**ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ**

# КВАЛІФІКАЦІЙНА РОБОТА

на тему: **«Штучний інтелект у забезпеченні високої продуктивності хмарних додатків»**

на здобуття освітнього ступеня бакалавра

зі спеціальності *126 Інформаційні системи та технології*

*(код, найменування спеціальності)*

освітньо-професійної програми *Інформаційні системи та технології*

*(назва)*

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

| | |
|---|---|
| | Дмитрієва Анастасія |
| *(підпис)* | *Ім'я, ПРІЗВИЩЕ здобувача* |

Виконав: здобувач вищої освіти групи ІСД-42 Дмитрієва Анастасія

*Ім'я, ПРІЗВИЩЕ*

Керівник: д.т.н.,професор Каміла Сторчак

*науковий ступінь, вчене звання* — *Ім'я, ПРІЗВИЩЕ*

Рецензент: _____

*науковий ступінь, вчене звання* — *Ім'я, ПРІЗВИЩЕ*

**Київ 2025**

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**

**НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

Кафедра Інформаційних систем та технологій

Ступінь вищої освіти Бакалавр

Спеціальність 126 Інформаційні системи та технології

Освітньо-професійна програма Інформаційні системи та технології

**ЗАТВЕРДЖУЮ**

Завідувач кафедрою Каміла Сторчак

*(Ім'я, ПРІЗВИЩЕ)*

«____» _____ 20__ р.

**ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Дмитрієва Анастасія Анатоліївна

*(прізвище, ім'я, по батькові здобувача)*

1. Тема кваліфікаційної роботи: Штучний інтелект у забезпеченні високої продуктивності хмарних додатків

керівник кваліфікаційної роботи Каміла Сторчак, д.т.н, професор

*(Ім'я, ПРІЗВИЩЕ, науковий ступінь, вчене звання)*

затверджені наказом Державного університету інформаційно-комунікаційних технологій від «___» _____ 20__ р. № ___

2. Строк подання кваліфікаційної роботи «___» _____ 20__

3. Вихідні дані до кваліфікаційної роботи: Аналіз продуктивності хмарних обчислень та платформ (AWS, Azure, Google Cloud), вивчення сучасних проблем кібербезпеки, витрат і управління ресурсами, дослідження методів автоматизації та використання штучного інтелекту для виявлення аномалій, прогнозування навантаження та оптимізації інфраструктури хмарного середовища.

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

1. Аналіз хмарних технологій та проблем продуктивності
2. Розробка AI-рішення для оптимізації та безпеки
3. Апробація рішення на прикладі хмарної платформи

5. Перелік ілюстративного матеріалу: презентація

6. Дата видачі завдання «___» _____ 20__ р.

КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів кваліфікаційної роботи | Строк виконання етапів роботи | Примітка |
|---|---|---|---|
| 1 | Формулювання теми, мети, завдань роботи та ознайомлення з вимогами до оформлення | 01.05 – 03.05.2025 | |
| 2 | Аналіз проблем хмарних технологій, кіберзагроз та витрат, огляд сучасних платформ (AWS, Azure тощо) | 04.05 – 06.05.2025 | |
| 3 | Обґрунтування вибору технологій, створення архітектури рішення, підготовка середовища | 07.05 – 10.05.2025 | |
| 4 | Реалізація AI-модуля: виявлення аномалій, прогнозне масштабування, оптимізація витрат | 11.05 – 13.05.2025 | |
| 5 | Тестування, збір результатів, побудова дашбордів, аналіз ефективності | 14.05 – 16.05.2025 | |
| 6 | Оформлення тексту дипломної роботи, написання вступу, висновків, підготовка до захисту | 17.05 – 21.05.2025 | |

Здобувач вищої освіти _____ Анастасія Дмитрієва

*(підпис)* *(Ім'я, ПРІЗВИЩЕ)*

Керівник кваліфікаційної роботи _____ Каміла Сторчак

*(підпис)* *(Ім'я, ПРІЗВИЩЕ)*

# РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня бакалавра : 67 сторінок, 15 рисунків, 5 таблиць, 20 джерел.

*Об'єкт дослідження – процес обробки та оптимізації навантаження в хмарному середовищі із застосуванням штучного інтелекту.*

*Предмет дослідження – моделі та методи застосування алгоритмів штучного інтелекту для виявлення аномалій,прогнозування навантажень та автоматичного масштабування хмарних додатків*

*Мета роботи – розробити систему з використанням штучного інтелекту для підвищення продуктивності хмарного додатку шляхом виявлення критичних навантажень, збереження метрик та автоматичної реакції на аномальні події у реальному часі.*

*Методи дослідження – аналітичнийогляд сучасних підходів до оптимізації хмарних систем за допомогою штучного інтелекту; моделювання взаємодії користувача з системою; проектування та реалізація бази даних; програмна імітація дій користувача; візуалізація результатів у формі SQL-запитів.*

*Питання підвищення продуктивності хмарних додатків є актуальним у зв'язку зі зростанням обсягів даних,кількості користувачів і потреб у стабільній роботі цифрових сервісів. Традиційні методи не завжди дозволяють вчасно реагувати на пікові навантаження, тому виникає потреба у впровадженні «розумних» підходів до управління ресурсами.*

*У цій роботі запропоновано метод оптимізації хмарного середовища шляхом інтеграції штучного інтелекту для виявлення аномалій та аналізу системних метрику реальному часі.Створено базу даних, яка збергігає показники продуктивності і реалізовано систему, яка ефективно реагує на зміни. Апробація рішення підтвердила ефективність для забезпечення стабільної роботи.*

*Галузь використання – Інформаційні системи, хмарні обчислення, кібербезпека, управління ресурсами, DevOps.*

КЛЮЧОВІ СЛОВА: ХМАРНІ ТЕХНОЛОГІЇ, ШТУЧНИЙ ІНТЕЛЕКТ, КІБЕРБЕЗПЕКА, МАСШТАБУВАННЯ, ПРОГНОЗНА АНАЛІТИКА, AWS, AZURE, GOOGLE CLOUD, ОПТИМІЗАЦІЯ ВИТРАТ, АВТОМАТИЗАЦІЯ, ВИЯВЛЕННЯ АНОМАЛІЙ, МОНІТОРИНГ ТРАФІКУ, МАШИННЕ НАВЧАННЯ

## Abstract

Text part of the bachelor level qualification work: 67 pages, 15 figures, 5 tables, 20 sources.

*Object of the research – general-purpose cloud computing platforms (AWS, Azure, Google Cloud).*

*Subject of the research – approaches to improving the performance of cloud systems using artificial intelligence methods.*

*Aim of the work – to develop an intelligent system for monitoring, resource optimization, and cyber threat detection in a cloud environment by integrating machine learning algorithms and predictive analytics.*

*Research methods – analytical review of current technologies, modeling of cloud infrastructure, development of intelligent resource distribution algorithms, building a system prototype, and evaluating its effectiveness under real conditions.*

*The issue of performance and security in cloud technologies is increasingly relevant due to the global digitalization of businesses, the growing volume of data, and the need for automated infrastructure management. Existing approaches often fail to handle dynamic load balancing or respond effectively to cyber threats.*

*This work proposes an AI-based solution that includes a traffic monitoring system for anomaly detection, a predictive resource scaling module, and automated cost management. All components were tested within a cloud platform environment, with performance, efficiency, and cost-effectiveness taken into account.*

*Field of application – Information systems and technologies, cloud computing, cybersecurity, intelligent control systems.*



KEYWORDS: CLOUD COMPUTING, ARTIFICIAL INTELLIGENCE, CYBERSECURITY, RESOURCE SCALING, PREDICTIVE ANALYTICS, AWS, AZURE, GOOGLE CLOUD, COST OPTIMIZATION, AUTOMATION, ANOMALY DETECTION, TRAFFIC MONITORING, MACHINE LEARNING.

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**
**НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

**ПОДАННЯ**
**ГОЛОВІ ЕКЗАМЕНАЦІЙНОЇ КОМІСІЇ**
**ЩОДО ЗАХИСТУ КВАЛІФІКАЦІЙНОЇ РОБОТИ**
**на здобуття освітнього ступеня бакалавра**

Направляється здобувач Дмитрієва А. А. до захисту кваліфікаційної роботи

*(прізвище та ініціали)*

за спеціальністю *126 Інформаційні системи та технології*

*(код, найменування спеціальності)*

освітньо-професійної програми *Інформаційні системи та технології*

*(назва)*

на тему: **«**Штучний інтелект у забезпеченні високої продуктивності хмарних додатків**».**

Кваліфікаційна робота і рецензія додаються.

Директор ННІ _____ _____

*(підпис)* *(Ім'я, ПРІЗВИЩЕ)*

**Висновок керівника кваліфікаційної роботи**

Здобувач _____

_____

_____

_____

Все це дозволяє оцінити виконану кваліфікаційну роботу здобувача _____ на оцінку «_____» та присвоїти йому кваліфікацію _____.

Керівник кваліфікаційної роботи _____ Каміла Сторчак

*(підпис)* *(Ім'я, ПРІЗВИЩЕ)*

«____» _____ 20___ року

**Висновок кафедри про кваліфікаційну роботу**

Кваліфікаційна робота розглянута. Здобувач(ка) прізвище та ініціали допускається до захисту даної роботи в Екзаменаційній комісії.

Завідувач кафедрою _____ _____ _____

*(назва)* *(підпис)* *(Ім'я, ПРІЗВИЩЕ)*

**ВІДГУК РЕЦЕНЗЕНТА**
**на кваліфікаційну роботу**
**на здобуття освітнього ступеня бакалавра**

здобувача вищої освіти Дмитрієва Анастасія Анатоліївна

*(прізвище, ім'я, по батькові)*

на тему: «Штучний інтелект у забезпеченні високої продуктивності хмарних додатків»

**Актуальність.**

**Позитивні сторони.**

1.
2.
3.

**Недоліки.**

1.
2.

Відзначені зауваження не впливають на загальну позитивну оцінку кваліфікаційної роботи бакалаврської

.

**Висновок**: *кваліфікаційна робота на здобуття ступеня бакалавра заслуговує оцінку "_____", а здобувач(ка) _____ заслуговує присвоєння кваліфікації: ……..*

Рецензент: _____ _____
*науковий ступінь, вчене звання* *підпис* *Ім'я, ПРІЗВИЩЕ*

**ЗМІСТ**

# ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

IT – Information Technology

SaaS – Software as a Service

IaaS – Infrastructure as a service

PaaS – Platform as a Service

AI – Artificial Intelligence

SIEM – Security Information and Event Management

IDS/IPS – Intrusion Detection System/ Intrusion Prevention System

SQL – Structured Query Language

ER – Enity-Relationship

DDoS – Distributed Denial of Service

API – Apllication Programming Interface

# ВСТУП

Наш світ постійно зазнає змін, включаючи інтернет-простір, де великі IT-компанії пропонують новітні продукти, які повністю змінюють наше життя, роблячи його більш цифровим і доступним для обробки будь-якої інформації в будь-якій сфері. ШІ – це інновація, яка виступає одним із найважливіших інструментів для оптимізації і підвищення продуктивності хмарних додатків. Завдяки інтеграції штучного інтелекту у хмарні додатки відкриваються унікальні можливості для оптимізації ресурсів, підвищення продуктивності, надійності та безпеки цифрових сервісів. Дослідивши ринок ШІ, можна зробити висновок, що він щорічно зростатиме на понад 30%, а саме через популярність хмарних обчислень і збільшенню кількості хмарних сервісів.
При розробці та підтримці хмарних додатків постають такі низки проблем, як нестабільність навантаження, обробка великих обсягів даних, проведення діагностики проблему у режимі реального часу. У більшості випадків саме ці технологічні труднощі знижують загальну продуктивність системи. Та застосовуючи штучний інтелект для виявлення вузьких місць, оптимізації запитів, управління ресурсами, ми можемо покращити роботу хмарних додатків. Використання таких систем передбачає спільний доступ до важливої інформації - журналів подій, статистики, продуктивності сервісів. Весь цей об'єм даних потребує централізованої бази даних, яка ефективно обробляє і зберігає матеріал.

Метою даної дипломної роботи є розробка та впровадження інтелектуальної системи, яка забезпечує високу продуктивність хмарних додатків шляхом автоматизації ресурсів, підвищення рівня безпеки. При виконанні роботи був проведений аналіз хмарних обчислень, а саме виявлення основних проблем продуктивності.

Описано предметну область: сутності, які їй належать та їх взаємодії. Описано процес розробки системи та структуру бази даних, яка ефективно інтегрує з алгоритмами штучного інтелекту, тестування функціональності бази даних.

Результатом роботи є створення прототипу системи, що поєднує можливості хмарного зберігання, бази даних і штучного інтелекту для виявлення аномалій, моніторингу метрик і автоматичного реагування на критичні навантаження. Реалізовано функціональну структуру бази даних, проведено тестування її ефективності, запропоновано архітектуру взаємодії компонентів та підтверджено доцільність застосування інтелектуального підходу для підвищення продуктивності хмарного середовища.

# 1. АНАЛІЗ ПРОБЛЕМ ПРОДУКТИВНОСТІ ХМАРНИХ ТЕХНОЛОГІЙ

Підвищення продуктивності є одним із найважливіших показників, який відображає виконаний обсяг роботи за одиницю часу, враховуючи фактори: середовища, капіталу, технологій та людських ресурсів. Від продуктивності залежить підвищення гнучкості та масштабованості, Швидкість доступу до ресурсів, ефективність обробки запитів користувачів.

Важливим завданням хмарних платформ, в умовах стрімкого розвитку технологій, забезпечити стабільність та продуктивність при одночасному обслуговуванні великої кількості клієнтів, адже користувачі щодня звертаються до хмарних сервісів задля отримання, зберігання та обробки даних, які зберігаються в Data Centre.

На відміну від традиційних серверів, хмарні платформи пов'язані з розподіленими ресурсами, які водночас використовує велика кількість користувачів. При такому колосальному навантаженні, це дало поштовх для створення особливих вимог, щоб ефективно балансувати та управляти хмарними обчисленнями. Перевага високої продуктивності мінімізує затримку у виконанні запиту, забезпечує стійкість, таким чином клієнти обслуговуються без втрат якості.

Незважаючи на багато переваг, в хмарній інфраструктурі існує низка проблем, які впливають на різні аспекти бізнесу та продуктивність. Серед них :

- надмірне навантаження серверів, що призводить до перевантаження окремих вузлів.

- затримка мережі, яка зумовлена нестабільною пропускною здатністю,

- недостатня оптимізація програмного забезпечення, що призводить до зниження оперативності

-обмеження ресурсів, в моменти високої активності, коли хмара не адаптується під запити

-неякісне балансування навантаження, коли трафік не перенаправляється на інші функціональні сервери.

Звідси випливає, що експлуатація та впровадження хмарних систем передбачає аналіз і вирішення проблем продуктивності, як основне із завдань. Оптимізація цих процесів не тільки підвищує ефективність роботи інфраструктури, але й дозволяє забезпечити надійність, масштабованість та економічні рішення у довгостроковій перспективі.

### 1.1 Огляд сучасного стану хмарних обчислень

Хмарні технології у 2025 році являються основною складовою IT-ландшафту. Це сфера, в яку інвестують мільярди доларів. Завдяки технологіям, організації швидко адаптуються до змін на ринку. В даний час, хмарні технології поділяються на різні типи, враховуючи розташування ресурсів для надання користувачу.

- Публічна хмара. Модель надання ресурсів усім користувачам. Послугами даної хмари може користуватись будь-хто, однак приватна інформація зберігає конфіденційність. Публічна хмара надає можливості доступності, швидкості, оплата тільки за використання ресурсів. Таким чином бюджет чітко розрахований на різні періоди.

- Гібридна хмара. Поєднує в собі приватну та публічну модель, завдяки цьому вона є гнучкою,розподіляючи інформацію до приватного сектору або публічного за допомогою шлюзів. Дана комбінація має переваги в: керуванні власними ресурсами, підлаштування під потреби користувача, високий рівень контролю та захисту.

- Приватна хмара. Середовище, яким володіє одна організація. Саме така архітектура підходить для збереження чутливих даних. Для налагодження інфраструктури її функціональності , вимагає від компаній багато часу та грошей. Користування приватною хмарою надає привілеї до управління високою інфраструктурою, контроль компаній над ресурсами, багаторівневий доступ.

- Мульти хмара. Своєю назвою позначає гнучку інфраструктуру із використанням декількох постачальників хмарних послуг. Замовник отримує

систему, яка оптимізована під власне підприємство. В цій моделі можна підкреслити такі переваги, як надійність, оптимізація витрат.

- Спільна хмара.Розроблена та налаштована для використання кількома компаніями, сприяє можливості контролю інфраструктури, висока ефективність використання даних, безпека.

Щоб користувач міг взаємодіяти із хмарними платформами, є способи розгортання моделей за принципами IaaS, PaaS, SaaS. Ці моделі надають широкий спектр послуг.

- IaaS- інфраструктура як послуга. Суть якої полягає в отриманні певної кількості потужності необхідній в мережі. В якості послуги надаються середовище для збережної інформації або віртуальні машини.

- PaaS - платформа як послуга. Користувачі отримують доступ до віртуальної платформи, де в них є можливість розробляти та тестувати додатки.

-SaaS- програмне забезпечення як послуга. Дана модель надає готовий додаток користувачам без необхідності встановлення. Основні характеристики: оновлення, технічне обслуговування та розміщення програмного забезпечення відбуваються через постачальника. Доступ здійснюється через інтернет.

Amazon Web Services, Microsoft Azure та Google Cloud Platform – найбільші постачальники на ринку хмарних послуг. Вони пропонують для розробників, аналітиків та IT-команд великий спектр інструментів з інтеграцією штучного інтелекту, аналізу великих даних, машинним навчанням.

Таблиця 1.1 – Порівняння постачальників

| Постачальник | Частка ринку(%) | Основні сервіси | Особливості |
|---|---|---|---|
| Amazon Web Services | 32% | EC2, S3,RDS, Sage Maker | Загальнодоступна, гнучка,широкий набір AI-рішень. |
| Microsoft Azure | 20% | Blob Storage,App Services, Azure ML | Надійні аналітичні, обчислювальні мережеві рішення, сильна інтеграція з Office та |

| | | | |
|---|---|---|---|
| | | | Windows. |
| Google Cloud Platform | 10% | Vertex AI,Cloud Functions, BigQuery | Гнучка, інноваційна в рішеннях аналітики, високий рівень безпеки. |

Для задоволення потреб бізнесу недостатньо мати один провайдер. Щоб уникнути залежності від постачальника та підвищити рівень безпеки і гнучкості, великих обертів набирає тенденція мульти хмари. Ця стратегія дозволяє розподіляти навантаження між кількома постачальниками, утворюючи єдину екосистему. Особливу увагу заслуговують гібридні моделі, використання яких швидко зростає на ринку. Це тип інфраструктури, який комбінує кілька видів платформ одночасно. Наприклад, поєднання локальної серверної інфраструктури із публічною хмарою. Такий підхід задовольняє потреби, оптимізовуючи витрати та забезпечуючи масштабованість, безперервність бізнес-процесів, навіть у разі збою одного з компонентів. Кожен постачальник має свій сервер для конкретного завдання, що сприяє покращенню продуктивності.

На 2025 рік актуальними рішеннями в хмарних обчисленнях є:

- Edge Computing: наближення зберігання та обробки даних до місця, де вони потрібні.

- Автоматизація процесу розробки та впровадження програмного забезпечення: взаємодія та зближення робочих процесів між командою розробників та інженерами з інфраструктури.

- Нативна архітектура програмного забезпечення (Cloud Native): включає в себе підходи мікросервісів, контейнерів, інфраструктуру як код, автоматизовані пайплайни, моніторинг та спостереження.

- Безпека як сервіс: комплексні заходи, шифрування, виявлення загроз, моніторинг подій.

- Інтеграція штучного інтелекту: машинне навчання, прогноз пікових навантажень, аналіз подій.

Можливості технологій та потреби в них зростають з кожним днем все більше, що означає високу вразливість в напрямку продуктивності, безпеки та витрат. В даному випадку, автоматизація відіграє важливу роль в оптимізації

процесів, як в балансуванні, так і в управлінні ресурсами. Саме пошук ефективного виходу у вирішенні проблем, з якими стикаються хмарні технології, є критично важливим шляхом для покращення та вдосконалення хмарних обчислень.

### 1.2 Основні проблеми хмарних технологій

Хмарні обчислення впроваджені майже у всіх галузях. При такому широкому зростанні попиту, виникають низки системних проблем, які стримують продуктивність, надійність та безпеку. В умовах критичного навантаження, масштабування і взаємодії з важливими даними, ці проблеми стають особливо відчутними. На етапах проєктування архітектури компанія визначає, як будуть організовані компоненти хмарної системи, наприклад: сховища, база даних, сервіси тощо. Тому перед розгортанням важливо визначити, наскільки інфраструктура відповідає вимогам компанії, як відбуватиметься масштабування при зростанні навантаження, і чи буде підтримуватись система при впровадженні нових алгоритмів.

Після розгортання важливо подбати про обчислювальні ресурси, сховище, центральний процесор, оперативну пам'ять, пропускну здатність. На цьому етапі невід'ємною частиною є автоматичне масштабування, яке дозволяє системі вчасно реагувати на надмірне навантаження. Використовуючи методи і віртуалізації та контейнеризації, управління ресурсами спрощується, забезпечується гнучкість системи.

Безпека даних в хмарному середовищі потребує висококомплексного підходу. До таких підходів відносять шифрування даних, політика аутентифікації та авторизації, передбачення захисту від зовнішніх атак, моніторинг та виявлення загроз.

Ефективна робота Хмари залежить від оптимізації використання ресурсів. Нераціональне використання або у випадку простою, неминучим стануть зайві витрати, адже вчасний та регулярний аналіз компонентів запобігають виникненню труднощів.

Окремою проблемою хмарних технологій вважається обмеженість досвіду та знань спеціалістів, а саме через складність використання хмарних можливостей.

### 1.2.1 Кібербезпека: загрози, витоки даних та вразливості

Онлайн робота та спілкування включає в себе обмін, збереження та обробку певних даних, що являється потенційною причиною атак на хмарні середовища. Найбільш поширеними загрозами є:

- DDoS-атаки, які спрямовані на виведення з ладу хмарних сервісів шляхом перевантаження їх трафіком із тисяч або мільйони джерел. Представлена атака спричиняє простої у неможливий доступ до критично важливих ресурсів.

- SQL-ін’єкції. Даний тип атаки, при якому вводиться SQL-код у форму воду або URL запити, з метою доступу до бази даних. Якщо в системі відсутня належна перевірка вхідних запитів, це дозволяє зловмиснику змінювати вміст інформації або встановлювати контроль над сервером.

- а так і через помилки конфігурації. Достатньо некоректно налаштувати сервіс,відсутність обмежень на мережевий трафік, облікові записи без зміни паролю, як це створить шлях до зловмисного втручання.

- Компрометація облікових записів. Ризик, який виникає внаслідок фішингових атак, повторного використання паролів чи слабку аутентифікацію. Внаслідок подібної атаки зловмисник може отримати доступ до інформації і змінити налаштування хмарної інфраструктури або знищити інформацію.

Конфіденційність даних є дуже важливою для користувачів. Часто причиною витоків інформації, пов'язані з людським фактором: відсутність політики шифрування, неправильне налаштування віртуальних сервісів, помилкове надання прав доступу. Іноді хмара стає мішенню для внутрішніх загроз, коли співробітники свідомо передають конфіденційні дані стороннім особам.

Підхід, який застосовується у вирішенні цієї проблеми, є використання систем із виявленням аномалії за допомогою штучного інтелекту. Алгоритми

машинного навчання можуть аналізувати моделі поведінки користувачів, визначати підозріли дії та реагувати на загрози в режимі реального часу. Наприклад, система IDS-IPC з компонентами штучного інтелекту може не тільки зазначити атаку, але й заблокувати доступ, який її спричинив.

Атаки та загрози зростають у великій кількості. Тому безпека у хмарі повинна бути спроєктована на рівні від шифрування даних до використання нульової моделі, концепція якої складається із принципу «ніколи не довіряй, постійно перевіряй», тобто всі потребують постійної перевірки. Ця модель дотримується принципів:

- Доступ з найменшими привілеями – модель, яка надає мінімальний рівень доступу необхідному користувачу.

- Мікро сегментація.Підхід передбачає розділення охоронних периметрів на зони, що забезпечує окремий доступ.

- Багатофакторна автентифікація. Ця методика сприяє підтримці балансу безпеки без впливу на продуктивність, надсилаючи перевірку у вигляді запиту пароля, повідомлення на телефон тощо.

- Постійний моніторинг реалізується в реальному часі задля перевірки стану безпеки. Таким чином забезпечується швидке виявлення загроз.

- Умовний доступ працює за рахунок механізмів автентифікації без впливу користувача.

### 1.2.2 Управління витратами та економічна ефективність

Економічна доцільність – одна з причин переходу бізнесу до хмари. Організація, яка тільки починає зростати, повинна розвивати гнучкість в багатьох аспектах. Не дивлячись на вартість послуг хмарної інфраструктури, яка в рази дорожче за серверні рішення , користувач отримує якісний вбудований функціонал. Компанії із фіксованим бюджетом найчастіше стикаються із проблемою неправильного управління ресурсами, що призводить до непередбачуваних витрат. Як правило,такі витрати зумовлені не вимкненими або невикористовуваними віртуальними машинами, застарілі сховища даних, резервні

копії з відсутнім строком очищення, неефективні запити на базу даних, тим самим створюючи надмірне навантаження.

Будь-які витрати потребують аналітики та моніторингу, а більшість компаній ігнорують ці критерії під час користування хмарними сервісами. В результаті отримує неправильне прогнозування потреб і додаткові витрати на велику суму.

Впровадження інструментів керування витратами із поєднанням автоматизації, запобігає труднощам з економічною ефективністю. Алгоритми штучного інтелекту із сучасними інструментами здатні аналізувати витрати, прогнозувати критичні періоди, контролювати бюджет, пропонувати вигідні варіанти масштабування . Отже, аналіз та перевірка витрат вважається не тільки технічним завданням, а й логічно розробленою стратегією, яка повинна слугувати для фінансової стабільності компанії.

### 1.2.3 Обмеженість ресурсів та дефіцит експертних знань

Хмарні технології все більше розширюють свої можливості, забезпечуючи клієнтам більший доступ до ресурсів та гнучкі рішення зберігання даних. Разом із зростанням, компанії стикаються з досить поширеною проблемою, як нестача кваліфікованих спеціалістів. В результаті обмеження експертних знань, виникають ускладнення побудови архітектури, організації уникають впровадження автоматизації або відмовляються від повноцінної міграції у хмару. Наслідком нестачі експертних знань, призводить до перевантаження, технічних збоїв, фінансових втрат на неефективну продуктивність. Крім неполадків з технічними процесами, також виникають непорозуміння між бізнесом і технічною командою, коли складно дійти узгодження у доцільному використанні хмарних рішень.

Покращити обставини в сфері спеціалістів хмарного середовища можна за допомогою інвестування в навчання та підвищення кваліфікацій співробітників. Сертифікаційні програми сприятимуть покращенню навичок в обслуговуванні хмарної інфраструктури. Партнерство з провайдерами послуг підготує

спеціалістів до практичного досвіду у хмарних сервісах. Велику увагу заслуговують інноваційні методи автоматизації. Нововведення з елементами штучного інтелекту зменшують потребу в ручному масштабуванні, архітектура будується за певними шаблонами, SQL-запити більш оптимізовані щодо конфігурації у реальному часі.

Таким чином, підхід активного навчання та автоматизації, зменшує залежність від вузьких спеціалістів, пришвидшуючи виконання складних завдань за допомогою автоматизованих систем або сертифікованих працівників.

### 1.3 Підходи до вирішення цих проблем

Кібербезпека, фінансова неефективність, нестача технічних знань – три категорії проблем, які супроводжуються в хмарних середовищах. Точним методом, в боротьбі з даними викликами, стане постійний внесок в інноваційні рішення на основі автоматизації процесів та штучних інтелектів.

Щоб зменшити ризик кібератак на хмарне середовище, компанії застосовують SIEM-системи для виявлення загроз, перш ніж вони можуть нашкодити бізнес-процесам. Система керування захистом інформації також застосовує засоби керування подіями безпеки в одне рішення. Головними принципами роботи інструментів – збирання та аналіз даних від програм до користувача організації в реальному часі, щоб прискорити процес виявлення та блокування фахівцями кібератак.

Завдяки можливостям SIEM-систем, робота з великими даними стає простішою, впорядкувавши їх, а потім визначаються ознаки загроз, атак чи порушень. Щоб визначити зв'язки між даними, відбувається процес сортування, що пришвидшує виявлення ризиків.

Після чого, система продовжує відстежувати інциденти, сповіщає і перевіряє всі дії, пов'язані з інцидентом.

Кожний бізнес має свою цінність, і щоб підвищити цю цінність в хмарному середовищі, організації переходять до операційної платформи Financial Operations. Ця модель допомагає організація покращити управління фінансами

хмарних технологіях на основі даних, об’єднуючи фінансові,інженерні та бізнес команди над оптимізацією витрат на хмарні сервіси.

Додаткова мета підходу не тільки керує витратами, а й збільшує прибутки через хмару. Прикладами інструментів є Amazon Web Services, Cost Explorer, Google Cloud або Google Health. Завдяки автоматизації, дані сервіси відслідковують , прогнозують та оптимізують витрати без необхідних навичок співробітника.

Тенденція «низький код/без коду» або «low-code/no-code» несе нове рішення у світ технологій. Під даним висловом маються на увазі платформи для створення програмного продукту з мінімальним кодуванням, маючи в наявності готові шаблони та компоненти. Таким чином розробка пришвидшується, такі компанії, як Microsoft Power Platform, OutSystem, Mendix дають можливість працівникам створювати бізнес-задатки чи проводити автоматизацію у хмарі.

**Висновки до розділу**

У першому розділі було розглянуто та здійснено аналіз сучасного стану хмарних обчислень, його роль, переваги, основні проблеми та підходи вирішення цих проблем.

Особливу увагу було перевернуто до труднощів, з якими найчастіше стикаються хмарні технології – це кібератаки, продуктивність, економічні ризики, дефіцит кваліфікованих спеціалістів. В якості вирішення даних перешкод були запропоновані методи, які ґрунтуються на внесенні новітніх методик,насамперед-автоматизація систем, завдяки чому покращується рівень безпеки, аналіз витрат, прогнозування надмірних навантажень.

# 2. ОБҐРУНТУВАННЯ ВИБОРУ ТЕХНОЛОГІЙ ДЛЯ ПОКРАЩЕННЯ ПРОДУКТИВНОСТІ

До оптимізації хмарної інфраструктури належить ключовий етап обґрунтування вибору технологій. Спостерігаючи за вимогами до обробки даних, які постійно зростають, в зв'язку безпеки та масштабованості,недостатньо впроваджувати нові рішення. Варто збільшити контроль аналізу відповідності до бізнес-процесів організації. В цьому розділі описана доцільність обраних технологій, включно з інтеграцією штучного інтелекту.

## 2.1 Використання штучного інтелекту та автоматизації для кібербезпеки

Чутлива інформація в хмарному середовищі завжди була мішенню для кібератак. Традиційні системи захисту працюють за принципами правил та фільтрів, шукаючи вже відомі шаблони загроз, відомим прикладом є антивірус Avast, який використовує сигнатуру для виявлення шкідливого програмного забезпечення. В той же час, як кіберзагрози стають дедалі серйознішими та непередбаченими, тому застосування штучного інтелекту у сфері кіберзахисту – це не просто актуальність, а необхідність в безпеці.

Автоматизація кардинально змінює підхід до захисту інформації, виявляючи можливі загрози ще до того, як вони почнуть діяти. Штучний інтелект все найдалі розкриває потенціал в аналізі великих обсягів даних, фіксуючи різного роду поведінку системи,що стає основою досвіду для машини.

Сценарії використання штучного інтелекту передбачають:

- Систему виявлення атак та аномалій. За допомогою аналізу, штучний інтелект аналізує поведінку користувача або системи, фіксуючи відхилення. Це може бути несанкціонований доступ в систему чи мережу, нетиповий час доступу, кількість запитів, місцезнаходження. На відміну від класичного антивірусу, штучний інтелект виявляє небезпеку або несанкціонований доступ раніше.

- Система виявлення вторгнень. Якщо класичні системи працюють за встановленими правилами, то засіб виявлення загроз в поєднанні з штучним

інтелектом опановує виявлення невідомих атак, застосовуючи аналіз вхідних потоків і запитів.

- Соціальна інженерія в ідентифікації фішингових атак. Технологія обробки природної мови - Natural Language Processing, присвячена розумінню машини людської мови, допомагає розпізнавати шкідливі посилання, фішингові листи, повідомлення.

- Прогнозування вразливості.Інформація про вразливості надходить з інтернет-ресурсів. З цього виходить, що машинне навчання може використовуватись для аналізу вхідних даних та отримувати результат прогнозу загорост та вразливостей.

Перевагою використання штучного інтелекту в кіберзахисті хмарних середовищ забезпечує миттєву реакцію на виявлення загрози до того, як системі буде завдана шкода. Моделі штучного інтелекту постійно навчаються, ця важлива характеристика допомагає адаптуватись до нових типів загроз та ефективно реагувати, блокуючи їх.

Саме ефективне реагування автоматизованих алгоритмів, зменшує залежність від людського фактора, який здатний в складних інцидентах уповільнити прийняття рішення. Таким чином зменшується навантаження на співробітників.

## 2.2 Оптимізація витрат на хмарні обчислення

Хмарні сервіси потребують якісної стратегії управління витратами, яка проводить моніторинг аналіз ресурсів. Щоб оптимізувати витрати, необхідно здійснити перший крок – це зрозуміти, на що витрачаються кошти. Для цього існують окремі інструменти, як Amazon Web Services Cost Explorer, який заощадить бюджет та надасть аналітику розходжень на хмарні сервіси. Отримання послуг від провайдерів закріплюється ліцензією, яка входить до витрат. Щоб уникнути неочікуваних економічних втрат, варто правильно підібрати розмір та тип інстанту для робочого навантаження, надмірна кількість провокує додаткові витрати, а їхня нестача знижує продуктивність.

В періоди застою необхідно відключити певний функціонал,в результаті, плата проходить за сховище та резервування. В разі неактивних ресурсів використовується автоматичне масштабування, яке зменшить обчислювальну потужність або видалить, залежно від потреби.

### 2.2.1 Методи ефективного масштабування ресурсів

Ознака правильного масштабування базується на налаштуванні хмарного середовища на основі навантаження. Це дає можливість експлуатувати необхідні ресурси стільки, скільки потрібно. Масштабування буває горизонтальним – додавання нових інстантів, та вертикальним – акцент збільшення потужності направлений в бік наявного ресурсу. Автоматичне масштабування можна здійснити за допомогою спеціальних інструментів провайдерів,це:

- Amazon Web Services Auto Scaling за лічені хвилини виконує моніторинг додатків, налаштовує масштабування, володіє потужним інтерфейсом, який в свою чергу створює нові ES2 інстанції, коли навантаження на піковій точці.

- Azure Virtual Machine Scale Sets – створює та управляє групу віртуальних машин, одночасно балансуючи навантаження. Відповідно до запитів, кількість віртуальних машин може автоматично зменшуватись або збільшуватись.

- Google Cloud Compute Engine AutoScaler пропонує можливості масштабування на основі виміряного навантаження та заздалегідь налаштованих параметрів.

- Kybernetes Horizontal Pod Autoscaler відповідно до попиту автоматично масштабує навантаження, адаптуючи кількість контейнерів у кластері, коли змінюється трафік.

Для мінімізації ризиків, що виникають через неправильне масштабування ресурсів, необхідно переконатись, що оплата здійснюється включно за ту потужність, яка використовується. Аби зберегти продуктивність, також потрібно переконатися, чи не виникає дефіцит обчислювальних ресурсів.

Отже, аналітика прогнозу навантаження ризиків значно покращує планування, масштабування і фінансову доцільність у хмарних сервісах. Представлені рішення забезпечують стабільну роботу системи.

### 2.2.2 Використання прогнозної аналітики для управління витратами

Прогнозна аналітика – вид аналітики даних, спрямований на прогнозування майбутніх результатів чи подій на основі даних. Великі дані зазвичай використовуються в аналітичному процесі, мета прогнозної аналітики – це передбачення майбутніх подій, щоб вдосконалити процеси прийняття рішень в управлінні, визначити один або кілька факторів, що впливають на передбачувану подію. Цей підхід застосовується в сферах фінансів, медицині, освіті, правоохоронних органах тощо. Найпоширенішими сучасними методами для здійснення прогнозування вважають регресійний аналіз, деревоподібні моделі рішень, нейронні мережі.

- Регресійний аналіз становлює залежності між змінними. Використовується для прогнозування прибутку або витрат. Дана регресія підходить як для простих, так і складних сценаріїв.

- Деревоподібна модель здійснює прогноз на основі набору умов. Структура має вигляд «гілок»,кожна точка в розгалуженні відповідна певному критерію. Для візуалізації процесу прийняття рішення цей метод найбільш доречний.

- Нейронні мережі – це один з важливих інструментів для прогнозного аналізу. Нейроні мережі мають перевагу в здатності моделювати нелінійні залежності між вхідними та вихідними даними. Мережі можуть вчитися з історичних даних та використовувати шаблон автоматичного виявлення, щоб точно передбачити майбутні цінності навіть у ситуаціях, коли традиційні методи неефективні.

До інструментів, які передбачають та реалізують аналітику, відносить Amazon Web Services Cost Explorer в поєднанні з моделями машинного навчання створюють графіки витрат, формують прогнози на основі минулих даних. Моделі машинного навчання проводять аналіз за останні тижні або місяці, враховуючи тип інстантів, пропонує прогноз на майбутній період.

Google Cloud Recommender – сервіс, який здатний автоматично оцінювати ресурси, розміщені в Google Cloud. На основі оцінювання генеруються рекомендації, ціль яких – надати поради щодо продуктивності та витрат на навантаження. Завдяки необхідним ресурсам пропонуються рішення, які сприяють змінам навантаження і якими методами.

Azure Advisor виступає в якості помічника налаштувань конфігурацій та рекомендацій в розгортані, безпеці та заощаджень фінансів. Надає поради з ефективного розподілу навантаження та зменшення витрат на обчислення, зберігання та мережу.

Тому прогнозована аналітика не тільки допомагає зменшити витрати, але й допомагає планувати масштабування, керувати ризиками та покращити продуктивність.

## 2.3 Використання штучного інтелекту для розподілу ресурсів та автоматизації

Організації завжди прагнуть у своїх проєктах зменшити ризики та збільшити ефективність. Щоб досягти успіху та стабільності серед конкурентів, штучний інтелект надає велику перевагу компаніям в управлінні проектами. На основі реальних даних керівники приймають обґрунтовані рішення завдяки штучному інтелекту, який швидко реагує на зміни в системі.

Оптимізація хмарних середовищах надає перевагу в покращенні процесу управління ресурсами, оцінює поведінку користувачів або активність на основі накопичених даних. Саме ці алгоритми машинного навчання прогнозують навантаження і автоматично відтворюють коригування додаючої інстанції та розподіляють навантаження. Такий метод передбачає уникненню збоїв у процесах та знижує фінансові витрати.

Автоматизація розподілу ресурсів відтворюється в декілька етапів, забезпечуючи стабільну динаміку в роботі інфраструктури.

Першим етапом є збір метрик та моніторинг. Щоб сформувати базу на основі даних та аналітики, існує безліч сервісів, таких як Amazon CloudWatch,

Prometheus, які відстежують стан системи за критеріями: пам'ять, завантаженість процесора, кількість запитів та сесій.

Наступним етапом слідує прогнозування навантаження. Цей рівень включає машини навчання, де система за певною статистикою вживає певні заходи та приймає рішення.

Прийняття рішення відбувається на основі поведінки та прогнозів системи і обирає необхідні дії, наприклад: зміна трафіку на інший вузол, активізація резервних копій, додавання інстанцій тощо. А саме політика автомасштабування встановлює дані цілі, тобто застосовує зміни, що вважається наступним етапом в розподілі ресурсів. Тобто система самостійно виконує необхідні команди без участі адміністратора, таким чином ризик виникнення помилок мінімізується, а реагування на зміни стає швидшим.

Фінальною фазою вважається машинне навчання моделі, при якому штучний інтелект аналізує впровадженні зміни, редагує параметри в разі перевантаження, що робить систему більш удосконаленою.

Приклади технологій для автоматизації, які керують інфраструктурою без впливу людини:

Таблиця 2.1 – Технології, які керують інфраструктурою без впливу людини

| Платформа | Призначення |
|---|---|
| Amazon web Services Auto Scaling | Автоматичне масштабування |
| Azure Machine Learning + Virtual Machine Scale Sets | Прогнозування навантаження, самостійне навчання моделі |
| Google Cloud Vertex AI | Машинне навчання, інтеграція з BigQuery та Dataflow |
| Kuberntes Horizontal Pod Autoscaler | Автоматичне масштабування на основі пам’яті та процесора |

Наочним прикладом виступає медійний провайдер Netflix, який використовує розгортання резервних серверів у моменті релізу нових фільмів.

## 2.4 Вибір платформи для реалізації системи (aws, azure, google cloud)

Для збереження та обробки даних достатньо використовувати хмарні платформи і не обов'язково мати власну техніку, адже хмарні сервіси повністю забезпечують всім необхідним. Amazon Web Services, Microsoft Azure, Google Cloud Platform – ідеальне рішення для аналізу даних і відкриття нових можливостей.

Зазвичай просторові дані проходять етапи збереження, обробки, візуалізації, також інтеграції. Вищезгадані платформи призначені саме для таких завдань.

Для збереження геоданих можна використати Cloud Storage на Google Cloud Platform,Blob Storage на Azure.

Amazon Web Services – лідер на ринку хмарних сервісів, який пропонує широкий спектр інструментів для різного роду задач.

до ключових сервісів відносять:

- Elastic Compute Cloud масштабує серверні потужності у вигляді віртуальних машин. Опираючись на потреби, клієнти налаштовують інстанції, які мають різні типи, які адаптовані для обчислень, зберігання та пам'яті.

- Lambda – без серверна служба, яка виконує код без необхідності в наявності серверу. Оплата знімається за час запуску і завершення коду.

- Simple Storage Service – відомий доступністю та захистом даних. Веб-сервіс надає сховище для зберігання і отримання інформації в будь-який момент.

- Auto Scaling використовує автоматичне масштабування для збільшення місткості обмежених ресурсів в залежності від навантаження, підтримує стабільність програм за допомогою відстеження додатків, що дозволяє оптимізовувати додавання або видалення ресурсів.

- CloudWatch проводить аналітику метрик і подій, щоб в подальшому автоматизовані дії реагували на зміни в системі.

Microsoft Azure – хмарна платформа, яка здебільшого засереджена на глибоку інтеграцію з продуктами Microsoft, надає рішення для спрощення процесів на основі штучного інтелекту.

До ключових сервісів відносять:

- Virtual Machines призначений для створення и управління машинами.

- Azure Machine Learning – сервіс, який призначений для машинного навчання.

- Azure Monitor надає комплексні рішення моніторингу додатків, а саме їх продуктивності, доступності в локальних системах.

- Virtual Machine Scale Sets.Віртуальні машини запускають декілька копій додатку, у разі пошкодження хоча б однієї копії, користувач не помітить збоїв. Таким чином гарантує стабільність в доступності та масштабованості.

Сервіси із штучного інтелекту:

- Azure Machine Learning Studio - ресурс, який передбачає машини навчання без потреб глибоких знань в програмуванні. Простий інтерфейс із принципом «перетягування блоків». Інтегрується з іншими системами, що дозволяє зберігати великий обсяг даних.

Google Cloud Platform – це набір хмарних служб, запропонованих Google. Платформа володіє можливостями в галузі штучного інтелекту та машинного навчання.

До ключових сервісів відносять:

- Google Compute Engine надає інфраструктуру як сервіс, де користувачі обирають оптимальні рішення для своїх завдань.

- Google Cloud Functions – без серверна платформа, яка дозволяє запускати код або функції без середовища та керування сервером, забезпечує високий рівень ефективності завдяки ресурсам, які використовуються під час процесу.

- BigQuery – хмарний сервіс, призначений для виконання різних аналітичних запитів великих даних, без необхідності наявності серверу. Можливість BigQuery передбачає інтеграцію із додатковими інструментами.

- Google Cloud Monitoring – інструмент, який відстежує стан хмарних додатків за допомогою збору та аналізу метрик, візуалізації і трасувань ресурсів. Завдяки чому вчасно відбувається реакція на зміни в системі.

Дані платформи корисні тим, що надають сучасні інструменти для роботи з великими даними і впровадження штучного інтелекту. Таким чином, інфраструктура ефективно масштабується та оптимізовує фінансові витрати.

**Висновки до розділу**

Оптимізація та автоматизація стали одним із головних пріоритетів розвитку інформаційних технологій. Використовуючи штучний інтелект, значно підвищується безпека та рівень захисту, хмарні сервіси стають більш гнучкими. Ефективне виявлення аномалій у системі та загроз забезпечує миттєву реакцію на негативні події.

Amazon Web Services, Azure Virtual Machine ScaleSets, Google Compute Engine, AutoScaler – це інструменти, які мають автоматичне масштабування. Налаштувавши правильно масштабування, клієнт заощаджує фінанси, уникає дефіциту ресурсів та має високу продуктивність в інфраструктурі.

Amazon Web Services Cost Explorer, Google Cloud Recommender та Azure Advisor відносяться до напрямку моніторингу прогнозу навантажень та управління витратами. Ці інструменти аналізують історичні дані і на їх основі формують рекомендації щодо витрат та нераціонального використання ресурсів. Прогнозна аналітика, яка побудована на методах регресії, дерева рішень і нейронних мереж, точно і якісно враховує динаміку процесів.

Для прогнозу навантаження чи збору метрик можна виконати без участі адміністратора, за допомогою інтегрованих платформ Amazon CloudWatch і Prometheus. Штучний інтелект зменшує ризик виникнення помилок без впливу людського фактору.

Великі дані для роботи потребують потужні інструменти, такі як Amazon Web Services, Microsoft Azure та Google Cloud Platform. Платформи володіють машинним навчанням, тим самим забезпечуючи надійне функціонування бізнес-процесів, адаптуючись під вимоги організацій.

# 3. РОЗРОБКА AI-РІШЕННЯ ДЛЯ МІНІМІЗАЦІЇ ПРОБЛЕМ ХМАРНИХ ОБЧИСЛЕНЬ

Хмарні обчислення постійно зіштовхуються з нестабільним навантаженням на систему, що зазвичай призводить до перебоїв у сервісах. Рішенням цієї проблеми виступає штучний інтелект, завдяки якому система аналізується в реальному часі та змінює налаштування параметрів. Машинне навчання відіграє роль у створенні політики масштабування , враховуючи типи задач та історичні дані.

## 3.1 Архітектура запропонованого рішення

Оптимізація хмарних обчислень включає декілька компонентів, які між собою мають взаємодію в архітектурі AI-рішень. Щоб забезпечити ефективність в управлінні ресурсами та безпеки, було запропоновано дані модулі:

1. Модуль моніторингу і збору даних. Забезпечує безперервний процес збору метрик із запитів до баз даних, завантаженості процесора, оперативної пам'яті. Моніторинг сесій користувачів та оперативні відповіді здійснюється допомогою Amazon CloudWatch,Azure Monitor, Prometheus платформ.

2. Аналітичний модуль з використанням штучного інтелекту. Здатність цього модуля полягає в аналізі і в отриманих даних для виявлення аномалій або пікових навантажень. Така модель передбачає майбутнє навантаження, що готує систему до змін заздалегідь.

3. Модуль прийняття рішення. Автоматизована система на основі проаналізованих даних приймає рішення на рахунок збільшення чи зменшення інстанцій, налаштування змін конфігурації віртуальних машин, зміни напрямку трафіку на інші вузли, вимикання негативного ресурсу.

4. Модуль реалізації змін у хмарі. Компонент, який відповідає за мінімізацію ризиків. Після того, як прийняли рішення, запускається запит, який діє в залежності від зміни. Наприклад, створює новий інстанс, масштабує базу даних тощо.

5. Модель самонавчання. При отриманні нових результатів чи даних, система на основі цього постійно вдосконалюється, тобто адаптується під параметри зворотного зв'язку. В той час як зворотній зв'язок надає повний фідбек ефективності прийнятих рішень.

Ця архітектура забезпечує автоматичне виправлення помилок, покращує правила масштабування та безпеки. Яскравим прикладом моніторингу є сервіс Amazon CloudWatch, який інтегрований із штучним інтелектом, створює зворотній зв’язок на основі навантажень. Наступним кроком відбувається зміна параметрів масштабування для зменшення ризику перевантаження в подальшому.

### 3.2 Автоматизована система моніторингу кіберзагроз

Захист хмарних інфраструктур має велике значення в сфері технологій. Традиційні методи захисту не дають високого рівня безпеки, адже кібератаки постійно зростають, як і їхня складність. В даному випадку, автоматизована система моніторингу побудована на використанні штучного інтелекту, створює надійний захист. При виявленні підозрілої активності аномалія швидко нейтралізується без участі людини.

#### 3.2.1 Використання штучного інтелекту для виявлення аномалій у трафіку

Штучний інтелект має велику значимість у виявленні аномалій у мережевому трафіку. На основі стандартної поведінки користувачів або системи, моделі штучного інтелекту створюють профіль із звичайним функціоналом, але будь-які відхилення можуть означати загрозу. Наприклад, до конкретного ресурсу різко надсилається велика кількість запитів або не характерна геолокація доступів. Ці ознаки можуть свідчити про DDoS атаку або несанкціонований доступ. Щоб змоделювати такі ситуації, використовується метод кластеризації для того, щоб система могла швидко розпізнати нові загрози.

Головна мета кластеризації - виявити зашифровану структуру даних, поділяючи об’єкти на кластери, тобто групи, але поділити потрібно так, щоб межі

одного кластеру були дуже схожими один на одного. Таким чином, система ефективно знаходить закономірності без надання прикладу або маркування.

Самонавчання відбувається на основі нових даних, завдяки чому система постійно покращується. В поєднанні нових та минулих даних формується більш точна та ефективна модель захисту, яка не пропускає критичні інциденти і зменшує хибні сигнали.

### 3.2.2 Алгоритми детекції зловмисних атак

Для того, щоб виявляти атаки, системи кібербезпеки застосовують складні алгоритми. Вони не тільки розпізнають відомі атаки, але й визначають нові. Найбільш ефективними методами вважають нейронні мережі, дерева, рішень, підтримувальні фактори, градієнтний бустинг і байєсівські мережі.

Таблиця 3.1 – Методи детекції атак

| Модель | Опис |
|---|---|
| Нейронні мережі | Складається із кількох прихованих шарів, застосовується в обробці зображень, відео, мови |
| Дерева рішень | Ієрархічна модель, яка поступово поділяє дані за ознаками і доходить до кінцевого результату. Застосовується в класифікації даних. |
| Підтримувальні вектори | Розділяє дані на класи за допомогою оптимальних площин. Знаходячи найближчі точки, які називають підтримувальні вектори. Застосовується в розпізнаванні облич. |
| Градієнтний бустинг | Будується на слабких моделях, щоб кожна виправляла помилки попередньої. Застосовується в прогнозуванні фінансових ринків. |
| Байєсівські мережі | Графова модель, яка представляє ймовірні залежності між змінними. Застосовується в медичних діагнозів та системах підтримки рішень. |

Ці алгоритми дозволяють зробити аналіз поведінки користувача або системи, щоб виявити загрозу.

Наприклад, у системі програмного забезпечення було виявлено фішингову атаку. Система автоматично блокує атаку і відправляє на аналіз фахівцями з безпеки. Також у разі виявлення небезпеки, система:

- автоматично блокує користувача,

-застосовує політику обмеження доступу,

-надсилає необхідну інформацію до служби безпеки,

-вносить ситуацію в систему (SIEM).

### 3.3 Оптимізація витрат за допомогою штучного інтелекту

Штучний інтелект надає великі можливості для ефективного управління витратами. Перевага використання штучного інтелекту автоматизує оптимізацію витрат, раціонально використовує ресурси та зменшує витрати.

Нові та історичні дані дають можливість AI-системам проаналізувати використання ресурсів, визначають неефективність та рекомендують можливість стратегії розподілу ресурсів. Дане рішення знижує фінансові витрати на інфраструктуру без втрати якості надання послуг.

#### 3.3.1 Прогнозне масштабування ресурсів

Прогнозне масштабування – це процес гнучкого управління ресурсами обробки даних хмарної інфраструктури за результатами аналітичного прогнозу системного навантаження. Масштабування діє на випередження, тим самим уникаючи зайвих витрат.

Прогнозне масштабування має в наявності моделі машинного навчання, які аналізують історичні дані: активність користувачів, кількість запитів, поведінку системи, пікові навантаження. Щоб передбачити майбутнє навантаження на систему, використовують регресійні моделі, нейронні мережі або моделі часових рядів, як Prophet. Тобто система відслідковує навантаження і зберігає необхідні дані. Наприклад , якщо щосереди вранці активність стабільно зростає на 40%, то алгоритми навчаються це передбачити. За необхідний проміжок часу, система самостійно додає обчислювальні ресурси, створює сервери, запускає процес

кешування або масштабує базу даних. Коли активність зменшується - додаткові ресурси вимикаються автоматично.

Такі компанії, як Airbnb, застосовують нейронні мережі для прогнозування попиту в залежності від місцезнаходження, сезону і пристрою. Netflix також завчасно налаштовує інфраструктуру до високого навантаження під час прем'єр фільмів.

На практиці частіше всього використовують дані провайдери, як:

- Amazon Web Services Predictive Scaling - автоматично створює нові інстанси за 10-30 хвилин до навантаження.

- Google Cloud Operations Suite інтегрується із Vertex AI, щоб навчати моделі керування масштабування.

- Azure AutoScale – інструмент, який дозволяє створювати власні моделі на основі зібраних метрик.

Прогнозне масштабування передбачає переваги зниження витрат, підвищення надійності, тобто користувачі уникають проблеми із затримками, навіть коли інфраструктура перебуває у стані пікового навантаження. Ця тема дуже важлива для електронної комерції, стримінгових сервісів, банківських систем, де довіра і стабільність понад усе.

Таким чином, застосування штучного інтелекту в області прогнозного масштабування ресурсів доводить, що забезпечується інфраструктурна гнучкість та постійна доступність хмарних сервісів.

### 3.3.2 Автоматичне вимкнення невикористовуваних ресурсів

Однією із поширених причин несподіваного збільшення витрат у хмарних сервісах є неефективно використовувані ресурси, які залишені без нагляду. Ці ресурси надалі використовують обчислювальні потужності та накопичують витрати. Ця проблема часто виникає із-за людського фактора. Наприклад, адміністратори вмикають інстанси для розробки чи тестування, пізніше не вимикаючи їх. В такому випадку компоненти працюють в режимі простою.

Щоб уникнути подібних ситуацій, застосовують штучний інтелект, який в свою чергу автоматично виявляє та вимикає невикористані ресурси. Такі системи складаються з алгоритмів, які мають здатність аналізувати різні активності на сервері, сховищі, базі даних, віртуальні машини та інші компоненти, використовуючи шаблони для порівняння з поточним навантаженням.

Або якщо віртуальна машина цілодобово працює, але за крайні кілька днів не обробила жодного запиту, або її процесор чи оперативна пам'ять застосовує менше ніж 1%, вона відноситься до варіанту на вимкнення або бюджетний тариф. Дана система ефективно створює прогноз тривалості бездіяльності, за допомогою вибору оптимального часу для вимкнення.

Приклад стандартних ресурсів, які автоматично вимикаються з допомогою оптимізації:

Таблиця 3.2 – Ресурси з автоматичним вимкненням

| Тип ресурсу | Ознаки неефективності | Автоматизація системи |
|---|---|---|
| Віртуальні машини | Відсутність активних підключень, процесор < 2% | Вимкнення або перехід системи в режим “очікування” |
| База даних | Відсутність запитів, копії | Призупинення або об’єднання копій |
| Сховища(Blob Storage) | Дані не були використані більше 30 днів | Процес архівації в бюджетний варіант зберігання |
| Контейнери (Docker) | Відсутність запитів, перехід допустимого обсягу | Перезапуск з обмеженнями або зупинка |
| Балансування навантаження | Без трафіку | Пауза або вимкнення |

Щоб реалізувати ці системи в роботі, використовують такі сервіси, як:

- Amazon Web Services Trusted Advisor - проводить детальний аналіз ресурсів, створює рекомендації для невикористаних еластичних обчислювальних сервісів та блочних сховищ, сервіс реляційних баз даних.

- Google Cloud Recommender - штучний інтелект надає поради щодо економії, також використання процесору.

- Azure Cost Management та Advisor - розпізнає та припиняє використання ресурсів із низькими навантаженнями.

Окрім цього, штучний інтелект може автоматично вносити зміни до тарифного плану, обираючи більш дешевий варіант у випадку прогнозу мінімальної активності на довгий час. Такий підхід зменшує витрати без втрати якості продуктивності.

Автоматизація вимкнення надає адміністраторам значну перевагу, оскільки вони отримують звіти про дії системи, налаштовують особливі винятки.

Автоматичне вимкнення – це інструмент для підтримки оптимального стану інфраструктури без надмірних витрат.

### 3.4 Розподіл роботи на основі інтелектуальних алгоритмів

В складних хмарних сервісах в один момент працюють сотні компонентів. Щоб забезпечити високу продуктивність між ними, використовують розподіл завдань. Таким чином навантаження розподіляється рівномірно. Розподіляти ресурси самостійно в умовах непередбаченого трафіку чи піку часто вважається неефективним підходом. Тому все більше організацій переходять на розумні алгоритми розподілу на основі штучного інтелекту.

Сенс цього підходу полягає в тому, що моделі штучного інтелекту аналізують нинішні навантаження на певні компоненти системи та в реальному часі приймають рішення, куди саме направляти нові запити, обробка або збереження інформації. В результаті досягнуто рівномірний розподіл ресурсів та вузли запобігли перенавантаженню.

Розумний розподіл може включати різні параметри, одні із них:

- Стан процесору та оперативної пам'яті, - зафіксовані звіти навантажень

-пріоритет задач

-місце знаходження користувача

-правила безпеки та резерву.

Load Balancer – підхід, який не тільки рівномірно поділяє запити, а й коригує шлях, де знаходиться найбільша ймовірність швидкої обробки. На платформі Google Cloud, Load Balancing застосовує алгоритми штучного інтелекту для підлаштування до типу трафіку в даний момент. Коли Amazon Web

Services Application Load Balancer реалізує перенаправлення згідно з вмістом, що проаналізовано моделями машинного навчання.

У більш складних ситуаціях, наприклад, мікросервісна архітектура із контейнера, в даному випадку, на основі аналізу даних ,розподіляються задачі між подами. Алгоритм може передбачати навантаженість контейнера і система перенаправить задачу в інший більш вільний контейнер.

Алгоритми також беруть роль планування та балансування задач з високим споживанням ресурсів. Наприклад, при масштабній обробці даних, штучний інтелект самостійно обирає конкретне місце обробки задачі, враховуючи вартість ресурсу і продуктивність.

Розподіл задач за допомогою алгоритмів штучного інтелекту значно покращує процес забезпечення стійкості до перевантажень і результати адаптації гнучкості досягають високого рівню. Таким чином забезпечується стабільна робота додатків без участі інженерів.

**Висновки до розділу**

В даному розділі було представлено рішення штучного інтелекту, які сприяють вирішенню проблем хмарного середовища - перевантаження, неефективне використання ресурсів та кіберзагрози. Продемонстровано архітектуру системи, що поєднує в собі автоматизацію процесів моніторингу, аналізу, прийняття рішень та самостійне навчання.

Також були розглянуті механізми розпізнавання аномалій та незаконні дії за допомогою штучного інтелекту, що ефективно допомагає реагувати на кіберзагрози в даному моменті. Окрім цього, були описані підходи,які оптимізують витрати завдяки прогнозному масштабуванню і автоматизованому вимкненню ресурсів. Запропоновані методи покращують стабільність роботи та мінімізує фінансові витрати.

Інтелектуальний розподіл, який підтримується на основі алгоритмів, що дозволяє аналізувати дійсні навантаження, виставляти пріоритет задач. Цей підхід здатний до швидкої адаптації та продуктивності.

В результаті, впровадження штучного інтелекту створило перевагу в оновленні концепції управління хмарними технологіями.

# 4. РОБОТА КОРИСТУВАЧА З РОЗРОБЛЕНОЮ СИСТЕМОЮ

Основна взаємодія користувача з системою відбувається на основі запитів до бази даних, що надає можливість працювати з даними, фіксувати сесії та виявляти аномалії. Даний підхід описує акцент на структуровану базу даних, яка не тільки зберігає сесії використання, а й також інтегрує алгоритми штучного інтелекту для виявлення аномалій.

Розумні механізми, які були внесені в систему, аналізують показники в реальному часі, встановлюють великі навантаження і за допомогою записів фіксують аномалії. Саме ця реалізація системи працює через SQL-запити. Також для подальшого застосування, систему можна реалізувати додавши інтерфейс або API.

## 4.1 Робота з базою даних

База даних - це впорядкована структура для зберігання, керування та обробки даних. Вона дозволяє систематизувати велику кількість інформації об’єднавши в єдину базу. База даних має перевагу у швидкості введення та застосування необхідної інформації. Щоб швидко знайти потрібну інформацію, використовують спеціальні алгоритми в базі даних.

База зберігає:

- користувачів системи
- сесії доступу
- метрики
- чіткі значення метрик
- аномалії, знайдені AI-модулем

Таблиця в реляційній базі даних - це структура, яка зберігає необхідну інформацію, впорядковує її та фільтрує.

```sql
-- Користувачі
CREATE TABLE users (
    id SERIAL PRIMARY KEY,
    email VARCHAR(100) NOT NULL UNIQUE,
    password_hash VARCHAR(255) NOT NULL,
    role VARCHAR(50) DEFAULT 'user' -- admin / user
);

-- Сесії
CREATE TABLE sessions (
    id SERIAL PRIMARY KEY,
    user_id INT REFERENCES users(id),
    start_time TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    ip_address VARCHAR(45),
    device_info VARCHAR(100) -- браузер, ОС
);

-- Метрики
CREATE TABLE metrics (
    id SERIAL PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    unit VARCHAR(20),
    threshold FLOAT DEFAULT NULL -- для автоматичного виявлення аномалій
);

-- Значення метрик
CREATE TABLE metric_values (
    id SERIAL PRIMARY KEY,
    session_id INT REFERENCES sessions(id),
    metric_id INT REFERENCES metrics(id),
    value FLOAT NOT NULL,
    recorded_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

Рисунок 4.1 – Таблиця баз даних

```sql

-- Аномалії
CREATE TABLE anomalies (
    id SERIAL PRIMARY KEY,
    session_id INT REFERENCES sessions(id),
    metric_id INT REFERENCES metrics(id),
    anomaly_type VARCHAR(100),
    description TEXT,
    severity VARCHAR(50) DEFAULT 'medium', -- low / medium / high
    detected_by_ai BOOLEAN DEFAULT TRUE,
    detected_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

Рисунок 4.2 – Таблиця баз даних

Призначення кожної таблиці:

**users** - зберігає особисту інформацію користувачів, у тому числі унікальну електронну адресу та надійний пароль. Може також зберігати про положення користувача (“user” або ”admin”).

**sessions** - записує всі активні сеанси користувачів: час входу та виходу, IP-адресу, тип браузера чи пристрою. Таким чином відстежується активність та рівень безпеки значно підвищується.

**metrics** - інформаційна таблиця, в якій знаходять назви метрик та одиниці вимірювання. В доповнення може містити граничні значення для того, щоб автоматично виявляти аномалії.

**metric_values** - зберігає важливі значення метрик в рамках кожного сеансу. На основі цих даних, які постійно оновлюються, будуються звіти.

**anomalies** - реєструє відхилення від норми, які можуть бути виявлені не тільки штучним інтелектом, а й користувачем. В реєстрації про відхилення описано аномалію, її серйозність час виникнення та тип.

```
46  SELECT table_name
47  FROM information_schema.tables
48  WHERE table_schema='public';
49
50  INSERT INTO users (email, password_hash, role)
51  VALUES ('test@example.com', 'hashed_password_123', 'admin');
```

Рисунок 4.3 – Додаваня користувача

Додавання нового користувача  в сесію з ID = 1. Фіксується IP- адреса та тип пристрою, з якого було здійснено вхід. Таким чином зберігається історія активності користувача для аналітики та безпеки.

```
55  -- Створення сесії
56  INSERT INTO sessions (user_id, ip_address, device_info)
57  VALUES (1, '192.168.0.10', 'Chrome on Windows');
```

Рис. 4.4 - Створення сесії

```
59  -- Додавання метрик
60  INSERT INTO metrics (name, unit, threshold)
61  VALUES
62  ('CPU usage', '%', 85),
63  ('RAM usage', 'MB', 4000),
64  ('Disk I/O', 'MB/s', 100);
```

Рис. 4.4 - Додавання метрик

Створює інформаційні записи для метрик. Після чого, до кожної метрики додається вимірювання і граничне значення, якщо воно вище норми, система зафіксує аномалію.

```sql
-- Додавання значення метрик
INSERT INTO metric_values (session_id, metric_id, value)
VALUES
(1, 1, 92.3), -- CPU (аномалія)
(1, 2, 3600), -- RAM
(1, 3, 110);  -- Disk I/O (аномалія)
```

Рис. 4.5 - Додавання метрик

Метрики набувають конкретних значень,які фіксуються під час сесії. Наприклад, процесор використовується на 91,4 % - це перевищує норму, що є важливо для наступних аналізів.

```sql
INSERT INTO anomalies (session_id, metric_id, anomaly_type, description, severity)
VALUES
(1, 1, 'CPU spike', 'CPU usage exceeded 85%', 'high'),
(1, 3, 'Disk I/O overload', 'Disk throughput exceeded threshold', 'medium');
```

Рис. 4.6 - Виявлення аномалій

Робить записи виявлених аномалій на основі перевищення порогового значення. Зберігається точний опис аномалії її тип та рівень критичності (high або low).

```sql
SELECT * FROM sessions;
SELECT * FROM metrics;
SELECT * FROM metric_values;
SELECT * FROM anomalies;
```

Рис. 4.7 - Перевірка даних

Ці запити дають змогу переглянути всі наявні записи у таблицях. Це дуже корисно на етапі тестування, щоб звірити дані та їх зв’язки.

```sql
-- Середнє використання CPU
SELECT AVG(value) FROM metric_values WHERE metric_id = 1;
```

Рис. 4.8 - Аналітичні SQL-запити

Обчислює середнє навантаження процесора за всі сеанси. Оцінює загальну продуктивність системи.

```sql
-- Кількість аномалій по кожній метриці
SELECT metric_id, COUNT(*) FROM anomalies GROUP BY metric_id;
```

Рис. 4.9 - Підрахунок аномалій

Підраховує кількість кожного типу аномалій. Таким чином, аналізуються проблеми, які виникають частіше.

```sql
-- Максимальне значення RAM
SELECT MAX(value) FROM metric_values WHERE metric_id = 2;

SELECT recorded_at, value FROM metric_values WHERE metric_id = 1 ORDER BY recorded_at;
```

Рис. 4.10 - Підрахунок максимального значення

Показує максимальне значення оперативної пам’яті, що є показником витоку пам’яті або високе навантаження.

```
+----+------------------+----------------------+-------+
| id | email            | password_hash        | role  |
+----+------------------+----------------------+-------+
|  1 | test@example.com | hashed_password_123 | admin |
+----+------------------+----------------------+-------+

+----+---------+---------------------+--------------+--------------------+
| id | user_id | start_time          | ip_address   | device_info        |
+----+---------+---------------------+--------------+--------------------+
|  1 |       1 | 2025-05-27 22:42:48 | 192.168.0.10 | Chrome on Windows |
+----+---------+---------------------+--------------+--------------------+

+----+-----------+------+-----------+
| id | name      | unit | threshold |
+----+-----------+------+-----------+
|  1 | CPU usage | %    |        85 |
|  2 | RAM usage | MB   |      4000 |
|  3 | Disk I/O  | MB/s |       100 |
+----+-----------+------+-----------+

+----+------------+-----------+-------+---------------------+
| id | session_id | metric_id | value | recorded_at         |
+----+------------+-----------+-------+---------------------+
|  1 |          1 |         1 |  92.3 | 2025-05-27 22:42:48 |
|  2 |          1 |         2 |  3600 | 2025-05-27 22:42:48 |
|  3 |          1 |         3 |   110 | 2025-05-27 22:42:48 |
+----+------------+-----------+-------+---------------------+
```

Рис. 4.11 – Результат

```
+----+------------+-----------+-----------------+-----------------------------------+----------+---------------+---------------------+
| id | session_id | metric_id | anomaly_type    | description                       | severity | detected_by_ai | detected_at         |
+----+------------+-----------+-----------------+-----------------------------------+----------+---------------+---------------------+
|  1 |          1 |         1 | CPU spike       | CPU usage exceeded 85%            | high     |             1 | 2025-05-27 22:42:48 |
|  2 |          1 |         3 | Disk I/O overload | Disk throughput exceeded threshold | medium |             1 | 2025-05-27 22:42:48 |
+----+------------+-----------+-----------------+-----------------------------------+----------+---------------+---------------------+
+-------------------+
| AVG(value)        |
+-------------------+
| 92.30000305175781 |
+-------------------+
+-----------+----------+
| metric_id | COUNT(*) |
+-----------+----------+
|         1 |        1 |
|         3 |        1 |
+-----------+----------+
+------------+
| MAX(value) |
+------------+
|       3600 |
+------------+
+---------------------+-------+
| recorded_at         | value |
+---------------------+-------+
| 2025-05-27 22:42:48 |  92.3 |
+---------------------+-------+
```

Рис. 4.12 – Результат

## 4.2 Основні сценарії використання

Для кращого розуміння концепції створеної системи, а саме принципів взаємодії користувачів із системою та механізмом доступу до окремих компонентів бази даних, було використано візуальні засоби моделювання. За їх допомогою побудовано відповідні діаграми, які наочно і структуровано демонструють логічні зв'язки між об'єктами системи, взаємодії між ними, а також процеси, що відбуваються під час користування функціоналом системи. **ER-Діаграма** – це схема, на якій описана структура бази даних у вигляді таблиць, тобто сутності. Ці сутності мають атрибути та зв'язки. Наприклад, кожна сесія прив'язана до певного користувача, в той час як метрика може бути пов'язана з багатьма сеансами.

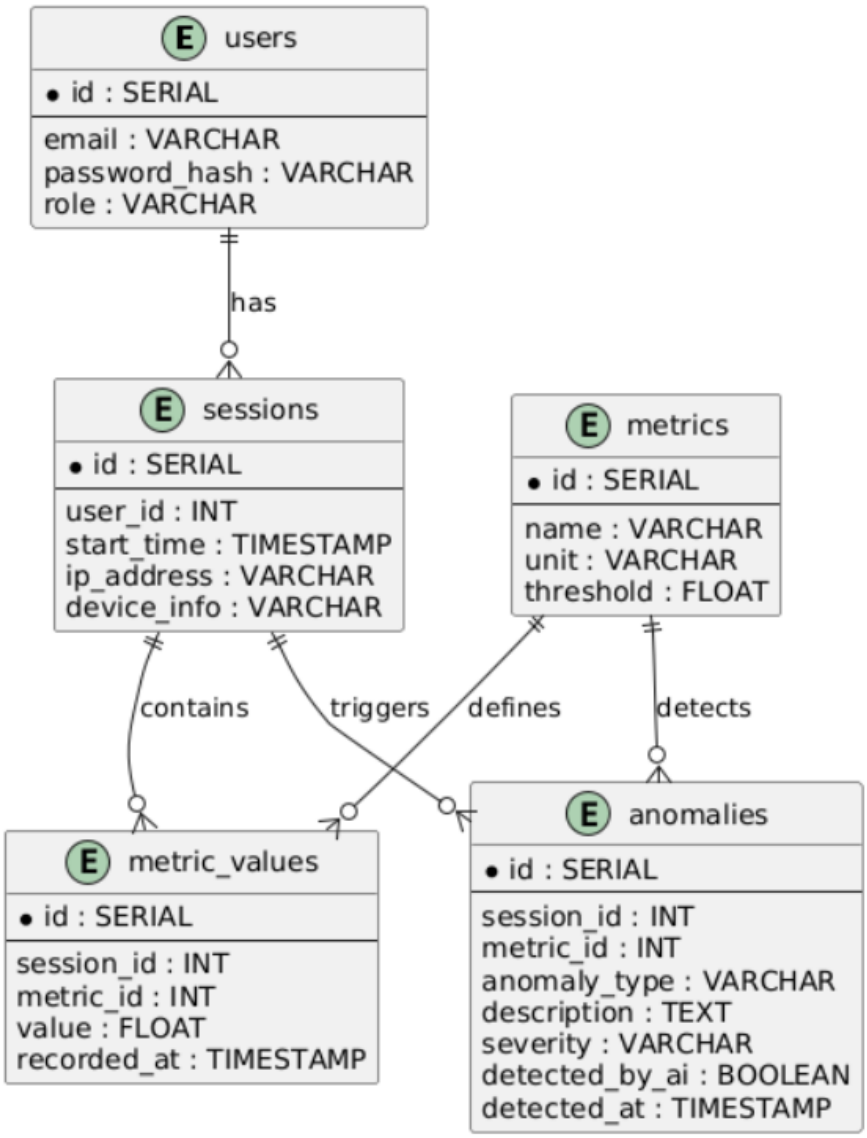



Рис. 4.13 - Діаграма сутності

Представлена діаграма показує ключові сутності системи, їх атрибути, включаючи зв'язки між компонентами.

1. Users.

- Зберігає особисту інформацію з зареєстрованих користувачів

- Поля: email, password_hash, role.

- Має зв'язок один до багатьох із таблиці Sessions.

2. Sessions

- Представляє кожну сесію користувача: час початку, IP адреса, пристрій

- Має зв'язки з таблицями Users, Metric_Value та Anomalies.

3. Metrics.

- Містить перелік метрик (наприклад, процесор, оперативна пам'ять і порогові значення)

- Пов'язані з Metric_Values» і Anomalies.

4. Metric_Values

- Зберігає конкретні значення метрик у межах певної сесії,

- Зв'язана із sessions, до якої належать метрики, і metrics, які саме метрикою вмірювались.

5. Anomalies

- Фіксує виявлені аномалії тип, опис, серйозність.

- Вказує, у якій сесії і у якій метриці було виявлено проблему.

Опис зв’язків:

Users → Sessions - Один користувач може мати багато сесій.

Sessions → Metric_Values - Одна сесія може вмістити багатозначний метрик.

Metrics → Metric_Values - Одна метрика використовується у багатьох записах.

Sessions → Anomalies - Аномалії виникають у конкретних сесіях. Metrics→ Anomalies - Кожна аномалія пов'язана з певною метрикою.

Основний принцип роботи:

1. Користувач реєструється в системі, його дані потрапляють у таблицю Users. Далі кожен користувач має свій “email”, зашифрований “password_hash” та роль.

2. Створення нової сесії при вході, яка зберігається у таблиці “sessions”. Фіксується час початку, IP-адресу та інформацію про пристрій.

3. Коли сесія почалась, відбувається збір метрик, це навантаження на процесор та використання оперативної пам’яті. Всі метрики описані в таблиці “metrics”.

4. Особливі значення метрик зберігаються у таблиці “metric_values”. Кожне з яких значення підв’язане до “sessions” і “metrics”? таким чином відслідковуються час активності метрики.

5. Якщо значення метрики перевищує норми, то система визначає це як аномалію. Фіксація аномалії записується в таблицю “anomalies”, де вказується тип.

Таблиця 4.1 – Події діаграми сутності

| Крок | Подія | Запис в таблиці |
|---|---|---|
| 1 | Авторизація | users |
| 2 | Початок нової сесії | sessions |
| 3 | Вимірювання системних показників | metrics, metric_values |
| 4 | Аналіз метрик | |
| 5 | Виявлення аномалій | anomalies |
| 6 | Формування звіту | |

**Діаграма зв’язків** - демонструє логічну структуру зв’язків між таблицями в базі даних.

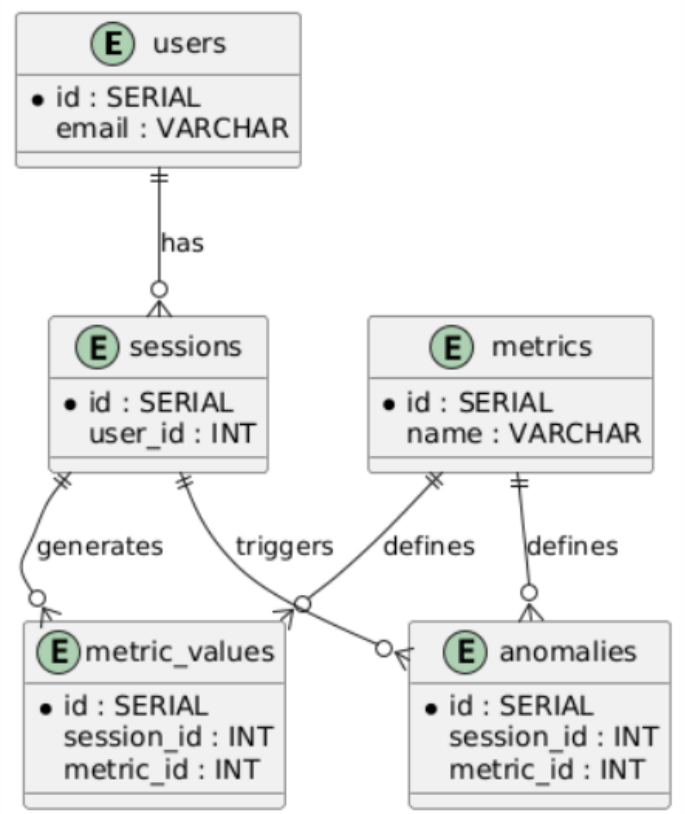



Рис. 4.14 - Діаграма зв’язків

1. Певний користувач створює велику кількість сесій.
2. Кожна сесія містить в собі багато значень метрик.
3. Таблиця “metrics” є інформаційною, тому робить опис показників, які вимірюються конкретними одиницями та порогами.
4. В результаті перевищення визначених меж утворюються аномалії.

**Діаграма послідовності** - визначає поетапну взаємодію користувача із системою в певному часі. Вона демонструє, які об’єкти обмінюються запитами та відповідями.

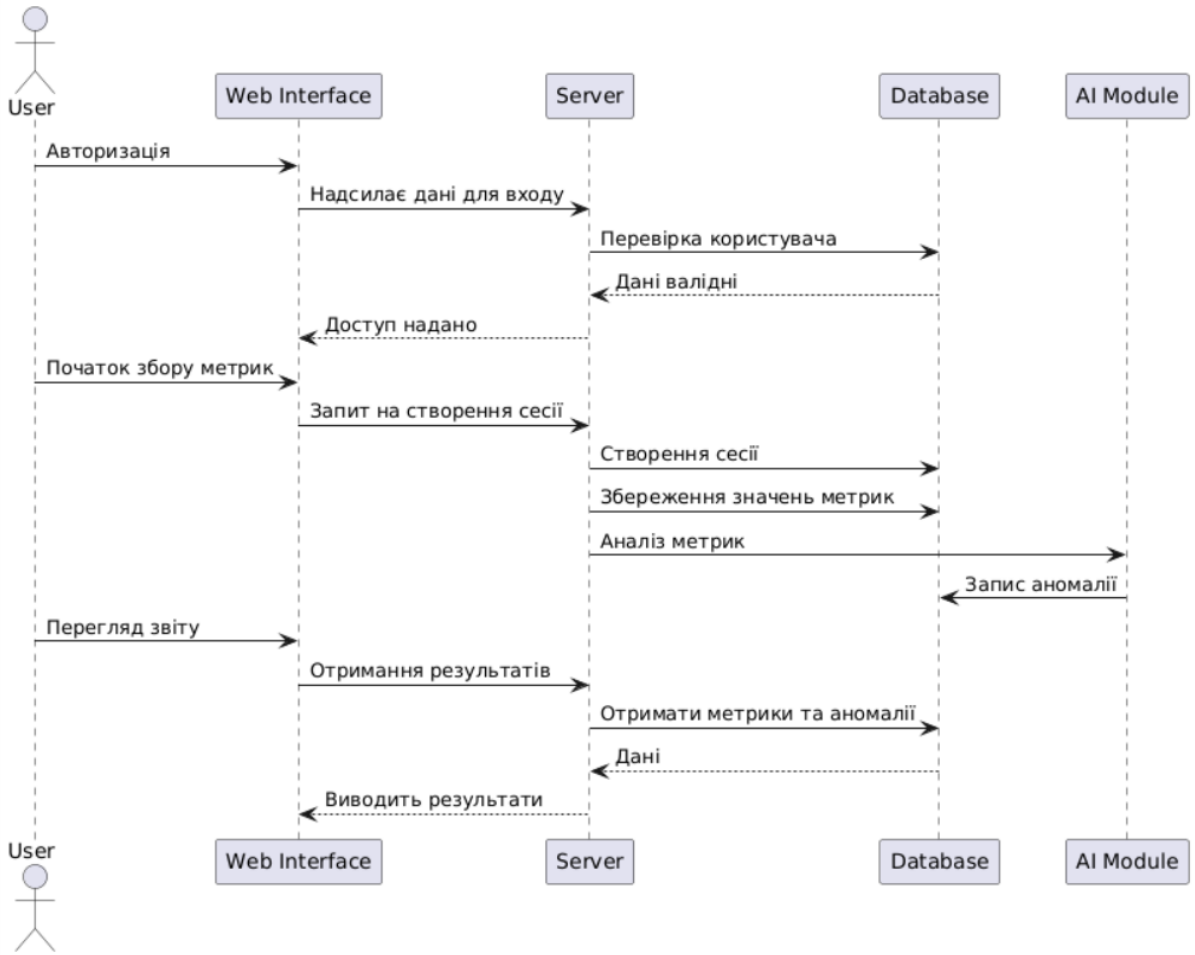



Рис. 4.15 - Діаграма послідовності

1. Користувач авторизується через веб-інтерфейс. Дані проходять перевірку у базі.
2. За умови успішної авторизації, користувач запускає процес моніторингу.
3. На сервері створюється сесія в базу даних та починається збір метрик.
4. При виявленні аномалій, вони фіксуються та записуються в базу.
5. Користувач отримує звіт.

## 4.3 Інтеграції з хмарними платформами

Інтеграція розроблених систем з хмарною платформою забезпечує масштабованість, надійність та гнучкість у доставці та обслуговуванні.

Багато процесів можна автоматизувати за допомогою хмарних служб, наприклад: резервне копіювання, моніторинг, обробка подій або забезпечення безпеки.

Розгортання бази даних у хмарі забезпечує ефективну доступність та масштабованість. Сервіси, які володіють даними можливостями, це:

- Amazon Relations Database Service, популярний в підтримці бази даних на платформах MySQL, PostgreSQL, Oracle, автоматично створює резервні копії, масштабує та оновлює без електроніки.

- Microsoft Azure SQL Database – це керована система управління баз даних, яка автоматично масштабує, надає високу доступність та налаштовані механізми безпеки.

- Google Cloud SQL. Підтримує SQL Server, PostgreSQL і MySQL. Застосовує інтеграцію з різними сервісами Google, автоматично створює резервні копії.

Такі сервіси спрощують та вдосконалюють використання системи без значних втрат ресурсів на обслуговування інфраструктури.

Серверлес-функції надають можливість для реалізації обробки інформації та фіксацію аномалій у реальному часі.

- Amazon Web Services Lambda.Якщо в базі даних відбулись зміни або надійшли нові дані, платформа запускає код в якості реакції на події.

- Azure Functions підтримує інтеграцію з багатьма сервісами, створює функції для реагування на події або запити.

- Google Cloud Functions створює функції, які мають реакцію на певні події в Cloud Storage, Pub/Sub та інші. Автоматично масштабує та підтримує інтеграцію з іншими сервісами Google.

Щоб забезпечити надійний та контрольований доступ у системи, виконується інтеграція через API-шлюзи.

-Amazon Web Services API Gateway - Створює, захищає і підтримує RESTful і WebSocket API. Інтегрований з платформами Amazon Web Services Lambda та іншими сервісами.

Такі сервіси спрощують та вдосконалюють використання системи без значних втрат ресурсів на обслуговування інфраструктури.

Серверлес-функції надають можливість для реалізації обробки інформації та фіксацію аномалій у реальному часі.

- Amazon Web Services Lambda.Якщо в базі даних відбулись зміни або надійшли нові дані, платформа запускає код в якості реакції на події.

- Azure Functions підтримує інтеграцію з багатьма сервісами, створює функції для реагування на події або запити.

- Google Cloud Functions створює функції, які мають реакцію на певні події в Cloud Storage, Pub/Sub та інші. Автоматично масштабує та підтримує інтеграцію з іншими сервісами Google.

Щоб забезпечити надійний та контрольований доступ у системи, виконується інтеграція через API-шлюзи.

-Amazon Web Services API Gateway - Створює, захищає і підтримує RESTful і WebSocket API. Інтегрований з платформами Amazon Web Services Lambda та іншими сервісами.

- Azure API Management створює, публікує та контролює API, при цьому є дотримання політики безпеки, обмеження швидкості та аналізи використання.

- Google Cloud Endpoints. Використовує OpenAPI для створення і моніторингу API, має інтеграцію з Cloud Functions, App Engine та інші сервіси.

Для підтримки стабільної роботи необхідно володіти оперативним реагуванням на проблеми, ефективними інструментами моніторингу вважають: - Amazon CloudWatch. Відслідковує журнал подій, збирає метрики, фіксує події, може налаштувати сповіщення та автоматизовувати дії у відповідь на непередбачувані ситуації.

- Azure Monitor займається збором аналізу та візуалізацією даних з ресурсів. - Google Cloud Monitoring за допомогою сповіщень,на основі метрик, надає можливості моніторингу.

**Висновки до розділу**

Хмарні платформи з можливістю інтеграції мають великі переваги в гнучкості, масштабованості та надійності. Керовані сервіси, серверлес- функції,

API-шлюзи покращують функціональність системи без потреби великих фінансових вкладень на обслуговування інфраструктури. Вибір сервісу залежить від потреб та вимог організації і проєкту.

# ВИСНОВКИ

На основі результатів написання цієї дипломної роботи розроблено інтелектуальне рішення для підвищення продуктивності хмарних додатків з використанням методів штучного інтелекту.

Запропонована система дозволяє виявляти аномалії в

навантаженні, прогнозувати потреби в ресурсах і автоматично масштабувати хмарну інфраструктуру. Такий підхід допомагає зменшити затримку, оптимізувати витрати та підвищити стабільність цифрових сервісів.

На початку роботи було проаналізовано поточний стан хмарних обчислень і визначено ключові проблеми: кіберзагрози, неефективне використання ресурсів, труднощі з масштабуванням і прогнозуванням навантаження. Було розглянуто методи штучного інтелекту, які могли б усунути вищезазначені проблеми, та обґрунтували вибір відповідних інструментів для впровадження системи (зокрема з використанням модулів штучного інтелекту, баз даних, інтерфейсів та хмарних платформ AWS, Azure, Google Cloud.

У процесі впровадження було створено архітектуру системи, яка об’єднує збір метрик, структуроване зберігання даних бази даних, автоматичне виявлення аномалій та вихідним результатом системи - звіт.

Розроблено базу даних, що містить інформацію про користувачів, сесії, значення метрик і виявлені відхилення.

Створено ER діаграми та діаграми послідовності для візуалізації сценаріїв використання системи.

Система тестувалася в умовах змінного навантаження, що дозволило перевірити її стабільність, точність виявлення критичних подій та ефективність ресурсозбереження.

Проаналізовано інтеграцію рішення з хмарними сервісами та розглянуто можливі сценарії подальшого розширення функціональності, зокрема створення повноцінного інтерфейсу

користувача.

Результати цієї роботи можуть стати основою для подальших досліджень у сферах хмарних обчислень, кібербезпеки, DevOps та управління IT-інфраструктурою.

# ПЕРЕЛІК ПОСИЛАНЬ


1. Вовк, Ірина Петрівна, and Ирина Петровна Вовк. *"Проблеми автоматизації управління ресурсами підприємства засобами ERP-систем."* (2011).
2. Погорєлий, М. А. *"Прогнозування властивостей багатокомпонентних сплавів з використанням алгоритмів штучного інтелекту."* (2020).
3. Колодчак, О. "Сучасні методи виявлення аномалій в системах виявлення вторгнень." *Вісник Національного ун-т «Львівська політехніка». Комп'ютерні системи та мережі* 745 (2012): 98-104.
4. Корєнев, Олександр Олександрович. *"Система моніторингу продуктивності роботи додатків у хмарних середовищах."* (2021).
5. ЯРОВОЙ, Тихон Сергійович. *"Можливості та ризики використання штучного інтелекту в публічному управлінні." Economic Synergy* 2 (2023): 36-47.
6. Гасанов, Вадим Анверович. *"Мінімізація ризиків інформаційної безпеки корпорації на основі хмарних обчислень."* (2018).
7. Дубина, В. П. "РОЗРОБКА ТА ВПРОВАДЖЕННЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ МОНІТОРИНГУ ТА ПРОГНОЗУВАННЯ КІБЕРЗАГРОЗ ДЛЯ ЗАХИСТУ ІНФОРМАЦІЙНИХ РЕСУРСІВ ВИЩИХ НАВЧАЛЬНИХ ЗАКЛАДІВ УКРАЇНИ." *Інформаційні технології і автоматизація–2024/Матеріали XVII міжнародної науково-практичної конференції. Одеса, 31 жовтня-1 листопада 2024 р.-Одеса, Видавництво ОНТУ, 2024 р.–847 с. Збірник включає матеріали доповідей учасників конференції, які об'єднані за тематичними напрямками конференції.* (2024): 190.
8. Грицак, А. В., Я. Ю. Яремчук, and В. М. Білоус. "ПІДВИЩЕННЯ СТІЙКОСТІ ВІРТУАЛЬНИХ СЕРВЕРІВ ДО DDOS-АТАК НА ОСНОВІ МАСШТАБУВАННЯ ОБЧИСЛЮВАЛЬНИХ РЕСУРСІВ КЛАСТЕРА." *Тези доповідей* (2023): 83.
9. Patterson, Scott. *Learn AWS Serverless Computing: A Beginner's Guide to Using AWS Lambda, Amazon API Gateway, and Services from Amazon Web Services*. Packt Publishing Ltd, 2019.

10. Kumar, Varun, and Ketan Agnihotri. *Serverless computing using Azure Functions: Build, deploy, automate, and secure serverless application development with Azure Functions (English Edition)*. BPB Publications, 2021.
11. Олексюк, Василь Петрович. "Упровадження технологій хмарних обчислень як складових ІТ-інфраструктури ВНЗ." *Інформаційні технології і засоби навчання* 41, вип. 3 (2014): 256-267.
12. Стельмащук, Р. Ю., & Булатецька, Л. В. (2022). Огляд хмарних сервісів для побудови ER-діаграм. *Математика. Інформаційні технології. Освіта*.
13. Конопліцька-Слободенюк, О. К., and А. Я. Клюй. "ОГЛЯД ІНСТРУМЕНТІВ ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ ОПТИМІЗАЦІЇ РОБОТИ З БАЗАМИ ДАНИХ." *Тези доповідей* (2025): 39.
14. Shah, Harshal. "Cloud Computing And Next-Generation AI-Creating The Intelligence Of The Future." *INTERNATIONAL RESEARCH JOURNAL OF ENGINEERING & APPLIED SCIENCES* 6.3 (2018): 10-55083.
15. Kampars, Jānis, and Krišjānis Pinka. "Auto-scaling and adjustment platform for cloud-based systems." *ENVIRONMENT. TECHNOLOGIES. RESOURCES. Proceedings of the International Scientific and Practical Conference*. Vol. 2. 2017.
16. Ковбас, Галина Іванівна, and Ірина Валеріївна Прокопенко. "Інтелектуальні алгоритми та аналітика даних у сучасних маркетингових системах." *Академічні візії* 41 (2025).
17. González-Granadillo, Gustavo, Susana González-Zarzosa, and Rodrigo Diaz. "Security information and event management (SIEM): Analysis, trends, and usage in critical infrastructures." *Sensors* 21.14 (2021): 4759.
18. Xin, Reynold S., et al. "Shark: SQL and rich analytics at scale." *Proceedings of the 2013 ACM SIGMOD International Conference on Management of data*. 2013.
19. Підгорна, А. О. "Використання інформаційно-комунікаційних технологій в процесі виконання проектів з архітектурного проектування." *Науковий часопис НПУ імені МП Драгоманова. Серія 2: Комп'ютерно-орієнтовані системи навчання* 15 (2015): 160-164.

20. Бідюк, Петро Іванович, et al. "Системи і методи підтримки прийняття рішень." (2022).

**ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ(Презентація)**

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ**

НА ТЕМУ

# ““ШТУЧНИЙ ІНТЕЛЕКТ У ЗАБЕЗПЕЧЕННІ ВИСОКОЇ ПРОДУКТИВНОСТІ ХМАРНИХ ДОДАТКІВ””

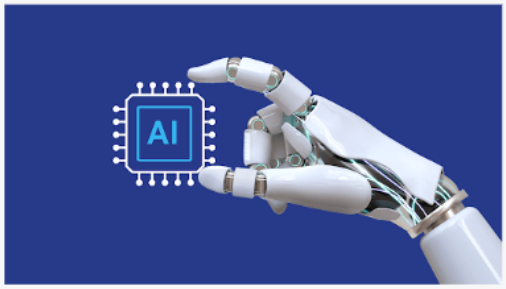
**ВИКОНАЛА: ЗДОБУВАЧ ВИЩОЇ ОСВІТИ ГР.ІСД-42**
**АНАСТАСІЯ ДМИТРІЄВА**
**КЕРІВНИК: Д.Т.Н.,ПРОФЕСОР КАМІЛА СТОРЧАК**



## АКТУАЛЬНІСТЬ ТЕМИ

Штучний інтелект - додатковий інструмент, який може автоматично виявляти аномалії, прогнозувати навантаження та адаптувати систему в режимі реального часу.

**Мета роботи** — створити базу даних для хмарного застосунку, що використовує AI для аналізу навантаження та фіксації аномалій.




**Основні завдання:**

- Вивчити вимоги до зберігання метрик;
- Побудувати ER-діаграму системи;
- Реалізувати базу даних і протестувати її;
- Візуалізувати взаємодію користувача

Актуальність теми полягає в необхідності забезпечення стабільної та безпечної роботи хмарних сервісів в умовах зростання навантаження.

# АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА ТЕХНОЛОГІЙ

**Типи хмар:**
- публічні
- приватні
- гібридні
- мультихмари

**Популярні моделі:**
- IaaS
- PaaS
- SaaS


Серед провідних платформ — **AWS, Azure і Google Cloud**, які інтегрують аналітику, автоматизацію та штучний інтелект.

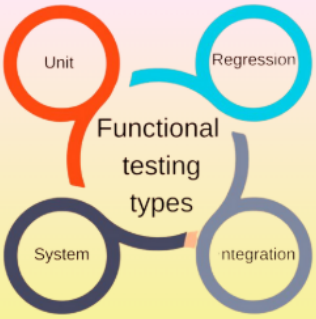
**Сучасні проблеми:**
- зростання навантаження
- необхідність використання ресурсів
- кіберзагрози
- спеціалісти

**Рішення:**
- застосування SIEM-систем
- моделі FinOps
- low-code/no-code платформи



# АРХІТЕКТУРА СИСТЕМИ, ІНТЕЛЕКТУАЛЬНА ОПТИМІЗАЦІЯ ТА РЕЗУЛЬТАТИ ТЕСТУВАННЯ

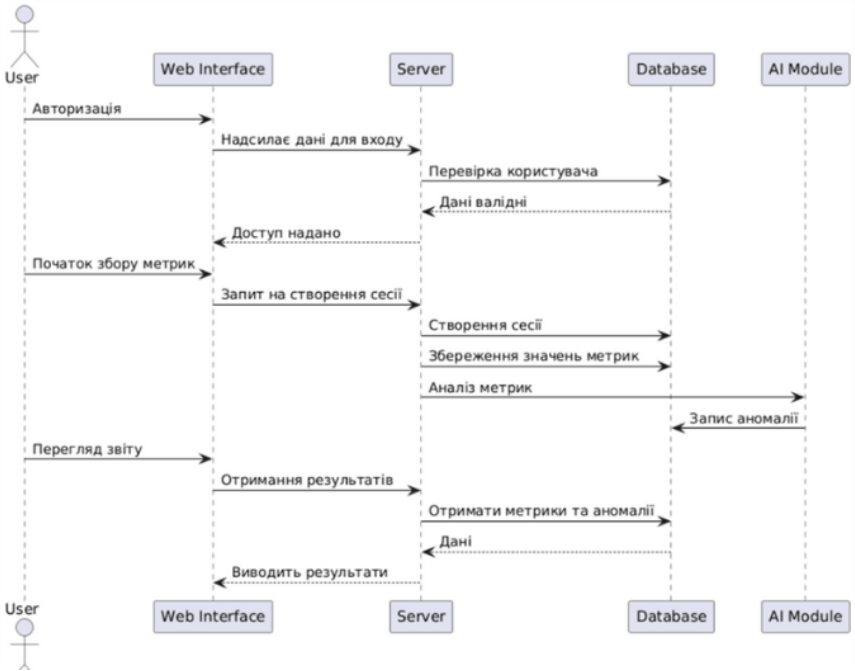
Розроблена система складається з чотирьох ключових компонентів:
- інтерфейсу користувача
- серверної частини
- бази даних
- модулі машинного навчання.




Було проведено функціональне та навантажувальне тестування системи. Результати показали високу продуктивність і стабільність навіть при пікових навантаженнях.



# ВЗАЄМОДІЯ КОРИСТУВАЧА ІЗ СИСТЕМОЮ




**Головні етапи:**
- Авторизація
- Запуск сесії збору метрик
- Аналіз даних
- Визначення можливих аномалій

```sql
CREATE TABLE users (
    id SERIAL PRIMARY KEY,
    email VARCHAR(100) NOT NULL UNIQUE,
    password_hash VARCHAR(255) NOT NULL
);

CREATE TABLE sessions (
    id SERIAL PRIMARY KEY,
    user_id INT REFERENCES users(id),
    start_time TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    ip_address VARCHAR(45)
);

CREATE TABLE metrics (
    id SERIAL PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    unit VARCHAR(20) -- наприклад: %, MB, ms
);

CREATE TABLE metric_values (
    id SERIAL PRIMARY KEY,
    session_id INT REFERENCES sessions(id),
    metric_id INT REFERENCES metrics(id),
    value FLOAT NOT NULL,
    recorded_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

CREATE TABLE anomalies (
    id SERIAL PRIMARY KEY,
    session_id INT REFERENCES sessions(id),
    anomaly_type VARCHAR(100),
    description TEXT,
    detected_by_ai BOOLEAN DEFAULT TRUE,
    detected_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

5

- **users** — зберігає інформацію про користувачів
- **sessions** — фіксує сеанси роботи користувачів з IP-адресою та часом початку.
- **metrics** — довідкова таблиця з назвами метрик, які ми вимірюємо (наприклад, CPU або RAM).
- **metric_values** — містить усі виміряні значення для кожної сесії та кожної метрики.
- **anomalies** — зберігає інформацію про виявлені відхилення у роботі системи, включаючи опис і час.

6

```sql
-- Користувач
INSERT INTO users (email, password_hash)
VALUES ('newuser@example.com', 'secure_hash_002');


-- Сесія користувача (user_id = 1)
INSERT INTO sessions (user_id, ip_address)
VALUES (1, '10.0.0.1');

-- Метрики (CPU, RAM)
INSERT INTO metrics (name, unit) VALUES
('CPU usage', '%'),
('RAM usage', 'MB');

-- Значення метрик у сесії (session_id = 1)
INSERT INTO metric_values (session_id, metric_id, value)
VALUES
(1, 1, 67.5),    -- CPU usage
(1, 2, 4096);    -- RAM usage

-- Аномалія, виявлена ШІ
INSERT INTO anomalies (session_id, anomaly_type, description)
VALUES
(1, 'CPU spike', 'CPU usage exceeded 90% for 3+ minutes');
```

1) Додавання нового користувача
2) Створення сесії з IP-адресою
3) Додавання показників використання RAM і CPU
4) запис сесії  в таблицю з метриками
5) Визначення аномалії

# ПРОВЕДЕНІ ДОСЛІДЖЕННЯ

```
+----+---------------------+-----------------+
| id | email               | password_hash   |
+----+---------------------+-----------------+
|  1 | newuser@example.com | secure_hash_002 |
+----+---------------------+-----------------+
+----+---------+---------------------+------------+
| id | user_id | start_time          | ip_address |
+----+---------+---------------------+------------+
|  1 |       1 | 2025-05-15 05:57:45 | 10.0.0.1   |
+----+---------+---------------------+------------+
+----+-----------+------+
| id | name      | unit |
+----+-----------+------+
|  1 | CPU usage | %    |
|  2 | RAM usage | MB   |
+----+-----------+------+
+----+------------+-----------+-------+---------------------+
| id | session_id | metric_id | value | recorded_at         |
+----+------------+-----------+-------+---------------------+
|  1 |          1 |         1 |  67.5 | 2025-05-15 05:57:45 |
|  2 |          1 |         2 |  4096 | 2025-05-15 05:57:45 |
+----+------------+-----------+-------+---------------------+
+----+------------+--------------+-----------------------------------+---------------+---------------------+
| id | session_id | anomaly_type | description                       | detected_by_ai | detected_at         |
+----+------------+--------------+-----------------------------------+---------------+---------------------+
|  1 |          1 | CPU spike    | CPU usage exceeded 90% for 3+ minutes |             1 | 2025-05-15 05:57:45 |
+----+------------+--------------+-----------------------------------+---------------+---------------------+
```

В результаті ми отримуємо результат успішного наповнення та зчитування даних із бази даних інформаційної системи моніторингу продуктивності.



# ВИСНОВОК

В результаті виконання бакалаврської роботи було проведено аналіз вимог до зберігання метрик у хмарному середовищі, що дало змогу сформувати основу для побудови системи. Розроблено ER-діаграму, яка відображає логіку взаємодії між сутностями. Реалізована база даних протестована на коректність структури та функціонування. Окрім того, створено візуалізацію взаємодії користувача з системою, що дозволяє чітко простежити процес обробки даних та виявлення аномалій.

Отримані результати підтверджують доцільність використання запропонованого підходу для побудови продуктивної та масштабованої системи.

# АПРОБАЦІЯ




ТЕМА

СИСТЕМИ ШТУЧНОГО ІНТЕЛЕКТУ У ЗАБЕЗПЕЧЕННІ ВИСОКОЇ ПРОДУКТИВНОСТІ ХМАРНИХ ДОДАТКІВ

ст. 382