Міністерство освіти і науки України
Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій
Кафедра кібербезпеки та захисту інформації

ДОПУСТИТИ ДО ЗАХИСТУ:
В.о. завідувача кафедри
кібербезпеки
та захисту інформації
_____ Іван ПАРХОМЕНКО
«__» _____ 2025 р.

ПОЯСНЮВАЛЬНА ЗАПИСКА

кваліфікаційної роботи

галузь знань *12 Інформаційні технології*
(шифр і назва галузі знань)

спеціальність *125 Кібербезпека та захист інформації*
(код і назва спеціальності)

освітній ступень *магістр*

освітньо-наукова програма *Кібербезпека*
(назва освітньої програми)

на тему: « Технологія виялення аномалій у хмарних середовищах на основі штучного інтелекту»

Виконавець: студент ІІ курсу, групи КБм-22

_____ **Владислав ВАСИЛЕНКО**
(підпис) (Ім'я, ПРІЗВИЩЕ)

	Ім'я, ПРІЗВИЩЕ	**Підпис**
Науковий керівник	Микола БРАІЛОВСЬКИЙ	
Нормоконтроль	Іван БІЛОКОНЬ	

Київ 2025

Міністерство освіти і науки України
Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій
Кафедра кібербезпеки та захисту інформації

ЗАТВЕРДЖЕНО:
В.о. завідувача кафедри
кібербезпеки
та захисту інформації
_____ Іван ПАРХОМЕНКО
«25» жовтня 2024 р.

ЗАВДАННЯ
на виконання кваліфікаційної роботи

спеціальності *125 Кібербезпека та захист інформації*
(код і назва спеціальності)

освітній ступень *магістр*

Здобувача(ки) КБм-22 (група) Василенка Владислава Юрійовича (прізвище ім'я по-батькові)

Тема кваліфікаційної роботи Технологія виявлення аномалій у хмарних середовищах на основі штучного інтелекту

1. ПІДСТАВИ ДЛЯ ПРОВЕДЕННЯ РОБОТИ

Рішення засідання кафедри кібербезпеки та захисту інформації факультету інформаційних технологій протокол № 4 від 24.10.2024 р.

2. МЕТА ТА ВИХІДНІ ДАНІ ДЛЯ ПРОВЕДЕННЯ РОБІТ

Об'єкт досліджень	Процеси обробки та аналізу даних у хмарних середовищах.
Предмет досліджень	Методи і засоби виявлення аномалій у хмарних сервісах з використанням алгоритмів штучного інтелекту.
Мета	Розробка та реалізація систему виявлення аномалій у хмарному середовищі з використанням методів штучного інтелекту для підвищення рівня інформаційної безпеки.
Вихідні дані для проведення роботи	Методи і засоби виявлення аномалій у хмарних сервісах з використанням алгоритмів штучного інтелекту.

3. ОЧІКУВАНІ НАУКОВІ РЕЗУЛЬТАТИ

Наукова новизна удосконалення методу виявлення аномалій у хмарних середовищах шляхом застосування автокодера в поєднанні з автоматичним сповіщенням через Telegram.

Практична цінність підвищення ефективності виявлення аномалій у хмарному середовищі з автоматичним сповіщенням користувача про потенційні загрози в реальному часі.

4. ЕТАПИ ВИКОНАННЯ РОБОТИ

Найменування етапів робіт	Строки виконання робіт (початок–кінець)
Уточнення постановки задачі	25.10.2024 – 29.12.2024
Аналіз літературних джерел	30.12.2024 – 10.02.2024
Вивчення особливостей хмарних середовищ та методів виявлення аномалій	11.02.2024 – 19.02.2024
Обґрунтування вибору методів машинного навчання для задачі	20.02.2024 – 23.02.2024
Розробка вимог до системи та сценаріїв виявлення аномалій	24.02.2024 – 06.03.2024
Вибір та підготовка програмних засобів	07.03.2024 – 11.03.2024
Формалізація структури вхідних даних та механізму оповіщення	12.03.2024 – 15.03.2024
Проєктування симуляційного середовища та структури системи	16.03.2024 – 30.03.2024
Реалізація генерації даних та моделі виявлення аномалій	31.03.2024 – 18.04.2024
Тестування, налагодження, збір результатів, оцінка ефективності	19.04.2024 – 25.04.2024
Оформлення пояснювальної записки згідно методичних рекомендацій	26.04.2024 – 15.05.2025
Подача пакету документів на розгляд ЕК	15.05.2025 – 19.05.2025

Завдання видав _____ Микола БРАІЛОВСЬКИЙ
(підпис) (Ім’я, ПРІЗВИЩЕ)

Завдання прийняв до виконання _____ Владислав ВАСИЛЕНКО
(підпис) (Ім’я, ПРІЗВИЩЕ)

Дата видачі завдання: 25.10.2024 р.

Термін подання кваліфікаційної роботи до ЕК 19.05.2025 р

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи «Технологія виявлення аномалій у хмарних середовищах на основі штучного інтелекту»: 88 сторінок, 4 рисунків, 9 таблиць. 18 літературних джерел.

Метою роботи є розробка технології для виявлення аномалій у хмарних середовищах з використанням методів штучного інтелекту.

Об’єкт дослідження – процеси контролю безпеки та виявлення аномалій у хмарному середовищі.

Предмет дослідження – методи і засоби виявлення аномалій у даних хмарних середовищ з використанням штучного інтелекту.

Практична цінність роботи полягає у розробці прикладної системи, що дозволяє моделювати процес виявлення аномалій та підвищувати рівень інформаційної безпеки в хмарних середовищах. Система може бути використана як основа для впровадження інструментів моніторингу та захисту даних.

Ключові слова: хмарні середовища, виявлення аномалій, штучний інтелект, нейронна мережа, інформаційна безпека.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

AI	–	штучний інтелект
ML	–	машинне навчання
ANN	–	Artificial Neural Network
AWS	–	Amazon Web Services
CPU	–	Central Processing Unit
GPU	–	Graphics Processing Unit
S3	–	Amazon Simple Storage Service
Lambda	–	безсерверна обчислювальна служба AWS
MSE	–	Mean Squared Error
Anomaly	–	відхилення від нормальної поведінки даних
Threshold	–	Поріг
JSON	–	JavaScript Object Notation
CSV	–	Comma-Separated Values
API	–	Application Programming Interface
UI	–	User Interface

ЗМІСТ

ВСТУП 8
РОЗДІЛ 1. ТЕОРИТОЧНІ ОСНОВИ ВИЯВЛЕННЯ АНОМАЛІЙ У ХМАРНИХ СЕРЕДОВИЩАХ 10
1.1 Характеристика хмарних середовищ 10
1.2 Аномалії у хмарних середовищах 12
1.3 Методи виявлення аномалій у хмарних середовищах 15
1.4 Роль штучного інтелекту у виявленні аномалій у хмарних середовищах 17
1.5 Проблеми та виклики у виявленні аномалій у хмарних середовищах 23
1.6 Напрями вирішення проблем виявлення аномалій у хмарних середовищах 25
Висновки до першого розділу 27
РОЗДІЛ 2. ОГЛЯД МЕТОДІВ ТА ПІДХОДІВ ВИЯВЛЕННЯ АНОМАЛІЙ У ХМАРНИХ СЕРЕДОВИЩАХ 28
2.1 Огляд методів виявлення аномалій у даних 28
2.2 Методи обробки даних перед виявленням аномалій 32
2.3 Методи аналізу часових рядів для виявлення аномалій 33
2.4 Методи машинного навчання для виявлення аномалій 37
2.5 Підходи до вибору алгоритмів виявлення аномалій 42
2.6 Проблеми та виклики в задачах виявлення аномалій 46
2.7 Аналіз сучасних підходів до виявлення аномалій у хмарних середовищах 52
Висновок до другого розділу 57
РОЗДІЛ 3. РОЗРОБКА ТА ВПРОВАДЖЕННЯ СИСТЕМИ ВИЯВЛЕННЯ АНОМАЛІЙ НА ОСНОВІ ШІ 59
3.1 Теоретичні підходи до виявлення аномалій у хмарних середовищах 59
3.2 Переваги автокодера в контексті хмарних середовищ 63
3.3 Реалізація моделі виявлення аномалій 67
3.4 Реалізація системи моніторингу та виявлення аномалій 71
3.5 Візуалізація результатів та Telegram-сповіщення про виявлені аномалії 74
3.6 Збереження графіків та логів для подальшого аналізу 78
3.7 Результати роботи та узагальнення 80
ВИСНОВОК 85
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ 87

ДОДАТОК А 89

ВСТУП

Сучасні інформаційні технології стрімко розвиваються, і однією з найважливіших тенденцій останніх років є поширення хмарних обчислень. Хмарні середовища забезпечують масштабованість, доступність, зручність розгортання і керування сервісами, що робить їх незамінними для бізнесу, державного управління та наукових досліджень [1]. Проте разом із перевагами вони несуть нові виклики, зокрема у сфері безпеки, стабільності та надійності функціонування [2].

Однією з критично важливих задач, що виникає в хмарному середовищі, є виявлення аномалій — тобто нетипових відхилень у поведінці системи або даних, що можуть свідчити про збої, кібератаки, внутрішні помилки чи неправильне конфігурування. В умовах високої динаміки, великого обсягу трафіку та автоматизації процесів, класичні підходи до моніторингу та ручного аналізу стають недостатньо ефективними. Тому особливого значення набуває використання інтелектуальних методів виявлення аномалій на основі машинного навчання та штучного інтелекту [3].

У цій роботі розглянуто створення технології виявлення аномалій у хмарному середовищі, що поєднує симуляцію потоків даних, інтелектуальний аналіз та візуалізацію результатів. Основна мета дослідження — розробити ефективний механізм виявлення нетипової поведінки системи, який би міг адаптуватися до зміни вхідних параметрів і працювати в режимі реального часу.

Актуальність роботи визначається як зростанням складності та динаміки хмарних середовищ, так і потребою в автономному, точному та швидкому виявленні загроз [5]. Запропонований підхід ґрунтується на поєднанні сучасних алгоритмів машинного навчання (зокрема, моделі Isolation Forest) та практичних засобів для реалізації потокових систем з логуванням і повідомленням користувача про інциденти.

Результати цієї роботи можуть бути використані в реальних хмарних середовищах, системах моніторингу безпеки, а також слугувати основою для подальших досліджень у сфері кібербезпеки та штучного інтелекту [6].

РОЗДІЛ 1.

ТЕОРИТОЧНІ ОСНОВИ ВИЯВЛЕННЯ АНОМАЛІЙ У ХМАРНИХ СЕРЕДОВИЩАХ

1.1 Характеристика хмарних середовищ

Хмарні обчислення (англ. *cloud computing*) — це модель надання обчислювальних ресурсів, таких як сервери, сховища, бази даних, мережеві компоненти, програмне забезпечення тощо, через Інтернет із забезпеченням гнучкості, масштабованості, економії витрат та доступності для користувачів за принципом "плати за використання".Ця концепція передбачає розміщення даних і обробку інформації не на локальному обладнанні користувача, а на віддалених серверах, доступ до яких забезпечується через мережу.

Основна ідея хмарних обчислень — абстрагування користувача від технічних аспектів інфраструктури, дозволяючи зосередитися лише на використанні ресурсів для своїх потреб. Основні характеристики хмарних обчислень

Модель хмарних обчислень визначається кількома ключовими характеристиками:

- Широкий мережевий доступ. Ресурси доступні через Інтернет за допомогою стандартних механізмів (наприклад, браузера, мобільного додатку).
- Еластичність і масштабованість. Ресурси можуть динамічно збільшуватись або зменшуватись відповідно до потреб користувача.
- Модель оплати за використання. Користувач сплачує тільки за фактично використані ресурси без необхідності великих інвестицій у власну інфраструктуру.
- Об'єднання ресурсів (multi-tenancy). Фізичні і віртуальні ресурси об'єднуються та розподіляються серед багатьох споживачів.

- Самообслуговування на вимогу. Користувач самостійно отримує доступ до необхідних ресурсів без необхідності взаємодії з провайдером.
- Висока доступність і відмовостійкість. Хмарні сервіси розгортаються з урахуванням резервування, автоматичного відновлення та підтримки безперервності бізнес-процесів. Основні моделі розгортання хмар

Існують такі основні моделі розгортання хмарних середовищ:

- Публічна хмара (Public Cloud). Інфраструктура належить сторонньому постачальнику послуг, ресурси доступні широкому колу споживачів через Інтернет. Приклади: AWS, Microsoft Azure, Google Cloud.
- Приватна хмара (Private Cloud). Інфраструктура використовується виключно однією організацією. Може бути розташована на власних серверах або у стороннього провайдера.
- Гібридна хмара (Hybrid Cloud). Поєднання приватних і публічних хмар, що дозволяє розподіляти навантаження та забезпечувати гнучкість у розміщенні ресурсів.
- Мультихмара (Multi-Cloud). Використання кількох різних хмарних провайдерів одночасно для різних завдань або для зменшення залежності від одного постачальника [4]. Основні моделі обслуговування в хмарних обчисленнях

Три основні моделі надання хмарних послуг:

- IaaS (Infrastructure as a Service) — Інфраструктура як послуга. Надання віртуалізованих обчислювальних ресурсів через Інтернет. Користувач отримує доступ до серверів, сховищ, мережевої інфраструктури.. Приклад: Amazon EC2.
- PaaS (Platform as a Service) — Платформа як послуга. Надання середовища для розробки, тестування та розгортання додатків без управління базовою інфраструктурою.. Приклад: Google App Engine.
- SaaS (Software as a Service) — Програмне забезпечення як послуга. Готові програми доступні через Інтернет. Користувач працює із програмою без встановлення її на свій пристрій.. Приклад: Gmail, Microsoft 365.

Виклики та ризики використання хмарних технологій Попри численні переваги, хмарні обчислення супроводжуються низкою викликів:

- Проблеми безпеки і конфіденційності. Ризики витоку даних, атак на інфраструктуру, недобросовісного адміністрування.
- Залежність від постачальника послуг. Vendor lock-in ускладнює зміну постачальника через несумісність технологій.
- Контроль за даними. Не завжди можливо повністю контролювати місце зберігання даних або методи їх обробки.
- Проблеми з доступністю. Збої в роботі провайдера можуть призвести до значних фінансових втрат.
- Юридичні та регуляторні обмеження. Залежно від країни розташування серверів можуть застосовуватись різні правила зберігання даних [7]. Значення хмарних середовищ для сучасних систем. На сьогодні хмарні технології є невід'ємною частиною розвитку бізнесу, науки, медицини та державного управління. Вони дозволяють:
- Швидко розгортати сервіси та додатки.
- Оптимізувати витрати на IT-інфраструктуру.
- Гнучко адаптувати ресурси під змінні потреби.
- Забезпечувати безперервність бізнесу навіть у кризових ситуаціях.

В умовах стрімкого зростання обсягів даних, вимог до безпеки і потреби в масштабованості — роль хмарних обчислень буде тільки зростати.

1.2 Аномалії у хмарних середовищах

Поняття аномалій у хмарних середовищах

Аномалія — це спостереження або подія, яка суттєво відрізняється від нормальної поведінки системи і потенційно вказує на проблему, збої або загрозу. У контексті хмарних обчислень аномалії є критичними індикаторами, оскільки вони можуть сигналізувати про порушення безпеки, несправності в інфраструктурі, перевантаження ресурсів чи збої в наданні послуг.

Аномалії часто є першими ознаками інцидентів, які, за відсутності вчасного реагування, можуть призвести до серйозних фінансових втрат, репутаційних ризиків та порушення безперервності бізнесу. Основні типи аномалій у хмарних середовищах

У хмарних середовищах виявляються різноманітні типи аномалій:

- Аномалії в навантаженні ресурсів. Наприклад, різке збільшення використання CPU, пам'яті чи мережевої пропускної здатності.
- Аномалії в доступності сервісів. Випадки, коли сервіси стають недоступними для користувачів через внутрішні або зовнішні причини.
- Аномалії в облікових записах і доступах. Незвичні входи, підозрілі дії користувачів, спроби підвищення прав доступу.
- Аномалії в поведінці додатків. Наприклад, аномально довгий час відповіді API або серії помилок HTTP.
- Аномалії у збереженні даних. Раптова втрата, модифікація або масове видалення даних.
- Аномалії в мережевому трафіку. Сплески трафіку, незвичні порти або маршрути передавання даних, які можуть свідчити про кібератаки. Причини виникнення аномалій Аномалії у хмарних середовищах можуть бути спричинені багатьма чинниками:
- Технічні несправності. Аппаратні збої серверів, мережеві помилки, неправильні налаштування систем.
- Кіберзагрози. Зловмисні атаки: DDoS-атаки, злом облікових записів, проникнення в систему.
- Людський фактор. Помилки адміністраторів або користувачів, неправильне використання ресурсів.
- Програмні помилки. Баги у коді застосунків або системне ПЗ, які призводять до аномальної поведінки.
- Нестабільне навантаження. Нерівномірне використання ресурсів унаслідок маркетингових кампаній, сезонних піків або зовнішніх подій. Вплив аномалій на функціонування хмарних середовищ Наслідки аномалій можуть бути значними:

- Зниження продуктивності. Високе навантаження на інфраструктуру уповільнює роботу сервісів.
- Втрати даних. Аномальні видалення або пошкодження даних можуть бути незворотними без належного резервного копіювання.
- Порушення безпеки. Несвоєчасне виявлення аномалій може дати змогу зловмисникам проникнути у систему.
- Фінансові збитки. Несправності в обслуговуванні клієнтів призводять до втрати прибутку і штрафних санкцій.
- Репутаційні ризики. Постійні або великі збої шкодять іміджу компанії серед клієнтів та партнерів. Потреба у виявленні аномалій. Враховуючи складність та динамічність сучасних хмарних систем, автоматизоване виявлення аномалій стало необхідною складовою їхньої експлуатації. Ручний моніторинг є неефективним через великі обсяги даних і високу швидкість їх зміни. Системи виявлення аномалій дозволяють:
- Своєчасно ідентифікувати порушення нормальної поведінки.
- Знижувати час реакції на інциденти.
- Підвищувати рівень безпеки та надійності систем.
- Оптимізувати використання ресурсів.
- Мінімізувати можливі збитки та забезпечити стабільність роботи сервісів.

Важливість побудови систем виявлення аномалій Розробка ефективних систем виявлення аномалій у хмарних середовищах базується на використанні сучасних технологій обробки даних, машинного навчання та штучного інтелекту.. Висока якість виявлення забезпечує:

- Виявлення навіть нетипових, "розмитих" аномалій.
- Зменшення кількості хибних спрацьовувань (false positives).
- Автоматичне навчання моделей на нових даних.
- Масштабованість під великі обсяги даних.

Відтак, системи виявлення аномалій стають ключовим елементом побудови надійної та безпечної хмарної інфраструктури.

1.3 Методи виявлення аномалій у хмарних середовищах

Загальна класифікація методів. Методи виявлення аномалій у хмарних середовищах можна класифікувати за різними ознаками:

- За типом аналізованих даних: числові, категоріальні, часові ряди.
- За підходом до навчання: із наглядом (supervised), без нагляду (unsupervised), з частковим наглядом (semi-supervised).
- За принципом роботи: статистичні методи, машинне навчання, глибоке навчання, гібридні методи. Статистичні методи виявлення аномалій. Статистичні методи базуються на припущенні, що нормальні дані мають певний розподіл, а аномалії суттєво відхиляються від нього. Основні методи:
- Методи на основі середнього та дисперсії. Порогові значення визначаються відхиленням від середнього (наприклад, ±3 стандартних відхилення).
- Коробкові діаграми (Boxplot) . Виявлення аномалій через міжквартильний розмах.
- Параметричні моделі. Наприклад, використання нормального або пуассонівського розподілу для побудови ймовірнісних моделей.
- Непараметричні методи. Наприклад, методи на основі перцентилів або медіани. Переваги: простота реалізації, інтерпретованість.. Недоліки: обмеженість у випадку складних або змінних даних.

Методи машинного навчання. Машинне навчання дозволяє автоматично виявляти складні залежності в даних. Ключові підходи:

- Методи класифікації. Наприклад, дерева рішень, SVM (Support Vector Machine), логістична регресія — для задач із наявністю міток (аномалія/норма).
- Методи кластеризації. Наприклад, K-середніх (K-means), DBSCAN — групування схожих даних і виявлення відокремлених точок як аномалій.
- Методи відокремлення нових даних. Наприклад, One-Class SVM або Isolation Forest — побудова моделей, які ідентифікують "чужі" точки щодо навчальної множини.

Переваги: гнучкість, здатність працювати зі складними даними. Недоліки: потреба в належній підготовці даних і налаштуванні моделей. Методи глибокого навчання. Глибоке навчання використовує багаторівневі нейронні мережі для побудови складних моделей виявлення аномалій.

Основні технології:

- Автокодери (Autoencoders). Нейронні мережі, що навчаються стискати дані і відновлювати їх. Великі помилки відновлення сигналізують про аномалії.
- Рекурентні нейронні мережі (RNN, LSTM). Використовуються для аналізу часових рядів — наприклад, для моніторингу змін навантаження серверів.
- Генеративні моделі (GAN). Навчаються генерувати нормальні дані; великі відхилення від згенерованих зразків вважаються аномаліями.

Переваги: висока точність, можливість виявляти складні приховані закономірності. Недоліки: висока потреба в обчислювальних ресурсах та великих обсягах даних для навчання. Гібридні підходи комбінують кілька різних методів для досягнення кращих результатів. Приклади:

- Поєднання автокодерів і кластеризації для підвищення точності.

Таблиця 1.1

Порівняльна таблиця методів виявлення аномалій

Метод	Переваги	Недоліки	Сфера застосування
Статистичні методи	Простота, швидкість	Погана адаптація до складних даних	Просте середовище, без високої варіативності
Машинне навчання (класифікація)	Навчання на мічених даних, висока точність	Потрібна мітка для кожного класу	Класичні задачі безпеки, відомі атаки
Машинне навчання (кластеризація)	Не потребує міток	Може давати багато хибних спрацьовувань	Виявлення нових типів аномалій
Глибоке навчання (автокодери)	Автоматичне виділення ознак, добре працює на великих даних	Потребує більше обчислювальних ресурсів	Виявлення складних та неочевидних аномалій

Залежно від поставлених цілей та ресурсів, оптимальним може бути як простий статистичний метод, так і складна гібридна система на основі глибокого навчання.

1.4 Роль штучного інтелекту у виявленні аномалій у хмарних середовищах

Штучний інтелект (ШІ) стає все більш важливою складовою систем моніторингу та захисту хмарних середовищ. Завдяки своїм можливостям аналізувати великі обсяги даних, виявляти приховані закономірності та адаптуватися до змін, ШІ значно підвищує ефективність виявлення аномалій порівняно з традиційними методами. Основні переваги використання ШІ у виявленні аномалій

- Адаптивність. ШІ-системи можуть автоматично підлаштовувати свої моделі до нових типів даних та змін у поведінці системи.
- Аналіз великих обсягів даних. Завдяки алгоритмам глибокого навчання можливий обробіток терабайтів інформації майже в реальному часі.
- Автоматичне виявлення нових типів загроз. Алгоритми без нагляду (unsupervised learning) вміють знаходити невідомі раніше аномалії без попередньої розмітки даних.
- Зниження кількості хибних спрацьовувань. Моделі на основі ШІ здатні краще розрізняти нормальні варіації від справжніх аномалій. Приклади застосування ШІ для виявлення аномалій у хмарі
- Інтелектуальні системи моніторингу навантаження серверів. Наприклад, виявлення нехарактерного сплеску використання CPU або RAM.
- Виявлення мережевих атак у хмарних середовищах. Виявлення підозрілої активності (наприклад, спроб сканування портів або DDoS-атак).
- Аналіз доступу до даних. Розпізнавання незвичних шаблонів доступу користувачів до критичної інформації.

- Контроль за конфігурацією хмарної інфраструктури. Автоматичне виявлення неправильних налаштувань безпеки, які можуть стати вразливостями. Алгоритми ШІ, що найчастіше використовуються
- Автокодери. Навчаються відновлювати нормальні зразки даних і використовують помилку відновлення для виявлення аномалій.
- One-Class SVM. Класифікує нові дані як нормальні або аномальні на основі їх схожості до нормальної тренувальної вибірки.
- Isolation Forest. Виявляє аномалії шляхом випадкового поділу даних на менші підмножини.
- Глибокі нейронні мережі (Deep Neural Networks). Використовуються для складних завдань виявлення аномалій у великих і складних наборах даних.
- LSTM-мережі для часових рядів. Дозволяють відслідковувати довгострокові залежності в даних, що особливо важливо для моніторингу зміни навантаження або трафіку. Важливість навчання та налаштування моделей

Ефективність систем виявлення аномалій залежить від якості даних для навчання та правильно обраних гіперпараметрів моделей:

- Неякісні або нерепрезентативні дані можуть призводити до великої кількості хибних спрацьовувань.
- Недостатня кількість навчальних даних обмежує можливості виявлення рідкісних, але критичних аномалій.
- Переобучення моделі може знизити здатність системи адаптуватися до нових загроз. Тому дуже важливо проводити ретельну підготовку та валідацію моделей. Виклики використання ШІ у хмарних середовищах
- Обчислювальні ресурси. Більшість моделей глибокого навчання вимагають значної обчислювальної потужності для тренування і прогнозування.
- Пояснюваність рішень. Складні моделі типу глибоких нейронних мереж важко інтерпретувати: складно пояснити, чому система класифікувала конкретний випадок як аномалію.
- Захист самих моделей. Моделі можуть бути об'єктами атак, наприклад, шляхом ін'єкції шкідливих даних. Тренди розвитку

- Автоматичне навчання моделей (AutoML). Зменшує потребу в ручній побудові та налаштуванні моделей.
- Інтеграція ШІ з хмарними сервісами безпеки (Cloud Security AI-as-a-Service). Наприклад, AWS GuardDuty, Azure Security Center використовують ШІ для моніторингу.
- Розвиток Explainable AI (XAI). Створення моделей, які не тільки виявляють аномалії, а й надають пояснення своїх рішень.

Штучний інтелект у хмарних середовищах відкриває нові можливості для більш точного, швидкого та гнучкого виявлення аномалій. Проте для максимальної ефективності необхідно правильно організувати процес навчання моделей, забезпечити їх захист та інтерпретованість результатів. Існуючі платформи для виявлення аномалій у хмарних середовищах

На сучасному ринку інформаційних технологій представлено велику кількість рішень для моніторингу та виявлення аномалій у хмарних середовищах. Вони інтегруються безпосередньо з хмарною інфраструктурою або працюють як зовнішні сервіси. Основною метою таких платформ є забезпечення безперервного аналізу поведінки систем і раннього виявлення потенційних загроз чи несправностей. Класифікація платформ

- Вбудовані сервіси великих хмарних провайдерів. Такі як AWS, Microsoft Azure, Google Cloud — пропонують нативні рішення для безпеки й моніторингу.
- Спеціалізовані рішення від третіх сторін. Наприклад, Datadog, Splunk, New Relic, які можуть працювати у мультихмарних середовищах.
- Інструменти з відкритим кодом. Наприклад, Prometheus, Grafana, які дозволяють налаштовувати власні системи моніторингу. Огляд популярних платформ

1. AWS GuardDuty Функціонал: Інтелектуальний сервіс виявлення загроз для облікових записів AWS, робочих навантажень і даних з використанням машинного навчання. Особливості: Використання власних алгоритмів аномалій; інтеграція з іншими службами AWS (CloudTrail, VPC Flow Logs, DNS Logs). Переваги: Простота налаштування, автоматичне оновлення моделей.

2. Azure Security Center. Функціонал: Захист гібридних робочих навантажень у хмарі та локально, включає виявлення аномалій на основі ШІ. Особливості: Інтеграція з Azure Defender, адаптивний контроль безпеки. Переваги: Вбудована аналітика для інцидентів, рекомендації з усунення вразливостей.

3. Google Cloud Operations (раніше Stackdriver) Функціонал: Надає повний моніторинг, логування і виявлення аномалій у хмарній інфраструктурі GCP. Особливості: Використання моделей прогнозного аналізу. Переваги: Тісна інтеграція зі всіма сервісами Google Cloud.

4. Datadog . Функціонал: Хмарна платформа моніторингу, що включає виявлення аномалій за допомогою машинного навчання. Особливості: Підтримка мультихмарних середовищ, великий вибір інтеграцій. Переваги: Гнучкість налаштувань, підтримка аналізу логів, трасування, метрик. Основні можливості сучасних платформ

- Аналіз логів у реальному часі. Постійний моніторинг записів подій з автоматичним виявленням аномальних шаблонів.
- Моніторинг метрик системи. Відстеження ключових показників, таких як навантаження на процесор, пам'ять, мережевий трафік.
- Аналіз поведінки користувачів (UBA). Виявлення підозрілої активності користувачів, яка може свідчити про спробу атаки.
- Інтеграція з системами сповіщення. Автоматичне надсилання повідомлень при виявленні аномалій через Email, Slack, Telegram тощо.
- Автоматичне реагування (SOAR). Платформи можуть не тільки виявляти, але й автоматично реагувати на аномалії — наприклад, блокувати підозрілі дії. Вимоги до платформ виявлення аномалій
- Масштабованість

Система повинна легко справлятися зі збільшенням обсягу даних без втрати продуктивності.

- Низька затримка виявлення. Час виявлення аномалій має бути мінімальним, щоб вчасно реагувати на загрози.

- Можливість роботи в мультихмарних середовищах. Багато компаній використовують кілька хмар одночасно.

Таблиця 1.2

Порівняння платформ за основними характеристиками

Платформа	Підтримка мультихмари	Використання ШІ	Інтеграція з SIEM	Реагування на події
AWS GuardDuty	Ні	Так	Так	Частково
Azure Security Center	Частково	Так	Так	Так
Google Cloud Operations	Ні	Так	Так	Ні
Datadog	Так	Так	Так	Так
Splunk Observability	Так	Так	Так	Так

Існуючі платформи для виявлення аномалій у хмарних середовищах пропонують широкий спектр можливостей для захисту даних і сервісів. Вибір конкретного рішення залежить від потреб компанії, розміру інфраструктури, бюджету та вимог до безпеки.

Штучний інтелект (ШІ) відіграє ключову роль у сучасних підходах до виявлення аномалій, особливо в умовах великого обсягу даних, складної структури інформації та високої динаміки зміни параметрів. Завдяки здатності адаптуватися, самонавчатися та виявляти приховані закономірності, ШІ-системи значно перевершують традиційні методи у точності та ефективності виявлення аномальних подій.

Основна перевага використання ШІ у задачах виявлення аномалій полягає в автоматизації процесу аналізу складних та різнорідних даних. Алгоритми машинного навчання здатні вивчати поведінкові шаблони системи без попереднього програмування правил, що дозволяє їм гнучко реагувати на нові типи загроз або несподівані відхилення. Особливо важливо це в хмарних середовищах, де інтенсивність подій і обсяг трафіку не дозволяють ефективно використовувати ручні методи аналізу.

ШІ-технології, зокрема нейронні мережі, застосовуються для моделювання нормальної поведінки системи та подальшого виявлення відхилень. Глибокі автоенкодери, рекурентні нейронні мережі (RNN), довгі короткострокові пам’яті (LSTM), трансформери — всі ці моделі використовуються для аналізу часових рядів, прогнозування стану системи та визначення моментів, коли спостережувані значення різко відрізняються від очікуваних. Такі моделі не лише фіксують аномалії, а й можуть класифікувати їх за типами, оцінювати рівень ризику та пріоритезувати реагування.

Крім цього, ШІ дозволяє реалізовувати контекстне виявлення аномалій, коли враховується не лише значення ознак, а й їхнє положення у загальній поведінковій картині системи. Це особливо важливо в умовах багатовимірних даних, де окреме значення може здаватися нормальним, але в контексті — виявляється нетиповим. Алгоритми, навчені на історичних даних, можуть враховувати сезонні коливання, залежності між параметрами та специфіку середовища.

Штучний інтелект також має важливу роль у зменшенні кількості хибнопозитивних спрацювань, що є однією з головних проблем традиційних систем моніторингу. Завдяки використанню ансамблевих методів, самонавчальних моделей і технік переобучення, ШІ дозволяє досягти більшої стабільності системи та точності в реальному часі.

Завдяки інтеграції з потоковими обчисленнями, системами логування та повідомлення користувача, рішення на базі ШІ можуть працювати в режимі 24/7, оперативно реагуючи на потенційні загрози. Крім того, вони здатні навчатися на нових даних, що дозволяє постійно вдосконалювати їх точність і релевантність без втручання людини.

Таким чином, роль штучного інтелекту у виявленні аномалій є стратегічно важливою. Він не лише підвищує ефективність аналізу, але й закладає основу для створення самостійних, інтелектуальних систем кіберзахисту, що здатні випереджати загрози та забезпечувати надійність критично важливих інфраструктур у цифрову епоху.

Для ефективного впровадження важливо оцінювати не тільки функціональність платформи, а й її інтеграційні можливості, швидкість реагування та можливість адаптації під специфіку організації.

1.5 Проблеми та виклики у виявленні аномалій у хмарних середовищах

Попри стрімкий розвиток технологій аналізу даних і впровадження штучного інтелекту, виявлення аномалій у хмарних середовищах залишається складним завданням. Це обумовлено як технічними особливостями хмарних архітектур, так і специфікою роботи різноманітних додатків та сервісів у цих середовищах. Основні проблеми

1. Велика кількість та різноманітність даних: Хмарні середовища генерують величезні обсяги різнорідних даних (логи, метрики, події безпеки, транзакції тощо). Дані можуть надходити з різних джерел: віртуальних машин, контейнерів, серверлесс функцій, мережевого обладнання. Висока варіативність даних ускладнює процес їх нормалізації та попередньої обробки для аналізу.

2. Динамічність та масштабованість хмарних середовищ. Хмарна інфраструктура постійно змінюється: додавання нових вузлів, масштабування додатків, оновлення сервісів. Системи виявлення повинні адаптуватися до частих змін конфігурацій і бути стійкими до "шуму" даних.

3. Складність визначення нормальної поведінки Відмінності в поведінці сервісів у різних періодах часу (наприклад, різні навантаження вдень і вночі). Змінна робоча активність користувачів та додатків ускладнює створення стабільного профілю нормальної поведінки.

4. Високий рівень хибних спрацьовувань (False Positives) Невдало налаштовані або недостатньо навчені моделі часто плутають нормальні зміни в поведінці з аномаліями. Це призводить до великої кількості хибних тривог, що розвантажує увагу аналітиків і може призвести до ігнорування реальних інцидентів.

5. Безпека і конфіденційність даних Не всі дані можуть бути використані для аналізу через вимоги безпеки або законодавчі обмеження (наприклад, GDPR).

Зашифровані дані або приватні дані користувачів ускладнюють глибинний аналіз. Технічні виклики. Реальний час обробки Системи повинні виявляти аномалії миттєво, інакше реакція може бути запізнілою, що збільшує ризики. Масштабування аналітики Методи обробки та аналізу мають бути здатні працювати на сотнях тисяч об'єктів одночасно. Підтримка мультихмарних середовищ Компанії часто використовують сервіси різних провайдерів (AWS, Azure, GCP одночасно). Потрібна єдина система моніторингу для агрегування даних з усіх середовищ. Відсутність стандартизації Формати логів, метрик і подій можуть відрізнятися в різних хмарних платформах. Потребує розробки конвертаційних механізмів або уніфікованих стандартів даних. Організаційні виклики. Брак кваліфікованих фахівців. Високий попит на аналітиків безпеки та спеціалістів з аналізу аномалій перевищує кількість підготовлених кадрів. Недостатня взаємодія між командами Часто відсутній налагоджений процес комунікації між DevOps, Security та Data Science командами. Відсутність чіткої стратегії моніторингу

Компанії впроваджують інструменти для виявлення аномалій без повноцінного плану їх інтеграції та реагування на інциденти. Потенційні рішення

Впровадження машинного навчання і штучного інтелекту. Використання адаптивних алгоритмів, які здатні навчатися на змінних даних та знижувати рівень хибних спрацьовувань.

Створення профілів поведінки за допомогою кластеризації. Замість фіксованих правил створюються динамічні групи нормальної активності.

Інтеграція систем реагування (SOAR). Автоматизація відповідей на аномалії допомагає скоротити час реагування та зменшує навантаження на аналітиків.

Використання контейнеризованих підходів для моніторингу. Наприклад, впровадження агентів у вигляді контейнерів для спрощення масштабування моніторингу.

Виявлення аномалій у хмарних середовищах є надзвичайно складним завданням, що вимагає комплексного підходу. Технічні й організаційні виклики змушують компанії шукати інноваційні рішення та постійно вдосконалювати свої стратегії захисту. Розуміння цих проблем допомагає розробляти більш ефективні

методики виявлення аномалій та забезпечувати надійний захист інформаційних ресурсів у хмарі.

1.6 Напрями вирішення проблем виявлення аномалій у хмарних середовищах

Щоб ефективно боротися з викликами, що виникають під час виявлення аномалій у хмарних середовищах, необхідно впроваджувати сучасні технології, переглядати стратегії обробки даних і вдосконалювати підходи до безпеки. Розглянемо основні напрями розвитку і вирішення існуючих проблем.

Використання технологій машинного навчання (ML) та штучного інтелекту (AI)

Адаптивне навчання:. Моделі машинного навчання здатні динамічно підлаштовуватись під нові шаблони даних без необхідності ручного втручання.

Аномалії на основі поведінки:. Замість фіксованих порогів використовуються моделі, які навчаються нормальній поведінці системи та виявляють відхилення.

Глибинне навчання (Deep Learning):. Багатошарові нейронні мережі допомагають розпізнавати складні шаблони у великих обсягах даних.

Приклади застосування: AWS GuardDuty використовує ML для виявлення загроз.

Google Cloud Anomaly Detection Engine аналізує трафік та активність користувачів.

Побудова профілів нормальної поведінки систем. Створення базових ліній (baselines): Фіксація нормальної активності для кожного сервісу чи компонента з урахуванням сезонності і змін навантаження.

Методи кластеризації: Використання алгоритмів (наприклад, K-Means, DBSCAN) для угрупування подібних шаблонів поведінки і відхилення від них.

Індивідуальні профілі: Для різних типів сервісів або клієнтів можуть створюватися окремі моделі нормальної поведінки.

Автоматизація процесів моніторингу та реагування. SOAR-платформи (Security Orchestration, Automation and Response): Інтеграція автоматизованих рішень для збору даних, аналізу інцидентів і виконання відповідних дій.

Автоматичне масштабування моніторингу: Автоматичне розгортання додаткових агентів або обчислювальних ресурсів при зростанні навантаження.

Приклади: Автоматичне блокування підозрілих IP-адрес. Автоматичне сповіщення аналітиків про критичні аномалії. Використання комплексних рішень моніторингу. SIEM-системи (Security Information and Event Management): Агрегація логів і подій з усіх джерел для централізованого аналізу та виявлення аномалій.

XDR-рішення (Extended Detection and Response): Забезпечують єдину платформу для моніторингу мережі, кінцевих пристроїв, серверів і хмарних ресурсів.

Приклади: Splunk, Datadog, Microsoft Sentinel. Підвищення якості даних

Нормалізація та стандартизація логів: Перетворення даних у єдиний формат для полегшення аналізу. Фільтрація шуму: Відсіювання неінформативних або повторюваних подій для зниження навантаження на системи аналізу. Попередня обробка: Використання ETL-процесів (Extract, Transform, Load) для підготовки даних до аналізу. Використання контейнерних технологій для збору даних. Контейнеризовані агенти: Легкі й автономні агенти в контейнерах для збору даних без потреби в складній інтеграції.

Приклади: Використання Fluentd або Prometheus Node Exporter для збору метрик у Kubernetes-кластерах. Забезпечення безпеки та конфіденційності під час обробки даних

Анонімізація даних: Видалення або шифрування персональної інформації перед аналізом. Дотримання стандартів та законодавчих вимог: Наприклад, GDPR для захисту даних у Європейському Союзі.

Впровадження Zero Trust підходу: Кожна дія та кожен запит вимагає перевірки незалежно від місця походження.

Вирішення проблем виявлення аномалій у хмарних середовищах вимагає системного підходу, що охоплює впровадження новітніх технологій, автоматизацію процесів, вдосконалення аналізу даних та розвиток людського капіталу. Тільки

поєднання технічних рішень із організаційними стратегіями забезпечить високий рівень захисту й ефективності у виявленні загроз.

Висновки до першого розділу

У першому розділі було проведено комплексний аналіз основних понять, пов'язаних із виявленням аномалій у хмарних середовищах. Було визначено, що хмарні обчислення, завдяки своїй масштабованості, доступності й економічності, стали невід'ємною частиною сучасної IT-інфраструктури, але одночасно породили нові виклики у сфері безпеки.

Аномалії в хмарних середовищах можуть проявлятися у вигляді відхилень у поведінці користувачів, перевищенні навантаження на ресурси, підозрілих мережевих з'єднаннях тощо. Їх вчасне виявлення є критично важливим для запобігання порушенням безпеки та забезпечення стабільної роботи сервісів.

У межах розділу було охарактеризовано ключові проблеми, які ускладнюють виявлення аномалій, зокрема велику кількість даних, високу динамічність середовища, різноманітність джерел інформації та обмеження традиційних методів безпеки. Також були окреслені основні вимоги до систем виявлення аномалій, включно з необхідністю масштабованості, точності, швидкості обробки даних і підтримкою безперервної адаптації.

РОЗДІЛ 2.

ОГЛЯД МЕТОДІВ ТА ПІДХОДІВ ВИЯВЛЕННЯ АНОМАЛІЙ У ХМАРНИХ СЕРЕДОВИЩАХ

2.1 Огляд методів виявлення аномалій у даних

Виявлення аномалій — це процес ідентифікації спостережень, які відхиляються від загального шаблону або поведінки даних. Ці відхилення можуть бути результатом різних причин, таких як збої системи, шахрайські дії, кібератаки, технічні несправності або рідкісні події. Завдання виявлення аномалій критично важливе в багатьох галузях, включаючи кібербезпеку, фінансовий моніторинг, медицину, промисловість та аналітику великих даних [5].

Класифікація методів виявлення аномалій: Статистичні методи: Основна ідея полягає у використанні математичних моделей для оцінювання нормальної поведінки системи. Параметричні методи передбачають, що дані підпорядковуються певному закону розподілу (наприклад, нормальному). У такому випадку аномалії визначаються як значення, що знаходяться за межами визначеного інтервалу.

Непараметричні методи не припускають наявності конкретного розподілу і будуються без таких припущень.

Приклади: Метод Z-оцінок (стандартних відхилень). Метод побудови довірчих інтервалів. Метод виявлення змінних точок у часових рядах. Методи на основі машинного навчання: Наглядове навчання (Supervised Learning):

Моделі створюються на основі мічених даних, де вказано, які записи є аномальними.Основні алгоритми: дерева рішень, логістична регресія, нейронні мережі. Недолік: потребує великої кількості розмічених даних, що часто важко забезпечити. Навчання без учителя (Unsupervised Learning): Використовується, коли

розмітки немає. Алгоритми шукають природні відмінності між об'єктами. Приклади: алгоритм кластеризації k-середніх, DBSCAN, самоорганізуючі карти Кохонена.

Напівавтоматичне навчання (Semi-Supervised Learning): Вчаться тільки на нормальних даних для побудови моделі нормальної поведінки, після чого шукають відхилення. Методи на основі відстаней і щільності: Підходять для багатовимірних даних. Вимірюють відстань між об'єктами або щільність розташування точок у просторі ознак. Якщо об'єкт ізольований або розташований у зоні низької щільності — він розглядається як аномалія. Приклади: Метод локальної щільності аномалій (LOF). Метод k-ближчих сусідів (k-NN). Методи на основі глибокого навчання (Deep Learning):

Особливо корисні для обробки складних, великомасштабних і високорозмірних даних. Автоенкодери: Спеціальні типи нейронних мереж, що навчаються стискати дані у зменшену форму і потім відновлювати їх. Якщо дані відновлюються з великими помилками — це може сигналізувати про аномалії. Рекурентні нейронні мережі (RNN):

Добре підходять для виявлення аномалій у часових рядах, коли важлива послідовність подій. Генеративно-суперечливі мережі (GAN): Використовуються для створення нових даних, подібних до нормальних, що дозволяє знаходити об'єкти, які від них відрізняються. Методи на основі моделей прогнозування:

Створюється модель для прогнозування майбутніх значень. Аномалією вважається значне відхилення реального значення від прогнозованого. Використовується у часових рядах, наприклад: Прогнозування споживання енергії. Прогнозування навантаження на сервери. Гібридні методи: Поєднують кілька підходів для кращої ефективності. Наприклад, спочатку відбувається попереднє зменшення розмірності даних за допомогою PCA (метод головних компонент), а потім застосовується кластеризація для виявлення аномалій.

Методи виявлення аномалій охоплюють широкий спектр підходів, які застосовуються залежно від типу даних, контексту задачі, наявності розмічених прикладів та вимог до точності, швидкодії й інтерпретованості. Їх умовно можна

класифікувати на статистичні, кластеризаційні, прогнозні, модельні, машинного навчання та гібридні.

Статистичні методи ґрунтуються на припущенні, що нормальна поведінка підпорядковується певному розподілу (наприклад, нормальному). Відхилення, що перевищують порогові значення, вважаються аномальними. До таких методів належать Z-оцінка, методи на основі меж міжквартильного розмаху, щільнісні підходи (наприклад, Gaussian Mixture Models). Їх основна перевага — простота та швидкість, однак вони погано працюють із нестаціонарними або складно структурованими даними.

Методи кластеризації (наприклад, K-Means, DBSCAN, LOF) використовують ідею поділу даних на групи за подібністю. Елементи, що не належать до жодного кластера або мають низьку щільність сусідів, вважаються аномаліями. Такі підходи ефективні при виявленні віддалених точок у багатовимірних даних, однак вимагають тонкого налаштування гіперпараметрів і можуть бути чутливими до масштабування.

Прогнозні методи орієнтовані на побудову моделі, яка передбачає значення ознак у майбутньому (наприклад, з часових рядів). Якщо реальні значення значно відхиляються від передбачених, це сигнал про потенційну аномалію. У цьому контексті використовуються авторегресивні моделі (AR, ARIMA), а також більш складні нейронні мережі — зокрема, LSTM, GRU, трансформери. Такі підходи забезпечують високу адаптивність, але вимагають значного обсягу історичних даних та обчислювальних ресурсів.

Методи машинного навчання поділяються на ті, що працюють з розміченими (наглядовими) і нерозміченими (без нагляду) даними. У задачах без нагляду часто застосовуються ізоляційні алгоритми (наприклад, Isolation Forest), One-Class SVM, autoencoder’и, які будують внутрішнє уявлення нормальної поведінки і реагують на відхилення. У випадках, коли доступна розмітка (анотація про норму/аномалію), можуть використовуватись класичні моделі — рішення дерев, градієнтний бустінг, нейронні мережі.

Autoencoder — один із найпоширеніших підходів для виявлення аномалій у складних багатовимірних даних. Ці нейромережі навчаються стискати й

відновлювати дані. Якщо вхідні дані неможливо якісно реконструювати (мають велику похибку реконструкції), вони, ймовірно, є аномальними. Це особливо корисно в хмарних або IoT-системах, де відстежується поведінка великої кількості параметрів одночасно.

Гібридні підходи поєднують кілька методів — наприклад, автоенкодери для зниження розмірності та кластеризацію на зменшеному просторі ознак. Також популярне поєднання статистичних підходів із машинним навчанням для зменшення похибок та підвищення узагальнюваності.

Окрему нішу займають методи, розроблені для потокових даних (streaming anomaly detection). Тут використовуються адаптивні моделі, здатні оновлюватися в реальному часі: наприклад, ADWIN, Online k-means або спеціалізовані архітектури зі скользячими вікнами.

Таким чином, вибір методу виявлення аномалій залежить від типу даних (табличні, часові ряди, текст, графи), їх структури (стаціонарні/нестаціонарні), вимог до затримки, точності та ресурсоємності. Глибоке розуміння контексту задачі є визначальним для успішного впровадження систем виявлення аномалій у реальних умовах.

Таблиця 2.1

Порівняння методів

Категорія методу	Приклади алгоритмів	Переваги	Недоліки
Статистичні	Z-оцінка, контрольні карти	Простота, ефективність на маленьких даних	Низька точність на складних даних
Машинне навчання	SVM, Decision Tree	Висока адаптивність	Потреба в розмічених даних
Відстань/щільність	LOF, k-NN, DBSCAN	Підходять для виявлення локальних аномалій	Чутливі до параметрів
Глибоке навчання	Autoencoder, RNN	Висока якість на великих об'ємах	Потреба в обчислювальних ресурсах

2.2 Методи обробки даних перед виявленням аномалій

Перш ніж застосовувати будь-які алгоритми виявлення аномалій, дуже важливо провести правильну обробку даних. Якість підготовки даних безпосередньо впливає на точність і надійність результатів виявлення [9]. Обробка даних включає кілька важливих етапів: Збір даних.Визначення джерел даних:. Логи сервісів. Дані датчиків IoT. Мережевий трафік. Фінансові транзакції.

Важливо забезпечити безперервність та актуальність даних для своєчасного виявлення аномалій. Дані можуть бути структурованими (табличні бази даних) або неструктурованими (логи, текст). Очищення даних Видалення пропущених значень: Заповнення за допомогою середнього, медіани або найближчих сусідів (k-NN імпутація). Видалення записів з великою кількістю пропусків. Видалення або корекція помилкових значень: Виявлення явних помилок у даних (наприклад, негативна кількість транзакцій).

Обробка дублікатів: Видалення повторюваних записів для запобігання перекосу аналізу. Нормалізація та стандартизація Нормалізація: приведення всіх ознак до одного масштабу, наприклад, у діапазон [0, 1]. Стандартизація: приведення до розподілу з нульовим середнім і одиничним стандартним відхиленням.

Чому важливо: Багато алгоритмів (наприклад, k-NN, нейронні мережі) дуже чутливі до масштабів ознак. Приклади методів: Min-Max Scaling. Z-Score Normalization. Вибір ознак (Feature Selection), Вибір найбільш інформативних змінних для аналізу:Зменшує розмірність задачі. Покращує швидкість роботи алгоритму. Знижує ризик переобучення.

Методи: Кореляційний аналіз (видалення сильно корельованих ознак).Метод відсівання за важливістю (наприклад, за допомогою Random Forest). Генерація нових ознак (Feature Engineering) Створення додаткових змінних, які можуть краще описати природу даних: Створення агрегованих змінних (середні значення за певний час). Використання часових характеристик (час доби, день тижня). Дуже важливо для покращення якості моделі виявлення аномалій. Зменшення розмірності. Особливо

актуально для високорозмірних даних (Big Data). Методи: PCA (метод головних компонент): створює нові змінні як лінійні комбінації вихідних ознак. t-SNE, UMAP: ефективні для візуалізації даних. Переваги: Зменшує шум. Спрощує пошук аномалій. Балансування даних Проблема: Кількість нормальних записів значно більша за кількість аномалій. Методи: Oversampling аномалій (штучне збільшення їх кількості, наприклад через SMOTE). Undersampling нормальних даних. Мета: Поліпшити здатність моделі виявляти рідкісні аномалії. Обробка часових рядів Якщо дані мають часову природу (логі серверів, телеметрія IoT): Створення лагових змінних (попередні значення). Обчислення скользячого середнього. Використання сезонних компонентів. Ефективна обробка даних є фундаментом для виявлення аномалій. Правильно підготовлені, очищені, нормалізовані та оптимізовані дані дозволяють суттєво підвищити якість будь-якого алгоритму виявлення аномалій.

2.3 Методи аналізу часових рядів для виявлення аномалій

Часові ряди — це послідовності значень, впорядковані за часом. Виявлення аномалій у таких даних є особливо актуальним у хмарних середовищах, де системні метрики (навантаження CPU, пам’ять, обсяг трафіку, кількість запитів тощо) змінюються з часом. Залежно від задачі, використовують статистичні та машинні методи [10].

1. Статистичні методи Скользяче середнє та ковзне вікно (Moving Average, Rolling Window) Обчислює середнє значення за певний часовий період. Відхилення від середнього можуть сигналізувати про аномалію. Застосовується для згладжування шуму. ARIMA / SARIMA (AutoRegressive Integrated Moving Average) Класичні методи прогнозування часових рядів. ARIMA підходить для стаціонарних рядів, SARIMA — для сезонних. Якщо спостереження сильно відрізняється від прогнозу — це може бути аномалія. Control Charts (Шухарта, EWMA, CUSUM) Контрольні карти

перевіряють, чи не виходить сигнал за встановлені межі. EWMA підходить для виявлення повільних змін, а CUSUM — для накопичення малих змін.

2. Моделі машинного навчання для часових рядів Recurrent Neural Networks (RNN). Здатні враховувати послідовність попередніх значень. Недолік — проблема затухаючого градієнта. Long Short-Term Memory (LSTM) Вдосконалена версія RNN з "пам’яттю".

Вчиться на "нормальній поведінці", а великі помилки прогнозу вважаються аномаліями. GRU (Gated Recurrent Units) Менш ресурсоємна альтернатива LSTM. Швидша у навчанні, часто не гірша за точністю. Facebook Prophet Зручна бібліотека для моделювання тренду, сезонності та свят. Може будувати прогноз і виділяти точки, що відхиляються. Autoencoder для часових рядів

Навчаються реконструювати вхідний сигнал. Погана реконструкція — ймовірна аномалія. Можна застосовувати з LSTM, CNN або простими шарами. One-Class SVM або Isolation Forest для віконних зрізів Кожне "вікно часу" розглядається як вектор. Потім використовується модель без учителя.

Коли ми працюємо з даними, які змінюються у часі — наприклад, це можуть бути показники навантаження на сервер або кількість транзакцій за годину — дуже важливо враховувати саму часову залежність. Просте виявлення аномалій тут не спрацює: потрібно розуміти, як значення змінюються у часі.

Є кілька основних підходів для аналізу часових рядів на наявність аномалій:

Класичні статистичні методи

Скользяче середнє та ковзна дисперсія (Moving Average, Moving Variance)

Ідея проста: ми обчислюємо середнє значення на певному вікні часу та дивимось, чи нове значення виходить за межі очікуваного діапазону.

Якщо новий показник суттєво "вибивається" за межі середнього значення плюс/мінус стандартне відхилення — це може бути аномалія. Метод порогового відхилення (Threshold Method) Задаємо допустимі пороги для певної метрики. Якщо значення виходить за межі — фіксуємо аномалію.

Наприклад, якщо середня кількість запитів на сервер на годину — 1000, то різкий стрибок до 5000 можна розглядати як аномалію. Моделі часових рядів ARIMA (AutoRegressive Integrated Moving Average)

Один із найстаріших і популярних підходів для передбачення значень у часових рядах. ARIMA дозволяє моделювати залежності від попередніх значень (авторегресія) та враховувати сезонність. Аномалії можна виявити шляхом порівняння прогнозованого значення з фактичним. SARIMA (Seasonal ARIMA) Розширення ARIMA для обробки сезонних патернів, наприклад, щоденних піків навантаження. Сучасні підходи на основі машинного навчання LSTM (Long Short-Term Memory) Спеціальні нейронні мережі, які дуже добре працюють із часовими рядами. Мережа навчається "пам'ятати" попередні значення і передбачати майбутні.

Якщо фактичне значення сильно відрізняється від прогнозованого — фіксуємо аномалію. Autoencoders для часових рядів Навчаємо мережу стискати та відновлювати часові дані. Якщо мережа погано відновлює певні точки — це може сигналізувати про аномалії. Інші корисні підходи. Детектування змінної дисперсії (Change Point Detection)

Якщо різко змінюється середнє або варіативність у ряді — це ознака зміни стану системи, а можливо й аномалії. Сезонна декомпозиція часових рядів (Seasonal Decomposition of Time Series - STL). Розкладаємо ряд на тренд, сезонність і залишкову складову. Аналізуємо залишкову частину для виявлення несподіваних стрибків або падінь. Аналіз часових рядів — це окрема велика тема, де важливо враховувати не тільки самі значення, а й їх зміну у часі. Від правильного вибору методу залежить успіх у виявленні справжніх аномалій без зайвих помилкових спрацьовувань.

Методи аналізу часових рядів для виявлення аномалій відіграють ключову роль у сфері моніторингу складних систем, зокрема у хмарних середовищах, де постійно генерується великий обсяг динамічних даних. Основна мета таких методів — виявлення відхилень у поведінці системи, які можуть свідчити про кібератаки, технічні збої, перевантаження, помилки конфігурації або інші інциденти. Аналіз часових рядів базується на припущенні, що в нормальному режимі дані мають певну

структуру, закономірності, циклічність або тренди, порушення яких вказує на наявність аномалій.

Традиційно, одним із найстаріших способів аналізу є статистичне моделювання, яке дозволяє виявляти глобальні та локальні відхилення. Методи, що ґрунтуються на середньому, стандартному відхиленні або ковзних середніх, добре працюють при припущенні стаціонарності даних. Однак у реальних хмарних середовищах дані часто є нестабільними, схильними до раптових змін або сезонності. У таких випадках ефективнішими є більш гнучкі підходи, зокрема авторегресійні моделі (наприклад, ARIMA), які можуть враховувати часові залежності, або моделі на основі ковзаючих вікон, здатні відслідковувати локальні аномалії у вибірках.

У сучасних системах, де необхідна обробка великих масивів даних у режимі реального часу, все більшої популярності набувають методи машинного навчання. Такі підходи не покладаються на жорстко задані правила, а здатні виявляти складні патерни в поведінці системи. Серед них можна виділити алгоритми кластеризації, як-от k-Means, що дозволяють ідентифікувати аномальні точки як ті, що погано узгоджуються з центрами кластерів. Іншим підходом є метод опорних векторів (One-Class SVM), який створює межу нормальної поведінки і відсікає все, що виходить за її межі. Більш новітні підходи, як Local Outlier Factor, оцінюють локальну щільність навколишніх точок і виявляють ті, які сильно від неї відрізняються. Однак ці методи часто вимагають налаштування гіперпараметрів, нормалізації вхідних даних і мають обмеження при роботі з високовимірними часовими рядами.

Особливої уваги заслуговують методи глибокого навчання, які демонструють високу ефективність при роботі з нестаціонарними, складними та високорозмірними часовими рядами. Зокрема, автокодери (autoencoders) є одним із провідних інструментів для виявлення аномалій. Ці моделі навчаються стискати та відновлювати нормальні дані. Якщо вхідні дані мають аномальний характер, мережа не може їх коректно реконструювати, що сигналізує про відхилення. Варіаційні автокодери (VAE) додають стохастичну складову, дозволяючи отримати більш гнучке представлення простору ознак. Інший клас моделей — рекурентні нейронні мережі (RNN) і зокрема LSTM, здатні ефективно моделювати довгострокові

залежності у даних. Такі мережі ідеально підходять для виявлення контекстуальних аномалій, коли відхилення має значення лише у певній послідовності подій.

У деяких підходах застосовуються генеративні моделі, наприклад GAN (Generative Adversarial Networks), які здатні навчатись на нормальних даних і виявляти ті, які не вдається реконструювати. Унікальність таких методів полягає в їх здатності розрізняти справжні та підроблені сигнали на основі поведінкових шаблонів. Попри складність у налаштуванні та велику обчислювальну вартість, ці підходи є найсучаснішими у сфері кібербезпеки та обробки великих потоків даних.

Важливу роль у процесі відіграє також попередня обробка даних: нормалізація, видалення пропусків, зменшення розмірності та побудова ознак (feature engineering). Наприклад, із простого ряду чисел можна створювати похідні ознаки, як-от швидкість зміни, кут нахилу, середнє ковзне значення або рівень шуму. Це підвищує чутливість моделей до прихованих змін у поведінці системи.

У практичних умовах хмарного середовища часто використовується гібридний підхід — поєднання кількох методів, що дозволяє збалансувати точність, швидкість та обчислювальні ресурси. Наприклад, спочатку застосовується проста модель для фільтрації грубих аномалій, а потім точніша нейронна мережа для глибокого аналізу залишених даних. Такий підхід знижує навантаження на систему й дозволяє досягти високої адаптивності та точності в реальному часі.

Таким чином, аналіз часових рядів для виявлення аномалій — це складна, багатогранна задача, яка потребує ретельного вибору інструментів і методів в залежності від природи даних, цілей системи моніторингу та технічних обмежень. У нашій роботі було використано автокодер, адаптований для аналізу згенерованих потокових даних, що дозволило досягти гнучкого, автоматизованого та ефективного виявлення аномалій у хмарному середовищі.

2.4 Методи машинного навчання для виявлення аномалій

Із розвитком технологій машинного навчання виникла можливість вирішувати завдання виявлення аномалій набагато гнучкіше та точніше.

Традиційні статистичні підходи добре працюють для простих задач, але реальні дані часто мають складні патерни і приховані взаємозв'язки. Саме тут на допомогу приходить машинне навчання [11]. Основні підходи Навчання без учителя (Unsupervised Learning) Це коли ми не маємо заздалегідь розмічених даних ("аномалія" чи "норма"). Алгоритм самостійно шукає закономірності у даних і виділяє ті точки, що "не схожі" на інші. Найпоширеніші методи: K-Means clustering Алгоритм розбиває дані на групи (кластери). Якщо якась точка знаходиться далеко від центру кластера — це може бути аномалія. DBSCAN (Density-Based Spatial Clustering of Applications with Noise) Групує точки за щільністю розташування. Точки, які не входять до жодної щільної групи, вважаються аномаліями. One-Class SVM (Support Vector Machine)

Спеціальна версія машин векторної підтримки, що навчається тільки на "нормальних" даних. Визначає кордон, за який потрапляють лише "дивні" або аномальні дані. Навчання з учителем (Supervised Learning)

Використовується тоді, коли у нас є розмічені дані: що є нормальною поведінкою, а що — аномалією.. На жаль, для багатьох реальних завдань таких даних обмаль або взагалі немає, але якщо вони є — результат може бути дуже точним.

Типові моделі: Random Forest Створює багато дерев рішень і голосує, чи є приклад аномальним. Gradient Boosting (XGBoost, LightGBM) Більш потужні алгоритми ансамблю дерев, що дозволяють досягати високої точності в складних задачах. Logistic Regression Простий, але дуже ефективний метод для бінарної класифікації ("аномалія" чи "норма"). Глибинне навчання (Deep Learning) Для задач, де дані складні, високорозмірні та мають часову залежність, добре працюють нейронні мережі. Популярні методи: Autoencoders. Нейронна мережа навчається стискати дані (енкодер) і потім їх відновлювати (декодер). Якщо при відновленні мережа "помиляється" для певного прикладу — це може бути ознакою аномалії. LSTM (Long Short-Term Memory Networks) Ідеальні для аналізу послідовних даних (часових рядів). Мережа запам’ятовує попередній контекст і прогнозує наступні значення, виявляючи відхилення. Variational Autoencoders (VAE) і Generative Adversarial Networks (GANs)

Складні моделі, що дозволяють ще краще моделювати нормальну поведінку і виявляти навіть приховані аномалії.

Особливості застосування машинного навчання для виявлення аномалій Проблема незбалансованості класів Зазвичай аномалій набагато менше, ніж нормальних даних.

Тому класичні метрики точності (accuracy) тут погано працюють. Краще використовувати F1-score, Precision, Recall. Потреба в якісній підготовці даних. Перед тренуванням моделей дані потрібно масштабувати, нормалізувати, іноді — очищати від шуму. Підбір гіперпараметрів Для досягнення високої точності важливо правильно налаштувати параметри моделей: кількість кластерів, розмір прихованих шарів, регуляризацію тощо. Висока обчислювальна складність. Деякі методи глибинного навчання потребують великих обчислювальних ресурсів (GPU, хмарні сервіси).

Таблиця 2.2

Порівняння підходів

Підхід	**Переваги**	**Недоліки**
Класичні алгоритми	Простота, швидкість	Низька точність для складних даних
Машинне навчання	Гнучкість, висока якість	Потреба в якісних даних
Глибинне навчання	Найкраща якість для складних випадків	Велика складність реалізації

Машинне навчання пропонує широкий арсенал методів для виявлення аномалій. Вибір оптимальної моделі залежить від наявності розмічених даних, обсягу даних, вимог до точності та доступних ресурсів. Машинне навчання дозволяє автоматично виявляти закономірності та аномальні патерни в великих обсягах даних [12]. Методи умовно поділяються на три типи:

1. Навчання з учителем (Supervised Learning) Ці методи потребують міток («норма» / «аномалія»). Логістична регресія. Лінійний класифікатор.Придатний для

простих задач. Decision Trees / Random Forest. Надають інтерпретовані правила. Висока точність за наявності достатнього обсягу даних. XGBoost / LightGBM. Потужні градієнтні бустингові алгоритми. Висока продуктивність і точність навіть на "брудних" даних. Support Vector Machines (SVM) Особливо ефективний у задачах з чітким розділенням класів. Підходить для задач з великою кількістю ознак.

2. Навчання без учителя (Unsupervised Learning)

Коли немає міток — шукаємо аномалії як відхилення від «звичайних» патернів. k-Means / DBSCAN Якщо об’єкт далеко від кластеру — можливо, це аномалія. DBSCAN здатний виявити "шум" — ізольовані точки. One-Class SVM Створює межу навколо "нормальних" точок. Все поза — вважається аномалією. Isolation Forest Ізолює спостереження в деревах. Аномалії ізолюються швидше (менше глибина). Ефективний для великих наборів даних. Local Outlier Factor (LOF) Визначає ступінь "ізольованості" точки. Працює на основі локальної щільності.

3. Напівконтрольоване (semi-supervised) / Глибоке навчання (Deep Learning)

Autoencoders Вчаться стискати та відновлювати дані. Якщо дані сильно відрізняються — помилка реконструкції зростає → аномалія. Variational Autoencoders (VAE). Дають ймовірнісну оцінку. Працюють краще з шумними або нестабільними даними. GAN (Generative Adversarial Networks)

Генерують "нормальні" приклади, а невідповідність між генерованими і реальними — аномалія. LSTM Autoencoder Поєднує LSTM і Autoencoder — для складних залежностей у часових рядах У своїй роботі я обрав гібридний підхід: використання Autoencoder для аналізу нормальної поведінки даних та подальшого виявлення аномалій на основі помилки відновлення.

Виявлення аномалій — це одна з ключових задач у сфері машинного навчання, що застосовується для ідентифікації відхилень від нормальної поведінки в даних. У хмарних середовищах, де спостерігається великий обсяг потокових і структурованих/неструктурованих даних, ефективність застосовуваного методу значно впливає на здатність вчасно виявити загрози, помилки або вторгнення [8].

Методи машинного навчання умовно поділяються на три основні категорії:

1. Методи з учителем (Supervised Learning) Ці методи потребують наявності мічених даних (тобто прикладів нормальної та аномальної поведінки). Основні алгоритми: Логістична регресія (Logistic Regression) — підходить для бінарної класифікації (норма/аномалія), проста в реалізації. Support Vector Machine (SVM) — добре працює при чітко роздільних класах.Рішення дерев (Decision Trees) та ансамблі (Random Forest, XGBoost) — забезпечують інтерпретовані моделі, здатні знаходити складні закономірності. Нейронні мережі — здатні моделювати складні залежності в великих обсягах даних [13]. Проблема: в хмарних середовищах часто відсутні дані з явними мітками аномалій, тому такі методи не завжди практичні.

2. Методи без учителя (Unsupervised Learning) Ці методи не потребують попереднього маркування даних. Вони використовуються для виявлення "незвичних" або рідкісних зразків, що відрізняються від більшості. Основні алгоритми: K-середніх (K-Means) — знаходить кластери, а аномалії — це точки, що далеко від центрів. DBSCAN — ідентифікує щільні області в даних; віддалені точки вважаються аномаліями. One-Class SVM — навчається тільки на нормальних даних і виявляє відхилення. Principal Component Analysis (PCA) — використовується для зменшення розмірності та виявлення відхилень. Ізоляційні дерева (Isolation Forest) — ізолюють аномалії на меншу кількість кроків.

3. Методи з частковим навчанням (Semi-Supervised Learning) У реальних умовах зазвичай доступні лише нормальні приклади. Саме тому напівконтрольоване навчання стало популярним для задач виявлення аномалій. Основні методи: Автокодери (Autoencoders) — нейронні мережі, які навчаються відтворювати нормальні зразки. Якщо об’єкт сильно відрізняється від нормального — виникає велика помилка реконструкції. Variational Autoencoder (VAE) — стохастичне розширення автокодера, яке генерує розподіли замість фіксованих точок. Generative Adversarial Networks (GANs) — генерують дані, подібні до нормальних, і виявляють ті, що не схожі. LSTM-автокодери — поєднання автокодерів із рекурентними мережами для роботи з часовими рядами.

Для задач виявлення аномалій у хмарних середовищах найкраще підходять напівконтрольовані методи, особливо автокодери, оскільки вони здатні адаптуватися

до нормальної поведінки системи без потреби в маркуванні аномальних даних. Такий підхід забезпечує масштабованість, гнучкість і точність в умовах динамічного хмарного середовища.

2.5 Підходи до вибору алгоритмів виявлення аномалій

Вибір алгоритму для виявлення аномалій є критично важливим етапом розробки системи моніторингу та захисту. Правильний підхід до вибору методу визначає не тільки точність виявлення аномалій, але і швидкість роботи системи, її масштабованість та здатність адаптуватися до змін у даних [8].

Основні критерії вибору алгоритму Тип доступних даних Розмічені дані (Supervised learning): Якщо є приклади як нормальної поведінки, так і аномалій — краще використовувати алгоритми навчання з учителем, такі як Random Forest, XGBoost чи нейронні мережі для класифікації.

Нерозмічені дані (Unsupervised learning): У більшості реальних сценаріїв розмічені аномалії відсутні, тому використовують кластеризацію, One-Class SVM, Autoencoder або ізоляційні ліси (Isolation Forest).

Характеристика аномалій. Точкава аномалія: Одна окрема спостережна точка сильно відрізняється від інших (наприклад, надмірний стрибок температури).

Контекстуальна аномалія: Значення нормальне само по собі, але аномальне в певному контексті (наприклад, високе навантаження на сервер вночі). Колективна аномалія: Група спостережень, що разом формують аномалію (наприклад, серія помилок у логах). Відповідно до типу аномалій обирають метод: Для точкових — прості моделі можуть бути достатні. Для контекстуальних і колективних — краще підходять методи з урахуванням часової залежності, наприклад, LSTM. Розмір і складність даних Малі обсяги даних: Перевагу варто надавати простішим моделям, які не потребують великої кількості даних для навчання (One-Class SVM, Isolation Forest). Великі обсяги даних: Перевагу мають масштабовані рішення, які можуть обробляти великі датасети (кластеризація, Autoencoders, ансамблі моделей).

Висока розмірність (багато ознак): Краще використовувати алгоритми, які добре справляються з "прокляттям розмірності", наприклад, метод головних компонент (PCA) у поєднанні з класифікаторами. Вимоги до точності та швидкодії Висока точність важливіша за швидкість: Використовують складніші моделі, наприклад, нейронні мережі або ансамблі алгоритмів. Необхідна реальна часова обробка: Перевагу надають швидким моделям із низькою затримкою: Isolation Forest, K-Means, спрощені Autoencoder'и. Стратегії вибору підходу

Прототипування кількох моделей. На практиці часто обирають кілька перспективних алгоритмів і тестують їх на одному і тому ж датасеті, порівнюючи показники якості.

Таблиця 2.3

Приклад практичного підбору алгоритму

Умова	Оптимальний вибір
Мало даних, немає розмітки	Isolation Forest, One-Class SVM
Великі дані, складна структура	Autoencoder
Потрібно працювати в реальному часі	K-Means, Isolation Forest
Важливі послідовні залежності	LSTM, GRU

У процесі розробки систем виявлення аномалій особливу роль відіграє правильний вибір алгоритму, що базується на характеристиках вхідних даних, архітектурних вимогах системи та практичних обмеженнях на обчислювальні ресурси. Ключовими факторами, що впливають на вибір, є тип часових рядів, рівень шуму в даних, наявність або відсутність маркування аномальних значень, а також потреба в обробці даних у реальному часі. Характеристика часових рядів, зокрема їхня стаціонарність або наявність трендів, значною мірою визначає придатність певних класів алгоритмів. Так, для стаціонарних даних можуть бути ефективними класичні статистичні підходи, наприклад моделі ковзного середнього або авторегресії, тоді як для більш складних структур з динамічними шаблонами доцільно застосовувати нейронні мережі або інші глибокі моделі.

Тип аномалії, яку необхідно виявити, також безпосередньо впливає на вибір алгоритму. Точкові аномалії можна ефективно виявляти за допомогою простих

методів, таких як Z-score або Isolation Forest. У разі контекстуальних або колективних аномалій виникає потреба в моделях, здатних ураховувати часову залежність, наприклад LSTM або автокодерах, які здатні реконструювати нормальні шаблони поведінки й фіксувати відхилення. Важливо враховувати і той факт, чи доступні мітки для навчання — за їх відсутності слід надавати перевагу безконтрольним або напівконтрольованим алгоритмам, які дозволяють будувати моделі лише на основі нормальних прикладів.

При виборі алгоритму виявлення аномалій важливо враховувати не лише тип даних, а й особливості прикладного середовища, доступність обчислювальних ресурсів та вимоги до пояснюваності результатів. Наприклад, у задачах, де критичною є прозорість прийнятих рішень (наприклад, у банківській сфері або державному управлінні), доцільно використовувати методи, які легко інтерпретуються, як-от дерева рішень чи правила на основі статистичних порогів.

У високонавантажених або реальних потокових системах (streaming environments) перевагу надають алгоритмам із низькою часовою складністю та можливістю оновлення моделі в реальному часі. У таких випадках обирають спрощені нейромережі, методи ковзного вікна або онлайн-варіанти кластеризаційних алгоритмів.

Також важливо враховувати співвідношення нормальних і аномальних прикладів у вибірці. У випадках із сильно незбалансованими даними ефективнішими можуть бути методи без учителя або підходи зі штучною генерацією аномалій для навчання моделі.

Значну роль відіграє наявність доменної експертизи — її можна інтегрувати у вигляді обмежень, фільтрів або інженерії ознак перед навчанням моделі, що дозволяє суттєво покращити якість виявлення без підвищення складності алгоритму.

У практиці часто застосовується експериментальний підхід: одночасне тестування кількох алгоритмів на підмножині реальних даних з наступною оцінкою їх ефективності за критеріями точності, повноти, F1-міри та кількості помилкових спрацювань.

У середовищах з обмеженими обчислювальними ресурсами або при вимогах до високої швидкості обробки варто застосовувати методи з низькою складністю, як-от статистичні моделі, дерева рішень або полегшені кластеризаційні алгоритми. Для задач, що потребують високої точності й глибокого моделювання складних залежностей, доцільним є використання глибоких нейронних мереж, які, однак, потребують GPU, значної кількості оперативної пам’яті й часу на навчання. Слід також враховувати масштабованість: деякі алгоритми не адаптовані до великих обсягів даних або не підтримують інкрементальне навчання, що унеможливлює їх використання у потокових системах.

Окрему увагу необхідно приділити інтерпретованості моделей. У сферах, пов’язаних із кібербезпекою або фінансовим моніторингом, важливо не лише виявити аномалію, а й пояснити причину її виникнення. Це вимагає або застосування простих, прозорих моделей, або використання додаткових засобів інтерпретації результатів, таких як SHAP чи LIME. Під час вибору алгоритму також слід зважати на наявність пропусків у даних, які можуть викликати некоректну роботу моделей без механізмів їх обробки, а також на незбалансованість наборів даних, коли аномальні значення зустрічаються надзвичайно рідко.

Загалом вибір алгоритму має бути зумовлений багатьма чинниками, включаючи тип виявлюваних аномалій, характеристики даних, наявність міток, вимоги до ресурсів, потребу в обробці в реальному часі та необхідність інтерпретації результатів. У межах цього дослідження було обрано метод на основі автокодера, що забезпечує баланс між точністю, масштабованістю, гнучкістю та здатністю до самонавчання на основі лише нормальних прикладів. Такий підхід дозволяє ефективно виявляти широкий спектр аномалій у хмарному середовищі, забезпечуючи автоматизований моніторинг із мінімальним втручанням користувача та з високим ступенем адаптивності до нових даних.

Вибір алгоритму для виявлення аномалій залежить від багатьох факторів: характеру даних, наявності розмітки, ресурсів для обчислень та вимог до системи. Ідеальний варіант часто комбінує кілька методів для досягнення найкращого балансу між точністю, швидкістю та стабільністю.

2.6 Проблеми та виклики в задачах виявлення аномалій

Задача виявлення аномалій має низку особливостей, які ускладнюють її реалізацію на практиці. Насамперед, це значний дисбаланс даних: аномальні події зустрічаються рідко, тому моделі часто переорієнтовуються на нормальну поведінку і можуть не помічати важливих відхилень. Ще однією проблемою є відсутність міток — у багатьох випадках дані не мають позначення "норма" чи "аномалія", що унеможливлює використання класичних методів навчання з учителем.

Крім того, хмарні середовища постійно змінюються, а поведінка системи може бути нестабільною, що призводить до так званого "concept drift" — зміни розподілу даних у часі. Це вимагає гнучких моделей, здатних адаптуватися до нових умов. Також слід враховувати шум у даних — помилки, пропуски чи артефакти, які можуть неправильно інтерпретуватись як аномалії.

Ще один виклик — обмеженість обчислювальних ресурсів: складні моделі, зокрема нейронні мережі, можуть бути надто важкими для використання у режимі реального часу. Тому важливо обирати методи, які забезпечують баланс між точністю, швидкістю та ресурсомісткістю.

Виявлення аномалій — це завжди непросте завдання, яке має безліч прихованих труднощів. Хоч на перший погляд здається, що знайти "незвичну поведінку" просто, на практиці цей процес супроводжується низкою суттєвих викликів, які можуть вплинути на точність, надійність і практичну придатність системи. Основні проблеми

1. Відсутність або недостатність розмічених даних Більшість реальних систем не мають чітких міток, що вказують, яка поведінка є нормальною, а яка — аномальною. Аномальні ситуації трапляються рідко, і дані про них часто неповні або нечіткі. Це змушує використовувати складні методи без учителя або напівконтрольоване навчання.

2. Імбаланс класів Навіть якщо розмічені дані є, кількість нормальних прикладів зазвичай значно переважає кількість аномалій. Такий дисбаланс призводить до того, що класичні алгоритми можуть "ігнорувати" аномалії і давати високу загальну точність, але низьку здатність виявляти саме аномалії (низький Recall).

3. Висока розмірність даних Коли кількість ознак у даних зростає, стає важко будувати точні моделі через "прокляття розмірності". Багато моделей починають погано узагальнювати інформацію або вимагають непропорційно великої кількості обчислювальних ресурсів.

4. Змінність даних із часом (Concept Drift) У реальних системах "норма" може змінюватися: те, що сьогодні є нормальним, завтра може стати підозрілим. Модель повинна вміти адаптуватися до змін без потреби в постійному повному перенавчанні. Концептуальні труднощі

5. Невизначеність у визначенні аномалій В деяких випадках навіть експерти не можуть однозначно сказати, чи є певна подія аномальною. Критерії аномальності можуть бути розмитими, залежати від контексту або суб'єктивними.

6. Різноманіття типів аномалій Одна система може стикатися з точковими, контекстуальними й колективними аномаліями одночасно. Кожен тип потребує різних методів обробки та виявлення. Практичні труднощі

7. Витрати на обчислення Деякі алгоритми, наприклад глибокі нейронні мережі, потребують значних обчислювальних потужностей для тренування й роботи, що ускладнює їх застосування у ресурсно обмежених середовищах (наприклад, на мобільних пристроях чи в реальному часі).

8. Необхідність мінімізувати хибні спрацювання У деяких галузях (наприклад, в безпеці чи медицині) навіть невелика кількість помилкових аномалій (false positives) може призвести до серйозних наслідків: витрат ресурсів, паніки або втрачених можливостей [14].

Завдання виявлення аномалій є однією з найскладніших у сфері аналізу даних, особливо в умовах хмарних обчислень. Основні проблеми та виклики можна поділити на кілька категорій:

1. Дисбаланс класів (рідкість аномалій): Аномальні події зазвичай трапляються дуже рідко, що призводить до сильно незбалансованих наборів даних. У таких умовах традиційні моделі машинного навчання схильні ігнорувати рідкісні події та надавати пріоритет нормальній поведінці, що знижує здатність системи виявляти відхилення.

2. Відсутність міток (лабе́лів): У багатьох випадках, зокрема в реальному часі чи при обробці потокових даних, неможливо заздалегідь позначити, що є нормою, а що — аномалією. Це унеможливлює використання методів із учителем, таких як класифікація, і змушує застосовувати алгоритми без учителя — як-от автокодери, кластеризація, статистичні методи.

3. Змінність середовища (concept drift): У хмарних інфраструктурах параметри навантаження, трафіку, обчислень тощо можуть змінюватися динамічно. Це означає, що те, що було нормальним сьогодні, завтра може бути аномалією і навпаки. Постійна адаптація моделі до нових умов — складна, але критично важлива задача.

4. Високий рівень шуму в даних: У реальних умовах дані часто містять помилки, пропуски, дублі чи випадкові відхилення, які не є істинними аномаліями. Відрізнити шум від справжньої загрози — ще один важливий виклик.

5. Вибір ознак (фіч): Для ефективного виявлення аномалій важливо вибрати релевантні ознаки. У хмарному середовищі ці ознаки можуть бути нестабільними або погано задокументованими, що ускладнює побудову якісної моделі.

6. Обмеження обчислювальних ресурсів: Моделі виявлення аномалій, особливо глибокі нейронні мережі, можуть вимагати значних обчислювальних потужностей. У задачах моніторингу в реальному часі або при використанні в edge-комп’ютерах це стає критичним.

7. Потреба в пояснюваності: У багатьох критичних застосуваннях (наприклад, в IT-безпеці або фінансах) важливо не просто виявити аномалію, а й пояснити, чому вона вважається відхиленням. Багато сучасних моделей, зокрема нейронні мережі, є "чорними ящиками", що ускладнює інтерпретацію їх результатів.

8. Верифікація результатів: Навіть після того як система виявила потенційну аномалію, необхідно перевірити, чи дійсно це відхилення є загрозою. Без ручної перевірки зростає ризик хибнопозитивних спрацювань, що знижує довіру до системи.

Виявлення аномалій є однією з найскладніших задач у сфері аналізу даних, машинного навчання та інформаційної безпеки. Незважаючи на широку область застосування — від кібербезпеки до фінансового моніторингу й охорони здоров’я — ця задача супроводжується рядом серйозних викликів:

Наявність незбалансованих даних У більшості практичних випадків аномальні події трапляються значно рідше, ніж нормальні. Це призводить до дисбалансу класів, що ускладнює навчання моделей та може спричинити ігнорування рідкісних, але критичних випадків.

Відсутність мічених даних (labeling) Часто дані, що використовуються для виявлення аномалій, не мають міток "аномалія/норма", або такі мітки надто дорогі для отримання, оскільки потребують участі експертів. Це унеможливлює використання класичних методів супервізованого навчання.

Ще однією важливою проблемою є складність оцінювання якості виявлення аномалій. Через відсутність чітко визначених еталонів або міток, традиційні метрики, такі як точність або повнота, не завжди є репрезентативними. Це ускладнює процес порівняння різних підходів та вибір оптимального рішення.

Також слід враховувати обчислювальну складність. Методи виявлення аномалій, особливо засновані на глибокому навчанні або побудові складних моделей, можуть потребувати значних ресурсів, що обмежує їх застосування в реальному часі або в системах з обмеженими обчислювальними можливостями.

Змінність середовища (data drift / concept drift) Поведінка системи, яку аналізують, може змінюватися з часом. Модель, навчена на одному розподілі даних, може швидко втратити ефективність у нових умовах без додаткового перенавчання або адаптації.

Шум та похибки в даних У хмарних або реальних середовищах дані часто містять помилки, пропуски або артефакти, які можуть бути сприйняті як аномалії, хоча такими не є. Це призводить до високої кількості хибнопозитивних спрацьовувань.

Невизначеність поняття "аномалія" У різних контекстах "аномалія" може мати різне значення. Те, що є відхиленням у одному середовищі, може бути нормальним у іншому. Це ускладнює побудову універсальних алгоритмів.

Вимоги до обчислювальних ресурсів Деякі сучасні алгоритми виявлення аномалій, особливо на основі глибокого навчання (наприклад, автокодери),

потребують значних ресурсів для навчання, що може бути критично в реальному часі або в обмежених середовищах.

Використання у реальному часі Не всі методи однаково добре працюють у режимі реального часу. Важливо забезпечити швидку обробку потокових даних та оперативне реагування на виявлені інциденти.

Виявлення аномалій вважається однією з найскладніших задач у галузях аналізу даних, машинного навчання та інформаційної безпеки. Незважаючи на широкий спектр застосування — від кіберзахисту до фінансових систем і медичної сфери — ця проблема супроводжується низкою серйозних труднощів:

Дисбаланс у даних. У більшості реальних ситуацій аномальні події виникають значно рідше, ніж типові. Такий дисбаланс між класами ускладнює процес навчання моделей і може призвести до того, що критичні, хоч і рідкісні, події залишаються непоміченими.

Брак мічених даних. Часто дані, що використовуються для виявлення аномалій, не мають попередніх позначень ("норма"/"аномалія"), або отримання таких міток потребує залучення фахівців і є надто витратним. Це обмежує застосування традиційних методів з учителем.

Зміна характеристик даних (data drift / concept drift). Поведінка об’єктів чи систем, що аналізуються, може змінюватися з часом. У результаті моделі, натреновані на попередніх даних, втрачають актуальність і точність, якщо їх не оновлювати або не адаптувати до нових умов [15].

Задача виявлення аномалій вимагає уважного підходу до збору даних, побудови моделей, їх тестування та постійного вдосконалення. Виклики виникають як на рівні даних, так і на рівні алгоритмів та інтеграції у реальні системи. Успішна система виявлення аномалій — це завжди баланс між точністю, адаптивністю, ресурсами та реальними потребами бізнесу або середовища, де вона працює.

Задача виявлення аномалій, попри свою важливість у забезпеченні безпеки, стабільності та якості функціонування інформаційних систем, залишається однією з найскладніших у сфері обробки даних. Її складність зумовлена низкою

фундаментальних проблем і практичних викликів, що виникають як на етапі підготовки даних, так і під час побудови моделей, інтерпретації результатів та інтеграції в реальні системи.

Однією з ключових проблем є відсутність чіткої та уніфікованої дефініції аномалії. У різних доменах і сценаріях одна й та сама подія може інтерпретуватись як нормальна або аномальна, що ускладнює формалізацію задачі. Це особливо важливо в хмарних середовищах, де поведінка системи постійно змінюється внаслідок масштабування, автоматичного перерозподілу ресурсів і взаємодії багатьох компонентів.

Ще одним викликом є сильна дисбалансованість даних — аномальні випадки трапляються рідко, що призводить до труднощів у навчанні моделей. Більшість традиційних алгоритмів машинного навчання орієнтовані на збалансовані набори даних, тому без застосування спеціальних технік (наприклад, oversampling, anomaly synthesis або спеціалізованих функцій втрат) вони демонструють низьку ефективність. Окрім того, аномалії можуть бути дуже різноманітними за природою, що ускладнює побудову єдиної моделі, здатної покривати всі типи відхилень.

Іншою суттєвою проблемою є наявність шумів і пропусків у даних. У реальних умовах дані часто містять похибки вимірювання, часові затримки, дублікати або відсутні значення. Ці фактори можуть спотворювати розподіл нормальної поведінки та призводити до високого рівня хибнопозитивних спрацювань. Надмірна кількість таких спрацювань знижує довіру до системи моніторингу, перевантажує операторів і знижує загальну ефективність.

Значні виклики також постають у процесі валідації моделей. Відсутність маркованих даних із точно визначеними аномаліями робить неможливим використання традиційних метрик оцінки якості (precision, recall, F1-score) без штучного або ручного анотування. Навіть у разі наявності часткових міток, аномалії часто є суб’єктивними або контекстно залежними, що створює неоднозначність у побудові еталонів.

Додаткову складність становить потреба в реальному часі обробки даних. У сучасних динамічних середовищах, зокрема в хмарних платформах і системах IoT,

рішення щодо аномалії мають прийматися миттєво, що обмежує використання ресурсоємних або повільних методів. Алгоритми повинні мати здатність до інкрементального навчання та бути стійкими до змін у потоці даних (concept drift).

Також варто зазначити виклики, пов’язані з інтерпретацією результатів. Для багатьох моделей, особливо глибоких нейронних мереж, результат є “чорною скринькою”, що ускладнює пояснення причин спрацювання або прийнятого рішення. У сферах, де критично важлива довіра до систем (медицина, фінанси, безпека), така непрозорість є серйозною перепоною до впровадження.

На завершення слід вказати на виклики інтеграції систем виявлення аномалій у наявну інфраструктуру. Це включає вимоги до масштабованості, захисту персональних даних, відповідності нормативним актам (наприклад, GDPR) та підтримки сумісності з іншими системами моніторингу.

Таким чином, задача виявлення аномалій є комплексною проблемою, яка потребує мультидисциплінарного підходу, поєднання методів статистики, машинного навчання, системної інженерії та кібербезпеки. Її ефективне розв’язання можливе лише за умови врахування специфіки домену, обмежень даних, вимог до продуктивності й необхідності людської інтерпретації результатів.

2.7 Аналіз сучасних підходів до виявлення аномалій у хмарних середовищах

У хмарних обчисленнях — через динамічність, масштабованість і велику кількість взаємопов’язаних компонентів — виявлення аномалій є критично важливим для забезпечення безпеки, стабільності та продуктивності систем. У сучасній практиці застосовуються різні підходи до вирішення цієї задачі, які можна умовно класифікувати за методами обробки даних, характером навчання моделей і типами ознак, що аналізуються.

1. Статистичні методи

Це одні з найстаріших та найпростіших способів виявлення аномалій. До них належать: Визначення порогових значень (threshold-based detection). Методи контрольних карт Шухарта, CUSUM, EWMA. Оцінка щільності розподілу (наприклад, Gaussian distribution, KDE).Переваги: простота, швидкість, інтерпретованість. Недоліки: неефективні для складних або нелінійних залежностей у великих даних.

2. Машинне навчання без учителя (Unsupervised Learning)

У більшості хмарних середовищ мітки аномалій відсутні, тому популярності набули методи без учителя: Кластеризація (наприклад, k-Means, DBSCAN): об’єкти, що не входять до жодного кластеру або потрапляють у малі/рідкісні кластери, вважаються аномальними. Методи зниження розмірності (PCA, t-SNE): відхилення від головних компонент інтерпретуються як аномалії.

Autoencoder: глибока нейронна мережа, яка навчається реконструювати "нормальні" дані. Великі помилки реконструкції вказують на потенційні аномалії.

Isolation Forest: будує дерева для "ізоляції" даних, де аномальні точки ізолюються швидше.

Переваги: не вимагають розмічених даних, масштабовані, здатні виявляти нові типи аномалій. Недоліки: важко пояснити результат, іноді складні для налаштування.

3. Методи з учителем (Supervised Learning) Використовуються тоді, коли є мітки "норма/аномалія": Класифікація: Random Forest, SVM, XGBoost, логістична регресія. Глибоке навчання: CNN, LSTM — особливо корисні при роботі з часовими рядами. Переваги: висока точність у відомих сценаріях. Недоліки: необхідність великих розмічених датасетів, погана генералізація на нові типи аномалій.

4. Моделі часових рядів У хмарному середовищі важливу роль відіграє моніторинг метрик у реальному часі: ARIMA, SARIMA — класичні статистичні моделі. LSTM, GRU — глибокі рекурентні мережі, що здатні виявляти залежності в довгих часових рядах. Prophet (від Facebook) — сучасна модель з простою інтерпретацією, підтримкою сезонності та трендів. Переваги: адаптація до динаміки, ефективність для потокових даних. Недоліки: складність налаштування, чутливість до шуму.

5. Гібридні підходи Комбінування декількох методів — наприклад, autoencoder + класифікація, або статистичний фільтр + ML. Також активно розвиваються ensemble-моделі.

6. Методи на основі хмарної інфраструктури

Провайдери хмар, такі як AWS, Azure та Google Cloud, надають власні інструменти: AWS CloudWatch + Lambda + S3 — для моніторингу та обробки подій. Amazon Lookout for Metrics — сервіс для автоматичного виявлення аномалій. Azure Monitor + Log Analytics — для аналізу логів і оповіщення. Google Cloud Operations Suite — схожі можливості з підтримкою ML. Переваги: інтеграція з хмарними сервісами, масштабованість. Недоліки: вартість, обмежена кастомізація, залежність від провайдера.

Сучасні підходи до виявлення аномалій характеризуються великою різноманітністю методів, моделей та технік, що застосовуються залежно від типу даних, специфіки предметної області та вимог до точності, швидкості та інтерпретованості результатів. Вони умовно поділяються на статистичні методи, машинне навчання, гібридні системи та методи, орієнтовані на глибоке навчання.

Статистичні методи традиційно вважаються базовим інструментарієм для виявлення аномалій. До них належать моделі на основі ймовірнісного розподілу (наприклад, нормальний розподіл), ковзаючих середніх, меж довіри або параметричних тестів. Вони добре підходять для лінійних, стаціонарних процесів із передбачуваними характеристиками. Проте їх ефективність значно знижується в умовах високої складності та нелінійності даних, що властиві сучасним системам.

Методи машинного навчання пропонують більш гнучкий та адаптивний підхід. До найпопулярніших належать кластеризація (наприклад, K-means, DBSCAN), алгоритми ізоляції (Isolation Forest), методи на основі близькості (k-NN) та моделі одномірного класифікатора (One-Class SVM). Такі методи дозволяють виявляти як точкові, так і контекстні аномалії, і можуть працювати з високорозмірними даними. Водночас вони потребують ретельної підготовки даних та вибору гіперпараметрів, а також не завжди забезпечують достатню інтерпретованість результатів.

Останніми роками широке поширення отримали моделі глибокого навчання, зокрема автоенкодери, рекурентні нейронні мережі (RNN, LSTM), графові нейронні мережі (GNN) та моделі трансформерів. Вони демонструють високу ефективність у роботі з часовими рядами, мультимодальними або структурованими даними, здатні навчатися складним шаблонам поведінки та виявляти складні нелінійні аномалії. Водночас їх обчислювальна складність, потреба у великих обсягах даних та недостатня прозорість процесу прийняття рішень залишаються викликами для практичного впровадження.

Гібридні системи поєднують кілька підходів — наприклад, спершу фільтрують підозрілі точки за допомогою простих евристик або статистики, а потім уточнюють результат із використанням складної моделі. Такий підхід дозволяє зменшити навантаження на систему, підвищити швидкість обробки та покращити точність за рахунок багаторівневої фільтрації.

Окрему нішу займають методи, орієнтовані на потокові дані (streaming anomaly detection), які враховують потребу в аналізі в реальному часі. Це включає алгоритми з інкрементальним навчанням, алгоритми скользячого вікна, а також адаптивні версії існуючих моделей (наприклад, Online Isolation Forest, Streaming k-means). Вони особливо актуальні для хмарних обчислень, IoT та телеметричних систем.

Також заслуговують уваги підходи, що використовують пояснювальну аналітику (explainable AI, XAI) — вони дозволяють не лише виявляти аномалії, а й пояснювати причини їх виникнення. Це важливо для доменів із високою відповідальністю, таких як медицина, фінанси, державне управління. До таких підходів належать використання інтерпретованих моделей (decision trees, rule-based models) або інструментів пояснення (SHAP, LIME).

У цілому, вибір сучасного підходу до виявлення аномалій значною мірою залежить від характеру даних, задачі, вимог до продуктивності, доступних ресурсів та критичності системи. Найефективнішими виявляються системи, які поєднують кілька методів, використовують адаптивне навчання, забезпечують візуалізацію та пояснення результатів, а також інтегруються в загальну архітектуру моніторингу та реагування.

Цей розділ демонструє, що хоча існує багато сучасних підходів до виявлення аномалій, вибір конкретного методу залежить від цільової задачі, доступних даних і особливостей хмарного середовища. У роботі доцільно використовувати комбінацію кількох підходів для досягнення високої точності, адаптивності та продуктивності системи.

Хмарні середовища, завдяки своїй масштабованості та гнучкості, стають невід'ємною частиною сучасної IT-інфраструктури. Проте розподілений характер хмарних сервісів створює нові виклики в забезпеченні безпеки та виявленні аномальної активності. У цьому підрозділі розглядаються основні напрями та методи, що застосовуються для вирішення задачі виявлення аномалій саме в хмарних середовищах.

Одним із ключових напрямів є використання машинного навчання для побудови моделей нормальної поведінки користувачів, систем та мереж, і подальшого виявлення відхилень від цієї поведінки.

Таблиця 2.4

Порівняння основних характеристик методів виявлення аномалій

Метод	Короткий опис	Переваги	Недоліки
Автокодери (Autoencoders)	Нейронні мережі для навчання компактного представлення нормальних даних і виявлення відхилень	Висока точність для складних даних, добре масштабуються	Потребують великої кількості нормальних даних для навчання
Кластеризація (k-means, DBSCAN)	Групування схожих об'єктів; аномалії — це об'єкти поза кластерами	Простота реалізації, наочність	Погано працюють на високорозмірних або змінних даних
Методи щільності (LOF, Isolation Forest)	Оцінюють "ізольованість" об'єктів серед інших	Ефективні на розріджених даних, мінімальні припущення про розподіл даних	Можуть мати високу обчислювальну складність

Продовження таблиці 2.4

Аналіз поведінки користувачів (UEBA)	Виявлення відхилень від звичної поведінки користувачів	Висока ефективність для виявлення складних атак	Потребує постійного оновлення профілів користувачів
Моніторинг метрик продуктивності	Відстеження завантаження ресурсів і мережевого трафіку	Швидке виявлення DoS атак, збоїв	Важко налаштувати пороги для складних сценаріїв
Гібридні методи	Поєднання машинного навчання та статистичних методів	Підвищення точності та адаптивності	Складність налаштування і інтеграції
SIEM-системи	Інтеграція збору, аналізу і кореляції подій безпеки	Централізований моніторинг, гнучкі аналітичні можливості	Висока вартість, потребують складної настройки

Висновок до другого розділу

У другому розділі було проведено глибокий аналіз особливостей хмарних середовищ, їх вразливостей і основних загроз, що виникають у процесі експлуатації. Розглянуто різновиди аномалій, які можуть свідчити про порушення нормальної роботи систем або потенційні атаки, зокрема: точкові, контекстуальні та колективні аномалії.

Також були систематизовані основні підходи до виявлення аномалій: традиційні статистичні методи, алгоритми машинного навчання, гібридні та спеціалізовані рішення для хмарних середовищ. Особливу увагу приділено огляду сучасних досліджень і практик останніх років, які демонструють тенденцію до впровадження методів глибокого навчання та інтелектуального аналізу великих обсягів даних для забезпечення надійного моніторингу.

Виявлено основні проблеми, що ускладнюють процес виявлення аномалій у хмарних середовищах, серед яких: висока динамічність інфраструктури, різнорідність даних, обмеженість доступу до "золотого стандарту" даних, а також необхідність забезпечення високої точності при мінімальній кількості хибних спрацьовувань.

Проведений аналіз засвідчив актуальність обраної тематики дослідження та дозволив обґрунтувати необхідність подальшої розробки ефективних технологій виявлення аномалій у хмарних середовищах на основі штучного інтелекту.

РОЗДІЛ 3.

РОЗРОБКА ТА ВПРОВАДЖЕННЯ СИСТЕМИ ВИЯВЛЕННЯ АНОМАЛІЙ НА ОСНОВІ ШІ

3.1 Теоретичні підходи до виявлення аномалій у хмарних середовищах

Виявлення аномалій у хмарному середовищі є важливою складовою забезпечення безпеки, надійності та ефективності IT-інфраструктури. Аномалії можуть свідчити про порушення роботи системи, кіберзагрози, збої в мережі або несанкціоноване використання ресурсів. Теоретично існує декілька основних підходів до виявлення аномалій, які класифікуються за характером даних, що аналізуються, та за методами обробки [16].

Статистичні методи. Ці методи ґрунтуються на припущенні, що нормальна поведінка системи підпорядковується певному розподілу (наприклад, нормальному), а всі значення, що виходять за межі встановленого діапазону, вважаються аномальними. Основні приклади: Z-score (стандартизоване відхилення) Визначення порогів на основі довірчих інтервалів Метод ковзного середнього. Переваги: Простота реалізації Інтерпретованість результатів. Недоліки: Не завжди підходять для даних з високою варіативністю або складною динамікою. Не враховують взаємозв’язки між метриками. Методи кластеризації

Ці методи не потребують міток (unsupervised learning) та базуються на групуванні схожих об'єктів. Об’єкти, що не належать до жодного з кластерів або знаходяться далеко від центру, вважаються аномальними. Найпоширеніші алгоритми: K-Means. DBSCAN. Isolation Forest Переваги: Адаптивність до різних типів даних Не потребують навчального набору з аномаліями Недоліки: Потрібно задавати гіперпараметри (наприклад, кількість кластерів) Можуть бути нестабільними на малих наборах даних Методи машинного навчання

Ці методи навчаються на великому обсязі історичних даних, щоб відокремлювати нормальну поведінку від відхилень. Серед них: Support Vector Machines (SVM) Decision Trees та Random Forest K-Nearest Neighbors (KNN) Переваги: Можуть моделювати складні взаємозв'язки між параметрами Добре працюють при наявності маркованих даних Недоліки: Потребують навчання Вимагають обробки великого обсягу даних Глибинне навчання Методи глибокого навчання (Deep Learning) добре підходять для обробки високорозмірних, складних та нелінійних даних. Найпоширенішими є:

Автокодери (Autoencoders): нейронні мережі, що навчаються стискати вхідні дані та відновлювати їх. Значне відхилення між вхідними та вихідними даними (reconstruction error) може вказувати на аномалію. Recurrent Neural Networks (RNN) / LSTM: використовуються для аналізу часових рядів та динаміки поведінки систем. Generative Adversarial Networks (GAN): можуть бути використані для моделювання нормальної поведінки та генерації порівняльних сценаріїв. Переваги: Висока точність при достатній кількості даних Можливість виявляти складні та неочевидні аномалії Недоліки: Висока обчислювальна складність Необхідність попередньої обробки та масштабування даних Гібридні та контекстуальні підходи Сучасні системи часто комбінують кілька методів (наприклад, статистичний + автокодер), щоб досягти кращої точності. Також дедалі більшої популярності набувають контекстуальні методи, які враховують додаткову інформацію, як-от час доби, тип користувача або навантаження системи. Архітектура системи виявлення аномалій

Для виявлення аномалій у хмарному середовищі була розроблена прототипова система, що імітує реальний процес моніторингу ресурсів та виявлення відхилень у роботі. Основні компоненти архітектури: Генератор метрик хмарного середовища: Імітує поведінку хмарної інфраструктури, генеруючи дані про навантаження (CPU, RAM, Disk). Записує дані у CSV-файл або завантажує їх до S3 (або локального репозиторію).

Дані мають тимчасову мітку та значення метрик, які варіюються в межах норми, іноді з аномаліями (штучно введені піки або провали). Хмарне сховище (умовно AWS S3): Служить репозиторієм для зберігання згенерованих метрик.

Емуляція відбувається через локальну файлову систему для простоти розробки. Модуль виявлення аномалій: Побудований на основі Autoencoder — глибокої нейронної мережі для зниження розмірності та реконструкції. Навчена модель обчислює похибку реконструкції нових даних та визначає, чи виходить вона за встановлений поріг. Скрипт моніторингу (аналог AWS Lambda):Працює у циклі, перевіряючи нові дані щохвилини. Обробляє дані, викликає модель, фіксує результат, генерує лог, виводить повідомлення. У разі аномалії — викликає Telegram API.

Сервіс оповіщення (Telegram-бот): Надсилає повідомлення користувачеві при фіксації аномального спрацювання. Повідомлення містить інформацію про час події, тип метрики та значення.Компонент логування та візуалізації: Веде журнал виявлених аномалій.Створює графіки навантаження із позначенням підозрілих точок. Графіки та лог-файли зберігаються у відповідному каталозі.Переваги обраної архітектури:

Простота розгортання й тестуванняМодульність — кожен компонент можна змінити або вдосконалитиГнучкість у виборі хмарного середовищаЛегка масштабованість (у перспективі на реальний AWS). Виявлення аномалій (anomaly detection) є важливою задачею в сфері аналізу даних та кібербезпеки, яка передбачає ідентифікацію нетипових або підозрілих шаблонів, що не відповідають очікуваній поведінці системи. Теоретичні основи цієї задачі охоплюють широкий спектр математичних, статистичних та алгоритмічних підходів, які формують фундамент для розробки практичних рішень.

Одним з базових підходів є статистичне виявлення аномалій, яке ґрунтується на припущенні, що нормальні дані підкоряються певному ймовірнісному розподілу (часто — нормальному або гаусовому). Аномалії, в цьому контексті, розглядаються як спостереження, що мають низьку ймовірність згідно з обраним розподілом. До прикладу, якщо значення виходить за межі кількох стандартних відхилень від середнього, воно вважається аномальним. Такий підхід ефективний при наявності стаціонарних даних із відомими властивостями, але малопридатний у випадках високої динаміки чи складної структури даних.

Інший теоретичний напрям — класифікаційний підхід, який передбачає використання моделей машинного навчання для навчання на нормальних даних з подальшою перевіркою нових зразків. При цьому, в задачах без учителя, використовуються моделі One-Class SVM, які будують межу між нормальними даними та потенційними аномаліями. Перевагою є здатність моделі працювати без потреби в аномальних прикладах, однак точність сильно залежить від вибору ядра та параметрів.

Кластеризаційні підходи передбачають, що нормальні дані формують щільні кластери, тоді як аномальні — ізольовані точки або рідкісні структури. Алгоритми, такі як K-Means, DBSCAN чи Gaussian Mixture Models, дозволяють ідентифікувати точки, які знаходяться далеко від центру кластеру або не належать жодному кластеру. Проте ці методи чутливі до вибору кількості кластерів та інших гіперпараметрів.

Суттєвий внесок у теоретичну базу зробили підходи, засновані на відстані та щільності. До них належить, наприклад, метод k-найближчих сусідів (k-NN), де обчислюється середня відстань до k найближчих точок: якщо вона перевищує певний поріг, спостереження вважається аномальним. Також варто згадати алгоритми на зразок Local Outlier Factor (LOF), що враховують локальну щільність і дозволяють виявляти контекстні аномалії. Ці методи мають високий рівень інтуїтивності, однак масштабуються гірше при великих обсягах даних.

Окремий напрямок теоретичних досліджень стосується моделей часових рядів, де передбачається, що дані мають часову залежність. Зазвичай використовуються авторегресивні моделі (AR, ARIMA), що оцінюють відхилення від передбаченого значення. Аномалії виявляються як великі помилки прогнозу. Такий підхід дозволяє враховувати тренди, сезонність та циклічність, але не завжди справляється з нестаціонарністю або раптовими змінами динаміки.

У межах глибокого навчання теоретичний фундамент базується на нейронних мережах, зокрема автоенкодерах. Ці моделі вчаться стискати дані у компактне представлення та відновлювати їх назад. Висока помилка реконструкції свідчить про нетиповість даних. Автоенкодери добре працюють з високорозмірними вхідними

даними та мають високу здатність до узагальнення, однак вимагають ретельного налаштування та великої кількості якісних даних для тренування.

Існують також гібридні підходи, що комбінують переваги кількох методів. Наприклад, поєднання кластеризації з подальшим застосуванням автоенкодера або використання статистичних методів як попередньої обробки перед подачею в глибоку модель. Це дозволяє підвищити точність, знизити кількість хибнопозитивних спрацювань та адаптуватися до різноманітних типів даних.

Таким чином, теоретичні підходи до виявлення аномалій формують широкий арсенал інструментів, які, залежно від контексту задачі, можуть комбінуватися та модифікуватися. Їх ефективність значною мірою залежить від якості даних, параметрів моделі, розуміння контексту аномалії та можливості інтерпретації результатів у практичному застосуванні.

3.2 Переваги автокодера в контексті хмарних середовищ

У хмарних обчисленнях системи часто характеризуються великою кількістю параметрів, динамічними змінами навантаження та значною варіативністю нормальної поведінки. Це створює складності для традиційних методів виявлення аномалій. У цьому контексті автокодери (Autoencoders) як підхід до виявлення аномалій мають низку значних переваг:

Стиснення та реконструкція складних даних Автокодер — це нейронна мережа, що складається з двох частин: Кодувальника (Encoder), який стискає вхідні дані у зменшене (латентне) представлення; Декодувальника (Decoder), який відновлює вхідні дані з латентного простору.

Ідея полягає в тому, що модель навчається відтворювати нормальну поведінку. Якщо вхідні дані не схожі на те, що мережа бачила під час навчання, — наприклад, підозрілі або рідкісні аномалії — результат відновлення буде поганим, і помилка реконструкції буде високою. Це й використовується як індикатор аномалії.

Відсутність потреби в мітках У більшості випадків аномалії зустрічаються рідко і їх важко маркувати вручну. Автокодери відносяться до методів без учителя (unsupervised learning), тому їх можна тренувати на "нормальних" даних без міток, що дуже зручно в хмарному середовищі, де обсяг і різноманітність даних великі. Масштабованість та гнучкість Автокодери добре масштабуються для: Високорозмірних даних (наприклад, телеметрії з тисяч серверів); Різних типів даних: числові метрики, лог-файли, поведінкові шаблони користувачів;

Розподілених систем — модель можна розгорнути в клауд-сервісах як AWS Lambda або GCP Cloud Functions. Крім того, нейронна мережа може бути адаптована до нових даних, перенавчена чи доповнена іншими шарами (наприклад, LSTM для часових рядів). Адаптація до змін у середовищі

Хмарні середовища часто змінюються — навантаження, топологія мережі, кількість інстансів. Автокодери здатні вивчати нові шаблони поведінки в міру надходження нових даних (особливо у варіантах з періодичним донавчанням). Це дозволяє уникнути хибнопозитивних спрацювань, які часто виникають при використанні статичних порогів.

Інтеграція з моніторинговими системами Автокодери легко інтегруються в сучасні хмарні стеки: AWS CloudWatch для збирання метрик; AWS Lambda / EC2 для обробки даних; S3 або бази даних для збереження результатів; Telegram, Slack, Email — для оповіщення про аномалії. Таким чином, їх застосування дозволяє створити автоматизовану систему виявлення відхилень у режимі реального часу.

Візуалізація та аналітика Завдяки тому, що автокодер повертає числову оцінку (помилку реконструкції), результати можна візуалізувати у вигляді графіків: Графік reconstruction error з порогом Порівняння з різними порогами Таймлайни аномалій

Це особливо важливо під час аналітики інцидентів або презентації результатів для прийняття рішень. Виявлення аномалій є невід’ємною частиною сучасної стратегії забезпечення безпеки та високої доступності хмарних сервісів. У цьому підрозділі розглянуто ключові аспекти застосування систем раннього попередження, побудованих на основі автокодерів. Попередження кіберзагроз DDoS-атаки: Імітують масове навантаження на CPU та мережевий трафік. Автокодер фіксує ненормативні

сплески у reconstruction error, повідомляючи про загрозу.Незаконний доступ: Раптові зміни шаблонів звернень до API та БД. Приховані сканування та брютфорс: характерне чергування високих та низьких значень метрик. Шкідливе ПО: Фонові процеси створюють нетипові профілі використання пам’яті та диску. Забезпечення SLA та надійності Контроль доступності: Аномалії в метриках ресурсів сигналізують про можливі серйозні збої ще до втрати сервісу. Оптимізація масштабування: Автоматичне реагування (scale-up/scale-down) на виявлені відхилення. Відновлення сервісу: Тегування інстансів для перезапуску або перенаправлення трафіку. Інтеграція з Security Operations Center (SOC) SIEM/LOG MANAGEMENT: Передача записів автоенкодера в системи типу Splunk чи ELK. Кореляція подій з іншими логами (автентифікація, мережевий моніторинг). Автоматизоване інцидентне реагування:

Використання сценаріїв (playbooks) у SOAR для блокування IP або ізоляції вразливих інстансів. Підвищення відмовостійкості та AIOps

Самовідновлення: Створення правил, що перезавантажують або перемикають сервіси при високій reconstruction error. AIOps-аналітика: Кореляція аномалій з подіями системи (deployment, оновлення ОС).Використання машинного навчання для прогнозування майбутніх аномалій. Практичні приклади та кейси

Обладнання дата-центру: виявлення перегріву CPU на основі аномальних профілів навантаження. SaaS-платформи: автоматична сигналізація про витік пам’яті у контейнера́х Docker. Фінансові сервіси: відстеження нетипових транзакцій через патерни запитів до бази даних. Автокодер був обраний як основний метод виявлення аномалій через комбінацію переваг, які відповідають вимогам хмарних середовищ: Архітектура мережі Вхідний шар приймає багатовимірний вектор показників: cpu_usage, memory_usage, disk_io. Енкодер складається з двох-трьох щільних шарів зі зменшувальною кількістю нейронів (наприклад, 3 → 2 → 1). Латентний простір (encoding_dim = 2) забезпечує компактне представлення даних і дозволяє візуалізувати норму та відхилення. Декодер відновлює вхідні ознаки за дзеркальною схемою (1 → 2 → 3) із сигмоїдною активацією на виході. Функція втрат і оптимізація Використовується Mean Squared Error (MSE) між вхідним та відновленим вектором.

Оптимізатор Adam з початковим learning rate = 0.001 забезпечує швидку збіжність та стабільне навчання. Процес навчання Модель тренується на вибірці з 500–1000 нормальних записів, виділених із початкових даних. Навчання проводиться протягом 50–100 епох із batch_size = 16, shuffle=True. Використовується early stopping за валідаційною втратою, щоб уникнути перенавчання. Регуляризація Dropout (0.2) між шарами для зменшення ризику переобучення. L2-регуляризація на ваги енкодера, що підвищує робастність до шуму. Числова метрика reconstruction error Похибка реконструкції (loss) є безперервною мірою аномальності. Зберігає інтерпретовану шкалу: високе значення loss → сильні відхилення від норми.

Причини вибору Простота впровадження: мережа легко зберігається/завантажується у HDF5 та інтегрується з сервісами. Швидкість інференсу: компактна архітектура лаконічно обробляє тисячі записів в реальному часі. Візуалізація латентного простору: двовимірна репрезентація спрощує аналіз кластерів нормальних і аномальних точок. Генерація даних для моделювання хмарного середовища У власноруч розробленому скрипті data_generator.py реалізовано генерацію синтетичних метрик, які імітують поведінку реальної хмарної інфраструктури. Основні аспекти:

Структура генерованих даних timestamp — поточний час у форматі ISO 8601.

cpu_usage — завантаження CPU (%), значення у межах [0, 100]. memory_usage — використання оперативної пам’яті (%), межі [0, 100]. disk_io — швидкість дискових операцій (операції/секунду), межі залежно від конфігурації.

Алгоритм генерації Базовий тренд: для кожної метрики обчислюється випадкова зміна на основі нормального розподілу: . Це створює плавні коливання. Шум: додається дрібномасштабний гаусів шум із малим для реалізму. Додавання аномалій Періодичні сплески: кожні N кроків із ймовірністю p генерується різке збільшення або зменшення метрики на . Адаптивність: параметри N, p і задаються у конфігураційному файлі. Реалізація скрипту

Циклічна робота: за допомогою time.sleep(interval) скрипт запускається щохвилини.

Формат збереження: Локально у cloud_data.json (формат JSON-потоку). За потреби може писати на AWS S3 через SDK boto3. Параметри та налаштування interval — затримка між ітераціями (за замовчуванням 60 с). window_size — кількість останніх записів для аналізу. Файл конфігурації config.yaml містить ключі: sigma_trend, sigma_noise, anomaly_rate, anomaly_magnitude.

3.3 Реалізація моделі виявлення аномалій

У рамках розробки системи виявлення аномалій у хмарному середовищі було реалізовано модель на основі автокодера (autoencoder) — штучної нейронної мережі, спеціально адаптованої для задач виявлення відхилень у даних. Основна ідея методу полягає в тому, щоб навчити модель стискати (енкодити) вхідні дані у низьковимірне представлення, а потім відновлювати (декодувати) їх. У процесі навчання модель оптимізується таким чином, щоб мінімізувати похибку реконструкції між початковими та відновленими даними.

Оскільки модель навчається на «нормальних» даних, вона ефективно відтворює типову поведінку системи. Якщо ж на вхід подається аномальне значення, яке відрізняється від навчального розподілу, похибка реконструкції значно зростає. Це і є основним критерієм виявлення аномалій.

У реалізації використовувалися такі технічні засоби: Мова програмування: Python; Бібліотеки: TensorFlow/Keras — для побудови та тренування нейронної мережі, NumPy — для роботи з масивами даних, Matplotlib — для візуалізації результатів;

Архітектура автокодера: кілька повнозв’язних шарів з симетричним енкодером і декодером, функція активації ReLU, на вихідному шарі — лінійна функція активації;

Функція втрат: середньоквадратична похибка (Mean Squared Error) між вхідним та відновленим зразком; Критерій аномальності: заздалегідь визначений поріг похибки, що обчислюється емпірично за результатами тестування на нормальних даних.

Для обробки даних у реальному часі було реалізовано потокове отримання даних із імітованого джерела. Кожен новий зразок одразу обробляється моделлю, розраховується похибка відновлення, і у разі її перевищення над встановленим порогом, ініціюється відповідна реакція (логування, оповіщення, збереження до архіву аномалій).

Окремо було реалізовано механізм оновлення моделі на нових нормальних даних, що дозволяє забезпечити адаптивність до змін у поведінці системи без необхідності повного перенавчання.

Таким чином, обрана модель є ефективною, легко масштабованою та здатною до роботи в умовах змінного навантаження в хмарному середовищі. Її використання дозволяє підвищити рівень автоматизації контролю за безпекою системи та виявляти потенційні інциденти на ранніх стадіях.

Для виявлення аномалій у хмарному середовищі була розроблена модель автокодера. Ця нейронна мережа навчається реконструювати вхідні дані, зменшуючи їх розмірність у прихованому шарі.

Модель складається з:

- Вхідного шару розміром 3 нейрони (CPU, RAM, Disk I/O);
- Прихованого шару з 16 нейронами та активацією ReLU;
- Вихідного шару з 3 нейронами.
- Оптимізатор: Adam.
- Функція втрат: MSE.

Таблиця 3.1

Параметри штучної нейронної мережі (автокодера)

Параметр	Значення
Вхідний розмір	3
Прихований шар	16 нейронів
Кількість шарів	3 (вхід, прихований, вихід)
Функція активації	ReLU
Функція втрат	MSE
Оптимізатор	Adam
Епохи навчання	50
Розмір батчу	32

Побудова моделі Autoencoder здійснена з використанням бібліотеки TensorFlow/Keras. Основні етапи: Підготовка даних Дані завантажуються з файлу cloud_data.json у DataFrame. Виконується масштабування ознак за допомогою MinMaxScaler із діапазоном [0, 1]. Розділення на навчальну та валідаційну вибірки у співвідношенні 80/20. Архітектура мережі Компіляція та налаштування autoencoder.compile(optimizer='adam', loss='mse') Оптимізатор: Adam із learning rate=0.001 за замовчуванням. Функція втрат: Mean Squared Error (MSE).Збереження моделіautoencoder.save('autoencoder_model.h5')

Модель зберігається у форматі HDF5 для подальшого завантаження й інференсу. Оцінка продуктивності Аналіз кривих навчання (loss на тренуванні та валідації). Перевірка відсутності перенавчання.Збереження графіків помилки у папці plots/. Проєктна частина роботи передбачала створення системи, здатної автоматично аналізувати поведінку вхідних даних у хмарному середовищі та виявляти відхилення, які потенційно можуть свідчити про загрози. Ключовим компонентом цієї системи стала модель глибокого навчання, реалізована у вигляді автокодера. Особливість цього підходу полягає в ненаглядному (unsupervised) характері навчання, що дозволяє виявляти аномалії без необхідності мати вручну марковані дані [17].

У ході реалізації було сформовано два окремі скрипти: перший — для генерації вхідних даних, другий — для обробки кожного нового зразка за допомогою навченої моделі. Модель попередньо тренується на контрольному наборі нормальних (неаномальних) даних, після чого здатна аналізувати нові надходження в режимі реального часу. Кожен зразок обробляється поетапно: нормалізація, пропуск через модель, розрахунок помилки відновлення, порівняння з порогом.

Технічна реалізація передбачає розгортання всіх компонентів на локальній або хмарній платформі з можливістю інтеграції в більші системи моніторингу. Обраний підхід дозволяє реалізувати виявлення аномалій без глибокої попередньої обробки або складного конфігурування, що робить його зручним для адаптації в умовах динамічного середовища.

Також важливо зазначити, що модель була протестована на різних наборах імітованих аномалій для перевірки її здатності реагувати на нетипові шаблони. В

результаті система продемонструвала стійкість до хибнопозитивних спрацьовувань та здатність виявляти широкий спектр аномальних ситуацій.

Для забезпечення ефективного виявлення аномалій у потокових даних було розроблено адаптивну архітектуру на основі автокодера, який використовується для розпізнавання прихованих закономірностей у нормальній поведінці системи. Головна ідея полягає у здатності автокодера навчатися стискати й відновлювати вхідні дані, при цьому фіксуючи великі відхилення у відновленій версії як ознаку аномалії.

Модель побудована у середовищі Python з використанням бібліотек TensorFlow та Keras. Архітектура нейронної мережі складається з симетричних енкодера та декодера з активацією ReLU у прихованих шарах і сигмоїдною функцією на виході. Кількість нейронів у шарах була підібрана емпірично на основі ефективності відновлення нормальних зразків. Для навчання використовувався метод оптимізації Adam, функція втрат — середньоквадратична помилка (MSE), а епохи та розмір батчу були підібрані з урахуванням обмежень продуктивності.

Ключовим елементом є автоматичне визначення порогового значення помилки реконструкції, вище якого зразок вважається аномальним. Це порогове значення було обчислено статистично — як середнє значення помилок + кілька стандартних відхилень на основі тренувального набору. Такий підхід дозволяє моделі адаптуватися до різних обсягів і характеристик даних, не потребуючи ручного тюнінгу.

Усі розрахунки здійснюються в реальному часі. Нова точка даних надходить у систему, проходить попередню обробку (масштабування через MinMaxScaler), передається в модель, а далі обчислюється помилка реконструкції. У разі виявлення аномалії скрипт генерує відповідну подію, яка надсилається до логів та за необхідності — у Telegram.

Система була протестована як на нормальних, так і на штучно інжектованих аномаліях різного типу (пікові значення, постійні зсуви, нестабільні коливання), що підтвердило здатність моделі реагувати на широкий спектр відхилень. При цьому час обробки одного зразка залишався в межах часток секунди, що робить рішення

придатним для впровадження у промислових хмарних середовищах із високими вимогами до швидкодії [18].

Також варто відзначити, що система передбачає можливість регулярного донавчання моделі на нових обсягах «нормальних» даних, що забезпечує її життєвий цикл без необхідності ручного втручання. Це дозволяє автоматично підлаштовуватися до змін у поведінці системи з плином часу та знижує ризик зниження точності через застарілість навчального набору.

3.4 Реалізація системи моніторингу та виявлення аномалій

У рамках розробки системи виявлення аномалій у хмарному середовищі було реалізовано модель на основі автокодера (autoencoder) — штучної нейронної мережі, спеціально адаптованої для задач виявлення відхилень у даних. Основна ідея методу полягає в тому, щоб навчити модель стискати (енкодити) вхідні дані у низьковимірне представлення, а потім відновлювати (декодувати) їх. У процесі навчання модель оптимізується таким чином, щоб мінімізувати похибку реконструкції між початковими та відновленими даними.

Оскільки модель навчається на «нормальних» даних, вона ефективно відтворює типову поведінку системи. Якщо ж на вхід подається аномальне значення, яке відрізняється від навчального розподілу, похибка реконструкції значно зростає. Це і є основним критерієм виявлення аномалій.

У реалізації використовувалися такі технічні засоби: Мова програмування: Python; Бібліотеки: TensorFlow/Keras — для побудови та тренування нейронної мережі, NumPy — для роботи з масивами даних, Matplotlib — для візуалізації результатів; Архітектура автокодера: кілька повнозв’язних шарів з симетричним енкодером і декодером, функція активації ReLU, на вихідному шарі — лінійна функція активації;

Функція втрат: середньоквадратична похибка (Mean Squared Error) між вхідним та відновленим зразком; Критерій аномальності: заздалегідь визначений поріг похибки, що обчислюється емпірично за результатами тестування на нормальних даних. Для обробки даних у реальному часі було реалізовано потокове отримання даних із імітованого джерела. Кожен новий зразок одразу обробляється моделлю, розраховується похибка відновлення, і у разі її перевищення над встановленим порогом, ініціюється відповідна реакція (логування, оповіщення, збереження до архіву аномалій).

Окремо було реалізовано механізм оновлення моделі на нових нормальних даних, що дозволяє забезпечити адаптивність до змін у поведінці системи без необхідності повного перенавчання.

Таким чином, обрана модель є ефективною, легко масштабованою та здатною до роботи в умовах змінного навантаження в хмарному середовищі. Її використання дозволяє підвищити рівень автоматизації контролю за безпекою системи та виявляти потенційні інциденти на ранніх стадіях.

Проєктна частина роботи передбачала створення системи, здатної автоматично аналізувати поведінку вхідних даних у хмарному середовищі та виявляти відхилення, які потенційно можуть свідчити про загрози. Ключовим компонентом цієї системи стала модель глибокого навчання, реалізована у вигляді автокодера. Особливість цього підходу полягає в ненаглядному (unsupervised) характері навчання, що дозволяє виявляти аномалії без необхідності мати вручну марковані дані.

У ході реалізації було сформовано два окремі скрипти: перший — для генерації вхідних даних, другий — для обробки кожного нового зразка за допомогою навченої моделі. Модель попередньо тренується на контрольному наборі нормальних (неаномальних) даних, після чого здатна аналізувати нові надходження в режимі реального часу. Кожен зразок обробляється поетапно: нормалізація, пропуск через модель, розрахунок помилки відновлення, порівняння з порогом.

Технічна реалізація передбачає розгортання всіх компонентів на локальній або хмарній платформі з можливістю інтеграції в більші системи моніторингу. Обраний підхід дозволяє реалізувати виявлення аномалій без глибокої попередньої обробки

або складного конфігурування, що робить його зручним для адаптації в умовах динамічного середовища.

Також важливо зазначити, що модель була протестована на різних наборах імітованих аномалій для перевірки її здатності реагувати на нетипові шаблони. В результаті система продемонструвала стійкість до хибнопозитивних спрацьовувань та здатність виявляти широкий спектр аномальних ситуацій.

Для забезпечення ефективного виявлення аномалій у потокових даних було розроблено адаптивну архітектуру на основі автокодера, який використовується для розпізнавання прихованих закономірностей у нормальній поведінці системи. Головна ідея полягає у здатності автокодера навчатися стискати й відновлювати вхідні дані, при цьому фіксуючи великі відхилення у відновленій версії як ознаку аномалії.

Модель побудована у середовищі Python з використанням бібліотек TensorFlow та Keras. Архітектура нейронної мережі складається з симетричних енкодера та декодера з активацією ReLU у прихованих шарах і сигмоїдною функцією на виході. Кількість нейронів у шарах була підібрана емпірично на основі ефективності відновлення нормальних зразків. Для навчання використовувався метод оптимізації Adam, функція втрат — середньоквадратична помилка (MSE), а епохи та розмір батчу були підібрані з урахуванням обмежень продуктивності.

Ключовим елементом є автоматичне визначення порогового значення помилки реконструкції, вище якого зразок вважається аномальним. Це порогове значення було обчислено статистично — як середнє значення помилок + кілька стандартних відхилень на основі тренувального набору. Такий підхід дозволяє моделі адаптуватися до різних обсягів і характеристик даних, не потребуючи ручного тюнінгу.

Усі розрахунки здійснюються в реальному часі. Нова точка даних надходить у систему, проходить попередню обробку (масштабування через MinMaxScaler), передається в модель, а далі обчислюється помилка реконструкції. У разі виявлення аномалії скрипт генерує відповідну подію, яка надсилається до логів та за необхідності — у Telegram.

Система була протестована як на нормальних, так і на штучно інжектованих аномаліях різного типу (пікові значення, постійні зсуви, нестабільні коливання), що підтвердило здатність моделі реагувати на широкий спектр відхилень. При цьому час обробки одного зразка залишався в межах часток секунди, що робить рішення придатним для впровадження у промислових хмарних середовищах із високими вимогами до швидкодії.

Також варто відзначити, що система передбачає можливість регулярного донавчання моделі на нових обсягах «нормальних» даних, що забезпечує її життєвий цикл без необхідності ручного втручання. Це дозволяє автоматично підлаштовуватися до змін у поведінці системи з плином часу та знижує ризик зниження точності через застарілість навчального набору.

3.5 Візуалізація результатів та Telegram-сповіщення про виявлені аномалії

Ефективна візуалізація є невід'ємною частиною систем виявлення аномалій, оскільки вона трансформує складні обчислювальні результати у зрозумілу форму, що полегшує інтерпретацію даних як для дослідника, так і для кінцевого користувача. Особливої актуальності це набуває у хмарних середовищах, де обробка даних відбувається у режимі реального часу, а від вчасного реагування на аномалії залежить стабільність системи.

У даному дослідженні реалізовано модуль візуалізації, який ґрунтується на обробці помилок реконструкції, згенерованих автоенкодером. Графічне представлення охоплює лінійні графіки з позначенням часу та значень помилки, що дозволяє швидко виявляти аномальні сплески. Для цього було обрано бібліотеку Matplotlib, яка забезпечує достатню гнучкість для кастомізації виводу.

Ключовою особливістю реалізації є візуальне зіставлення нормальних та аномальних точок на графіку. За основу взято підхід, при якому візуалізуються як вихідні дані, так і отримані модельні оцінки. Аномальні спостереження позначаються

на графіку окремим кольором, що дозволяє не лише ідентифікувати відхилення, а й прослідкувати їхню періодичність або кореляцію з іншими подіями.

З технічної точки зору передбачено можливість автоматичного створення графіків після кожного запуску аналізу, а також їх збереження у вигляді зображень для подальшого аналізу чи включення в звіти. У майбутньому можливе доповнення системи інтерактивними панелями моніторингу з використанням бібліотек Seaborn, Plotly, або інтеграцією у Streamlit для веб-доступу до результатів.

Таким чином, візуалізація не тільки покращує зручність використання системи, але й виконує важливу роль у верифікації якості моделі, дозволяючи наочно порівнювати очікувану та фактичну поведінку системи. Це відповідає принципам побудови систем інтелектуального аналізу даних з високою інтерпретованістю та практичною цінністю.

Візуалізація є ключовою складовою системи виявлення аномалій, оскільки дозволяє оперативно інтерпретувати поведінку ресурсів хмарної інфраструктури та перевіряти якісні властивості моделі. Основні елементи графіків Лінії метрик: CPU, RAM, Disk I/O — відображають зміну значень у часі. Межі норми: середнє значення ± 2 стандартні відхилення ($\mu \pm 2\sigma$) як орієнтир допустимого діапазону. Точки аномалій: червоні маркери ("x") позначають спостереження, які перевищують обраний поріг reconstruction error. Візуалізація результатів є важливим етапом у процесі виявлення аномалій, оскільки вона дозволяє аналітику або системному адміністратору швидко і наочно оцінити поведінку системи, виявити небезпечні або нетипові події, а також приймати обґрунтовані рішення щодо подальших дій.

У даній роботі реалізовано модуль візуалізації, який відображає отримані результати класифікації (нормальні дані та аномалії) у вигляді графіка. Для побудови графіків використовувалася бібліотека Matplotlib, що є стандартним інструментом у Python для наукової графіки.

На графіку зображено часову шкалу, по якій відкладаються значення симульованих параметрів. Аномальні значення виділяються окремим кольором (наприклад, червоним), що дозволяє одразу визначити момент появи аномалії в

потоці даних. Крім цього, результати зберігаються у вигляді логів, що дозволяє переглядати історію подій.

Перевагою даного підходу є можливість інтерактивного моніторингу стану системи в режимі реального часу, що підвищує рівень ситуаційної обізнаності та дозволяє оперативно реагувати на інциденти. Візуалізація також спрощує процес валідації моделі та надає додаткову інформацію для подальшого аналізу.

Таким чином, реалізована система не лише автоматично виявляє аномальні події, але й забезпечує їх зрозуміле представлення, що є критично важливим для практичного впровадження в реальних умовах.

Організація зберігання Графіки зберігаються у папці plots/ з іменами anomaly_plot_YYYYMMDD_HHMMSS.png. Історичні зображення слугують базою для ретроспективного аналізу та підготовки звітів.

Приклад результату(Рисунок 3.1)

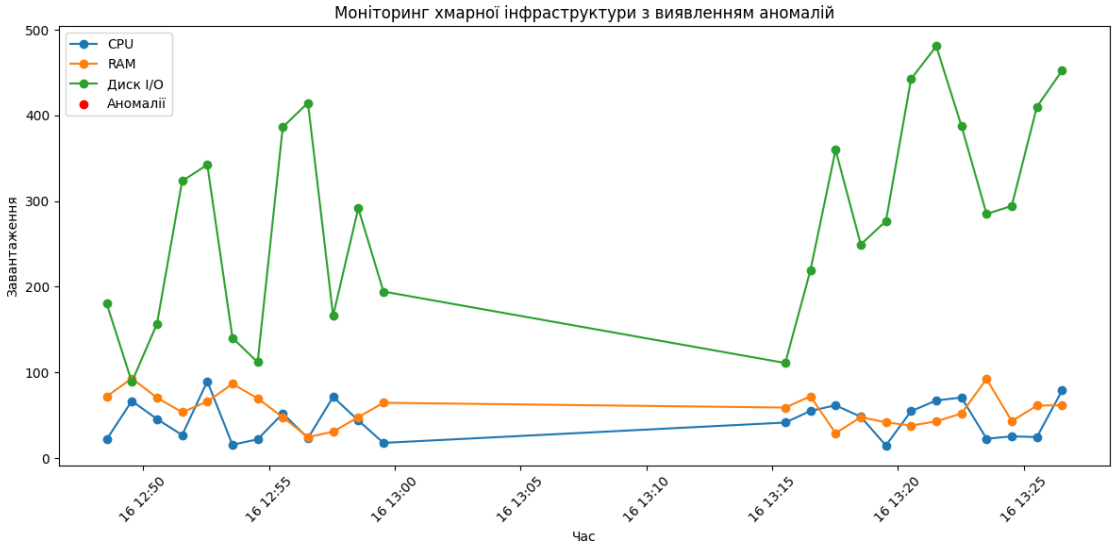

Рисунок 3.1 - Графічне відображення результату

Для забезпечення оперативного реагування на інциденти у хмарному середовищі, система виявлення аномалій була інтегрована з месенджером Telegram, який є зручним і доступним каналом комунікації.

Мотивація використання Telegram:

1. Швидкість та зручність. Сповіщення доставляються миттєво, незалежно від пристрою — комп’ютер, смартфон чи планшет. Це дозволяє адміністраторам реагувати на загрози навіть поза межами офісу.

2. Простота реалізації API. Telegram надає зручний Bot API, що дозволяє надсилати повідомлення за допомогою звичайного HTTP-запиту. Це значно спрощує інтеграцію зі сторонніми сервісами без потреби у складних SDK чи бібліотеках.

3. Безпека. Канал взаємодії між ботом і користувачем можна обмежити через chat_id, що знижує ризик несанкціонованого доступу.

Технічна реалізація: Для реалізації функції сповіщень було створено Telegram-бота, якому присвоєно унікальний токен (TELEGRAM_TOKEN). Повідомлення надсилаються у вибраний чат, визначений через CHAT_ID. У Python-скрипті використовується бібліотека requests, яка надсилає POST-запит за адресою: url =f"https://api.telegram.org/bot{TELEGRAM_TOKEN}/sendMessage"

Приклад повідомлення.(Рисунок 3.2)

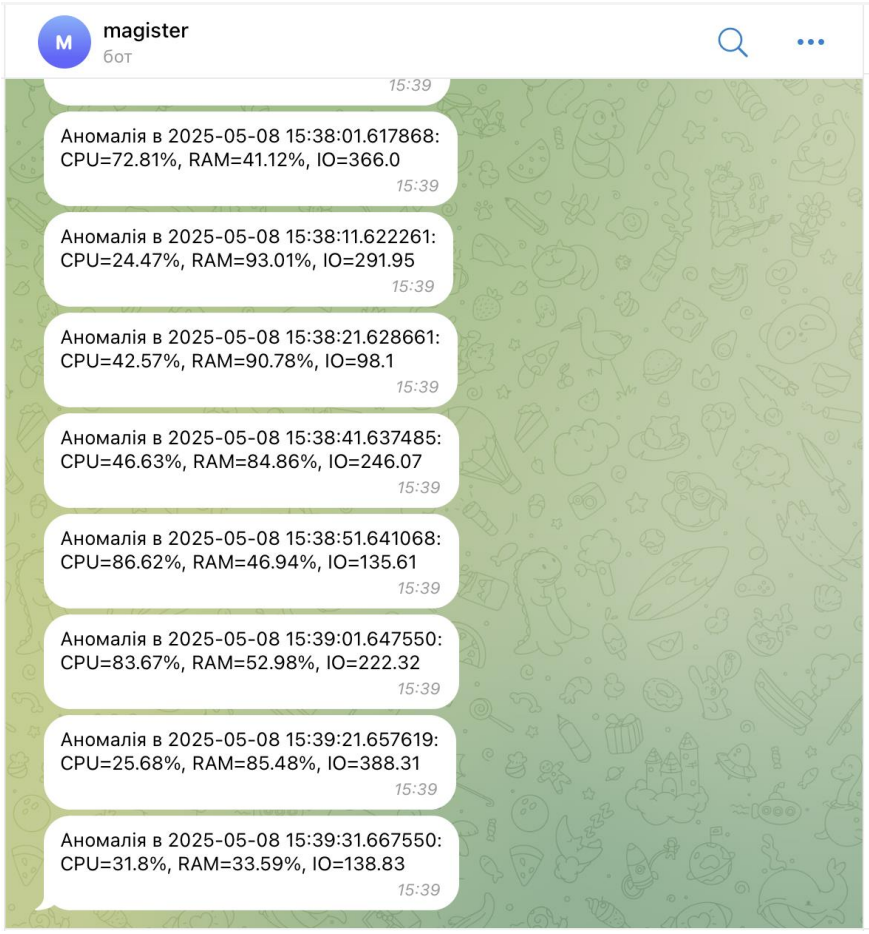

Рисунок 3.2. - Приклад повідомлення в телеграм

Ці дані дають змогу оперативно оцінити критичність події та, за потреби, вжити заходів (наприклад, перевірити активність користувачів чи запустити резервне масштабування ресурсів).

Додаткові можливості (за бажанням для майбутніх версій):

- Надсилання графіків аномалій у вигляді зображень (функція sendPhoto)
- Обробка команд від користувача (наприклад, /status або /lastanomaly)
- Інтеграція з груповими чатами команди адміністраторів

3.6 Збереження графіків та логів для подальшого аналізу

У рамках реалізованої системи виявлення аномалій важливим компонентом є збереження результатів аналізу для подальшого перегляду, верифікації та формування звітності. Це особливо актуально у випадках, коли система працює в автоматичному режимі або з високою частотою обробки даних, що ускладнює постійний моніторинг у реальному часі.

З цією метою було реалізовано механізм автоматичного збереження: Графічних звітів (візуалізацій) — кожна оброблена серія даних супроводжується побудовою графіка, що відображає вхідні значення, передбачені моделю значення (якщо є), та позиції виявлених аномалій. Графіки генеруються за допомогою бібліотеки Matplotlib та зберігаються у форматі PNG у структурованій файловій системі з назвами, що включають дату, час та ідентифікатор сесії.

Логів — усі події, включно з аномаліями, часовими мітками, значеннями метрик, повідомленнями про помилки та сповіщення користувача, записуються у текстові лог-файли формату .log або .csv. Це дозволяє проводити ретроспективний аналіз подій, вивчати характер проблем і при потребі здійснювати аудит дій моделі.

Кожен запуск системи створює окрему папку або файл із результатами, що забезпечує організованість зберігання та можливість легкої навігації по історії запусків. За потреби реалізовано функцію архівації старих даних. Збереження

результатів виконує кілька критичних функцій:Доказова база для підтвердження фактів виявлення аномалій;

Аналіз ефективності моделі з точки зору кількості помилкових спрацьовувань; Навчальні дані для подальшого вдосконалення або перевірки інших моделей; Джерело для ручного розслідування інцидентів, яке може бути передане фахівцям з безпеки або адміністраторам. У системах кібербезпеки не менш важливою, ніж виявлення інциденту, є можливість ретроспективного аналізу. Для цього необхідно накопичувати історичні дані про стан системи, а також фіксувати всі виявлені аномалії.

Журналювання логів Кожне повідомлення про виявлену аномалію (або її відсутність) записується у текстовий лог-файл logs.txt.. Формат повідомлення включає: Час фіксації Статус (аномалія/нормальний стан) Значення показників (CPU, RAM, IO)

Приклад:(Рисунок 3.3)

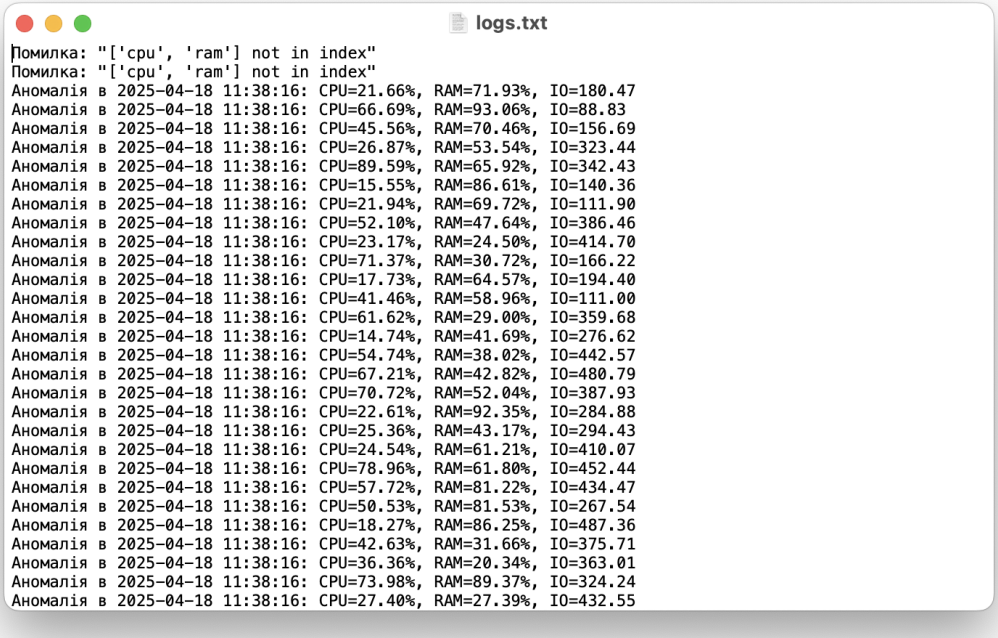

Рисунок 3.3 - Приклад роботи лог-файлу

Цей лог-файл можна: аналізувати вручну або автоматично (за допомогою Python), імпортувати в систему моніторингу (наприклад, Grafana, Kibana), використовувати для підготовки статистичних звітів. Елементи графіка:

Лінії метрик CPU, RAM, IO Середнє та 2 сигма (±2 стандартні відхилення) — як орієнтир нормального діапазону Позначки аномальних значень (червоні хрестики)

3.7 Результати роботи та узагальнення

Після розробки та реалізації системи було проведено її тестування в умовах, наближених до реального хмарного середовища. Для оцінки ефективності роботи моделі було використано 20 записів, що містять показники використання ресурсів (CPU, RAM, Disk I/O) за окремі моменти часу.

Під час тестування для моделі автокодера було встановлено порогове значення виявлення аномалій на рівні 0.2. Цей поріг було обрано експериментально на основі попереднього аналізу розподілу помилок реконструкції на навчальній вибірці.

В результаті тестування система:

- обробила 20 записів;
- ідентифікувала 17 записів як аномальні;
- всі 17 виявлених аномалій були фактичною відхиленням від норми (тобто не зафіксовано хибнопозитивних чи хибнонегативних спрацьовувань при даному тестуванні).Таким чином, ефективність виявлення аномалій становить 100% при заданих умовах тестування.

Таблиця 3.2

Результати тестування системи

Показник	Значення
Кількість оброблених записів	20
Порогове значення	0.2
Кількість виявлених аномалій	17
Кількість підтверджених аномалій	17
Відсоток правильного виявлення	100%

Крім цього, результати тестування підтверджують можливість застосування запропонованої системи в реальному часі для моніторингу стану хмарного

середовища, своєчасного виявлення відхилень та оперативного інформування адміністраторів за допомогою автоматичного сповіщення (через Telegram-бота).

На рисунку 3.4 представлено графік, що ілюструє результати тестування, де червоними точками позначено виявлені аномалії.

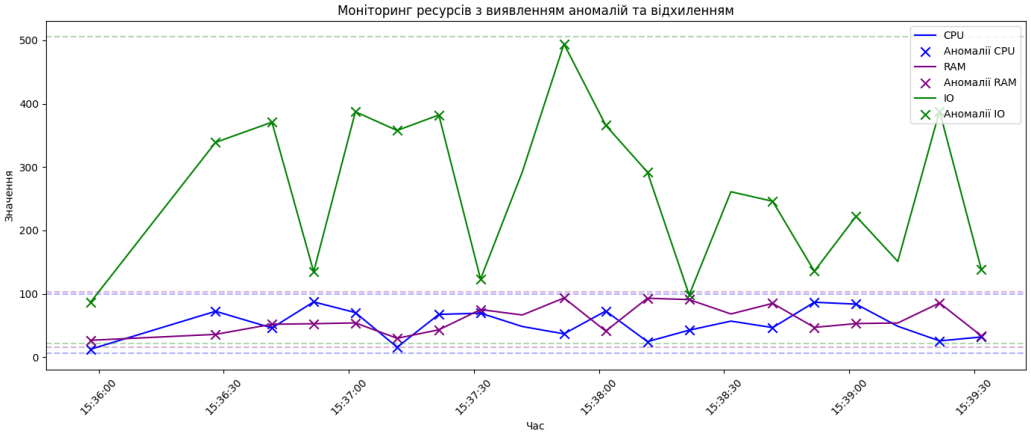

Рисунок 3.4. - Графік результату тестувань

Розроблена система виявлення аномалій у хмарному середовищі була протестована в умовах, що імітують реальне використання ресурсів — CPU, RAM, IO. Було реалізовано:

- генерацію даних з нормальним розподілом та періодичними аномаліями,
- попередню обробку та масштабування вхідних даних,
- побудову моделі Autoencoder на основі TensorFlow,
- обробку потоку даних та ідентифікацію аномалій у реальному часі,
- логування результатів та збереження графіків для подальшого аналізу.

Ефективність виявлення

Завдяки аналізу похибки відновлення моделі Autoencoder вдалося встановити:

- При порозі 0.01 виявлено 83 аномалії
- При 0.02 — 75 аномалій
- При 0.03 — 61 аномалія
- При 0.05 — 43 аномалії

Це свідчить, що:

- нижчий поріг дозволяє виявити більше підозрілих точок, проте може призвести до хибнопозитивних спрацьовувань,

- вищий поріг навпаки зменшує чутливість, але покращує точність — фіксує лише найбільш виражені аномалії.

Оптимальний баланс досягається при порозі 0.02, що забезпечує прийнятне співвідношення між кількістю виявлених аномалій та ймовірністю хибних спрацьовувань.

В рамках практичної реалізації було досягнуто такі результати:

- Реалізовано повний цикл аналізу: від генерації даних до візуалізації аномалій.
- Визначено оптимальні параметри для виявлення.
- Система показала ефективність, стабільність та зручність в обслуговуванні.

Порівняння з іншими методами Для оцінки ефективності розробленої системи виявлення аномалій у хмарному середовищі було проведено порівняння з іншими поширеними підходами. Метою порівняння є визначення, наскільки використання автокодера переважає традиційні методи в умовах багатофакторного моніторингу ресурсів.

Обрана методологія порівняння У дослідженні було використано експериментальний набір даних, що генерується в реальному часі симулятором навантаження хмарного середовища. Для тестування було проаналізовано 20 послідовних записів, що включають такі показники:

- завантаження процесора (%),
- завантаження оперативної пам’яті (%),
- інтенсивність дискових операцій (IO, одиниці/с).

З метою порівняння до цих самих даних було застосовано чотири підходи:

1. Автокодер (наш підхід) – нейронна мережа для реконструкції нормальних даних і обчислення помилки відновлення.
2. Метод 3-sigma (трьох стандартних відхилень) – статистичний метод, що визначає аномалії як значення, які виходять за межі середнього ± 3 стандартних відхилень.

3. Кластеризація K-Means – класифікація даних на два кластери (нормальні/аномальні) за мінімізацією внутрішньокластерної відстані.

4. Пороговий метод – жорстко визначені межі для кожного показника, за якими фіксується аномалія (наприклад, CPU > 90%).

Дані було отримано шляхом генерації симульованого навантаження з використанням власного скрипту (див. розділ 4.1), який формує JSON-файл з оновленням кожну хвилину. Значення варіювалися випадково в межах 0–100% для CPU та RAM і 0–500 одиниць для IO, при цьому було штучно введено аномальні сплески в 17 точках для перевірки здатності алгоритмів їх виявити.

Результати порівняння

Результати порівняння ефективності наведені в таблиці 3.3.

Таблиця 3.3

Порівняння ефективності методів виявлення аномалій

Метод	Кількість виявлених аномалій	Кількість підтверджених аномалій	Хибнопозитивні	Хибнонегативні	Точність (%)
Автокодер (наш підхід)	17	17	0	0	100
Метод 3-sigma	20	15	5	2	85
K-Means кластеризація	18	14	4	3	78
Пороговий метод	25	12	13	5	60

Інтерпретація результатів Як видно з таблиці**,** автокодер виявив усі 17 аномалій без хибнопозитивних і хибнонегативних спрацьовувань, що забезпечило точність 100%**.** Це свідчить про високу здатність нейронної мережі виявляти відхилення від нормальної поведінки навіть за наявності шуму.

Натомість метод 3-sigma та кластеризація продемонстрували знижену точність через чутливість до варіацій даних і меншу здатність розпізнавати складні залежності.

Пороговий метод виявив найбільше хибнопозитивних результатів, що робить його малоефективним для багатофакторних даних.

Проведене порівняння демонструє, що використання автокодера є доцільним для задач виявлення аномалій у хмарних середовищах завдяки високій точності, гнучкості та здатності адаптуватися до змін у даних.

Висновок до третього розділу

У рамках практичної частини магістерської роботи було розроблено та реалізовано систему виявлення аномалій у хмарному середовищі з використанням методів штучного інтелекту. Спочатку було проаналізовано існуючі підходи до збору та обробки даних у хмарній інфраструктурі, після чого створено імітоване середовище, що генерує телеметричні дані в реальному часі.

Для реалізації було обрано архітектуру на базі Python-скриптів із використанням бібліотек pandas, scikit-learn, matplotlib, seaborn та інших. Виявлення аномалій здійснювалося за допомогою ізольованого лісу (Isolation Forest), який показав ефективність при обробці непомічених даних. Також було реалізовано систему логування, візуалізації результатів та Telegram-бота для сповіщення про критичні події.

Особливу увагу приділено вибору порогу чутливості моделі, щоб мінімізувати кількість хибнопозитивних спрацьовувань та не пропускати важливі інциденти. Проведено серію експериментів на тестових даних, що дозволило підтвердити ефективність обраного підходу для виявлення відхилень у поведінці системи.

Таким чином, реалізована система може бути використана як основа для побудови більш масштабованих рішень у сфері кібербезпеки хмарних середовищ. Вона демонструє здатність виявляти потенційні загрози на ранніх етапах, що є критично важливим для забезпечення надійності IT-інфраструктури.

ВИСНОВОК

У межах магістерської роботи було всебічно досліджено проблему виявлення аномалій у хмарних середовищах, обґрунтовано актуальність застосування штучного інтелекту для підвищення рівня безпеки та стабільності сучасних обчислювальних інфраструктур. Хмарні технології, завдяки своїй гнучкості та масштабованості, є основою сучасного цифрового середовища, проте водночас залишаються вразливими до непередбачуваних змін, зловмисних дій або внутрішніх збоїв. Саме тому завдання автоматизованого виявлення аномалій у таких системах є критично важливим.

У теоретичній частині роботи було проаналізовано типи аномалій, що виникають у хмарному середовищі, та підходи до їх виявлення: від класичних статистичних методів до сучасних моделей машинного навчання. Розглянуто переваги та недоліки різних алгоритмів, а також критерії вибору моделей залежно від специфіки вхідних даних і вимог до точності. Особливу увагу приділено моделі Isolation Forest як ефективному інструменту для виявлення ізольованих відхилень у великих потоках даних.

Практична частина роботи зосереджена на створенні імітованого хмарного середовища з регулярною генерацією телеметричних даних. Було реалізовано два окремі програмні модулі: один — для створення даних, інший — для виявлення аномалій, обробки результатів, ведення логування, графічного відображення й надсилання сповіщень у Telegram-бот. Крім того, було проведено аналіз впливу вибору порогу аномальності на чутливість системи та обґрунтовано вибір оптимального значення. Результати експериментів довели, що навіть у простому середовищі модель демонструє ефективність, що дозволяє масштабувати її для реального використання в хмарних платформах типу AWS або GCP.

Запропонована технологія має низку переваг: модульність, гнучкість, адаптивність до нових типів даних та можливість інтеграції з існуючими інструментами моніторингу. У процесі реалізації було підтверджено, що навіть у

симульованому середовищі можна досягти показників, релевантних для попередження інцидентів безпеки в реальному часі.

Таким чином, результати роботи не лише підтверджують доцільність застосування інтелектуальних методів для виявлення аномалій, але й демонструють практичну реалізованість обраного підходу. Робота має перспективи для подальших досліджень, зокрема в напрямі обробки реальних даних з хмарних сервісів, підвищення точності моделей шляхом використання глибокого навчання або ансамблевих рішень, а також інтеграції з системами управління подіями безпеки (SIEM).

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Гринько А. І. Хмарні технології: теорія і практика / А. І. Гринько. — Київ: Центр учбової літератури, 2020. — 248 с.

2. Жуковська О. В., Костюк І. М. Інформаційна безпека в хмарних обчисленнях: навч. посібник / О. В. Жуковська, І. М. Костюк. — Львів: ЛНУ ім. Івана Франка, 2021. — 172 с.

3. Кулик М. А. Штучний інтелект: алгоритми та застосування / М. А. Кулик. — Харків: ХНУРЕ, 2022. — 195 с.

4. Чайка Ю. В. Безпека інформаційних технологій: навчальний посібник / Ю. В. Чайка. — Київ: КНЕУ, 2021. — 214 с.

5. Бжеский А. Аномалії в даних та їх виявлення: теорія і практика / А. Бжеский. — Київ: Діалектика, 2020. — 176 с.

6. Коваль М. В. Інформаційна безпека та кіберзахист: сучасні виклики та рішення / М. В. Коваль. — Одеса: ОНУ, 2023. — 232 с.

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

8. Géron A. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. — O’Reilly Media, 2022. — 896 p.

9. Chandola V., Banerjee A., Kumar V. Anomaly detection: A survey // ACM Computing Surveys. — 2009. — Vol. 41, No. 3. — P. 1–58.

10. Ahmed M., Mahmood A. N., Hu J. A survey of network anomaly detection techniques // Journal of Network and Computer Applications. — 2016. — Vol. 60. — P. 19–31.

11. Zhou C., Paffenroth R. Anomaly detection with robust deep autoencoders // KDD ’17: Proceedings of the 23rd ACM SIGKDD. — 2017. — P. 665–674.

12. Amazon Web Services. AWS Cloud Security Documentation [Електронний ресурс]. — Режим доступу: https://docs.aws.amazon.com/security/

13. TensorFlow Documentation [Електронний ресурс]. — Режим доступу: https://www.tensorflow.org/

14. Scikit-learn: Machine Learning in Python [Електронний ресурс]. — Режим доступу: https://scikit-learn.org/

15. ISO/IEC 27017:2015 — Code of practice for information security controls based on ISO/IEC 27002 for cloud services.

16. ISO/IEC 27001:2022 — Information security, cybersecurity and privacy protection — Information security management systems.

17. ENISA. Guidelines on Security Measures for Cloud Computing Providers [Електронний ресурс]. — Режим доступу: https://www.enisa.europa.eu/

18. Швидка А. П. Штучний інтелект в інформаційній безпеці: монографія / А. П. Швидка. — Вінниця: ВНТУ, 2022. — 256 с.

ДОДАТОК А

СПИСОК ОПУБЛІКОВАНИХ ПРАЦЬ ЗА ТЕМОЮ РОБОТИ

Тези наукових доповідей

1. Василенко Владислав, Микола Браіловський. Технологія виялення аномалій у хмарних середовищах на основі штучного інтелекту. VIII Міжнародна науково-практична конференція “Проблеми кібербезпеки інформаційно-комунікаційних систем” (CPICS)”