЗ	Арк.
						22
Змн.	Арк.	№ докум.	Підпис	Дата		

машин – це процес розгортання віртуальних машин на вибраному сервері, тоді як *міграція віртуальних машин* – це процес переміщення віртуальних машин з перевантажених/недозавантажених серверів на вибраний оптимальний сервер, а *консолідація віртуальних машин* – це техніка розгортання віртуальних машин на мінімальній кількості активних серверів для зменшення споживання енергії та втрат ресурсів. На рис. 1.2 показано існуючі методи розміщення та міграції віртуальних машин. Віртуальні машини можна розподілити між доступними серверами в кластері, використовуючи звичайні/традиційні методи, такі як First-Fit, Best-Fit, Random-Fit або Worst-Fit, де вимоги до ресурсів віртуальних машин задовольняються відповідним сервером без будь-якої конкуренції за ресурси. Віртуальні машини можна розміщувати відповідно до однієї мети, такої як термін виконання завдання користувача, максимальне використання процесора, знижене споживання енергії, безпека, надійність або максимальна доступність тощо. Крім того, віртуальні машини можна розподілити, задовольняючи кілька цілей одночасно, за допомогою пріоритетного або недомінованого сортування на основі парето-фронту. Для досягнення того ж результату зазвичай використовуються такі підходи, як еволюційна оптимізація (такі як PSO, ACO, Firefly, GA та гібридні алгоритми оптимізації тощо), нечітка кластеризація, розподіл віртуальних машин на основі теорії ігор, жадібний підхід, ймовірнісний метод та інші методи оптимізації. Розподіл віртуальних машин додатково оптимізується шляхом міграції віртуальних машин у разі конфлікту ресурсів за допомогою методу міграції віртуальних машин у реальному часі, тобто гарячої міграції віртуальних машин, та офлайн - міграції віртуальних машин, тобто холодної міграції відповідно до вимог або переміщення віртуальних машин, визначених блоком балансування навантаження. Крім того, віртуальні машини можна мігрувати, задовольняючи певні цілі, включаючи обмеження, пов'язані з енергоспоживанням, мережею, безпекою або надійністю тощо.

					БР.ІІІ - 95.00.00.000 ПЗ	Арк.
						23
Змн.	Арк.	№ докум.	Підпис	Дата		

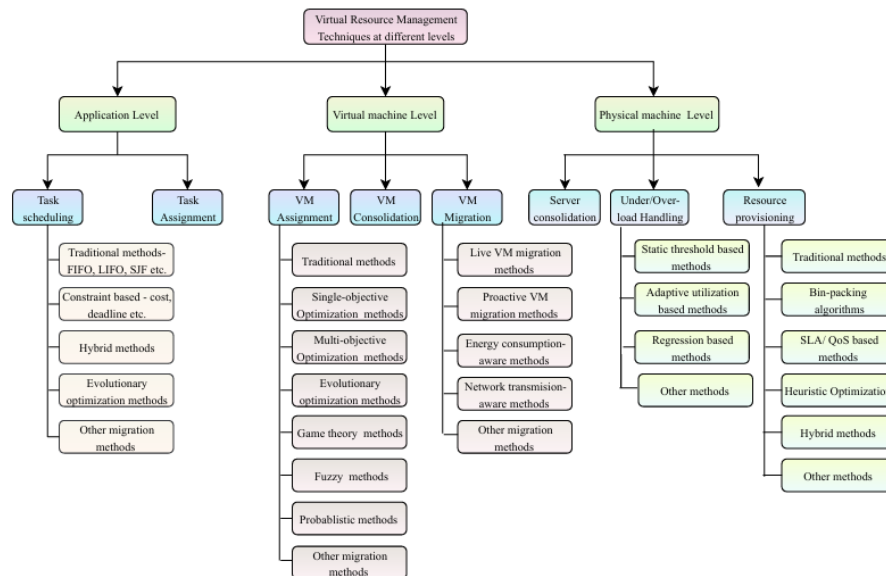


Рисунок 1.2 - Категоризація методів управління ресурсами

Основними операціями, що входять до управління на *рівні фізичної машини*, є консолідація серверів, обробка перевантаження/недовантаження на сервері та виділення ресурсів сервера. Консолідація серверів спрямована на мінімізацію кількості активних серверів шляхом застосування енергоефективних методів розміщення віртуальних машин [25]. Обробка перевантаження/недовантаження включає виявлення відповідного стану на етапі керування та пом'якшення його наслідків за допомогою міграції віртуальних машин. Надання ресурсів – це ефективний розподіл фізичних ресурсів між віртуальними машинами до появи фактичного попиту від користувача хмари. Різні методи обробки перевантаження/недовантаження та виділення ресурсів згадані на рис. 1.2. Проблема перевантаження /недовантаження вирішується шляхом міграції віртуальних машин з перевантажених/недовантажених серверів на вибраний ефективний сервер, де сервер призначення може бути обраний шляхом застосування різних підходів, таких як метод першого підходу, метод найкращого підходу, випадкове підлаштування, вибір на основі нечіткого підходу або евристична оптимізація тощо. Цю проблему можна вирішити реактивно або проактивно, де перший підхід мігрує віртуальну машину після виявлення перевантаження, тоді як другий підхід оцінює перевантаження/недовантаження до його виникнення та

пом'якшує його наслідки шляхом проактивного переміщення віртуальних машин. Фізичні ресурси сервера виділяються за допомогою традиційних методів, алгоритму упаковки в контейнери та методів евристичної оптимізації, одночасно задовольняючи обмеження рівня доступу до даних (SLA) та якості обслуговування (QoS).

1.4 Висновки по розділу

У першому розділі було розглянуто теоретичні основи прогнозування витрат на хмарні ресурси. Проаналізовано принципи функціонування хмарних обчислень, підходи до управління ресурсами та сучасні моделі оптимізації використання хмарної інфраструктури. Досліджено існуючі методи прогнозування витрат і визначено основні фактори, що впливають на формування вартості хмарних сервісів. Отримані результати дозволили сформулювати теоретичну базу для побудови систем прогнозування витрат із використанням методів машинного навчання.

					БР.ІП - 95.00.00.000 ПЗ	Арк.
						25
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2 . МЕТОДИ ТА ІНСТРУМЕНТИ ПРОГНОЗУВАННЯ ВИТРАТ ЗА ДОПОМОГОЮ МАШИННОГО НАВЧАННЯ

2.1 Методологія побудови систем прогнозування

Етапи методології дослідження для прогнозування використання ресурсів у хмарних обчисленнях наведені нижче та також представлені на рисунку 2.1:

- Розуміння дослідження: Метою цього дослідження є прогнозування точності використання процесора віртуальної машини та пропускну здатності мережі з використанням прогностичних моделей машинного навчання (ML) та навчання ділянці (DL) у середовищах хмарних обчислень, що зосереджено на досягненні ефективного використання ресурсів, підвищенні продуктивності та зниженні витрат.
- Збір даних: Якісні та кількісні дані з репозиторіїв з відкритим кодом, трасування Microsoft Azure з GitHub, що охоплюють використання процесора віртуальної машини та пропуску здатність мережі.
- Попередня обробка даних: очищення даних, виконання інженерії ознак для готовності моделі. Вирішення проблеми відсутніх значень та підготовка наборів даних для навчання прогностичних моделей машинного навчання (ML) та навчання з навчанням (DL) .
- Створення прогнозової моделі: Навчання прогнозних моделей машинного навчання (ML) та навчання ділянці (DL) з використанням вибраних наборів даних. BiLSTM обрано за його здатність фіксувати складні часові залежності, що має на меті точно прогнозувати використання процесора віртуальної машини та пропуску здатність мережі.
- Оцінювання: Метрики включають середньоквадратичну похибку (MSE), середню абсолютну похибку (MAE), середньоквадратичну похибку та значення R-квадрату (R²) для оцінки точності прогнозування.
- Критерії ефективності: спрямовані на досягнення ефективного використання ресурсів, підвищення продуктивності та зниження операційних витрат.

					БР.ІІІ - 95.00.00.000 ПЗ	Арк.
						26
Змн.	Арк.	№ докум.	Підпис	Дата		

Експериментування та зворотний зв'язок: Експериментальні сценарії ретельно розробляються з контрольованими змінними для ретельного тестування прогностичних моделей [26]. Ці експерименти спрямовані на підтвердження ефективності та надійності моделі у прогнозуванні використання ресурсів.

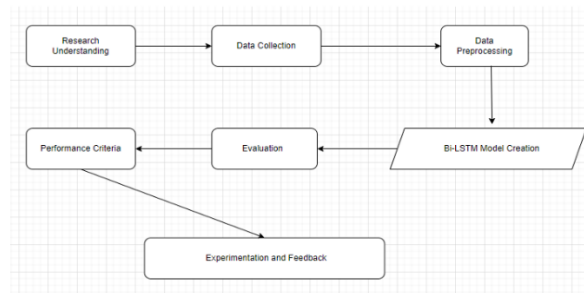


Рисунок 2.1 - Блок-схема методології дослідження

Опис набору даних

У цьому дослідженні використовуються два ключові набори даних: набір даних Bitbrains, та набір даних Microsoft Azure Traces, отриманий з GitHub. Набір даних Bitbrains спеціально зосереджений на прогнозуванні використання процесора та пропускної здатності мережі, а набір даних Microsoft Azure Traces зосереджений на використанні процесора в контексті часових рядів. Набір даних Bitbrains включає дані про позначку часу, використання процесора та пропускну здатність мережі як основні параметри.

Аналогічно, набір даних Microsoft Azure Traces 2017 надає дані про використання процесора для прогнозування моделей використання процесора та пропускної здатності мережі.

Набори даних надають доступ до ключових показників ресурсів, включаючи використання процесора, що є вирішальним для прогностичних моделей у цьому дослідженні. Для забезпечення узгодженості та порівнянності показники використання ресурсів у цих наборах даних нормалізуються, масштабуючи ці вимірювання в діапазоні від 0 до 1.

Кожен ресурс у наборах даних нормалізується окремо, причому значення 1 означає максимальну потужність ресурсу стосовно всіх машин, що

					БР.ІІІ - 95.00.00.000 ПЗ	Арк.
						27
Змн.	Арк.	№ докум.	Підпис	Дата		

охоплюються трасуванням. Ця методологія нормалізації сприяє стандартизованому підходу до аналізу та прогнозування ресурсів у наборах даних, забезпечуючи надійне та порівнянне розуміння моделей використання ресурсів.

Архітектура

Модель розподілу ресурсів у хмарних обчисленнях інтегрує різні прогнозні моделі машинного навчання (ML) та глибокого навчання (DL), включаючи лінійну регресію, регресію дерева рішень, регресію опорних векторів, регресію з градієнтним підвищуючим навантаженням, LSTM та Bi-LSTM. Ця модель діє як комплексна структура, яка керує розподілом та управлінням критично важливими ресурсами, такими як пропускна здатність мережі, ємності сховища та потужність процесора, на основі даних, отриманих з цих прогнозних моделей, на рисунку 2.2.

У хмарних середовищах доступність обчислювальних ресурсів, таких як мережеві можливості, ємності сховища та потужність процесора, є основою надання послуг. Прогнозні моделі машинного навчання (ML) та навчання з навчанням (DL) відіграють вирішальну роль, прогножуючи використання процесора та пропускну здатність мережі, що суттєво впливає на ці ресурси. Ці моделі дозволяють постачальникам хмарних послуг точніше передбачати потреби в ресурсах, оптимізуючи таким чином розподіл мережевих, сховищних та процесорних ресурсів відповідно до прогнозованого попиту [27]. Використовуючи дані цих моделей, хмарні середовища досягають покращеного використання ресурсів та ефективності розподілу.

Як у сценаріях на основі резервування, так і на вимогу, дані прогнозного моделювання мають значний вплив на методи розподілу ресурсів. Прогнозні моделі використовуються для точного прогнозування потреб у ресурсах з часом, що допомагає у стратегіях розподілу резервувань. Цей метод гарантує постійне використання відповідно до очікуваних потреб, забезпечуючи при цьому ефективне резервування ресурсів та мінімізуючи втрати [28]. Прогнозні моделі одночасно підтримують оптимізацію ресурсів у режимі реального часу для сценаріїв на вимогу, динамічно змінюючи ресурси процесора та мережі у відповідь на

					БР.ІП - 95.00.00.000 ПЗ	Арк.
						28
Змн.	Арк.	№ докум.	Підпис	Дата		

термінові потреби. Під час коливань вимог ця динамічна адаптивність значно покращує продуктивність системи та використання ресурсів.

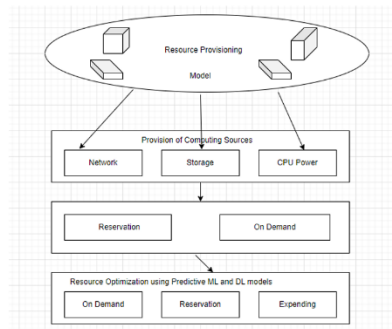


Рисунок 2.2 - Архітектура забезпечення ресурсами

Стратегії оптимізації, прогностні моделі машинного навчання та навчання на основі даних (ML) відіграють ключову роль у різних аспектах управління ресурсами. Оптимізація ресурсів на вимогу в режимі реального часу використовує ці моделі для швидкої адаптації ресурсів процесора та мережі до нагальних потреб. Таке гнучке налаштування значно покращує ефективність використання ресурсів, тим самим підвищуючи продуктивність системи під час коливань попиту. Крім того, оптимізація на основі резервування використовує прогностні моделі для мінімізації непотрібного резервування ресурсів, забезпечуючи стабільність системи, зберігаючи при цьому ефективність. Крім того, ці моделі сприяють розширенню ресурсів під час раптових стрибків попиту, дозволяючи проактивне масштабування для забезпечення адекватної продуктивності в години пік та оптимізуючи економічну ефективність шляхом динамічного регулювання розподілу ресурсів. Загалом, прогностні моделі ML та навчання на основі даних утворюють основу ефективних стратегій оптимізації ресурсів, що сприяє підвищенню продуктивності та економічної ефективності в середовищах хмарних обчислень.

Запропонований підхід

Багато досліджень заглибилися у використання потенціалу моделей машинного навчання (ML) та глибокого навчання (DL) для прогнозування використання ресурсів у системах хмарних обчислень. Ці моделі відіграють ключову

					БР.ІІІ - 95.00.00.000 ПЗ	Арк.
						29
Змн.	Арк.	№ докум.	Підпис	Дата		

роль у прогнозуванні та ефективному управлінні розподілом критично важливих ресурсів, таких як використання процесора та дані, що передаються через мережу. Це дослідження було зосереджено на вивченні комплексного набору прогнозних моделей, що охоплює чотири фундаментальні моделі ML: лінійну регресію, регресію дерева рішень, регресію з градієнтним підсиленням, регресію опорних векторів, а також дві впливові моделі DL: довгу короткочасну пам'ять (LSTM) та її розширення, двонаправлену довгу короткочасну пам'ять (BiLSTM), див. рисунок 2.3.

Лінійна регресія (LiR):

Лінійна регресія – це традиційна регресійна модель, метою якої є встановлення лінійного зв'язку між незалежними змінними (функціями) та залежною змінною (використанням ресурсів). При прогнозуванні використання процесора та пропускної здатності мережі в хмарних обчисленнях лінійна регресія намагається виявити прямі лінійні зв'язки між різними факторами, що впливають на використання ресурсів, та фактичним використанням (див. рис. 2.3а).

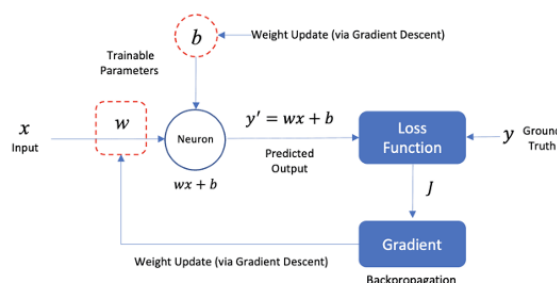


Рисунок 2.3 - Архітектура моделі лінійної регресії (LiR).

Регресія дерева рішень (DTR):

Регресія дерева рішень будує деревоподібну структуру на основі характеристик даних, що дозволяє прогнозувати використання ресурсів [29]. У контексті прогнозування хмарних ресурсів регресія дерева рішень створюватиме правила прийняття рішень на основі таких атрибутів, як пропускна здатність мережі, використання процесора, для прогнозування рівнів використання ресурсів (див. рис. 2.4).

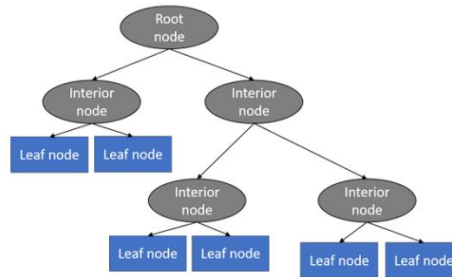


Рисунок 2.4 - Вихідний код архітектури моделі DTR

Градiєнтна регресія з підсилювальним градієнтом (GBR):

Градiєнтна регресія з посиленням будує ансамбль дерев рішень, ітеративно покращуючи прогнози шляхом мінімізації помилок. У прогнозуванні використання ресурсів хмарних обчислень ця модель поєднує кілька підходів для уточнення прогнозів шляхом розуміння складних взаємозв'язків між різними факторами, що впливають на використання ресурсів (див. рис. 2.5).

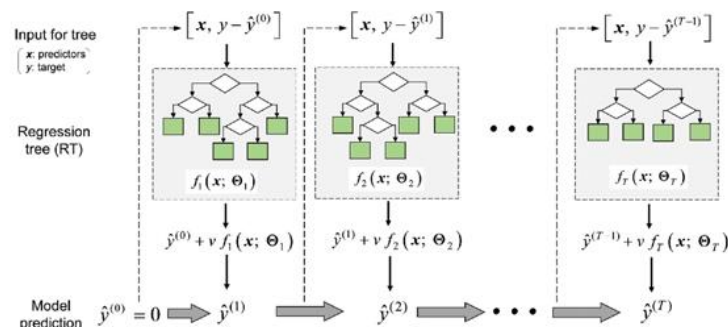


Рисунок 2.5 - Вихідний код архітектури моделі GBR

Регресія опорних векторів (SVR):

Регресія опорних векторів встановлює гіперплощину, яка найкраще відображає зв'язок між вхідними ознаками та використанням ресурсів. У контексті хмарних обчислень вона визначає високовимірну межу для прогнозування використання процесора та пропускної здатності мережі на основі різних факторів (див. рис. 2.6) .

Довготривала короточасна пам'ять (ДКМП):

•



BiLSTM розширює LSTM, обробляючи дані як у прямому, так і у зворотному напрямках, щоб одночасно фіксувати минулий та майбутній контекст. У прогнозуванні використання ресурсів хмарних обчислень BiLSTM всебічно аналізує часові залежності в обох напрямках.



Інструкції, що забезпечують більш цілісне розуміння використання процесора та моделей пропускнуї здатності мережі, на рисунку 2.8.

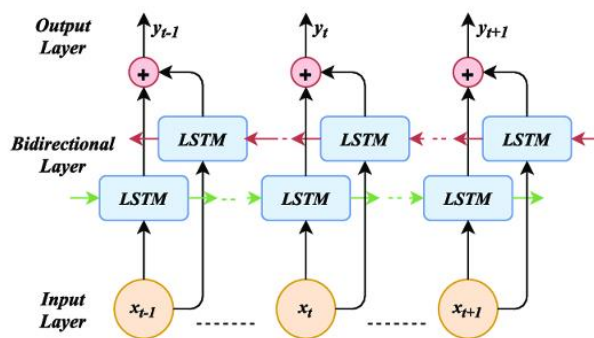


Рисунок 2.8 - Архітектура моделі BiLSTM

Порівняння традиційних регресійних моделей, таких як лінійна регресія, регресія дерева рішень, градієнтна регресія з підвищуючим градієнтом та регресія опорних векторів, з моделями глибокого навчання, такими як LSTM та BiLSTM, відображає оцінку простіших, заснованих на правилах підходів у порівнянні зі складними моделями з покращеною пам'яттю, в їхній здатності прогнозувати моделі використання ресурсів у хмарних обчисленнях, які особливо зосереджені на використанні процесора та пропускній здатності мережі.

Запропонований підхід ґрунтується на використанні моделі BiLSTM (дво-направлена довга короткочасна пам'ять) для підвищення точності прогнозів як щодо використання процесора віртуальної машини, так і щодо пропускної здатності мережі. По-перше, набори даних з Bitsbrain (для використання процесора та пропускної здатності мережі) та Microsoft Azure Traces 2017 (зокрема, для використання процесора) збираються та ретельно обробляються для забезпечення повноти та релевантності в контексті дослідження. Після цього прогностичні моделі впроваджуються та ретельно навчаються з використанням цих наборів даних. Основна увага приділяється оцінці ключових прогностичних метрик, таких як середня абсолютна похибка (MAE), середньоквадратична похибка (MSE), середньоквадратична похибка (RMSE) та показник R2, для оцінки точності моделі у прогнозуванні використання ресурсів віртуальної машини. Кінцева мета двояка: оптимізувати використання ресурсів та підвищити загальну продуктивність у середовищах хмарних обчислень [30]. Це досягається шляхом комплексного вивчення та порівняння моделі BiLSTM з усталеними фреймворками машинного

навчання, метою якого є визначення найефективнішої моделі для точного прогнозування та ефективних стратегій розподілу ресурсів.

2.2 Моделі та алгоритми машинного навчання для аналізу витрат

Запропонований алгоритм базується на гібридній моделі GA-PSO [9]. Гібридна модель інтегрує окремі прогностичні моделі, що дозволяє подолати обмеження прогностичної моделі з точки зору вартості та складності кінцевої моделі. Головною метою є вибір та відповідне поєднання набору методів прогнозування для підвищення точності кінцевого прогнозу. Поєднуючи переваги кожної прогностичної моделі та мінімізуючи її недоліки, можна покращити точність. У цьому розділі представлено детальну інформацію про різні компоненти запропонованого алгоритму [31].

Нейронні мережі

Нейронні мережі, натхненні біологічним мозком, є одним із найуніверсальніших та найвідоміших підходів до машинного навчання. Нейронні мережі широко застосовуються в різних галузях, включаючи регресію, кластеризацію, класифікацію, прогнозування, розпізнавання образів, навчання та робототехніку [32]. Стандартна нейронна мережа прямого зв'язку складається з вхідного шару, одного або кількох прихованих шарів та вихідного шару. В принципі, нейрон - це стратегічний блок обробки сигналів, який зчитує численні сигнали як вхідні, а потім використовує функцію активації (наприклад, сигмоподібну) для генерації сигналу як вихідного. Сенсорний шлях для мережі для виявлення середовища формується шляхом зчитування нормалізованого сигналу у вхідний шар. Наприклад, у задачі прогнозування використання процесора цей сигнал стосується вимог до процесора хоста з попередніх кроків часу. Цей сигнал спочатку надсилається вперед через зважені з'єднання через взаємопов'язані шари нейронів. Як тільки сигнал поширюється мережею, мережа генерує вихідний сигнал через кінцевий вихідний шар. Вихідний сигнал корелює з вимогами до процесора на

					БР.ІП - 95.00.00.000 ПЗ	Арк.
						34
Змн.	Арк.	№ докум.	Підпис	Дата		

майбутніх кроках часу. Кілька прикладів у літературі показують ефективно використання нейронних мереж для вирішення таких задач, як прогнозування використання процесора. Варто зазначити, що одним із найскладніших завдань у реалізації нейронних мереж є навчання вагових коефіцієнтів мережі. Навчання вагових коефіцієнтів мережі таким чином, щоб мережа могла генерувати високоточний вихід для будь-якого заданого вхідного сигналу, є складною задачею оптимізації, і тому передбачає потенційне використання ройових та оптимізаційних алгоритмів. Функціонально-ланкова нейронна мережа (FLNN) – це клас нейронних мереж, які використовують вищі комбінації вхідних даних. FLNN використовується в багатьох програмах, включаючи класифікацію та прогнозування [26].

Запропонований алгоритм

Запропонований алгоритм прогнозування використання ресурсів навчається за допомогою гібридного GA-PSO для кращого управління ресурсами. Через величезний розмір користувацьких програм та велику кількість ресурсів управління ресурсами стає проблемою. Точність моделі прогнозування ресурсів сильно впливає на використання ресурсів та інші фактори, пов'язані з продуктивністю хмари. Модель прогнозування повинна мати можливість обробляти кілька ресурсів. Алгоритми GA-PSO використовуються для поєднання двох моделей навчання, щоб допомогти ефективно керувати хмарними ресурсами. Література показує, що збільшення кількості ітерацій алгоритмів на основі GA призводить до покращених рішень порівняно з іншими алгоритмами. Однак велика кількість ітерацій призводить до більшого часу виконання для конвергенції до найкращих рішень. З іншого боку, PSO дає рішення за менший час порівняно з іншими алгоритмами. Однак PSO страждає від проблеми швидкої конвергенції та локального оптимуму. Запропонований алгоритм використовує переваги як GA, так і PSO для отримання кращих результатів. Гібридна модель використовує різноманітність та швидку конвергенцію до оптимальних рішень і досягає найкращого рішення порівняно швидше, ніж інші алгоритми. Гібридний алгоритм на основі GA-PSO складається з атрибутів як GA, так і PSO. Схеми алгоритму GA-PSO

					БР.ІІІ - 95.00.00.000 ПЗ	Арк.
						35
Змн.	Арк.	№ докум.	Підпис	Дата		

показана на рисунку 2.9.

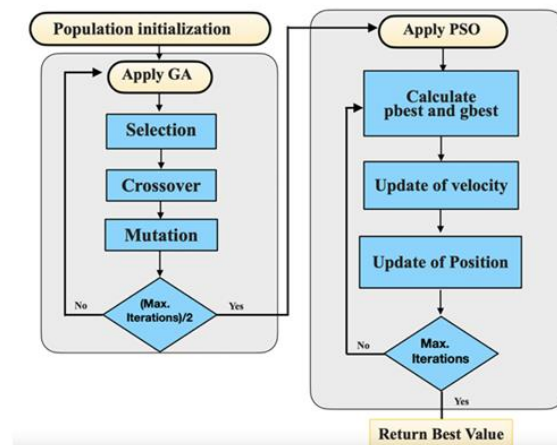


Рисунок 2.9 - Ілюстрація потоку алгоритму GA-PSO.

Процедура починається з генерації випадкової популяції та застосування генетичного алгоритму (ГА) до початкової популяції. Після того, як процедура досягає половини кількості ітерацій, вона зупиняється, а згенеровані рішення передаються до PSO, де обчислюються локальні та глобальні змінні, а змінні відповідно оновлюються. Процедура зупиняється, коли досягається максимальна кількість ітерацій. Ініціалізація в гібридному алгоритмі GA-PSO налаштовується на певну кількість ітерацій. На першій ітерації рішення запускається випадковим чином. В кінці першої ітерації генерується серія нових рішень. Ці рішення використовуються рекурсивним чином для формування серії нових рішень. У стандартному ГА популяція представлена як набір хромосом. Кожна хромосома складається з генів. У ГА еволюція хромосом відбувається за допомогою операторів кросовера та мутації, а також відбору потомства для наступних поколінь. У запропонованому алгоритмі використовується вибір за допомогою колеса рулетки. Процедура відбору використовує обертання колеса на основі значення придатності кожної хромосоми. Для кросовера використовується оператор кластерного кросовера. У цій процедурі вибираються точки розриву в кожній хромосомі, і нові хромосоми генеруються шляхом змішування вибраних точок. Для оператора мутації використовується swar, де вибирається пара генів, і гени в обох точках міняються місцями. Решта ітерацій виконуються з PSO, беручи популяцію,

згенеровану за допомогою GA. Положення та швидкість частинок у попередній ітерації використовуються для створення нових частинок. Положення та швидкість змінюються на основі рішень, отриманих за допомогою GA.

Рішення gbest - це рішення з найнижчим значенням придатності. Крім того, на кожній ітерації проводиться порівняння на основі значення придатності між раніше створеними частинками та новоствореними частинками. pbest містить частинку з найкращим значенням придатності. На кожній ітерації gbest зберігає найкращу частинку серед усього покоління, порівнюючи значення придатності з найкращим значенням. Це порівняння на кожному кроці гарантує, що всі частинки рухаються до найкращого рішення для досягнення оптимального рішення. Придатність розраховується в обох випадках за допомогою MAE. Алгоритм 2.1 показано реалізацію гібридного алгоритму GA-PSO, тоді як на рисунку 2.10 показано послідовність дій гібридної моделі під час навчання мережі.

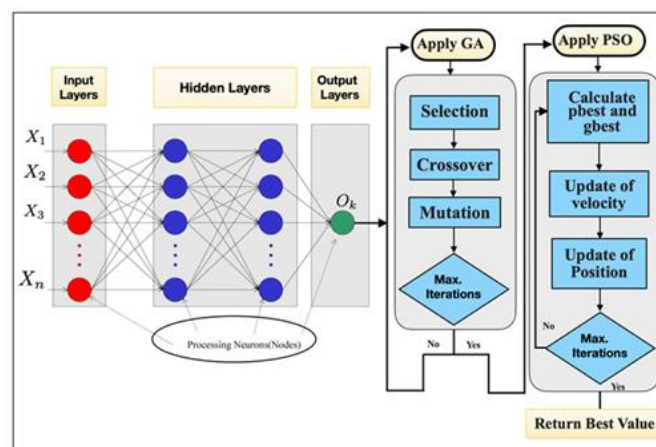


Рисунок 2.10 - Ілюстрація потоку гібридної моделі під час навчання мережі.

Фігура на рис. 2.11 показано систему прогнозування ресурсів для хмари [12]. Система складається з чотирьох модулів, а саме менеджера, препроцесора, тренера та предиктора. Модуль менеджера відповідає за збір даних моніторингу ресурсів. Зібрані дані зберігаються в репозиторії. База даних використовується для зберігання останніх даних моніторингу ресурсів.

```

Input :network weights
Output:best solutions
 $p_c$  crossover probability
 $p_m$  mutation probability
 $iter_{max}$  max number of iterations
initialize population  $p=E_1, \dots, E_{p_g}$ , each entity is a d-dimension vector
 $E_i = (e_{i1}, \dots, e_{id})$ 
 $iter=1$ 
while ( $iter \leq iter_{max}/2$ ) do
    calculate fitness with equation 4
    best chromosome  $E_{best}$ = fittest chromosome
    selection (according to the fitness function)
    crossover (according to crossover probability)
    mutation (according to mutation probability)
    select best chromosome ( $chromosome_i$ )
end
set particle  $p=(chromosome_i)$ 
calculate fitness of  $p$  according to Eq. 4
calculate velocity of  $p$  according to Eq.5
 $g_{best}$ =global best position
 $p_{best}$ =particles' best position
while not reach  $iter_{max}$  do
    initialize  $X_{ij}^t$  randomly
    initialize velocity  $v$  randomly
    evaluate  $p_i$ 
    update  $p_{best}$  and  $g_{best}$ 
end

```

Лістинг 2.1 - Гібридний алгоритм GA-PSO

Дані моніторингу, що зберігаються в базі даних, створюються як історичні дані моніторингу та часові ряди [32]. Модуль препроцесора створює ці форми даних моніторингу. Дані часових рядів у необробленому вигляді перетворюються модулем препроцесора на дані з наглядом для коригування вхідних даних нейронної мережі.

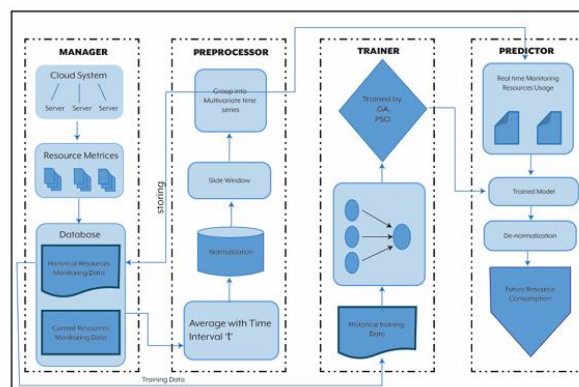


Рисунок 2.11 - Архітектура прогнозних систем.

Доступно розгортання різноманітних механізмів, які обробляють дані щодо робочого навантаження хмари. До них належать нормалізовані дані, усереднення

даних у довгостроковій перспективі, групування в багатовимірні часові ряди та ковзне вікно. Після обробки в модулі препроцесора дані, що виводяться як історичні дані ресурсів, надходять до колектора бази даних. Ці історичні дані ресурсів створюють модель для прогнозів, які виконуються в модулі тренера. Для прогнозування використання ресурсів дані передаються предикторам. Запропонована методологія включає метод навчання, доданий у модуль тренера, який використовує FLNN. Механізм GA-PSO навчає мережу один за одним для збільшення швидкості збіжності та підвищення точності прогнозування [33]. Розроблений механізм навчання називається FLGAPS ONN. Після завершення процесу навчання навчений модуль прогнозує використання ресурсів у майбутньому, що виконується в модулі прогнозування.

2.3 Проєктування та специфікація системи прогнозування

У специфікації проєкту пояснюється базова архітектура та вимоги до реалізації точного прогнозування використання ресурсів віртуальної машини, яке зосереджено на використанні процесора та пропускній здатності мережі в середовищах хмарних обчислень. У ній окреслено методи, структуру, архітектуру та основні вимоги. Після отримання наборів даних з репозиторіїв з відкритим кодом невід'ємний етап включає ретельну попередню обробку даних. Цей етап включає перевірку на наявність відсутніх значень, забезпечення повноти даних та застосування відповідних методів інженерії функцій для підготовки наборів даних до подальшого аналізу.

Регресія, регресія дерева рішень, регресія градієнтного підсилення, регресія опорних векторів, довга короткочасна пам'ять, двонаправлені мережі довгої короткочасної пам'яті (BiLSTM) для прогнозування точності використання процесора віртуальної машини та пропускної здатності мережі передачі даних у середовищах хмарних обчислень. Архітектура моделі включає двонаправлені рекурентні нейронні мережі, які дозволяють проводити комплексний аналіз часових залежностей у вхідних даних. Мережі BiLSTM включають як прямий, так і

					БР.ІІІ - 95.00.00.000 ПЗ	Арк.
						39
Змн.	Арк.	№ докум.	Підпис	Дата		

зворотний потік інформації, що дозволяє моделі фіксувати складні закономірності в даних часових рядів. Середовище реалізації служить основою для розробки, навчання та оцінки прогнозних моделей, сприяючи ефективному експериментуванню та ітеративному вдосконаленню моделі. Щоб визначити прогностичну точність та ефективність моделі BiLSTM, метрики оцінки кількісно визначають продуктивність моделі в прогнозуванні використання ресурсів, допомагаючи оцінити точність та надійність прогнозів.

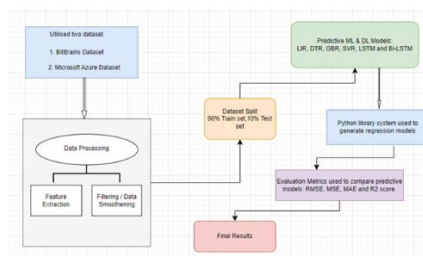


Рисунок 2.12 - Дорожня карта розробки специфікацій проектування для впровадження

Важливим аспектом є проведення порівняльного аналізу між моделлю BiLSTM, довгою короткочасною пам'яттю та лінійною регресією, регресією опорних векторів, регресією дерева рішень та регресією градієнтного підсилення. Це порівняльне дослідження має на меті підкреслити чудову прогностичну ефективність моделі BiLSTM у прогнозуванні використання процесора віртуальної машини та пропускну здатності мережі в середовищах хмарних обчислень [34]. Контрольовані експериментальні сценарії ретельно розроблені для полегшення всебічного порівняння прогностичних можливостей різних моделей. Ці контрольовані експерименти сприяють демонстрації ефективності моделі BiLSTM у прогнозуванні використання ресурсів, тим самим обґрунтовуючи її застосовність у реальних середовищах хмарних обчислень.

Впровадження. У цьому дослідницькому проєкті використовується мова програмування Python у Google Colab для прогнозування використання ресурсів. На рисунку 2.12 показано кроки впровадження. У цьому дослідженні використовуються два набори даних для використання ресурсів: набір даних Bitbrains

					БР.ІІІ - 95.00.00.000 ПЗ	Арк.
						40
Змн.	Арк.	№ докум.	Підпис	Дата		

вибирає параметри, такі як використання процесора, пропускна здатність мережі та позначка часу. Набір даних Microsoft Azure Traces вибирає параметри, такі як мінімальне використання процесора, максимальне використання процесора, середнє використання процесора та позначка часу. Для обох наборів даних реалізація виконується окремо: набір даних Bitbrains використовується для прогнозування використання процесора та пропускної здатності мережі, а Microsoft Azure Traces використовується для прогнозування використання процесора.

- Перед початком процесу моделювання ретельно виконуються ретельні перевірки на наявність відсутніх значень та обробка таких екземплярів, щоб забезпечити цілісність та повноту наборів даних. Для нормалізації даних як у наборах даних Bitbrains, так і в наборах даних Microsoft Azure Traces використовуються методи масштабування ознак, зокрема MinMaxScaler. Цей важливий крок попередньої обробки стандартизує атрибути, забезпечуючи однорідність та оптимальну продуктивність під час навчання моделі машинного навчання.

- Етап впровадження починається з масштабного дослідницького аналізу даних (EDA), спрямованого на всебічне розуміння наборів даних. Це поглиблене дослідження включає ретельне вивчення критичних деталей, таких як форма набору даних, розмір, типи даних, середні значення, назви стовпців, кількість, стандартні відхилення та діапазон між мінімальними та максимальними значеннями. Ці статистичні дані забезпечують цілісне уявлення про набори даних, необхідне для подальшого моделювання.

- Процес машинного навчання починається з визначення цільового стовпця, позначеного як «у», який буде передбачено моделями. Для полегшення навчання та оцінки моделі набір даних розділено на чотири підмножини: X-train, X-test, y-train та y-test. Цей поділ проводиться у співвідношенні 90-10, де 90% даних виділяється для навчання, а 10% - для тестування.

- Це сприяє ефективному впровадженню деяких процедур машинного та глибокого навчання, таких як лінійна регресія, регресія дерева рішень, градієнтна регресія з підвищуючим градієнтом, регресія опорних векторів, довготривала короткочасна пам'ять та двонаправлена довготривала короткочасна пам'ять.

					БР.ІІІ - 95.00.00.000 ПЗ	Арк.
						41
Змн.	Арк.	№ докум.	Підпис	Дата		

- Для виконання цих алгоритмів використовується потужна бібліотечна система, що забезпечує точні результати. Більше того, цей процес вміло налаштовує алгоритми, використовуючи можливості, що пропонуються бібліотечною системою.

- Такі показники оцінки, як середньоквадратична похибка, середня абсолютна похибка, похибка R-квадрат та середньоквадратична похибка, розраховуються для прогнозування точності кожної регресійної моделі.

- Тестування в режимі реального часу проводиться для перевірки ефективності моделей у практичних сценаріях з використанням тестових даних, що забезпечує їхню життєздатність та точність у реальному хмарному середовищі. Постійний моніторинг та оптимізація цих моделей залишаються ключовими, що дозволяє вносити корективи на основі динаміки хмарної інфраструктури та закономірностей у наборах даних, що розвиваються. Зрештою, ця реалізація має на меті забезпечити надійну прогнозну систему, що сприяє ефективному розподілу ресурсів, підвищенню продуктивності та зниженню витрат в інфраструктурах хмарних обчислень.

Експериментальна установка

Збір даних:

- *Набір даних Bitbrains:*
- Отримайте набір даних Bitbrains, переконавшись, що він містить відповідні показники, такі як використання процесора, пропускна здатність мережі та часова позначка.

- *Набір даних Microsoft Azure:*
- Зберіть набір даних Microsoft Azure, зосереджуючись на даних про використання процесора, забезпечуючи, щоб він включав схожий набір функцій для узгодженості та порівняння .

Попередня обробка даних:

- *Очищення та інтеграція даних:*
- Очистіть набори даних, щоб видалити відсутні значення, викиди або невідповідності.

					БР.ІІІ - 95.00.00.000 ПЗ	Арк.
						42
Змн.	Арк.	№ докум.	Підпис	Дата		

- Використання наборів даних Bitbrains та Microsoft Azure, що забезпечує сумісність з точки зору функцій та часових рамок використання процесора.
- *Інженерія функцій:*
- Вилучено відповідні характеристики для моделювання, такі як історичне використання процесора, моделі мережевого трафіку, час доби тощо.
- Трансформуйте категоріальні змінні, нормалізуйте/стандартизуйте числові змінні, якщо потрібно.

Експериментальна установка:

- *Вибір моделі:*
- Обрано чотири традиційні моделі машинного навчання: лінійна регресія (LIR), регресія дерева рішень (DTR), регресія опорних векторів (SVR), регресія градієнтного підсилення (GBR).
- Обрані дві моделі глибокого навчання: довга короткочасна пам'ять (LSTM) та двонаправлена довга короткочасна пам'ять (BiLSTM).
- *Розділення даних:*
- Розділив набори даних на навчальні, валідаційні та тестові набори, зберігаючи часовий порядок, якщо це можливо.
- *Модельне навчання:*
- Навчив кожну з вибраних моделей (LIR, DTR, SVR, GBR, LSTM, BiLSTM) окремо, використовуючи навчальні дані.
- *Оцінка моделі:*
- Оцінено продуктивність моделей за допомогою набору валідації для налаштування гіперпараметрів та забезпечення узагальнення.
- Порівнювали моделі на основі таких показників, як середньоквадратична похибка (MSE), середня абсолютна похибка (MAE), середньоквадратична похибка (RMSE) та коефіцієнт детермінації (R-квадрат).

Тестування конкретної моделі:

- Тестування набору даних Bitbrains:
- Протестував моделі машинного навчання (ML) та ді-логічного навчання (DL) (LIR, DTR, SVR, GBR, LSTM, BiLSTM) на наборі даних Bitbrains для

					БР.ІІІ - 95.00.00.000 ПЗ	Арк.
						43
Змн.	Арк.	№ докум.	Підпис	Дата		

прогнозування використання процесора та пропускної здатності мережі.

- Тестування набору даних Azure:
- Протестував моделі ML та DL (LIR, DTR, SVR, GBR, LSTM, BiLSTM) на наборі даних Microsoft Azure для прогнозування використання процесора.

Аналіз та висновки:

- Аналіз продуктивності:
- Проаналізовано та порівняно продуктивність кожної моделі в прогнозуванні використання процесора та пропускної здатності мережі.
- *Оцінка ефективності та економічної ефективності:*
- Розглянуто обчислювальну ефективність та економічну ефективність моделей щодо використання ресурсів.

Стек інструментів та технологій:

Мови програмування:

- Python: Він широко використовується для попередньої обробки даних, реалізації моделей (з використанням таких бібліотек, як scikit-learn, TensorFlow, Keras) та аналізу.

Бібліотеки та фреймворки:

- scikit-learn: Використовується для реалізації традиційних моделей машинного навчання (лінійна регресія, регресія дерева рішень, регресія опорних векторів, регресія з градієнтним підсилювачем).
- TensorFlow/Keras: Використовується для розробки та навчання моделей глибокого навчання (LSTM, BiLSTM).
- Pandas: Використовується для маніпулювання даними та їх попередньої обробки.
- NumPy: Використовується для числових обчислень.
- Matplotlib/Seaborn: Використовується для візуалізації даних.

Хмарні платформи:

- Google Colab: Використовується для впровадження, експериментів та співпраці.

					БР.ІІІ - 95.00.00.000 ПЗ	Арк.
						44
Змн.	Арк.	№ докум.	Підпис	Дата		

- Google Диск: Він використовується для зберігання та доступу до наборів даних.

Збір та управління даними:

- Набір даних Bitbrains: Керування та доступ до нього здійснюється через Google Диск.

- *Набір даних Microsoft Azure. Керування та доступ до нього здійснюється через Google Диск. Експерименти та аналіз:*

- Навчання та валідація моделей: Використовувалися бібліотеки машинного та глибокого навчання на основі Python для навчання та валідації моделей у Google Colab.

- Метрики оцінки моделі: Використовував бібліотеки Python для оцінки моделей на основі метрик продуктивності (MSE, RMSE, R-квадрат, MAE) в Google Colab.

Візуалізація даних: Matplotlib та Seaborn для візуалізації в блокнотах Colab.

2.3 Висновки по розділу

У другому розділі досліджено методи та інструменти прогнозування витрат на хмарні ресурси за допомогою технологій машинного навчання. Розглянуто методологію побудови систем прогнозування, алгоритми аналізу даних та моделі машинного навчання для оцінювання майбутніх витрат. Також виконано проектування структури системи прогнозування та визначено її функціональні компоненти. У результаті сформовано підхід до створення інтелектуальної системи, здатної аналізувати навантаження та прогнозувати витрати хмарної інфраструктури.

					БР.ІП - 95.00.00.000 ПЗ	Арк.
						45
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 3. РОЗРОБКА СИСТЕМИ ТА ОЦІНКА ЕФЕКТИВНОСТІ

3.1 Прогнозування робочого навантаження хмарної інфраструктури

Хмарні обчислення дозволяють автоматичне масштабування ресурсів для кожної системи онлайн-транзакцій, що є однією з ключових характеристик, що відрізняє хмарну платформу від традиційних обчислювальних моделей. Однак ініціалізація нового віртуального екземпляра в хмарі не відбувається миттєво, платформи хмарного хостингу вводять певний період затримки під час розподілу апаратних ресурсів. Тому аналіз використання ресурсів на основі прогнозування є ключем до кількох важливих рішень щодо проектування системи та розгортання віртуальних машин, таких як управління робочим навантаженням, визначення розміру системи, планування потужності та автоматична генерація правил у хмарі. рис. 3.1 ілюструє загальний сценарій прогнозування робочого навантаження в хмарному центрі обробки даних. Мільйони користувачів надсилають запити через Інтернет до хмарного центру обробки даних. Запити обробляються там, і всі запити, що надійшли протягом певного інтервалу прогнозування, агрегуються як історичні дані, які пізніше використовуються для прогнозування робочого навантаження в майбутньому. Історичні дані збираються та попередньо обробляються для їх нормалізації. Потім нормалізовані дані передаються до системи прогнозування для прогнозування майбутнього робочого навантаження. Це прогнозоване робоче навантаження дасть попередню інформацію про майбутнє навантаження, і це дозволяє ефективно використовувати час для підготовки та організації виділення ресурсів, прийняття рішень щодо керування станом увімкнення або вимкнення живлення [35].

Загальний прогноз застосунку означає прогнозування майбутньої поведінки застосунку різними способами, такими як очікуване робоче навантаження та продуктивність. Важливим кроком стає прогнозування застосунку в різних вимірах для ефективного управління ресурсами в хмарі. Відповідно до очікуваних потреб застосунку в майбутньому, можна виявити ефективне виділення ресурсів

					БР.ІІІ - 95.00.00.000 ПЗ	Арк.
						46
Змн.	Арк.	№ докум.	Підпис	Дата		

та запланувати/виділити достатню кількість ресурсів для досягнення параметрів QoS, таких як використання процесора, час відгуку, доступність, надійність та безпека.

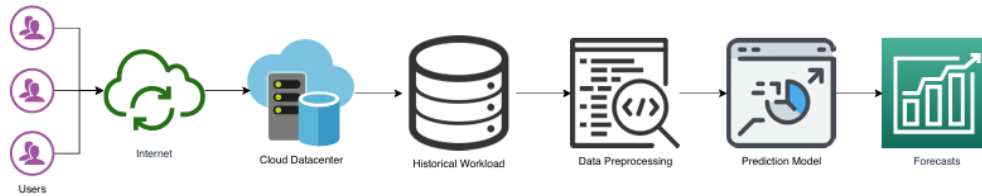


Рисунок 3.1 - Загальна модель прогнозування робочого навантаження

Різні виміри прогнозування робочого навантаження

Прогнозування може бути реалізовано на віртуальних та фізичних машинах. На рівні віртуальної машини необхідно підтримувати достатню кількість серверів для їхнього забезпечення, щоб збалансувати майбутнє навантаження в центрі обробки даних. Прогнозування робочого навантаження може бути розподілено за різними вимірами залежно від застосування наступним чином:

- *Потреба в прогнозуванні:* Дві основні вимоги, що мотивують прогнозування, - це управління ресурсами та управління додатками. Управління ресурсами робить акцент на ефективному використанні ресурсів для збільшення фінансових прибутків, запобігання порушенню SLA, підтримки постійної доступності ресурсів, запобігання порушенню умов SLA та уникнення марнування ресурсів [36]. Ще однією вимогою для точного прогнозування є управління додатками, яке стосується балансування навантаження додатків на віртуальних машинах, приймаючи рішення щодо міграції віртуальних машин, розподілу віртуальних машин та їх виділення на фізичних машинах.

- *Характеристики:* Різними характеристиками прогнозування робочого навантаження є точність результатів прогнозування, адаптивність відповідно до динамічних вимог до робочого навантаження з різними інтервалами прогнозування, проактивне прогнозування на випередження та ефективне використання історичних даних для оцінки майбутнього робочого навантаження.

- *Проблеми:* Зазвичай існують три проблеми прогнозування робочого навантаження. Часова та просторова складність, необхідна для обчислення прогнозованого робочого навантаження, повинна бути розумною, щоб його розгортання було доступним. Деталізація даних надзвичайно важлива для прогнозування робочого навантаження [37]. Грубозерниста та довгострокова вибірка даних призводить до того, що модель втрачає динамічну та адаптивну поведінку системи. З іншого боку, дрібнозернисте та короткострокове прогнозування займає надлишок часу та включає хибні деталі, які не є релевантними, що збільшує складність їх фіксації.

Ще однією великою проблемою є визначення довжини прогнозування. Коротка довжина прогнозування призводить до високої точності, але має підвищену складність. З більшою довжиною прогнозування кількість вибірок даних зменшується, а точність падає.

- *Метрики оцінювання:* Різні метрики для оцінки моделі прогнозування - це витрати, пов'язані з прогнозуванням робочого навантаження, досягнутий успіх у вигляді точності передбачення майбутніх потреб, прибуток, отриманий у вигляді інтелектуального управління ресурсами, та оцінена похибка під час прогнозування.

Найновіший сучасний стан та порівняння

Еволюційна нейронна мережа представлено підхід до прогнозування робочого навантаження для хмарних центрів обробки даних з використанням нейронної мережі та самоадаптивної диференціальної еволюції (SaDE). У цій роботі запити, що надходять від різних користувачів, агрегуються в певні одиниці часу як історичні дані. Потім виконується попередня обробка цих даних шляхом їх нормалізації в діапазоні $[0, 1]$. Тут модель прогнозування витягує закономірності з фактичного робочого навантаження та аналізує ,

попередні n значень робочого навантаження для прогнозування майбутнього робочого навантаження в центрі обробки даних у $n + 1$ момент часу. Потім для навчання нейронної мережі застосовується метод самоадаптивної диференціальної еволюції. Була запропонована модель прогнозування навченої

					БР.ІІІ - 95.00.00.000 ПЗ	Арк.
						48
Змн.	Арк.	№ докум.	Підпис	Дата		

нейронної мережі з двофазним алгоритмом адаптивної диференціальної еволюції (BaDE), яка використовує подвійну адаптацію, а саме: на рівні кросовера під час процесу експлуатації та мутації на фазі дослідження для підвищення ефективності навчання нейронної мережі. Ця робота перевершила [23] за точністю прогнозування. Пізніше, Була розроблена автоадаптивна нейронна мережа, де ваги мережевих з'єднань оптимізовані за допомогою алгоритму триадаптивної диференціальної еволюції (TaDE). Адаптація використовується на рівні кросовера, мутації та генерації контрольних параметрів, що дозволяє покращити навчання еволюційної нейронної мережі.

Багатовхідна та багатовихідна нейронна мережа

Запропоновано модель прогнозування на основі багаторесурсної еволюційної нейронної мережі прямого зв'язку (SISO) шляхом модифікації функціональності існуючої нейронної мережі прямого зв'язку з одним входом та одним виходом [38]. На входному, прихованому та вихідному рівнях використовуються набори нейронних вузлів замість звичайних вузлів для отримання кількох входних даних та прогнозування виходу на основі кількох атрибутів. Вона використовує комбіновані операції класифікації та прогнозування під час процесу навчання вагових зв'язків мережі та одночасно класифікує прогнозовану інформацію кількох атрибутів. Вона покращує ефективність з точки зору точності прогнозування з меншими вимогами до просторової та часової складності порівняно з еквівалентною кількістю нейронних мереж SISO, необхідних для прогнозування такої ж кількості атрибутів.

SVM та нейронна мережа прямого зв'язку

Запропоновано модель прогнозування навантаження хоста на основі зваженого вейвлет-перетворення та методу опорних векторів. Цей підхід поєднав функціональність вейвлет-перетворення та методу опорних векторів, а також присвоїв вагу вибірці, що відображало важливість різних вибірок та покращувало точність прогнозування навантаження. Байєсівський підхід був запропонований для прогнозування навантаження віртуальних машин у центрі обробки даних. Метод використовував прогнози попиту на ресурси для відповідного

					БР.ІІІ - 95.00.00.000 ПЗ	Арк.
						49
Змн.	Арк.	№ докум.	Підпис	Дата		

забезпечення ресурсами. Був розроблений метод прогнозування майбутнього навантаження, заснований на навчанні зворотного поширення помилки на тришаровій нейронній мережі, який забезпечив прогнозування навантаження з прийнятною точністю на трасах веб-журналів NASA HTTP для інтервалу прогнозування до 60 секунд. Застосовано моніторинг історичних даних для прогнозування навантаження сервера в майбутньому, застосовуючи метод машинного навчання, такий як Random Forest. Ця схема перевершила метод аналізу часових рядів з точки зору точності прогнозування навантаження.

Довгострокова короткострокова рекурентна нейронна мережа (LSTM-RNN). Модель довгострокової пам'яті в рекурентній нейронній мережі (LSTM-RNN) для дрібнозернистого прогнозування навантаження хоста. Хоча модель LSTM-RNN вивчає довгострокові залежності та забезпечує високу точність прогнозування навантаження сервера, вона страждає від тривалого часу обчислення під час навчання через використання алгоритму зворотного поширення між рекурентними шарами.

Комплекснозначна нейронна мережа

Запропоновано підхід до прогнозування робочого навантаження, заснований на використанні ресурсів хоста в центрі обробки даних. Для навчання моделі прогнозування вони застосували нейронну мережу на основі комплексних значень, яка є ефективнішою, ніж традиційна нейронна мережа на основі дійсних значень. Нейронна мережа на основі комплексних значень може бути додатково розроблена як квантова версія нейронної мережі, що забезпечить підвищення продуктивності та швидкості для центру обробки даних.

Структура прогнозування ансамблю

Було запропоновано фреймворк прогнозування робочого навантаження CloudInsight, який одночасно використовує інтегрований потенціал кількох предикторів машинного навчання (наприклад, SVM, нейронна мережа, випадковий ліс, лінійна регресія тощо) для точного прогнозування гетерогенного робочого навантаження в хмарному центрі обробки даних. Метод багатокласової регресії використовується для призначення ваг різним предикторам у фреймворку

					БР.ІІІ - 95.00.00.000 ПЗ	Арк.
						50
Змн.	Арк.	№ докум.	Підпис	Дата		

ансамблю-предикторів у режимі реального часу відповідно до поточної точності, досягнутої різними предикторами. Модель ансамблю-предикторів періодично оптимізується для врахування раптових змін майбутнього робочого навантаження. Цей фреймворк прогнозування натхненний тим фактом, що управління еластичними ресурсами неминуче залежить від предикторів робочого навантаження, які вимірюють короткострокові та довгострокові відхилення та коливання гетерогенних хмарних робочих навантажень. Однак, будучи попередньо оптимізованими для певного типу моделей робочого навантаження, вивчених під час навчання за допомогою історичних даних, вони недостатні для оцінки реальних хмарних робочих навантажень, моделі яких не видно, можуть динамічно змінюватися з часом і можуть бути нерегулярними. Отже, ці класичні предиктори часто призводять до надмірного/недостатнього виділення хмарних ресурсів. Експериментальне моделювання показало, що CloudInsight має на 13–27% вищу точність, ніж класичні предиктори на основі машинного навчання. Крім того, він мінімізував період недостатнього/надмірного виділення ресурсів, що сприяє високій оптимізації витрат та низькому рівню порушень SLA.

Квантова нейронна мережа

Запропоновано еволюційну квантову нейронну мережу (EQNN) для прогнозування різнорідних видів хмарних робочих навантажень, включаючи веб, кластерні та високопродуктивні обчислення тощо. EQNN – це інтелектуальна інтеграція специфічних принципів квантових обчислень та еволюційно-алгоритмічно навченої нейронної мережі, яка використовує C-NOT та вентиля обертання як функцію активації в нейронній мережі. Вважається, що квантові бітові значення (тобто кубіти) мають кращий потенціал для дослідження та використання, ніж дійсні числа, що дозволяє інтуїтивно вивчати вилучені шаблони під час процесу навчання. Попередньо оброблені вхідні дані для навчання передаються у вигляді кубітів, згенерованих за допомогою вентиля обертання, а також значення кубітів генеруються для мережевих підключень замість вагових коефіцієнтів у вигляді дійсних чисел. Наведено короткий виклад вищезгаданих підходів до прогнозування робочого навантаження на основі машинного навчання

					БР.ІП - 95.00.00.000 ПЗ	Арк.
						51
Змн.	Арк.	№ докум.	Підпис	Дата		

3.2 Управління еластичними ресурсами та оптимізація витрат

Еластичне управління ресурсами в хмарному середовищі складається з різних взаємопов'язаних операцій, таких як (i) *Призначення завдань*, (ii) *Міграція віртуальних машин*, (iii) *Прогнозування ресурсів* та (iv) *Розподіл віртуальних машин*, як показано на рис. 3.2.

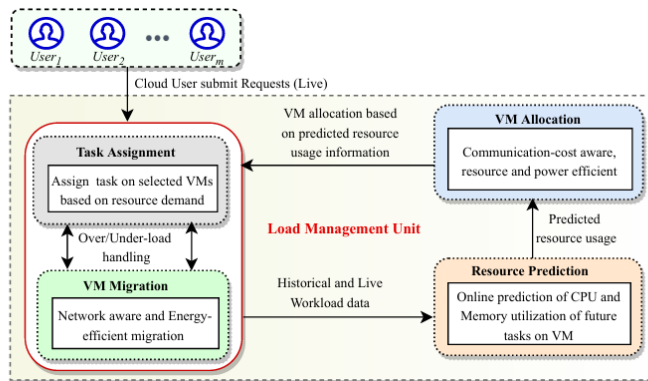


Рисунок 3.2 - Структура управління еластичними ресурсами

Усі ці операції взаємодіють одна з одною одночасно на спільній хмарній платформі, що дозволяє ефективно розподіляти та керувати доступними фізичними ресурсами. Розглянемо m хмарних користувачів, які подали свої запити на виконання в хмарному середовищі (рис. 3.2), де Блок управління навантаженням розподіляє різні запити (тобто програми, розділені на різні

завдання) на вибрані віртуальні машини, виконуючи призначення завдань з урахуванням обмежень ресурсної потужності. Загальні функції, що виконуються різними взаємодіючими блоками щодо управління навантаженням, такі:

- *Призначення завдань*: Завдання призначаються найбільш підходящій віртуальній машині на основі їхніх обчислювальних вимог та вимог до зберігання, що також зменшує втрату ресурсів. Наприклад, завдання з малими вимогами до ресурсів призначаються віртуальній машині малого розміру, тоді як великі завдання – віртуальній машині великого розміру тощо, де розмір означає ємність ресурсів.
- *Прогнозування ресурсів*: Очікуване навантаження на процесор та

					БР.ІП - 95.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		52

пам'ять майбутніми завданнями визначається на основі історичного та поточного виконання завдань відповідною віртуальною машиною. Крім того, інформація про прогнозування кожної віртуальної машини агрегується для попереднього виявлення будь-якого перевантаження/недовантаження сервера з метою попереднього пом'якшення його небажаного впливу на продуктивність усього блоку керування навантаженням.

- *Розподіл віртуальних машин:* Віртуальні машини, завантажені прогнозованим використанням ресурсів, призначаються серверам енергоефективним способом, щоб споживання енергії та витрати на зв'язок між взаємозалежними віртуальними машинами були мінімальними, а використання ресурсів максимальним завдяки застосуванню відповідної стратегії розміщення віртуальних машин.

- *Міграція віртуальних машин:* Для обробки будь-якого перевантаження/недостатнього навантаження (яке суттєво знижує продуктивність центру обробки даних) під час розподілу віртуальних машин, міграція віртуальних машин є найкращим рішенням. Інформація про прогнозоване використання ресурсів використовується для проактивного виявлення будь-якої такої ситуації та її вирішення шляхом застосування відповідної енергоефективної та ефективної з точки зору мережевого трафіку стратегії міграції віртуальних машин, щоб мінімізувати ймовірність зниження продуктивності.

Ефективне призначення та міграція віртуальних машин дозволяє оптимізувати енергоспоживання та мережевий трафік під час управління навантаженням, що додатково оптимізується шляхом застосування відповідного прогнозування ресурсів. Отже, метою інтеграції всіх цих операцій під час управління еластичними ресурсами є збереження оптимізації, досягнутої на кожному кроці, шляхом застосування різних жадібних, евристичних, нечітких підходів, методів машинного навчання та еволюційної оптимізації, і, нарешті, інкапсулює всю оптимізацію енергоспоживання для забезпечення плавної та економічно ефективної роботи в хмарному центрі обробки даних.

Найновіший сучасний стан та порівняння

					БР.ІІІ - 95.00.00.000 ПЗ	Арк.
						53
Змн.	Арк.	№ докум.	Підпис	Дата		

Ієрархічне управління ресурсами на основі SLA:

Представлено управління ресурсами на основі SLA, спеціально розроблене для зменшення порушень SLA та споживання електроенергії, включаючи пристрої охолодження та використання процесора в хмарному центрі обробки даних [39]. Автори вважають, що управління віртуальними машинами (VM) – це проблема розгортання VM на серверах та призначення їм ресурсів, що задовольняє критерії мінімізації загальних експлуатаційних витрат та порушення умов SLA, залежно від доступності ресурсів та обмежень QoS. У цьому підході набір децентралізованих менеджерів рішень генерує обмеження та працює у співпраці для прийняття ієрархічних рішень для управління VM, щоб зменшити експлуатаційні витрати центрів обробки даних. Крім того, вони застосовують періодичну оптимізацію для призначення нових VM та перерозподілу активних VM на фізичні машини відповідно до очікуваного робочого навантаження протягом наступної епохи. Результатом цієї роботи стало скорочення експлуатаційних витрат центру обробки даних на 40% та часу виконання, що в сім разів менше, ніж у централізованій структурі управління для тих самих операцій.

Енергоефективне розміщення віртуальної машини на основі евклідової відстані для багатьох цілей

Запропоновано багатоцільове енергоефективне розміщення віртуальних машин на основі евклідової відстані. Ця робота включає споживання енергії на основі використання процесора, за винятком пристроїв охолодження, багатоцільове розміщення віртуальних машин з мінімальним споживанням енергії та максимальним використанням ресурсів, планування завдань та міграцію віртуальних машин, що зменшує порушення угоди про рівень обслуговування (SLA). Автори запропонували алгоритм HGAPSO, поєднуючи генетичний алгоритм (GA) та оптимізацію рою частинок (PSO) для розподілу віртуальних машин на серверах, щоб мінімізувати втрати ресурсів та порушення угоди про рівень обслуговування (SLA). GA допомагає в міграції віртуальних машин з вихідного сервера на цільовий, але він повільний у збіжності, а PSO допомагає GA у виборі оптимального цільового сервера, дозволяючи розміщувати віртуальні машини з

					БР.ІІІ - 95.00.00.000 ПЗ	Арк.
						54
Змн.	Арк.	№ докум.	Підпис	Дата		

неенергоефективного на енергоефективний сервер. Планування завдань виконується з урахуванням обмежень термінів виконання програми, тим самим зменшуючи ймовірність порушення SLA. Коли завдання завершено, віртуальні машини завершуються, і після перерозподілу починається міграція віртуальних машин на енергоефективну фізичну машину [40]. Ця робота порівнюється з точним алгоритмом на основі гілок та меж, який покращує використання ресурсів та зменшує споживання енергії.

Розподіл ресурсів на основі експоненціально зваженої ковзної середньої

Динамічний розподіл ресурсів та розміщення віртуальних машин, який розподіляє ресурси центру обробки даних залежно від вимог програми та дозволяє енергоефективно використовувати ресурси. Ця робота сприяє уникненню перевантаження та запобіганню втратам ресурсів. По-перше, автори передбачили майбутні потреби в ресурсах, обчислюючи експоненціальну зважену ковзну середню (EWMA) з використанням минулої поведінки віртуальних машин. Цей прогнозований попит на ресурси пізніше використовується для відповідного розподілу віртуальних машин на серверах. Крім того, введено концепцію «асиметрії» для вимірювання нерівномірності або неоднорідності використання ресурсів сервера, що мінімізується для покращення загального використання ресурсів сервера. Щоб уникнути перевантаження серверів, ступінь перевантаження періодично вимірюється, і якщо досягнуто порогу використання ресурсів, віртуальні машини переносяться на менш використовувані сервери. Щоб забезпечити зелені обчислення, недостатньо використані сервери вимикаються після міграції працюючих віртуальних машин на енергоефективні сервери. Результатом є прогнозування навантаження із покращенням загального використання ресурсів, що дозволяє уникнути перевантаження порівняно зі сценарієм без прогнозування навантаження.

Стохастичне балансування навантаження для управління віртуальними машинами:

Представлено стохастичне балансування навантаження для вирішення проблеми перевантаження серверів через багатовимірні та динамічні робочі

					БР.ІІІ - 95.00.00.000 ПЗ	Арк.
						55
Змн.	Арк.	№ докум.	Підпис	Дата		

навантаження для управління віртуальними машинами в центрі обробки даних. Перевантажені сервери призводять до зниження продуктивності, оскільки кожна програма, що виконується на них, витрачає багато часу на відповідь та часті міграції віртуальних машин. По-перше, оцінюється очікуваний попит на ресурси в певному інтервалі прогнозування, а ймовірнісний розподіл помилок прогнозування доповнюється прогнозованим результатом, щоб уникнути порушень SLA, недооцінки та переоцінки потреб у ресурсах, що виникають через помилки під час прогнозування. Крім того, для обробки неправильних міграцій віртуальних машин, що виникають через перевантажені сервери та неефективне використання ресурсів, виявляються гарячі точки та пропонується евристичний алгоритм для визначення міграцій віртуальних машин з гарячої точки на недовикористані сервери та справедливого розподілу віртуальних машин на доступних серверах. Ця робота перевершує детерміновані схеми балансування навантаження, залежно від обробки перевантажених серверів, використання ресурсів та кількості міграцій віртуальних машин. Обмеженням роботи є те, що вони не враховують споживання енергії або аспект енергозбереження під час балансування навантаження, проблеми безпеки, вартість мережевого зв'язку між різними віртуальними машинами користувача, пропускну здатність мережі, необхідну під час міграції віртуальних машин.

Розміщення онлайн-віртуальної машини:

Запропоновано розміщення віртуальних машин онлайн для збільшення доходу постачальника хмарних послуг. Автори зазначають, що порушення SLA та міграція віртуальних машин разом зі споживанням енергії знижують фінансову вигоду постачальника послуг у центрі обробки даних. Щоб вирішити ці проблеми, вони представляють алгоритми першого підходу та гармонійні алгоритми для розміщення віртуальних машин онлайн без урахування міграції віртуальних машин.

Більше того, вони довели, що проблема максимізації загального доходу внаслідок міграції віртуальних машин є NP-складною. Вони розробили алгоритм Least Reliable First (LRF, використовуючи найменшу кількість міграцій) та

					БР.ІІІ - 95.00.00.000 ПЗ	Арк.
						56
Змн.	Арк.	№ докум.	Підпис	Дата		

Decreased Density Greedy (евристичний алгоритм DDG, для дрібнозернистих жадібних міграцій, що збільшують дохід постачальника хмарних послуг) для обробки процесу міграції віртуальних машин. Під час експериментальної оцінки було помічено, що гармонійний алгоритм має більший потенціал для збільшення доходу постачальника хмарних послуг порівняно з алгоритмом First-fit. DDG є доцільним для сценаріїв з високим штрафом за рівень обслуговування (SLA), а LRF підходить для навпаки. Обмеженням є те, що вони ігнорують вартість зв'язку між віртуальними машинами та вимоги до пропускну здатності під час міграції віртуальних машин.

Прогнозування та міграція віртуальних машин для перевантажених хмар:

Запропоновано енергоефективне прогнозування та міграцію віртуальних машин для перевантажених хмар. Перевантаження ресурсів означає розміщення віртуальних машин на серверах з потужністю, що перевищує їх фактичні потреби. Це призводить до непотрібного резервування ресурсів та надмірного енергоспоживання. Крім того, виникає нова проблема, тобто перевантаження материнської плати (PM) та накладні витрати на міграцію віртуальних машин. Для боротьби з цими проблемами на кожній віртуальній машині та материнській платі відповідно розгортаються прогнозувальник ресурсів та прогнозувальник перевантаження, який прогнозує майбутні потреби віртуальної машини в ресурсах, і відповідно віртуальні машини розміщуються на материнських платах з урахуванням обмежень материнської плати щодо ресурсної ємності та перевантаження. Прогнозувальники перевантаження всіх материнських плат підключені до менеджера кластера, який визначає процес міграції віртуальних машин. Для енергоефективності ідея полягає в розгортанні нової віртуальної машини або віртуальних машин, які мають мігрувати на вже запущені материнські плати, що задовольняють обмеження порогу перевантаження, а не в запуску нової материнської плати. Основним обмеженням цієї роботи є розгортання прогнозувальника ресурсів на кожній віртуальній машині та прогнозувальника перевантаження на кожному сервері, оскільки це призводить до непотрібних накладних

					БР.ІІІ - 95.00.00.000 ПЗ	Арк.
						57
Змн.	Арк.	№ докум.	Підпис	Дата		

витрат на обчислення, які споживають додатковий час та простір на кожній фізичній машині.

Безпечне та енергоощадне балансування навантаження (SEA-LB)

Представлено фреймворк безпечного та енергоощадного балансування навантаження (SEA-LB) для розміщення віртуальних машин, в якому вони представили концепцію безпеки шляхом мінімізації атак по бічних каналах разом з економією енергії та ефективним використанням ресурсів під час розподілу РМ між віртуальними машинами. Забезпечення безпеки кожної віртуальної машини здається прийнятним, проте це зменшує використання ресурсів. Насправді, не кожна віртуальна машина містить конфіденційні дані, і тому запобігання SCA не є важливою вимогою кожної віртуальної машини під час балансування навантаження. Крім того, вони проігнорували роль міграції віртуальних машин, використання пропускну здатності та витрат на мережевий зв'язок під час балансування віртуальних машин на РМ.

Багатоцільовий генетичний алгоритм:

Запропоновано багатоцільовий генетичний алгоритм для прогнозування та розподілу ресурсів, що враховує використання процесора та пам'яті віртуальними машинами та програмними моделями (ПМ), а також загальне споживання енергії центром обробки даних. Алгоритм генетичного аналізу (ГА) прогнозує використання ресурсів перед розподілом ПМ між віртуальними машинами. Потім застосовується алгоритм розміщення віртуальних машин для максимізації використання ресурсів та мінімізації споживання енергії. Обмеженням цієї роботи є те, що порушення SLA та міграції віртуальних машин можуть збільшитися через помилки прогнозування.

Багатоцільовий ACO:

Запропоновано багатоцільову оптимізацію мурашиної колонії (ACO) для оптимального розміщення віртуальних машин (VM) та ефективного споживання енергії. Мета полягає в оптимальному розміщенні VM на доступних РМ, щоб мінімізувати втрати ресурсів та споживання енергії. Недоліком ACO є те, що він залежить від кількості феромонів для пошуку оптимального рішення в просторі

					БР.ІП - 95.00.00.000 ПЗ	Арк.
						58
Змн.	Арк.	№ докум.	Підпис	Дата		

пошуку, і він непридатний для рекурсивного покращення використання ресурсів.

Розміщення віртуальної машини з керованим профілем

Нещодавно з трифазну структуру для розміщення віртуальних машин на основі профілювання програм, класифікації завдань та призначення програм. Однак, вона ігнорувала обробку міграції віртуальних машин під час ситуацій перевантаження/недостатнього навантаження.

Розміщення віртуальної машини з урахуванням трафіку

Зосереджено увагу на мінімізації мережевого трафіку та вивчали вплив мережевих ресурсів під час оптимізації міграції віртуальних машин на хост-машинах. Запропоновано дворівневий апроксимаційний алгоритм для вирішення проблеми розміщення віртуальних машин з урахуванням трафіку, що призвело до збільшення пропускної здатності та зниження вартості зв'язку.

Безпечне та багатоцільове розміщення віртуальних машин

Запропоновано генетичний алгоритм оптимізації китів (WOGA) для безпечного та багатоцільового розміщення віртуальних машин (тобто SM-VMP) з багатоефективною міграцією віртуальних машин. Ця робота зосереджена на мінімізації загроз безпеці, витрат на зв'язок, споживання енергії та одночасному покращенні використання ресурсів. Було представлено нову систему кодування-декодування на основі ймовірностей, яка кодує розподіл віртуальних машин як вектор позиції кита, враховуючи гетерогенне середовище віртуальних машин. Оцінка продуктивності та порівняння цієї роботи з різними існуючими багатоцільовими VMP, включаючи GA, оптимізацію китів, GA на основі недомінування (NSGA-II), PSO, гібрид GA та PSO, показує її перевагу в ефективному розподілі віртуальних машин.

Багатоцільове балансування навантаження на основі онлайн-прогнозування

Представлено енергоефективну та онлайн-прогнозовану систему багатоцільового балансування навантаження (OP-MLB), де розроблено та використано блок прогнозування на основі еволюційної нейронної мережі для оцінки майбутнього використання ресурсів віртуальних машин та проактивної оцінки

					БР.ІП - 95.00.00.000 ПЗ	Арк.
						59
Змн.	Арк.	№ докум.	Підпис	Дата		

перевантаження/недовантаження серверів. Для багатоцільового розміщення віртуальних машин застосовується генетичний алгоритм сортування на основі не-домінування. Крім того, запропоновано енергоефективний та мережево-ефективний алгоритм міграції віртуальних машин для мінімізації споживання енергії в центрі обробки даних. Він дозволяє економити енергію, виключаючи неактивні сервери та мінімізуючи кількість активних серверів, зменшуючи міграції віртуальних машин та максимізуючи використання ресурсів.

Проактивне автомасштабування віртуальних машин

Для задоволення динамічних потреб майбутніх застосунків розроблено енергоефективну структуру розподілу ресурсів. Ця структура вирішує такі проблеми, як марнування ресурсів, погіршення продуктивності та якості обслуговування (QoS), шляхом порівняння прогнозованих потреб застосунку в ресурсах з ресурсомісткістю віртуальних машин та консолідації всього навантаження на мінімальну кількість серверів. Для прогнозування численних потреб у ресурсах розроблено онлайн-нейронну мережу прямого зв'язку з кількома ресурсами (OM-FNN), і прогнозовані віртуальні машини розміщуються на енергоефективних серверах. Цей інтегрований підхід оптимізував використання ресурсів та споживання енергії.

Комунікація Економія коштів Ресурсоефективне балансування навантаження

Представлено фреймворк балансування навантаження з урахуванням витрат на зв'язок та ресурсоефективного балансування навантаження (CARELB) для зниження витрат на зв'язок між віртуальними машинами в центрі обробки даних, мінімізації споживання енергії та максимізації використання ресурсів. Цей фреймворк розподіляє віртуальні машини з високою спорідненістю та взаємозалежністю фізично ближче одна до одної. Для оптимізованого розміщення віртуальних машин запропоновано оптимізацію рою частинок та генетичний алгоритм сортування без домінування, тобто алгоритм PSOGA, де розподіли віртуальних машин кодуються як частинки, так і хромосоми. Експериментальний аналіз показує, що фреймворк CARELB досяг стабільної оптимізації споживання

					БР.ІІІ - 95.00.00.000 ІІЗ	Арк.
						60
Змн.	Арк.	№ докум.	Підпис	Дата		

енергії та використання ресурсів під час розміщення віртуальних машин.

Динамічний розподіл ресурсів, вбудований у безпеку

Щоб зменшити ризики безпеки під час розміщення віртуальних машин та спільного використання фізичних ресурсів з кількома користувачами, запропоновано модель динамічного розподілу ресурсів із вбудованою системою безпеки (SEDRA). Ця модель використовує аналізатор для ідентифікації спільно розташованих зв'язків між віртуальними машинами та виявлення сервера зі шкідливим гіпервізором для зменшення порушень безпеки. Детектор загроз віртуальних машин на основі випадкових дерев та передбачач робочого навантаження працюють у фоновому режимі, щоб забезпечити безпечний та ефективний розподіл фізичних ресурсів між віртуальними машинами. SEDRA захищає від порушень безпеки, спричинених відомими віртуальними машинами користувачів, та суттєво зменшує загрози спільного розміщення, спричинені невідомими віртуальними машинами. Крім того, ця модель удосконалена для забезпечення безпеки під час надходження навантаження до центру обробки даних та виявлення будь-яких конфліктів до їх виникнення.

У наведеному узагальненні порівняння пов'язаних робіт, обговорених вище, з урахуванням мети відповідної запропонованої роботи, використаного підходу, сильних та слабких сторін роботи.

3.3 Оцінка ефективності та аналіз результатів прогнозування

У цьому підрозділі візуалізація даних спрощує складну інформацію, перетворюючи її на візуальні зображення, допомагаючи нам краще розуміти дані. На рисунку 3.3, графік використання процесора на основі набору даних BitBrains, представлено часовий ряд зображень використання процесора протягом певного періоду. Графік демонструє часові варіації, що відображають відсоток використання процесора протягом різних інтервалів.

					БР.ІІІ - 95.00.00.000 ПЗ	Арк.
						61
Змн.	Арк.	№ докум.	Підпис	Дата		

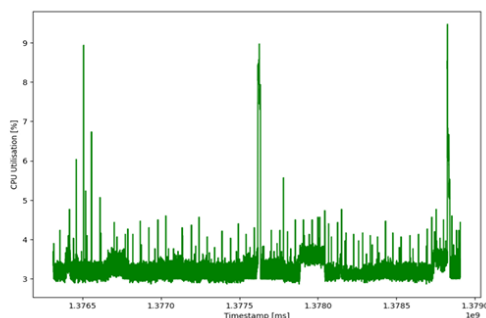


Рисунок 3.3 - Використання процесора набором даних BitBrains

На рисунку 3.4 показано пропускну здатність мережі в хмарному середовищі, демонструючи динамічні зміни та варіації швидкості передачі даних мережі протягом заданого періоду часу. Графік, який є візуальним

представленням фактичної статистики пропускну здатності мережі, надає уявлення про загальну продуктивність мережі та моделі передачі даних. Ці дані є критично важливими для розуміння ефективності мережі та використання потужності.

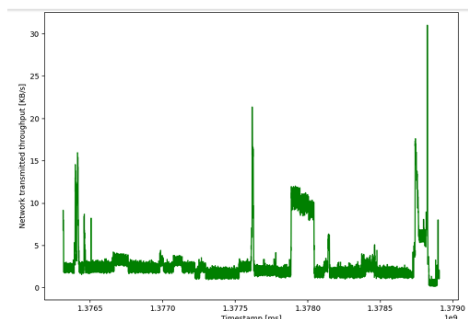


Рисунок 3.4 - Пропускна здатність мережі передачі даних BitBrains

На рисунку 3.5 представлено вичерпне зображення тенденцій використання процесора протягом заданого періоду, демонструючи найвищі піки, що позначають максимальне використання процесора, найнижчі мінімуми, що представляють мінімальне використання процесора, та траєкторію середнього використання процесора. Ці показники разом пропонують вичерпний огляд продуктивності процесора в середовищі Azure.

					БР.ІП - 95.00.00.000 ПЗ	Арк.
						62
Змн.	Арк.	№ докум.	Підпис	Дата		

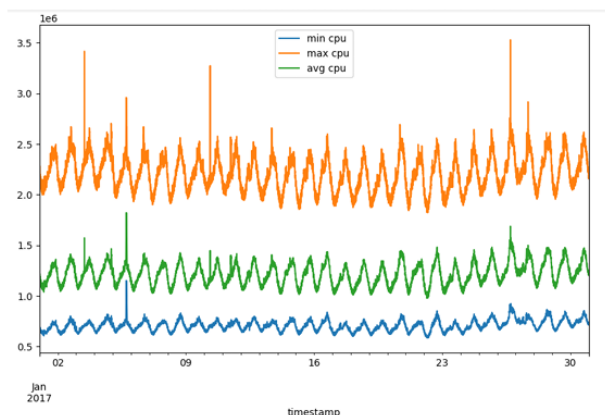


Рисунок 3.5 - Використання процесора набору даних Microsoft Azure

Оцінка використання ресурсів для моделей машинного та глибокого навчання

Таблиця 3.1 Для набору даних BitBrains - Порівняння алгоритмів машинного навчання та глибокого навчання для прогнозування використання процесора

Модель	MSE (Середньоквадратична помилка)	MAE (Середня абсолютна помилка)	RMSE (Корінь із середньоквадратичної помилки)	R2 score (Коефіцієнт детермінації)
LiR (Лінійна регресія)	0.0027	0.0215	0.0521	0.7794
DTR (Дерево рішень)	0.0099	0.0402	0.0995	0.1951
GBR (Гradientний бустинг)	0.0055	0.0294	0.0747	0.5465
SVR (Метод опорних векторів)	0.0030	0.0389	0.0556	0.7488
LSTM (Довга короткострокова пам'ять)	0.0026	0.0233	0.0515	0.7843
BiLSTM (Двонаправлена LSTM)	0.0024	0.0224	0.0490	0.8042

У цій роботі ми провели впровадження, використовуючи два загальнодоступні набори даних, а саме BitBrains та Microsoft Azure Traces 2017. Ми оцінили використання процесора та пропускну здатність мережі за допомогою набору даних BitBrains, а також оцінили використання процесора для набору даних Microsoft Azure.

Оцінка використання процесора

Оцінені результати використання процесора з використанням набору даних BitBrains детально представлені в таблиці 3.1 та на рисунку 3.6.

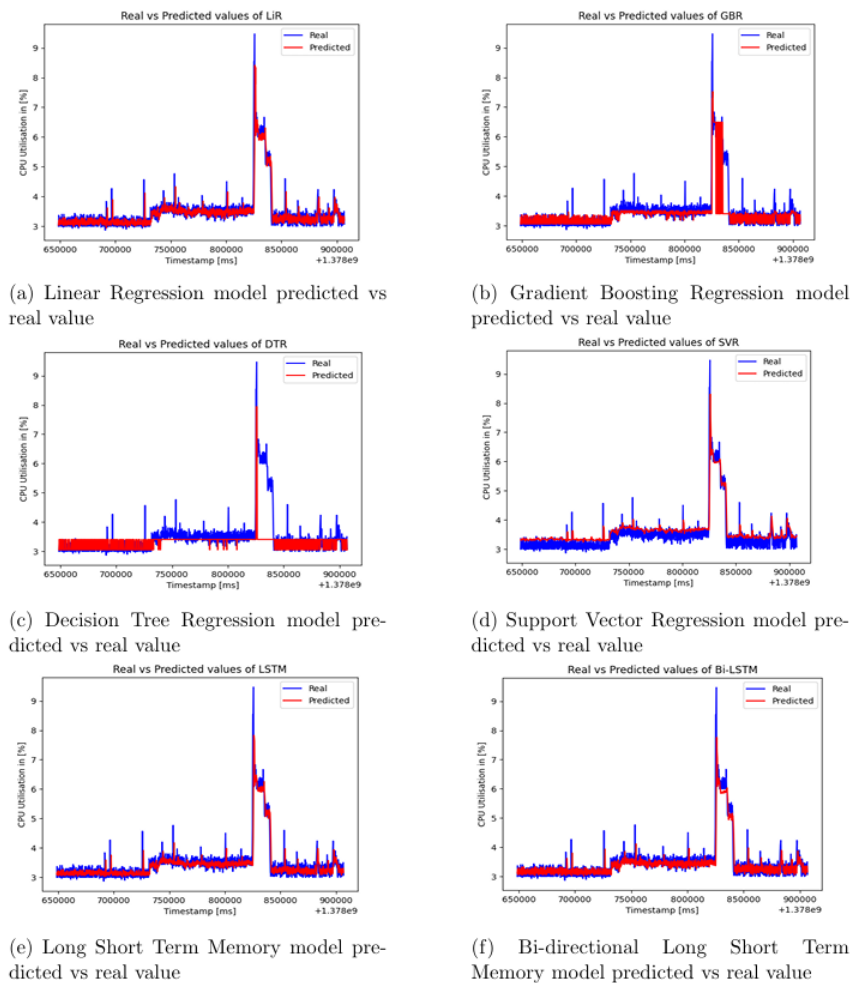


Рисунок 3.6 - Прогнозування використання процесора за допомогою набору даних BitBrains

Також для набору даних MicroSoft Azure оцінені результати використання процесора детально представлені в таблиці 3.2 та на рисунку 3.7. Метрики MSE, MAE, RMSE та R2 алгоритмів ML та DL, порівняних у цій роботі, наведено в таблиці 3.1 та таблиці 3.2.

Результати показують, що для алгоритмів RMSE, регресії дерева рішень, градієнтної регресії з підвищенням частоти мали вищі значення помилок порівняно з моделлю BiLSTM, яка працювала краще, ніж порівнювані моделі, показані в таблиці 3.1 та таблиці 3.2. Оцінені результати використання процесора для

прогнозованих та фактичних значень моделей машинного навчання представлені на рисунку 3.6 та рисунку 3.7. Отже, BiLSTM працював краще, ніж порівнювані підходи, а також лінійна регресія та LSTM.

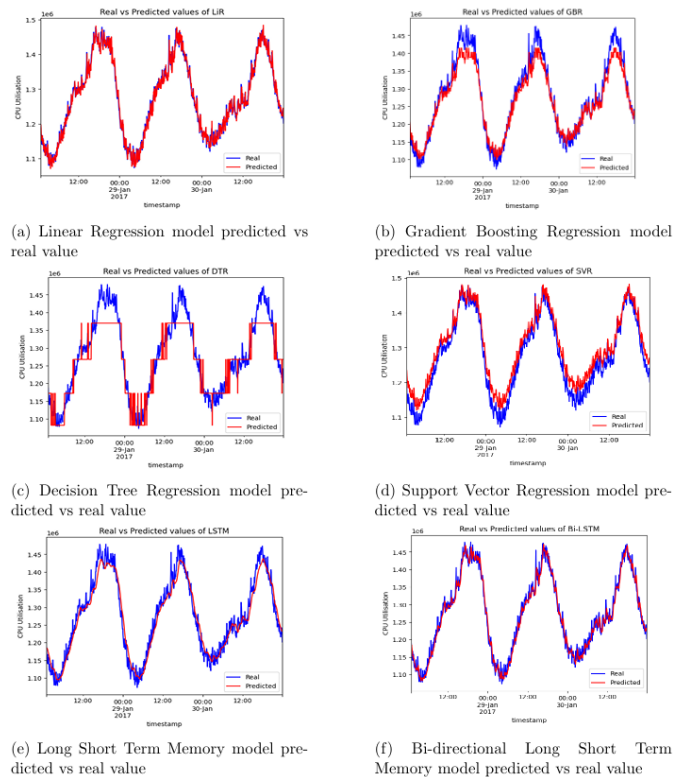


Рисунок 3.7 - Прогнозування використання процесора з використанням набору даних Microsoft Azure

Таблиця 3.2 Порівняння алгоритмів машинного навчання та глибокого навчання для прогнозування використання процесора

Модель	MSE (Середньоквадратична помилка)	MAE (Середня абсолютна помилка)	RMSE (Корінь із середньоквадратичної помилки)	R2 score (Коефіцієнт детермінації)
LiR (Лінійна регресія)	0.0002	0.0131	0.0169	0.9833
DTR (Дерево рішень)	0.0023	0.0390	0.0480	0.8661
GBR (Градiєнтний бустинг)	0.0010	0.0255	0.0321	0.9399
SVR (Метод опорних векторів)	0.0015	0.0337	0.0388	0.9127
LSTM (Довга короткострокова пам'ять)	0.0009	0.0239	0.0304	0.9462
BiLSTM (Двонаправлена LSTM)	0.0004	0.0169	0.0214	0.9732

Для набору даних Microsoft Azure

На рисунках 3.6 та 3.7 показано прогнози використання процесора за допомогою низки методів машинного та глибокого навчання, включаючи лінійну регресію (LIR), градієнтну регресію з підвищуючим навантаженням (GBR), регресію дерева рішень (DTR), регресію опорних векторів (SVR), довгострокову пам'ять (LSTM) та двонаправлену довгострокову пам'ять (BiLSTM). Для набору даних BitBrains див. табл. 3.3.

Таблиця 3.3

Порівняння алгоритмів машинного навчання та глибокого навчання для прогнозування пропускної здатності мережі

Модель	MSE Середньоквадратична помилка	MAE Середня абсолютна помилка	RMSE Корінь із середньоквадратичної помилки	R2 score Коefіцієнт детермінації
LIR (Лінійна регресія)	0.0012	0.0114	0.0359	0.9256
DTR (Дерево рішень)	0.0043	0.0473	0.0656	0.752
GBR (Градiєнтний бустинг)	0.00183	0.0259	0.0428	0.894
SVR (Метод опорних векторів)	0.0037	0.0495	0.0610	0.786
LSTM (Довга короткострокова пам'ять)	0.0026	0.0232	0.0513	0.848
BiLSTM (Двонаправлена LSTM)	0.00172	0.0203	0.0415	0.901

Ці прогнозні моделі дозволяють точно прогнозувати використання процесора, надаючи цінну інформацію про потреби в ресурсах та моделі використання в хмарному середовищі.

На рисунку 3.8 представлено прогнози щодо пропускної здатності мережі з використанням моделей машинного та глибокого навчання, включаючи лінійну регресію (LIR), градієнтну регресію з підвищуючим навантаженням (GBR), регресію дерева рішень (DTR), регресію опорних векторів (SVR), довгострокову пам'ять (LSTM) та двонаправлену довгострокову пам'ять (BiLSTM). Ці прогнози

					БР.ІІІ - 95.00.00.000 ПЗ	Арк.
						66
Змн.	Арк.	№ докум.	Підпис	Дата		

пропонують цінну інформацію про очікувані тенденції та закономірності пропускної здатності мережі в хмарних середовищах, допомагаючи в проактивному управлінні та оптимізації мережевих ресурсів.

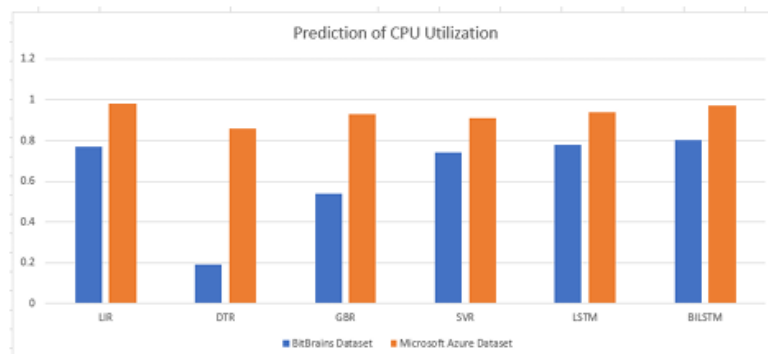


Рисунок 3.8 - Порівняння прогнозування використання процесора

Оцінка пропускної здатності мережі

Оцінені результати щодо пропускної здатності мережі представлені в таблиці 3.3 та на рисунку 3.8. З результатів видно, що BiLSTM має дуже близькі значення порівняно з фактичним значенням. BiLSTM досяг вищої точності прогнозування пропускної здатності мережі, ніж порівнювані моделі, з 0,9 для метрик R2 та нижчими коефіцієнтами помилок для RMSE та MAE. Рисунок 13а показує ті ж реальні та прогнозовані значення, оскільки точність лінійної регресії з 0,92 для метрик R2 вища, ніж у всіх алгоритмів для пропускної здатності мережі. Як BiLSTM, так і лінійна регресія дають майже однакові результати. BiLSTM має переваги над лінійною регресією, оскільки може працювати зі складними часовими закономірностями в майбутньому. Крім того, він краще працює там, де дані можуть демонструвати нелінійні закономірності.

Порівняльний аналіз наборів даних, що використовуються для прогнозування використання процесора

Під час порівняння моделей для прогнозування використання ЦП з використанням наборів даних BitBrains та Microsoft Azure, модель BiLSTM послідовно виділялася як найточніша (рис. 3.8). Порівняно з моделями лінійної регресії,

LSTM, SVR, GBR та DTR, BiLSTM послідовно демонструвала вищу продуктивність в обох наборах даних. Її сила у врахуванні складних часових залежностей дозволила точніше прогнозувати динаміку використання ЦП.

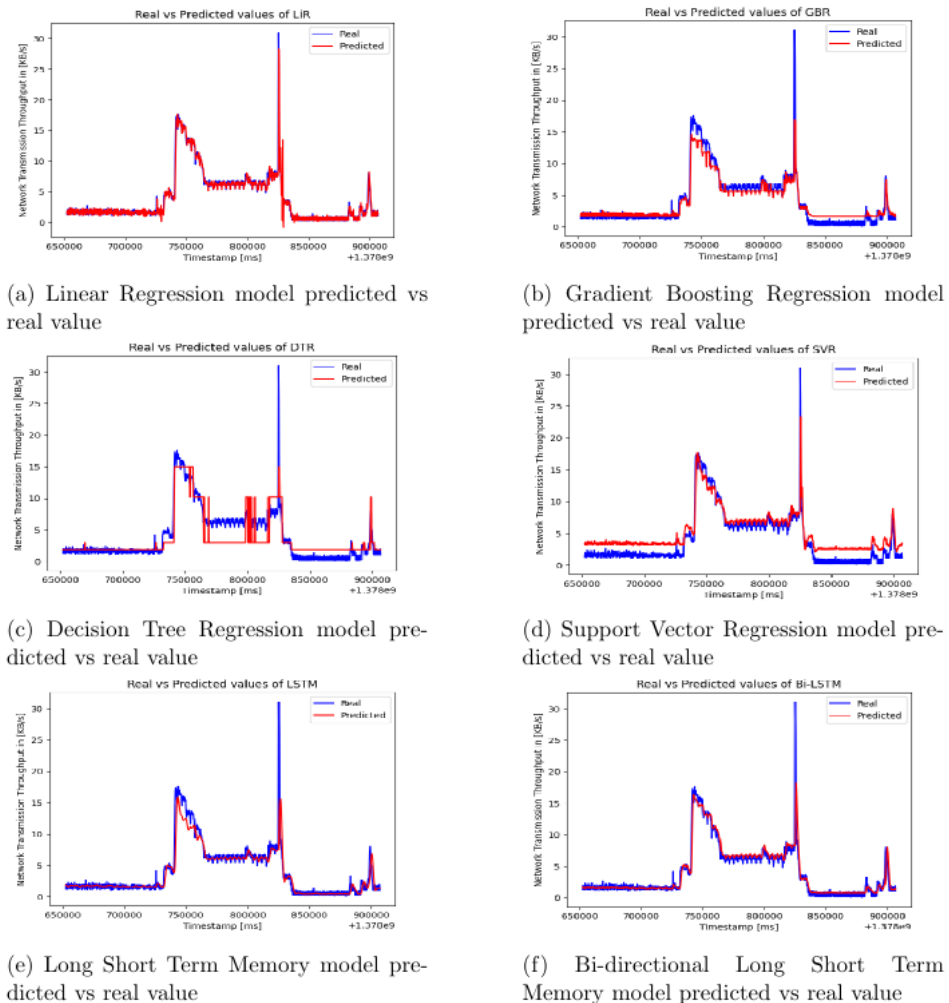


Рисунок 3.9 - Прогнозування пропускнуєї здатності мережі за допомогою набору даних BitBrains

Для набору даних BitBrains значення продуктивності досить відрізняється: GBR: 0,5465 та LSTM: 0,7843. Те саме для набору даних Azure: GBR: 0,9399 та LSTM: 0,9462. Існує незначна різниця в оцінці R2. Як GBR, так і LSTM прогнозують досить схожу оцінку R2 для використання ЦП. Навіть попри це, різниця між ними становить лише близько 1%. У колекції немає складних закономірностей, таких як (позначка часу, використання ЦП). В результаті, обидва алгоритми можуть працювати однаково, фіксуючи закономірності подібним

					БР.ІІІ - 95.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		68

чином. Також важливо зазначити, що результати для тих самих алгоритмів на наборі даних BitBrains значно відрізняються. Хоча інші моделі демонстрували багатообіцяючі результати різною мірою, жодна з них не досягла такої ж стійкості, як BiLSTM, в обробці складнощів у цих наборах даних. Це підкреслює ключову роль архітектури моделі в ефективному прогнозуванні використання процесора в різних наборах даних.

У цьому підрозділі викладено методології, що використовуються для розгортання чотирьох алгоритмів машинного навчання та двох алгоритмів глибокого навчання, а також демонстровано їхні відповідні результати. Комплексний аналіз підкреслює точність та надійність усього процесу, особливо виділяючи модель двонаправленої довгострокової пам'яті (BiLSTM). Примітно, що модель BiLSTM продемонструвала чудову продуктивність, про що свідчить її найближче до одиниці значення R-квадрату та найнижча середньоквадратична помилка (RMSE) серед усіх оцінених алгоритмів машинного навчання. Цей підхід передбачає програмування на Python, виконане в Google Collaboration Notebook, для прогнозування використання ресурсів. Використання ресурсів Набори даних BitBrains та Microsoft Azure, що охоплюють використання процесора, пропускну здатність мережі та відповідні позначки часу, дозволяють застосовувати методи машинного навчання, такі як лінійна регресія, регресія дерева рішень, регресія градієнтного підсилення, регресія опорних векторів та моделі глибокого навчання, включаючи довгу короткочасну пам'ять та двонаправлену довгу короткочасну пам'ять.

Це дослідження провело комплексне дослідження прогнозування використання ресурсів у рамках хмарних обчислень за допомогою різноманітних моделей машинного та глибокого навчання. Програмування на Python у Google Colab використовувалося разом із наборами даних BitBrains та Microsoft Azure, охоплюючи критичні показники, такі як використання процесора, пропускну здатність мережі та часові позначки. Результати дослідження чітко підкреслили ефективність моделі двонаправленої довгострокової пам'яті (BiLSTM), яка перевершує інші алгоритми машинного навчання за точністю та продуктивністю.

					БР.ІІІ - 95.00.00.000 ПЗ	Арк.
						69
Змн.	Арк.	№ докум.	Підпис	Дата		

Досягнуті значення R-квадрату та показники середньоквадратичної помилки (RMSE) підкреслюють виняткові прогностичні здібності моделі BiLSTM у прогнозуванні використання ресурсів, пропонуючи ключові висновки для оптимізації ефективності хмарних обчислень.

На основі цього дослідження існує низка цікавих напрямків для подальшого вивчення. Точність прогнозування можна ще більше підвищити, дослідивши стратегії ансамблевого навчання для інтеграції різних моделей. Більш глибоке розуміння моделей використання ресурсів можна отримати, розширивши охоплення набору даних за межі BitBrains та Microsoft Azure. Подальші дослідження інших аспектів даних у реальному часі можуть покращити точність прогнозування; проте покращення інтерпретованості моделей все ще є вирішальним кроком до глибшого розуміння.

У світлі існуючих підходів до управління еластичними ресурсами, для вдосконалення та розвитку більш перспективного та впливового управління еластичними ресурсами в хмарному середовищі визначено такі проблеми:

- *Підвищення точності прогнозування робочого навантаження:* Точність прогнозування ресурсів віртуальної машини та прогнозування робочого навантаження необхідно підвищити, щоб уникнути порушень угоди про рівень обслуговування (SLA) та забезпечити безконфліктне виконання користувацьких програм. Точна оцінка потреб у ресурсах у майбутньому є неминучою для забезпечення надійних та високодоступних хмарних сервісів під час ефективного управління ресурсами. Існуючі підходи є статичними, тобто працюють лише з історичним робочим навантаженням.
- *Перевантаження та погіршення продуктивності :* під час пікового навантаження сукупний попит на ресурси віртуальних машин перевищує доступну ресурсну потужність сервера, що призводить до перевантаження серверів, що спричиняє погіршення продуктивності та порушення SLA. Наприклад, деякі віртуальні машини можуть аварійно завершувати роботу, ресурси можуть бути довше недоступні, час відгуку може збільшитися тощо.

					БР.ІІІ - 95.00.00.000 ПЗ	Арк.
						70
Змн.	Арк.	№ докум.	Підпис	Дата		

- *Міграція віртуальних машин у реальному часі та час простою:* Щоб керувати динамічними та випадковими вимогами до надлишкових ресурсів або обробляти перевантаження, віртуальні машини мігрують у реальному часі з перевантаженого сервера на інший сервер з достатньою ресурсною потужністю, що призводить до затримки виконання. Ще однією важливою проблемою є вирішення питання про те, яку віртуальну машину слід мігрувати, коли мігрувати та на який сервер її слід мігрувати?

- *Розгляд одного ресурсу під час балансування навантаження:* більшість існуючих стратегій еластичного розподілу ресурсів враховують один ресурс, тобто лише процесор, тоді як у реальному світі управління хмарними ресурсами всі ресурси є значними, оскільки конкуренція за будь-який з ресурсів: пам'ять, пропускна здатність, використання дискового простору, процесор може стати вузьким місцем, що знижує загальну продуктивність системи.

- *Взаємодія та співпраця між пов'язаними операціями:* індивідуальний розгляд прогнозування робочого навантаження, виділення ресурсів, розміщення та міграції віртуальних машин, хоча всі ці операції служать спільній меті.

Тому всі пов'язані операції з управління ресурсами повинні використовуватися на єдиній платформі, щоб забезпечити взаємодію між ними для розробки прагматичного рішення для управління ресурсами в хмарному середовищі.

- *Відсутність безпеки під час розподілу ресурсів :* оскільки безпека даних є головною проблемою кожного користувача, розподіл фізичних ресурсів між віртуальними машинами різних користувачів повинен враховувати захист даних як основну перспективу разом з обмеженнями ресурсної потужності.

- Для заповнення прогалів у дослідженнях та вирішення вищезазначених проблем розроблені такі напрямки досліджень:

- Необхідно розробити вдосконалений метод прогнозування робочого навантаження з високою стабільністю продуктивності. Предиктор повинен динамічно навчатися в режимі реального часу, навчаючи та перенавчаючи себе на

основі історичних та реальних зразків використання ресурсів різними віртуальними машинами та серверами.

- Потрібен ефективний метод прогнозування недостатнього/перевантаження, який може проактивно оцінювати умови пікового навантаження та завантаження з високою точністю, що попереджатиме проактивну міграцію віртуальних машин до фактичного виникнення недостатнього/перевантаження та мінімізуватиме зниження продуктивності.
- Оптимізувати використання кількох ресурсів, включаючи процесор, пам'ять, пропускну здатність, одночасно під час балансування навантаження для безконфліктного виконання користувацьких програм, одночасно економлячи енергію та зменшуючи втрати ресурсів у хмарному середовищі.
- Інтегрувати прогнозування робочого навантаження з ефективним підходом до балансування навантаження або моделлю використання ресурсів на єдиній платформі для підвищення ефективності управління ресурсами.
- Керувати мережевим трафіком, одночасно вирішуючи проблему перевантаження ресурсів під час розподілу ресурсів та ефективного балансування навантаження мережі.
- Розробка безпечних моделей розподілу ресурсів для хмарного центру обробки даних, що забезпечують безпечне виконання робочого навантаження користувачів, безпечний зв'язок між взаємозалежними віртуальними машинами та запобігають несанкціонованому доступу.

Прогнозування використання хмарних ресурсів є важливим питанням для подолання невизначеності в середовищах хмарних обчислень. У хмарних обчисленнях ресурси розподіляються між користувацькими програмами, доступними з будь-якого місця через Інтернет. Для обробки великої кількості користувачів ресурси необхідно динамічно масштабувати для ефективного використання, зниження споживання енергії та витрат з кращою якістю обслуговування (QoS). Метою цієї дослідницької роботи є вивчення ефективності нейронних мереж для прогнозування використання багатьох ресурсів. Запропонована модель використовує гібридний GA-PSO для навчання ваг мережі та FLNN для прогнозування.

					БР.ІІІ - 95.00.00.000 ПЗ	Арк.
						72
Змн.	Арк.	№ докум.	Підпис	Дата		

Запропонована модель перевірена порівняльними експериментальними результатами. Результати показують, що запропонована гібридна модель забезпечує кращу точність порівняно з традиційними методами.

Потенційними напрямками майбутніх досліджень може бути подальша оцінка предикторів нейронних мереж в інших галузях хмарних обчислень, таких як прогнозування інших ресурсів, таких як використання дискового простору, економічна ефективність, мережа та зниження споживання енергії для зелених обчислень. Запропонована структура оцінюється за допомогою набору даних Google trace для використання пам'яті та процесора. Було б конструктивно підтримати запропонований еволюційний підхід нейронних мереж шляхом подальшої роботи з іншими багатовимірними наборами даних про використання ресурсів.

3.4 Висновки по розділу

У третьому розділі було реалізовано систему прогнозування витрат на хмарні ресурси та проведено оцінювання її ефективності. Розглянуто процес прогнозування робочого навантаження, механізми управління еластичними ресурсами та методи оптимізації витрат у хмарному середовищі. Проведений аналіз результатів підтвердив ефективність використання моделей машинного навчання для підвищення точності прогнозування та оптимізації використання ресурсів. Отримані результати демонструють перспективність застосування інтелектуальних методів для автоматизації управління витратами у хмарних системах.

					БР.ІІІ - 95.00.00.000 ІІЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		73

ВИСНОВКИ

У ході виконання бакалаврської роботи було досліджено та реалізовано підходи до прогнозування витрат на хмарні ресурси за допомогою методів машинного навчання, що є комплексним процесом, спрямованим на підвищення ефективності використання хмарної інфраструктури та оптимізацію фінансових витрат. У роботі проаналізовано сучасні підходи до управління хмарними ресурсами, досліджено переваги та недоліки різних моделей прогнозування, а також розглянуто можливості використання алгоритмів машинного навчання для аналізу навантаження та оцінки майбутніх витрат. Процес охопив аналіз вимог, моделювання системи прогнозування, реалізацію алгоритмів обробки даних, тестування моделей та оцінювання ефективності запропонованого рішення, що дозволило створити цілісну систему підтримки прийняття рішень у хмарних середовищах.

Ми розпочали з аналізу предметної області, визначивши функціональні та нефункціональні вимоги до системи прогнозування витрат на хмарні ресурси. Особливу увагу було приділено питанням масштабованості, точності прогнозування, швидкості обробки даних та можливості роботи з великими обсягами інформації. Аналіз існуючих рішень дозволив виявити основні проблеми сучасних систем управління витратами, серед яких складність прогнозування змінного навантаження, недостатня адаптивність до динамічних середовищ та висока вартість надлишкового використання ресурсів. Для структурування процесів були розроблені діаграми та моделі взаємодії компонентів системи, що стали основою подальшої реалізації.

У процесі дослідження було розглянуто основні концепції хмарних обчислень, моделі надання сервісів та методи управління ресурсами. Значну увагу приділено дослідженню алгоритмів машинного навчання для прогнозування витрат і навантаження, зокрема методам регресійного аналізу, часових рядів та алгоритмам класифікації. Використання моделей прогнозування дозволило аналізувати історичні дані використання ресурсів і формувати прогнози щодо

					БР.ІІІ - 95.00.00.000 ПЗ	Арк.
						74
Змн.	Арк.	№ докум.	Підпис	Дата		

майбутніх витрат, що є важливим для автоматизації процесів керування інфраструктурою.

Реалізація системи охопила створення механізмів збору та аналізу даних про використання хмарних ресурсів, модулів прогнозування навантаження та підсистеми управління еластичними ресурсами. Було забезпечено взаємодію між компонентами системи, що дозволило автоматизувати процес оцінювання використання обчислювальних потужностей та формування прогнозів витрат. Особлива увага приділялася точності роботи моделей, оптимізації продуктивності системи та можливості її адаптації до різних типів хмарної інфраструктури.

Ми провели тестування розробленого рішення, включаючи перевірку точності прогнозування, продуктивності обробки даних та ефективності управління ресурсами. У ході тестування було виявлено потенційні проблеми, пов'язані з нестабільністю навантаження та складністю прогнозування у випадку різких змін використання сервісів. Для покращення результатів були застосовані методи оптимізації моделей і додаткової обробки даних, що дозволило підвищити точність прогнозування та стабільність роботи системи.

Загалом, результати роботи підтверджують, що використання методів машинного навчання для прогнозування витрат на хмарні ресурси дозволяє підвищити ефективність управління інфраструктурою, оптимізувати використання обчислювальних ресурсів та знизити фінансові витрати на обслуговування хмарних сервісів. Перевагами запропонованого підходу є можливість автоматичного аналізу великих обсягів даних, адаптивність до змін навантаження та підтримка масштабування системи. Водночас існують певні обмеження, зокрема залежність точності прогнозування від якості вхідних даних і необхідність додаткової оптимізації моделей для складних сценаріїв використання. У майбутньому система може бути розширена шляхом інтеграції з платформами моніторингу хмарної інфраструктури, використання більш складних алгоритмів штучного інтелекту та підтримки автоматизованого прийняття рішень щодо розподілу ресурсів у реальному часі.

					БР.ІІІ - 95.00.00.000 ПЗ	Арк.
						75
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ПОСИЛАНЬ НА ДЖЕРЕЛА

1. Ian Foster, & Carl Kesselman. (2004). The Grid 2: Blueprint for a New Computing Infrastructure. elsevier.com. <https://www.elsevier.com/books/the-grid-2/foster/978-1-55860-933-4>
2. Han, J., Kamber, M., & Pei, J. (2022). Data Mining: Concepts and Techniques. morgankaufmann.com. <https://www.sciencedirect.com/book/9780128117606/data-mining>
3. Géron, A. (2022). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. oreilly.com. <https://www.oreilly.com/library/view/hands-on-machine-learning/9781098125967/>
4. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. deeplearningbook.org. <https://www.deeplearningbook.org/>
5. Amazon Web Services. (2025). AWS Cost Management Documentation. docs.aws.amazon.com. <https://docs.aws.amazon.com/cost-management/>
6. Microsoft Azure. (2025). Azure Cost Management and Billing Documentation. learn.microsoft.com. <https://learn.microsoft.com/en-us/azure/cost-management-billing/>
7. Google Cloud. (2025). Cloud Billing Documentation. cloud.google.com. <https://cloud.google.com/billing/docs>
8. IBM Cloud Docs. (2025). Cloud Monitoring and Resource Optimization. cloud.ibm.com. <https://cloud.ibm.com/docs>
9. Kubernetes Documentation. (2025). Resource Management for Pods and Containers. kubernetes.io. <https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/>
10. Docker Documentation. (2025). Docker Resource Constraints. docs.docker.com. https://docs.docker.com/config/containers/resource_constraints/
11. TensorFlow Documentation. (2025). Machine Learning Framework. tensorflow.org. <https://www.tensorflow.org/>

					БР.ІІІ - 95.00.00.000 ІІЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		76

12. PyTorch Documentation. (2025). Machine Learning Library. pytorch.org.
<https://pytorch.org/docs/stable/index.html>
13. Scikit-learn Documentation. (2025). Machine Learning in Python. scikit-learn.org. <https://scikit-learn.org/stable/>
14. Brownlee, J. (2021). Machine Learning Mastery with Python. machinelearningmastery.com. <https://machinelearningmastery.com/start-here/>
15. Chollet, F. (2021). Deep Learning with Python. manning.com.
<https://www.manning.com/books/deep-learning-with-python-second-edition>
16. Zhang, Q., Cheng, L., & Boutaba, R. (2010). Cloud Computing: State-of-the-Art and Research Challenges. Journal of Internet Services and Applications, 1(1), 7–18. <https://doi.org/10.1007/s13174-010-0007-6>
17. Armbrust, M., et al. (2010). A View of Cloud Computing. Communications of the ACM, 53(4), 50–58.
<https://doi.org/10.1145/1721654.1721672>
18. Buyya, R., Broberg, J., & Goscinski, A. (2011). Cloud Computing: Principles and Paradigms. wiley.com. <https://www.wiley.com/en-us/Cloud+Computing%3A+Principles+and+Paradigms-p-9780470887998>
19. Dean, J., & Ghemawat, S. (2008). MapReduce: Simplified Data Processing on Large Clusters. cacm.acm.org.
<https://cacm.acm.org/research/mapreduce-simplified-data-processing-on-large-clusters/>
20. Apache Hadoop Documentation. (2025). hadoop.apache.org.
<https://hadoop.apache.org/docs/>
21. Apache Spark Documentation. (2025). spark.apache.org.
<https://spark.apache.org/docs/latest/>
22. MongoDB Documentation. (2025). Cloud Data Platform. mongodb.com.
<https://www.mongodb.com/docs/>
23. PostgreSQL Documentation. (2025). postgresql.org.
<https://www.postgresql.org/docs/>

					БР.ІІІ - 95.00.00.000 ІІЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		77

24. Oracle Cloud Infrastructure. (2025). Cost Analysis and Budgets. oracle.com. <https://docs.oracle.com/en-us/iaas/Content/Billing/home.htm>
25. Red Hat OpenShift Documentation. (2025). Monitoring and Cost Management. docs.redhat.com.
https://docs.redhat.com/en/documentation/openshift_container_platform
26. Jain, A., & Kumar, S. (2024). Machine Learning Approaches for Cloud Cost Prediction. Journal of Cloud Computing, 13(2), 44–58.
<https://doi.org/10.1186/s13677-024-00521-7>
27. Patel, R., & Shah, D. (2023). Predictive Analytics for Cloud Resource Allocation. IEEE Access, 11, 55431–55445.
<https://doi.org/10.1109/ACCESS.2023.3278945>
28. Wang, T., & Li, X. (2025). AI-Based Optimization of Cloud Infrastructure Costs. Future Generation Computer Systems, 158, 210–223.
<https://doi.org/10.1016/j.future.2024.12.015>
29. Kumar, P., & Singh, V. (2024). Forecasting Cloud Workloads Using Neural Networks. ACM Transactions on Internet Technology, 24(3), 1–19.
<https://doi.org/10.1145/3651123>
30. Hassan, M., & Ahmed, F. (2023). Elastic Resource Provisioning in Cloud Computing Environments. IEEE Transactions on Cloud Computing, 11(4), 1890–1903. <https://doi.org/10.1109/TCC.2022.3147854>
31. Microsoft Learn. (2025). Azure Machine Learning Documentation. learn.microsoft.com. <https://learn.microsoft.com/en-us/azure/machine-learning/>
32. Amazon SageMaker Developer Guide. (2025). docs.aws.amazon.com. <https://docs.aws.amazon.com/sagemaker/>
33. Google Vertex AI Documentation. (2025). cloud.google.com. <https://cloud.google.com/vertex-ai/docs>
34. NIST. (2024). The NIST Definition of Cloud Computing. nist.gov. <https://csrc.nist.gov/publications/detail/sp/800-145/final>
35. ISO/IEC 17788:2014. (2014). Cloud Computing – Overview and Vocabulary. iso.org. <https://www.iso.org/standard/60544.html>

					БР.ІІІ - 95.00.00.000 ІІЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		78

36. ISO/IEC 17789:2014. (2014). Cloud Computing – Reference Architecture. iso.org. <https://www.iso.org/standard/60545.html>
37. OWASP Foundation. (2024). OWASP Top Ten Security Risks. owasp.org. <https://owasp.org/www-project-top-ten/>
38. Gartner. (2025). Trends in Cloud Financial Management and FinOps. gartner.com. <https://www.gartner.com/en/information-technology/insights/cloud-management>
39. FinOps Foundation. (2025). FinOps Framework. finops.org. <https://www.finops.org/framework/>
40. • Chen, Y., & Zhao, H. (2025). Intelligent Cost Optimization in Multi-Cloud Environments Using Machine Learning. IEEE Transactions on Services Computing, 18(1), 88–101. <https://doi.org/10.1109/TSC.2024.3398712>

					БР.ІІІ - 95.00.00.000 ІІЗ	Арк.
						79
Змн.	Арк.	№ докум.	Підпис	Дата		

БІБЛІОГРАФІЧНА ДОВІДКА

Тема бакалаврської роботи: " Прогнозування витрат на хмарні ресурси
за допомогою машинного навчання"

Обсяг пояснювальної записки: 75 аркушів

Дата закінчення бакалаврської роботи 10 червня 2026р.

Підпис студента _____