НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Факультет інформаційних технологій

ПОГОДЖЕНО
Декан факультету (Директор ННІ)
інформаційних технологій
(назва факультету (ННІ))

_____ Ігор Болбот
(підпис) (ім'я ПРІЗВИЩЕ)

"___" _____ 2025 р.

ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ
Завідувач кафедри
комп'ютерних наук
(назва кафедри)

_____ Белла Голуб
(підпис) (ім'я ПРІЗВИЩЕ)

"___" _____ 2025 р.

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему Програмне забезпечення для підвищення захисту інформаційних систем

Спеціальність 121 «Інженерія програмного забезпечення»
(код і найменування)

Освітня програма Програмне забезпечення інформаційних систем
(назва)

Орієнтація освітньої програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Гарант освітньої програми
к.ф.-м.н., доцент _____ Віктор Кириченко
(науковий ступінь та вчене звання) (підпис) (ім'я ПРІЗВИЩЕ)

Керівник магістерської кваліфікаційної роботи
к.ф.-м.н., доцент _____ Володимир СЕМКО
(науковий ступінь та вчене звання) (підпис) (ім'я ПРІЗВИЩЕ)

Виконав _____ Ігор ГУМЕНЮК
(підпис) (ім'я ПРІЗВИЩЕ здобувача)

КИЇВ – 2025

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Факультет інформаційних технологій

ЗАТВЕРДЖУЮ
Завідувач кафедри комп'ютерних наук
к.т.н., доцент ____ Белла Голуб
(науковий ступінь, вчене звання) (підпис) (ім'я ПРІЗВИЩЕ)
01" листопада 2024 року

З А В Д А Н Н Я

ДО ВИКОНАННЯ МАГІСТЕРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ ЗДОБУВАЧУ

Гуменюк Ігор Олександрович
(прізвище, ім'я, по батькові)

Спеціальність 121 «Інженерія програмного забезпечення»
(код і найменування)

Освітня програма Програмне забезпечення інформаційних систем
(назва)

Орієнтація освітньої програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Тема магістерської кваліфікаційної роботи Програмне забепечення для підвищення захисту інформаційних систем

затверджена наказом від "01" листопада 2024р. №1963 «С»

Термін подання завершеної роботи на кафедру 20.11.2025
(рік, місяць, число)

Вихідні дані до магістерської кваліфікаційної роботи Інформаційні журнали подій, телеметричні дані користувацьких сесій, специфікації протоколів OIDC/MFA, вимоги до інтеграції з платформами SIEM/SOAR, технічні характеристики серверного середовища та програмних компонентів системи кіберзахисту.

Перелік питань, що підлягають дослідженню:
Визначення ефективності алгоритмів машинного навчання для виявлення аномалій у потоковій телеметрії та оцінювання їх точності, стійкості та інтерпретованості. Дослідження архітектури, методів інтеграції та механізмів реагування експертної системи у контексті підвищення рівня захисту корпоративних інформаційних ресурсів.Перелік графічного матеріалу (за потреби)

Дата видачі завдання "01" листопада 2024 р.

Керівник магістерської кваліфікаційної роботи _____ Володимир СЕМКО
(підпис) (ім'я ПРІЗВИЩЕ)

Завдання прийняв до виконання _____ Ігор ГУМЕНЮК
(підпис) (ім'я ПРІЗВИЩЕ)

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

1. API — Application Programming Interface, програмний інтерфейс взаємодії.
2. ABAC — Attribute-Based Access Control, атрибутивна модель контролю доступу.
3. RBAC — Role-Based Access Control, рольова модель контролю доступу.
4. ML — Machine Learning, машинне навчання.
5. OCSVM — One-Class Support Vector Machine, однокласовий метод опорних векторів для виявлення аномалій.
6. SHAP — SHapley Additive exPlanations, метод пояснення рішень ML-моделей.
7. SIEM — Security Information and Event Management, система керування журналами та подіями безпеки.
8. SOAR — Security Orchestration, Automation and Response, платформа автоматизації та реагування на інциденти.
9. IdP — Identity Provider, сервіс автентифікації користувачів.
10. OIDC — OpenID Connect, протокол автентифікації поверх OAuth 2.0.
11. MFA — Multi-Factor Authentication, багатофакторна автентифікація.
12. TLS — Transport Layer Security, протокол захисту передавання даних у мережі.
13. HTTPS — захищений протокол передавання гіпертексту.
14. gRPC — протокол віддалених викликів процедур з підтримкою потокової взаємодії.
15. REST — Representational State Transfer, архітектурний стиль веб-сервісів.
16. CPU — Central Processing Unit, центральний процесор.

17. Docker — програмна платформа контейнеризації застосунків.

18. SQL — Structured Query Language, мова структурованих запитів до БД.

19. SLA — Service Level Agreement, договірний показник якості сервісу.

20. K-means — метод кластеризації на основі середніх значень.

21. OLAP — Online Analytical Processing, технологія аналітичної обробки даних.

22. Telemetrія — набір параметрів та подій, що надходять від систем/пристроїв у реальному часі.

ВСТУП

Традиційні засоби безпеки, орієнтовані на сигнатурне виявлення, не завжди здатні своєчасно реагувати на нові загрози, що призводить до витоку конфіденційної інформації, порушення цілісності даних і компрометації критичних ресурсів [1]. У цих умовах актуальним є створення програмного забезпечення, яке забезпечує інтелектуальне виявлення аномалій, автоматизований аналіз подій і адаптивне реагування на загрози в реальному часі, що підвищує загальний рівень кіберстійкості інформаційних систем [2].

Метою дослідження є розроблення програмного забезпечення для підвищення захисту інформаційних систем на основі інтеграції механізмів моніторингу, аналізу ризиків і керування інцидентами із застосуванням методів машинного навчання та інтелектуальної обробки даних.

Для досягнення цієї мети передбачено виконання комплексу **завдань**:

– провести аналіз предметної області інформаційної безпеки та існуючих програмних рішень;

– сформулювати вимоги до системи підвищення захисту інформаційних систем;

– спроєктувати архітектуру та моделі даних програмного комплексу;

– реалізувати модулі моніторингу, виявлення аномалій і керування інцидентами;

– оцінити ефективність запропонованого підходу на основі експериментальних даних.

Об’єктом дослідження виступає процес забезпечення захисту інформаційних систем у корпоративних і розподілених середовищах, де високі вимоги до безперервності функціонування поєднуються з потребою у швидкому реагуванні на загрози.

Предметом дослідження є методи, алгоритми та архітектурні рішення програмного забезпечення, орієнтовані на підвищення рівня безпеки

інформаційних систем шляхом динамічного виявлення аномалій і запобігання інцидентам.

Для реалізації поставлених завдань застосовуються **методи** системного аналізу, теорії інформаційної безпеки, машинного навчання, інтелектуального аналізу даних (Data Mining), статистичної фільтрації та кореляційного аналізу потоків подій. Проєктування здійснюється з використанням UML-діаграм, CASE-засобів і технологій розроблення на мовах Python та Java з опорою на архітектурні принципи мікросервісності та асинхронної обробки даних.

Наукова новизна роботи полягає у формуванні цілісної концепції побудови інтелектуального програмного забезпечення для забезпечення інформаційної безпеки, що поєднує алгоритми машинного навчання, моделі поведінкового аналізу та адаптивні механізми реагування на інциденти. Запропонований підхід дозволяє підвищити точність виявлення загроз, зменшити кількість хибнопозитивних спрацьовувань і забезпечити ефективне функціонування систем захисту в умовах мінливого кіберпростору.

Практична значущість роботи полягає у створенні прикладного програмного рішення, яке може бути інтегроване до існуючих інформаційних систем підприємств, державних установ або освітніх організацій для підвищення їхнього рівня кіберзахисту.

Отримані результати можуть бути безпосередньо застосовані для удосконалення корпоративних систем моніторингу безпеки (SIEM), побудови центрів оперативного реагування (SOC) або використані як навчальний і дослідницький інструмент у галузі інформаційної безпеки. Крім того, розроблені методи й архітектурні рішення можуть бути адаптовані до різних масштабів систем — від локальних корпоративних мереж до хмарних середовищ, що забезпечує універсальність і практичну корисність одержаних результатів.

1 СИСТЕМНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Опис предметної області системи підтримки прийняття рішень

У межах предметної області програмного забезпечення для підвищення захисту інформаційних систем розглядається сукупність процесів, які забезпечують виявлення, аналіз і реагування на події інформаційної безпеки в організаційному середовищі. Основу становить багаторівнева структура оброблення даних, де телеметрія з серверів, мережевого обладнання та прикладних систем надходить у підсистему збору й кореляції подій, яка виконує нормалізацію, агрегування та первинний аналіз інформаційних потоків (рис. 1.1). На цьому етапі здійснюється попереднє фільтрування та ідентифікація подій, що мають ознаки інцидентів безпеки, з подальшим занесенням у журнал подій.

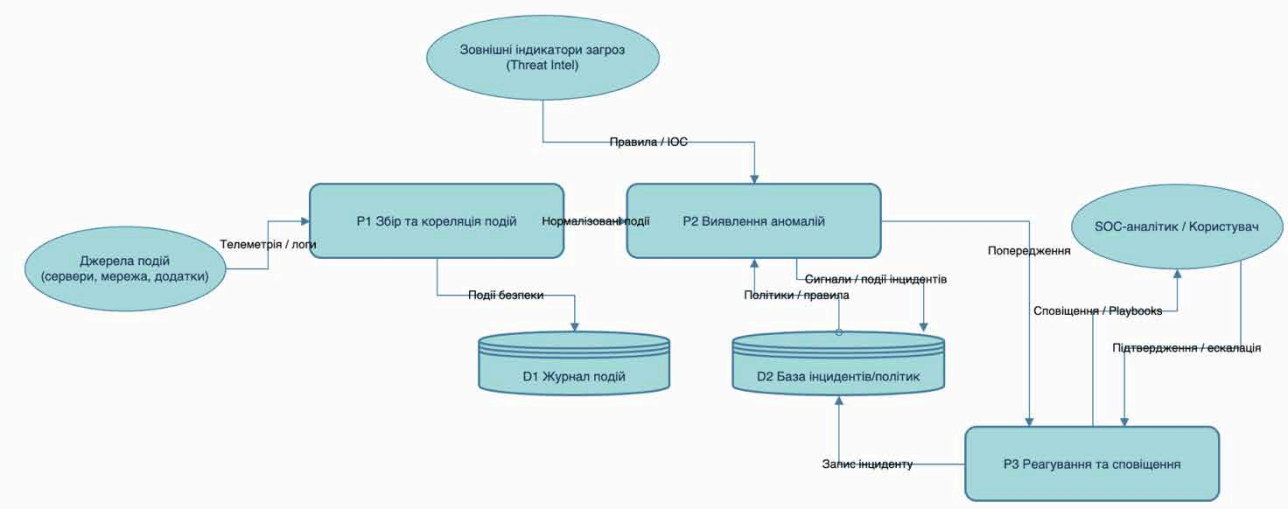

Рис. 1.1 – DFD-діаграма опису предметної області системи підвищення захисту інформаційних систем

Другий рівень функціонування системи передбачає модуль виявлення аномалій, який, використовуючи політики безпеки, індикатори компрометації (IOC) та зовнішні джерела Threat Intelligence, виконує поведінковий і статистичний аналіз для встановлення відхилень від нормальної активності. Результати аналізу формують базу інцидентів і політик, на основі якої будується

адаптивне реагування. Завершальна підсистема -реагування та сповіщення - генерує попередження, автоматизовані сценарії реагування (playbooks) і надсилає сповіщення SOC-аналітикам для підтвердження або ескалації інцидентів. Така організація процесів забезпечує зворотний зв’язок між користувачем і системою, дозволяючи постійно вдосконалювати правила виявлення загроз і мінімізувати ризики кібератак у реальному часі.

Узагальнена структура потоків даних наведена на рисунку 1.1, що відображає взаємодію зовнішніх джерел подій, процесів оброблення та внутрішніх сховищ інформації. Для характеристики інформаційних потоків наведено класифікаційну таблицю 1.1, яка формалізує основні типи даних, що циркулюють між компонентами системи.

Таблиця 1.1

Основні інформаційні потоки предметної області

№	Джерело / Призначення	Тип даних	Опис призначення
1	Сервери, мережеві пристрої, додатки	Телеметрія / логи	Потоки первинних подій безпеки, що підлягають кореляції
2	Підсистема збору подій → Журнал подій	Нормалізовані записи	Збереження уніфікованих подій для подальшого аналізу
3	Підсистема виявлення аномалій → База політик	Сигнали інцидентів	Результати аналітичного моделювання ризиків та виявлення загроз
4	Підсистема реагування → SOC-аналітик	Сповіщення / сценарії реагування	Автоматизовані дії та звіти про інциденти безпеки

Загалом предметна область системи охоплює процеси збору, аналітичної обробки, оцінки ризиків і реагування на загрози інформаційної безпеки у реальному часі. Вона базується на принципах безперервного моніторингу, машинного аналізу подій та динамічного формування політик реагування, що забезпечує підвищення ефективності виявлення загроз і мінімізацію людського фактора при прийнятті рішень

1.2 Теоретико-методологічні засади та стан наукових досліджень

Теоретико-методологічні засади розроблення програмного забезпечення для підвищення захисту інформаційних систем ґрунтуються на сучасних концепціях виявлення аномалій, машинного навчання та поведінкового аналізу, що дозволяють автоматизувати процес ідентифікації кіберзагроз у складних обчислювальних середовищах. У більшості досліджень акцент робиться на застосуванні алгоритмів кластеризації, нейронних мереж і статистичних моделей для виявлення відхилень у трафіку, затримках чи шаблонах поведінки користувачів [10]. Важливим теоретичним напрямом є використання візуальних моделей розподілу показників мережевої активності, що дозволяє проаналізувати кореляцію між пропускною здатністю, затримками, втратами пакетів і завантаженістю маршрутизаторів (рис. 1.2).

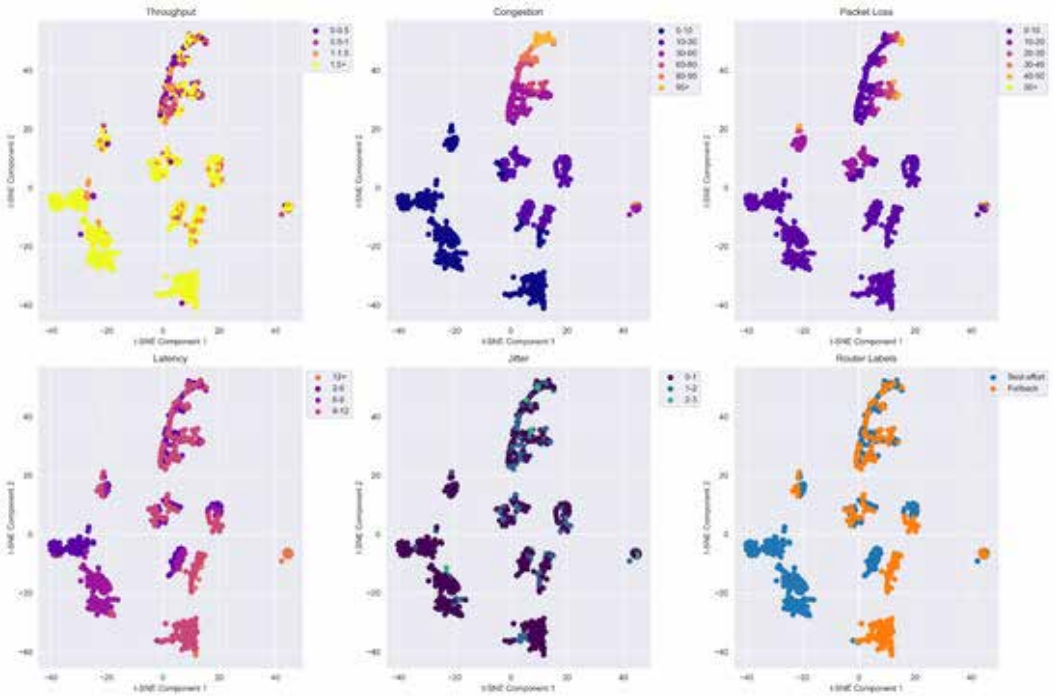

Рис. 1.2 – Візуалізація параметрів мережевого трафіку (Throughput, Congestion, Packet Loss, Latency, Jitter, Router Labels) для виявлення аномалій

Методологія аналізу аномалій у кіберзахисті передбачає інтеграцію часових і кореляційних характеристик потоків даних, що надає можливість побудови моделей прогнозування інцидентів на основі трендових залежностей. Зокрема, у працях [5] показано, що поєднання часових рядів із

класифікаційними моделями дозволяє ефективно виявляти зміни у поведінці мережі, пов’язані з атаками типу DoS або спуфінгом (рис. 1.3).

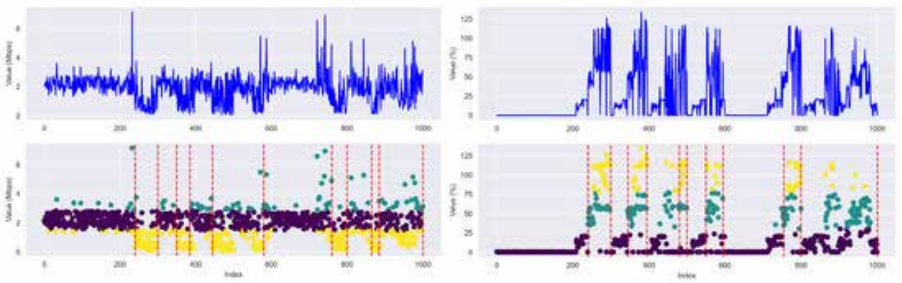

Рис. 1.3 – Графічне представлення часових рядів мережевих метрик та позначення інцидентів безпеки у часі

Суттєвий внесок у розвиток методології пояснюваного машинного навчання для інформаційної безпеки зробили дослідники Lundberg і Lee, які запропонували метод SHAP для кількісного визначення впливу кожної змінної на рішення моделі [6]. Цей підхід використовується для інтерпретації результатів класифікації мережевих подій за ознаками «нормальна», «модифікація даних», «спуфінг» та дозволяє формувати обґрунтовані політики реагування (рис. 1.4).

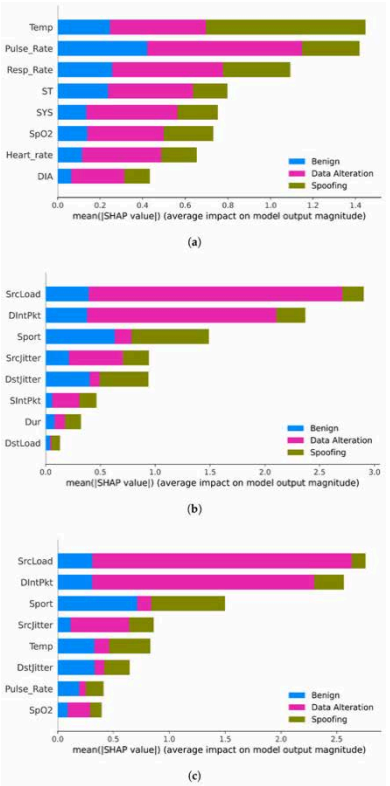

Рис. 1.4 – SHAP-аналіз впливу параметрів телеметрії на класифікацію інцидентів безпеки

Використання гібридних моделей, що поєднують статистичні підходи (наприклад, One-Class SVM) і методи штучного інтелекту, дало змогу покращити виявлення відхилень навіть за наявності обмеженої кількості маркованих даних [7]. На рисунку 1.5 показано приклад порівняння звичайної та аномальної активності, де алгоритм OCSVM забезпечує поділ простору даних відповідно до поведінкових закономірностей.

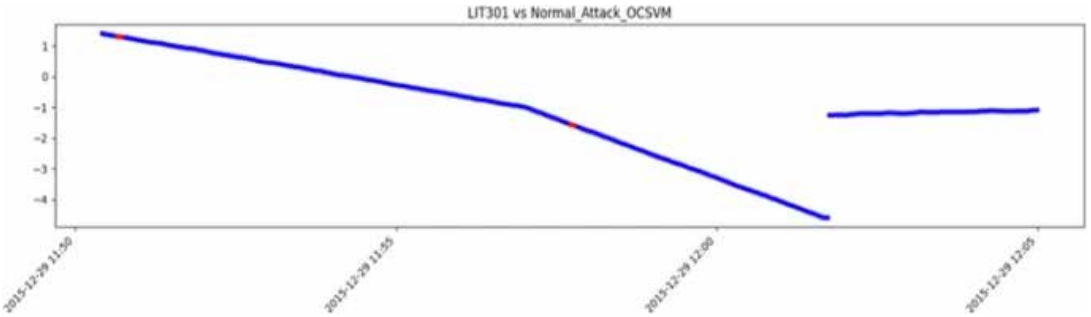

Рис. 1.5 – Модель відокремлення нормальної та аномальної активності на основі One-Class SVM

Теоретичні основи побудови систем виявлення вторгнень також спираються на парадигму комбінування різних методів штучного інтелекту - логіки нечітких множин, дерев рішень, кластерного аналізу, генетичних алгоритмів і нейронних мереж. Така інтеграція дає можливість створити багаторівневу структуру виявлення, де кожен метод забезпечує аналіз певного аспекту поведінки системи [8]. Узагальнену концептуальну схему класів алгоритмів IDS наведено на рисунку 1.6.

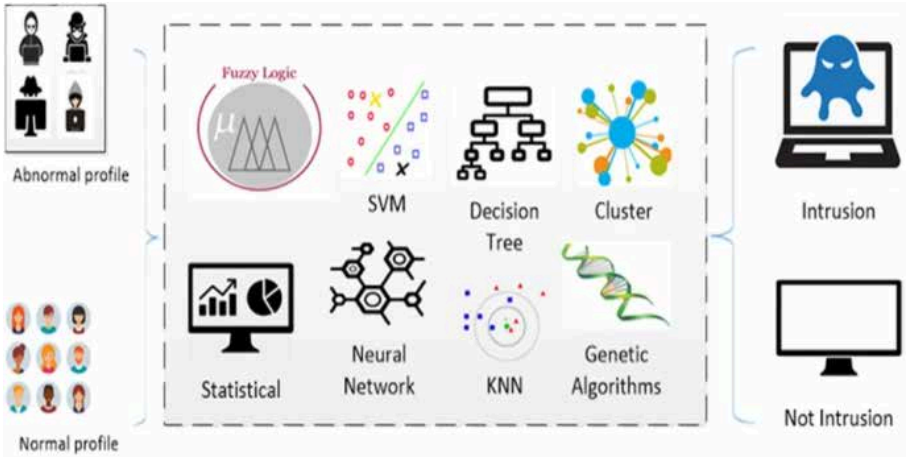

Рис. 1.6 – Концептуальна схема методів штучного інтелекту, що застосовуються у системах виявлення вторгнень

На основі проведеного аналізу можна зробити висновок, що сучасні дослідження у сфері кібербезпеки орієнтуються на використання глибоких і ансамблевих моделей машинного навчання, здатних адаптивно оновлювати політики виявлення. Проте більшість наявних систем залишаються орієнтованими лише на ідентифікацію подій, без інтеграції механізмів автоматичного реагування та зворотного навчання. Саме тому наукова новизна цієї роботи полягає у розробленні інтегрованого підходу, що поєднує поведінковий аналіз, методи інтелектуального оцінювання ризиків і адаптивне реагування на основі даних журналів подій, що дозволяє підвищити точність і швидкість реагування на загрози у реальному часі.

1.3 Аналіз існуючих рішень

Аналіз існуючих рішень у сфері забезпечення інформаційної безпеки показує, що нині найбільш поширеними є системи виявлення вторгнень і моніторингу подій безпеки, які поєднують аналіз мережевого трафіку, сигнатурне розпізнавання та поведінкову аналітику. До найвідоміших належать Snort, Suricata, OSSEC, Zeek та Cisco Secure IDS, кожна з яких реалізує власний підхід до виявлення загроз, логування та управління інцидентами [11].

Система Snort є класичним представником сигнатурно-орієнтованих IDS-рішень. Вона використовує базу правил для виявлення вторгнень, аналізує мережеві пакети на різних рівнях моделі OSI та формує звіти про потенційні загрози (рис. 1.7). Її інтерфейс дозволяє фільтрувати події, відслідковувати джерела атак і експортувати результати в журнали безпеки.

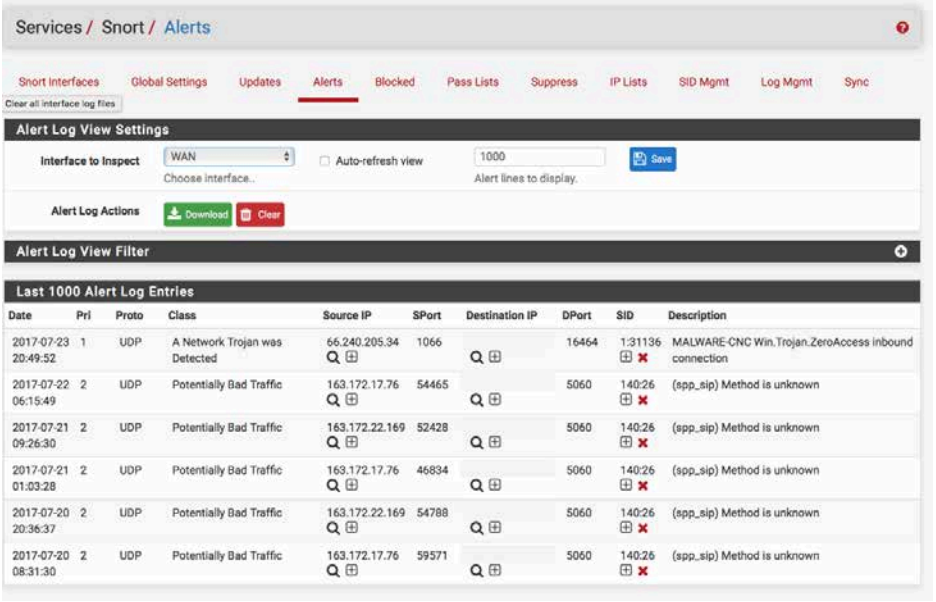

Рис. 1.7 – Інтерфейс системи Snort із журналом виявлених подій безпеки

Інша популярна система – Suricata – реалізує більш сучасний підхід, орієнтований на багатопоточність і розширену протокол-аналітику. Її архітектура дозволяє одночасно аналізувати тисячі потоків даних у реальному часі, підтримуючи SSL/TLS-декодування та інтеграцію з системами Kibana чи Elasticsearch (рис. 1.8). Ця гнучкість робить Suricata ефективною для великих корпоративних мереж із високим навантаженням [12].

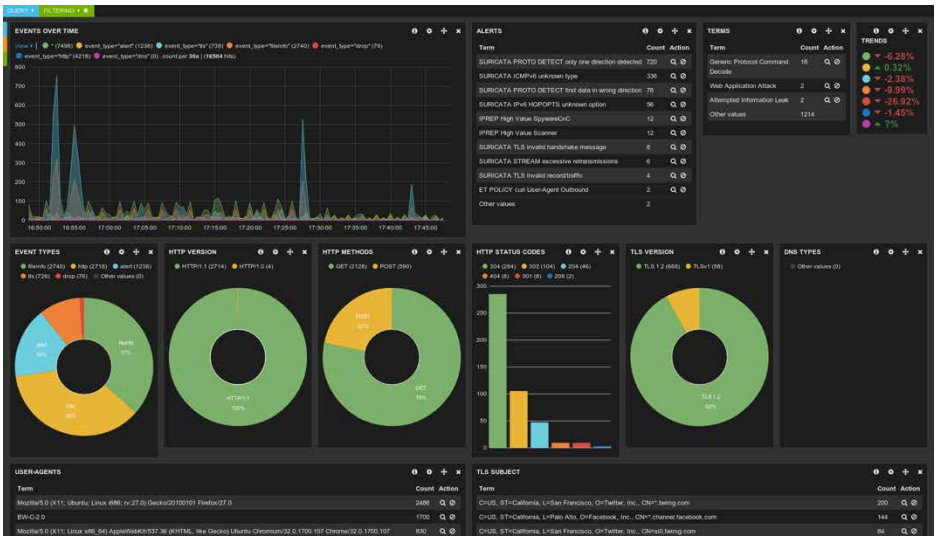

Рис. 1.8 – Панель аналітики Suricata з візуалізацією трафіку та подій безпеки

Система OSSEC, у свою чергу, фокусується на аналізі подій операційних систем і цілісності файлів, що робить її доцільною для використання в серверах

і хмарних середовищах. Вона забезпечує централізований контроль стану безпеки вузлів, формує звіти за CVE-уразливостями та дозволяє оцінити динаміку ризиків (рис. 1.9).

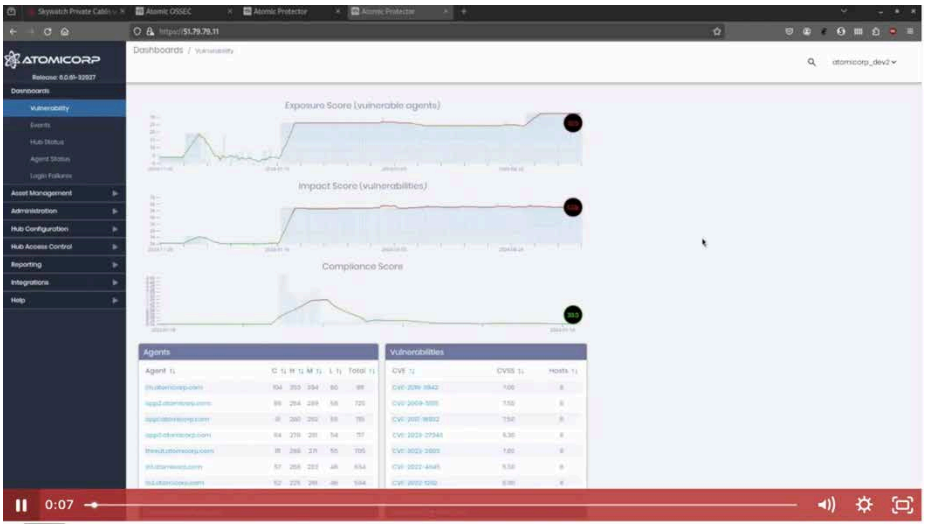

Рис. 1.9 – Приклад аналітичного дашборду OSSEC із показниками уразливостей і рівнем відповідності політикам

Система Zeek (раніше Bro) реалізує орієнтований на поведінковий аналіз підхід до моніторингу мережевого трафіку. Її аналітична панель дозволяє досліджувати SSL-сертифікати, відстежувати хости та сервіси, що полегшує ідентифікацію нетипової активності (рис. 1.10). Завдяки можливості гнучкої обробки логів Zeek часто інтегрується з SIEM-платформами як джерело первинної телеметрії [13].

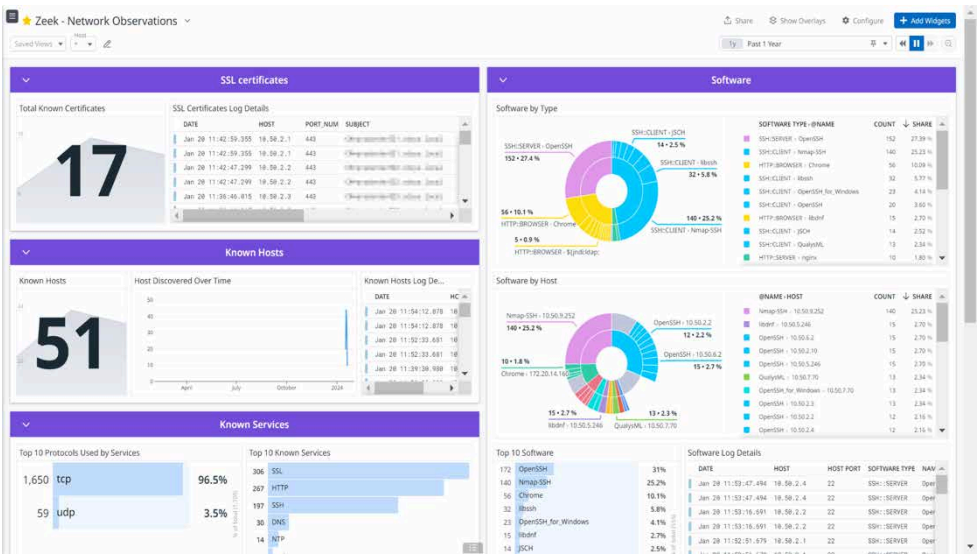

Рис. 1.10 – Дашборд Zeek із детальною аналітикою SSL-з'єднань і сервісів у мережі

І нарешті, Cisco Secure IDS є комерційним рішенням, яке поєднує функції виявлення атак, централізованого управління інцидентами та модулів комплаєнсу (рис. 1.11). Його перевагою є глибока інтеграція з корпоративною інфраструктурою Cisco, однак недоліком – висока вартість і обмежена відкритість архітектури.

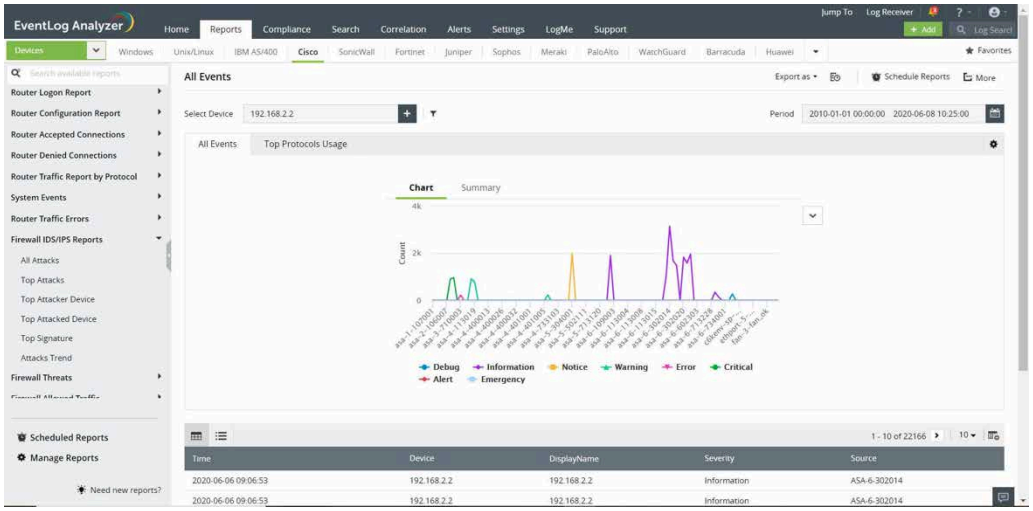

Рис. 1.11 – Панель Cisco Secure IDS із відображенням звітів за протоколами та критичністю подій

Для узагальнення проведено порівняльний аналіз п'яти основних рішень (див. табл. 1.2). У таблиці наведено їхні ключові характеристики за критеріями архітектури, типу аналізу, відкритості коду, підтримки AI-алгоритмів та інтеграції з SIEM-системами. До порівняння включено також розроблюване програмне забезпечення для підвищення захисту інформаційних систем, яке відрізняється інтеграцією адаптивного поведінкового аналізу, автоматизованого реагування та машинного навчання для зниження кількості хибних спрацьовувань.

Таблиця 1.2

Порівняльна характеристика існуючих систем захисту інформації

Система	Тип аналізу	Архітектура	Підтримка AI/ML	Інтеграція з SIEM	Особливості

Snort	Сигнатурний	Монолітна	Ні	Частково	Висока точність при відомих атаках
Suricata	Сигнатурний + мережевий	Багатопотокова	Обмежена	Так	TLS-декодування

Продовження таблиці 1.2

OSSEC	Хост-орієнтований	Агенто-центрична	Ні	Так	Контроль цілісності файлів і логів
Zeek	Поведінковий	Модульна	Частково	Так	Висока гнучкість і інтеграція з аналітичними системами
Cisco Secure IDS	Сигнатурно-аналітичний	Корпоративна	Ні	Так	Комерційне рішення з високим рівнем інтеграції
Розроблена система	Адаптивно-поведінковий + аналітичний	Мікросервісна	Так	Повна інтеграція	Інтелектуальне реагування та самонавчання модулів

Проведений аналіз засвідчує, що більшість існуючих рішень мають обмежену гнучкість у контексті самонавчання та автоматичного реагування. Вони здебільшого орієнтовані на пасивний моніторинг і не здатні адаптуватися до нових загроз без ручного оновлення сигнатур. На відміну від них, розроблена система пропонує інтелектуальний механізм динамічного аналізу ризиків та автоматизованого реагування з використанням моделей машинного навчання. Це забезпечує підвищення швидкості обробки інцидентів, скорочення кількості помилкових сповіщень і збільшення рівня загальної кіберстійкості організації.

1.4 Моделювання програмної системи

Моделювання предметної області системи підвищення захисту інформаційних систем здійснюється з метою формалізації взаємодії основних суб’єктів, процесів та інформаційних потоків, що забезпечують цілісність, конфіденційність і доступність даних у корпоративному середовищі. Для

побудови моделі використано методологію UML, яка дозволяє представити систему через різні аспекти – функціональний, поведінковий та інформаційний.

На діаграмі прецедентів (рис. 1.12) відображено ключових акторів і прецеденти, що описують основні сценарії взаємодії користувачів із системою. До основних суб'єктів належать адміністратор безпеки, аналітик SOC, користувач системи та зовнішній модуль SIEM/IDS. Система підтримує багатофакторну автентифікацію (MFA), керування користувачами за моделями RBAC/ABAC, моніторинг подій, реагування на інциденти, а також створення звітів відповідності стандартам GDPR, ISO та внутрішнім політикам НУБіП. Така модель відображає логіку комплексного керування подіями безпеки, включно з формуванням сповіщень, журналюванням і аудитом дій.

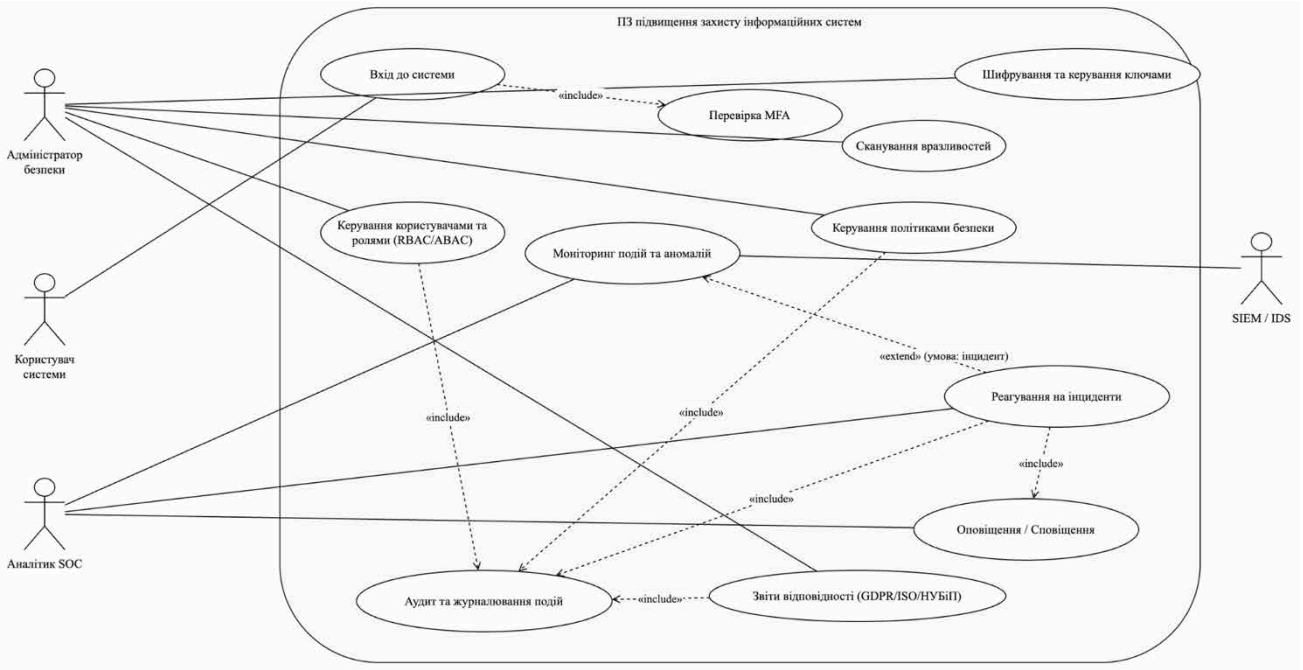

Рис. 1.12 – Діаграма прецедентів програмного забезпечення підвищення захисту інформаційних систем

Послідовність дій при автентифікації користувача та доступі до захищених ресурсів наведено на діаграмі послідовності (рис. 1.13). У ній проілюстровано повний цикл взаємодії між веб-клієнтом, Auth Service, сервісом MFA, модулем політик RBAC/ABAC, журналом аудиту та SIEM/SOAR-системою. Користувач надсилає запит авторизації, проходить перевірку MFA, після чого система формує токен JWT, що забезпечує

контрольований доступ до ресурсів. У випадку виявлення аномалії створюється інцидент безпеки, який передається у модуль SOAR для подальшого аналізу.

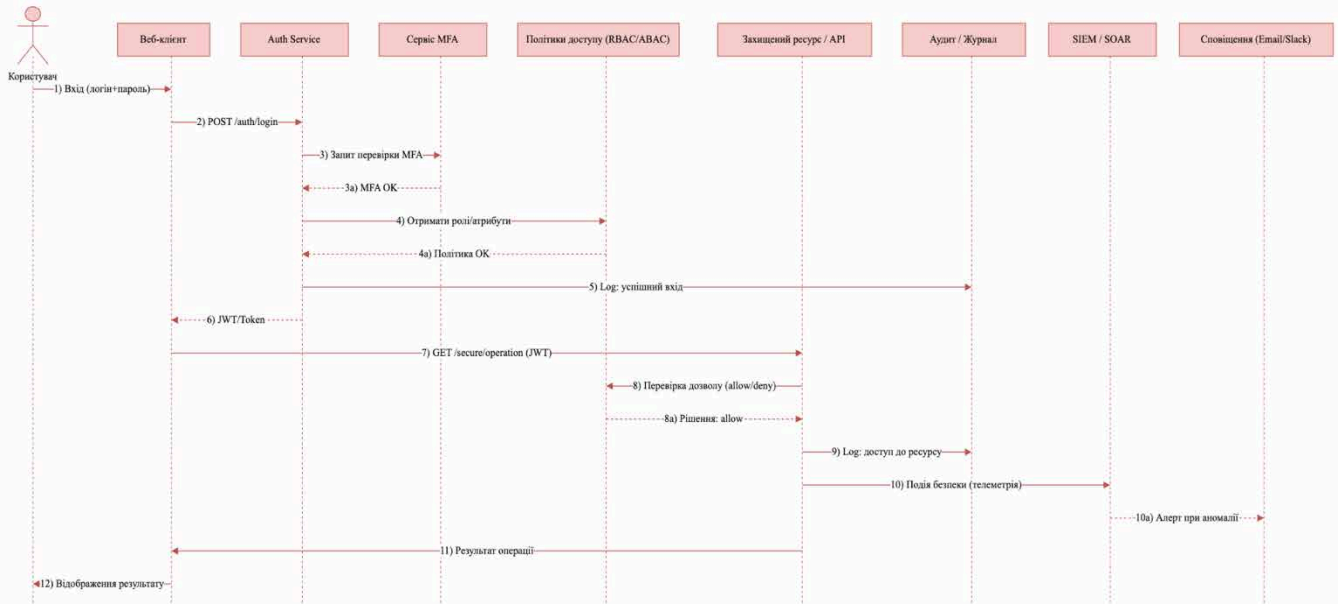

Рис. 1.13 – Діаграма послідовності взаємодії користувача з системою автентифікації та реагування

Логіка перевірки доступу і реакції на події безпеки деталізована на діаграмі активності (рис. 1.14). Процес починається з введення користувачем облікових даних і проходження перевірки MFA. Далі система оцінює політики доступу за атрибутами RBAC/ABAC, після чого або надає доступ, або реєструє відмову. Всі дії користувача журналюються, а при виявленні аномалії ініціюється створення інциденту в системі SOAR. Така схема демонструє інтеграцію механізмів автентифікації, контролю доступу й аналітичного реагування у єдиному безпековому контурі.

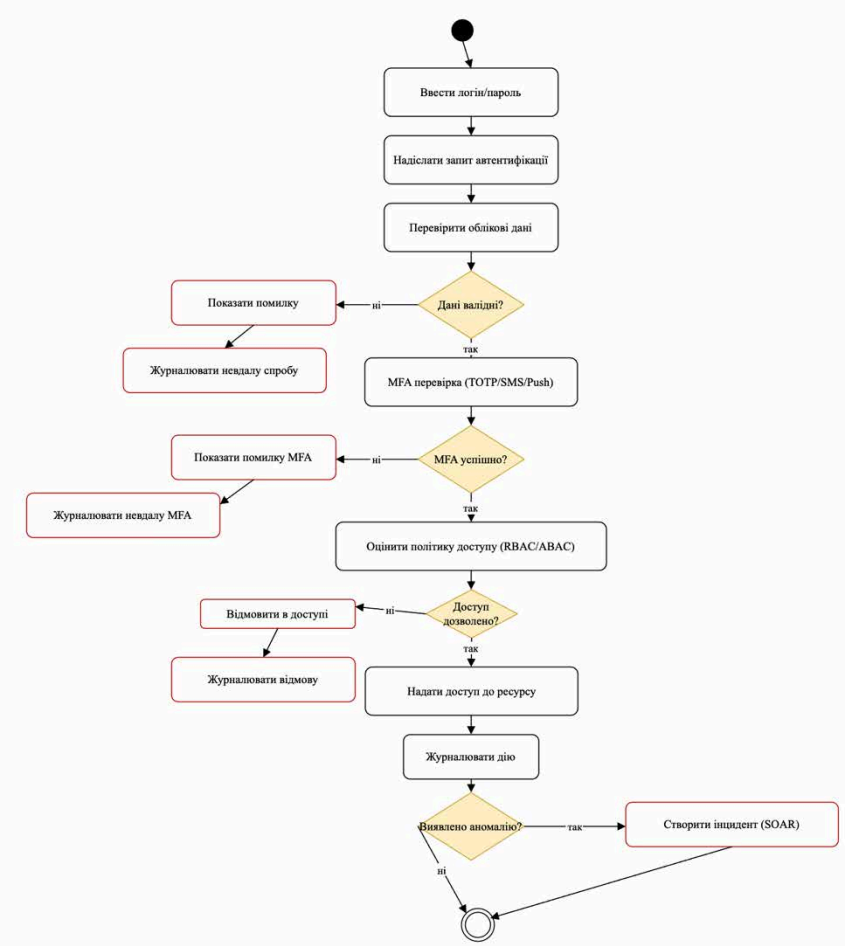

Рис. 1.14 – Діаграма активності процесу автентифікації та реагування на інциденти безпеки

Побудована модель предметної області описує узгоджену взаємодію компонентів системи підвищення захисту інформаційних систем, що забезпечує автоматизований моніторинг, контроль доступу та адаптивне реагування на події. Інтеграція політик RBAC/ABAC, модулів MFA та SIEM/SOAR створює передумови для реалізації інтелектуальної системи кіберзахисту, здатної до самоадаптації на основі аналізу журналів подій. Запропонована модель стане основою для подальшого проектування програмного забезпечення та доведення наукової новизни розробки.

1.5 Аналіз вимог захисту інформаційних систем

Аналіз вимог до експертної системи підвищення захисту інформаційних систем є ключовим етапом розроблення, який визначає структуру,

функціональні характеристики, обмеження та критерії якості майбутнього програмного забезпечення. Основна мета аналізу полягає у формалізації вимог користувачів і замовника з подальшим перетворенням їх у технічні специфікації, що забезпечують коректну роботу системи в умовах реального навантаження. Вимоги сформовано на основі принципів ISO/IEC 25010, методичних рекомендацій НУБіП України [16], а також сучасних підходів до побудови систем кіберзахисту [5], [7].

Функціональні вимоги визначають основні можливості системи, її сервіси та сценарії використання (див. табл. 1.3). Програмний продукт має забезпечувати автентифікацію користувачів, підтримку багатофакторної перевірки (MFA), управління ролями та атрибутами (RBAC/ABAC), моніторинг подій безпеки, реагування на інциденти, журналювання, а також генерацію звітів відповідності політикам GDPR/ISO. Особливістю системи є модуль експертного аналізу, який на основі накопичених журналів подій та моделей машинного навчання формує рекомендації для корекції політик безпеки й оптимізації рівня довіри до користувачів.

Таблиця 1.3

Функціональні вимоги до експертної системи

№	Функція	Опис	Очікуваний результат
1	Автентифікація користувачів	Підтримка входу з перевіркою облікових даних	Надання авторизованого доступу
2	Перевірка MFA	Підтримка TOTP, SMS або Push-перевірки	Підвищення рівня безпеки автентифікації
3	Керування ролями та політиками доступу (RBAC/ABAC)	Гранульоване налаштування дозволів	Контроль дій користувачів за атрибутами
4	Моніторинг подій і виявлення аномалій	Аналіз журналів та потоків подій у реальному часі	Виявлення підозрілої активності
5	Реагування на інциденти (SOAR)	Формування сценаріїв реагування, сповіщення	Автоматизоване усунення інцидентів
6	Журналювання та аудит	Збереження історії дій користувачів	Забезпечення прозорості контролю

Продовження таблиці 1.3

7	Звіти відповідності (GDPR, ISO)	Формування звітів за вимогами комплаєнсу	Підтримка стандартів і перевірок

Нефункціональні вимоги визначають якісні параметри системи, що характеризують її продуктивність, зручність використання, масштабованість і сумісність (див. табл. 1.4). Вони спрямовані на забезпечення стабільної роботи програмного комплексу при високих навантаженнях, мінімізації часу відгуку та гарантуванні відмовостійкості.

Таблиця 1.4

Нефункціональні вимоги до експертної системи

№	Категорія	Вимога	Показник / Критерій
1	Продуктивність	Час обробки запиту автентифікації	≤ 200 мс
2	Відмовостійкість	Система має відновлювати роботу після збою	≤ 10 сек. простою
3	Масштабованість	Підтримка до 10 000 одночасних сесій	Горизонтальне розширення кластерів
4	Зручність інтерфейсу	Інтуїтивна навігація, адаптивний дизайн	UX-індекс ≥ 0.85
5	Інтеграція	Сумісність із SIEM/SOAR, LDAP, REST API	Повна функціональна підтримка
6	Надійність зберігання	Використання транзакційної БД з ACID-властивостями	Без втрати даних при збої

Окрему категорію становлять вимоги до інформаційної безпеки, які встановлюють політику захисту даних, контроль доступу, шифрування та аудит (див. табл. 1.5). Для критичних компонентів системи передбачено використання TLS 1.3, криптографічних алгоритмів AES-256 та RSA-2048, а також механізмів обмеження прав доступу на основі атрибутів користувачів.

Таблиця 1.5

Вимоги до безпеки експертної системи

№	Категорія безпеки	Вимога	Реалізація
1	Конфіденційність	Шифрування каналів зв’язку	TLS 1.3, HTTPS

Продовження таблиці 1.5

2	Цілісність	Контроль хеш-сум даних і журналів	SHA-256, Merkle Tree
3	Доступність	Реплікація даних, моніторинг стану вузлів	Kubernetes + Prometheus
4	Аутентифікація	MFA, токени JWT	Auth Service + KeyStore
5	Авторизація	Модель RBAC/ABAC	Розподіл прав за атрибутами
6	Аудит і відповідність	GDPR/ISO 27001/NUBiP policy	Автоматизовані звіти комплаєнсу

Підсумовуючи результати аналізу вимог, слід зазначити, що система проєктується як інтелектуальний модуль адаптивного захисту, орієнтований на безперервне навчання та самооптимізацію механізмів реагування. Поєднання аналітичних методів, машинного навчання та формальної моделі політик доступу забезпечує високу точність і швидкодію при виявленні загроз, а також відповідність сучасним міжнародним стандартам кібербезпеки.

1.6 Постановка завдання

Постановка завдання для розроблення експертної системи підвищення захисту інформаційних систем полягає у формулюванні проблеми, визначенні мети, структури вхідних та вихідних даних, а також очікуваних функціональних результатів роботи програмного комплексу. Основним завданням є створення інтелектуального інструменту, який забезпечує автоматизований моніторинг подій безпеки, аналіз аномалій та адаптивне реагування на інциденти в реальному часі. Система повинна інтегруватися з існуючими корпоративними засобами кіберзахисту (SIEM, IDS/IPS) і підтримувати моделі керування доступом RBAC/ABAC, багатофакторну автентифікацію (MFA), а також механізми машинного навчання для аналізу поведінкових патернів користувачів.

У межах поставленої задачі необхідно забезпечити реалізацію таких функціональних модулів:

– модуль автентифікації та авторизації з перевіркою MFA;

– модуль моніторингу подій і потоків безпеки;

– аналітичний модуль експертної оцінки загроз на основі накопичених логів;

– модуль реагування на інциденти та формування рекомендацій;

– модуль звітності й аудиту відповідності вимогам безпеки.

Вхідними даними системи є:

– журнали подій та телеметрія мережевої активності (дані з SIEM/IDS, файлів логів, серверів і агентів моніторингу);

– результати перевірок автентифікації користувачів, включно з MFA-токенами та атрибутами доступу;

– інформація про конфігурацію системи безпеки, політики доступу, атрибути користувачів;

– набори навчальних даних для модулів машинного навчання (аномальні та нормальні шаблони поведінки);

– сигнали від зовнішніх джерел Threat Intelligence для збагачення контексту загроз.

Вихідними даними є:

– автоматично сформовані сповіщення та рекомендації для аналітиків SOC;

– результати виявлених інцидентів із класифікацією за рівнем критичності;

– оновлені політики доступу та правила реагування на події;

– аналітичні звіти про рівень захищеності, виявлені уразливості та ступінь відповідності вимогам GDPR/ISO;

– журнал аудиту дій користувачів і системних подій для подальшого аналізу.

Таким чином, система повинна забезпечувати замкнений цикл «виявлення → аналіз → реагування → вдосконалення політик», де кожен етап є взаємопов’язаний із попереднім через зворотні зв’язки. Очікуваним результатом

реалізації поставленого завдання є створення адаптивної експертної системи, здатної виявляти складні аномалії поведінки, автоматично ініціювати заходи реагування та забезпечувати відповідність міжнародним стандартам кіберзахисту. Реалізація системи здійснюється із застосуванням технологій Python, SQLLite, Flask/FastAPI, Docker, а також інтеграції з SIEM/SOAR-рішеннями для централізованого аналізу та обміну подіями безпеки.

1.7 Висновки до розділу 1

У результаті виконаного системного аналізу предметної області, проведеного у першому розділі, сформовано теоретико-методологічні засади, необхідні для розроблення експертної системи підвищення захисту інформаційних систем. У ході дослідження визначено основні напрями розвитку сучасних рішень у сфері кібербезпеки, здійснено аналіз існуючих платформ типу Snort, Suricata, OSSEC, Zeek та Cisco Secure IDS, які продемонстрували ефективність у виявленні мережевих загроз, але мають обмеження щодо адаптивності, масштабованості й інтелектуальної обробки даних. Це обґрунтовує потребу створення нової експертної системи, що поєднує методи машинного навчання, аналітичне оцінювання подій та автоматизоване реагування на інциденти.

У розділі виконано моделювання предметної області із застосуванням UML-діаграм, що дозволило описати структуру системи, її функціональні сценарії, потоки взаємодії користувачів та процеси перевірки доступу (MFA, RBAC/ABAC) і реагування на аномальні події. Розроблені діаграми прецедентів, послідовності та активності відобразили повний цикл роботи системи - від автентифікації користувача до автоматизованого інцидент-менеджменту через інтеграцію з SIEM/SOAR.

Проведено аналіз вимог, що включає функціональні, нефункціональні та вимоги до безпеки. Визначено, що система має забезпечувати час відгуку не

більше ніж 200 мс, підтримку до 10 000 одночасних сесій, шифрування каналів зв’язку за TLS 1.3, багатофакторну автентифікацію, повний аудит дій користувачів і відповідність стандартам GDPR/ISO 27001.

У постановці завдання сформульовано вхідні та вихідні дані системи, які охоплюють телеметрію, журнали подій, результати перевірок автентифікації та сигнали Threat Intelligence. Очікуваними вихідними результатами є автоматично сформовані сповіщення SOC-аналітику, оновлені політики безпеки, звіти комплаєнсу та рекомендації з підвищення рівня захищеності.

Перший розділ закладає науково-методологічну основу для подальшого проектування та реалізації експертної системи. Наукова новизна розробки полягає у поєднанні моделей RBAC/ABAC із машинним аналізом журналів безпеки, що дозволяє реалізувати адаптивну поведінкову модель реагування на інциденти в реальному часі. Результати розділу забезпечують повну обґрунтованість архітектури майбутнього програмного комплексу та визначають напрям подальшої розробки системи у другому розділі.

2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОГО ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Логічна модель даних у вигляді ER-діаграми

Логічна модель даних експертної системи підвищення захисту інформаційних систем відображає концептуальну структуру сховища даних, яка забезпечує узгодженість, нормалізацію та ефективність оброблення інформації у межах усіх підсистем. Під час моделювання застосовано методологію ER/UML-підходу, що дозволяє формалізувати сутності, атрибути, первинні та зовнішні ключі, а також типи зв’язків між таблицями. Основна мета побудови цієї моделі - створення цілісної логічної основи, що гарантує узгодженість даних під час моніторингу, аналізу інцидентів та адаптивного реагування.

На рис. 2.1 наведено узагальнену ER-діаграму логічної моделі даних, яка відображає взаємозв’язок між ключовими об’єктами системи: користувачами, ролями, політиками доступу, журналами подій, інцидентами, сповіщеннями, аналітичними моделями машинного навчання та звітами відповідності. Структура моделі побудована за принципами третої нормальної форми (3NF), що дозволяє уникнути надлишковості, забезпечує уніфікацію атрибутів і мінімізує ризик аномалій при оновленні або видаленні записів. Усі сутності мають унікальні первинні ключі (UUID) і підтримують зовнішні зв’язки для формування логічно цілісних транзакцій.

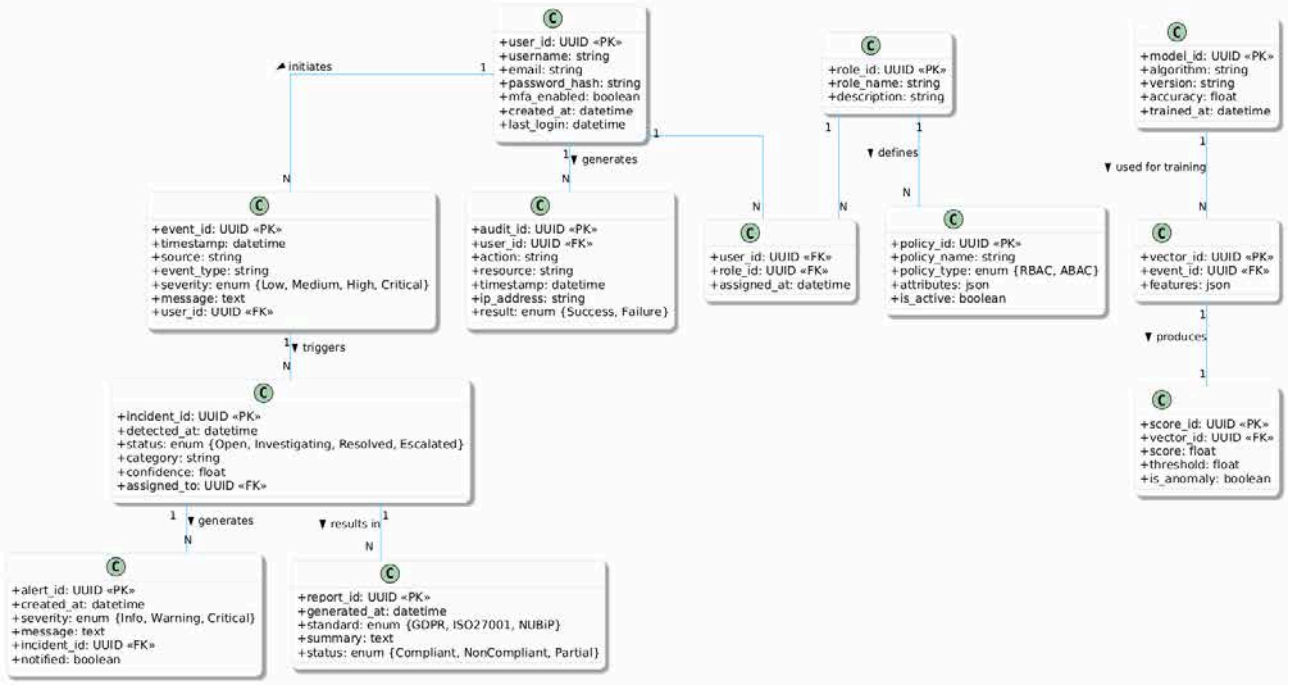

Рис. 2.1 – Логічна модель даних експертної системи підвищення захисту інформаційних систем

Для формалізації параметрів моделі створено класифікаційну таблицю 2.1, у якій подано основні сутності бази даних і відповідні домени атрибутів. Таблиця дозволяє системно визначити типи даних і взаємозв’язки, що формують ядро бази знань системи.

Таблиця 2.1

Основні сутності логічної моделі даних

№	Сутність	Призначення	Основні атрибути
1	User	Облік користувачів, автентифікація та MFA	user_id, username, email, password_hash, mfa_enabled
2	Role / AccessPolicy	Моделі RBAC/ABAC, контроль політик доступу	role_id, policy_id, attributes, is_active
3	EventLog	Реєстрація подій безпеки та телеметрії	event_id, timestamp, source, severity, message
4	Incident / Alert	Виявлення аномалій, формування інцидентів і сповіщень	incident_id, status, category, alert_id, severity
5	MLModel / FeatureVector / AnomalyScore	Аналітичне ядро ML-модулів, класифікація аномалій	model_id, vector_id, score, threshold

Продовження таблиці 2.1

6	AuditLog / ComplianceReport	Аудит дій користувачів і звіти відповідності GDPR/ISO	audit_id, action, result, report_id, standard

Побудована логічна модель формує єдиний семантичний простір даних, у якому кожен модуль системи (моніторинг, аналітика, реагування, аудит) працює з узгодженими структурами. Такий підхід забезпечує масштабованість і розширюваність схеми без порушення цілісності бази, а також створює передумови для реалізації ETL-процесів і подальшої побудови OLAP-кубу для стратегічного аналізу показників безпеки. Застосування принципів нормалізації та модульної структуризації дозволяє інтегрувати модель із зовнішніми системами SIEM/SOAR, гарантуючи узгодженість даних і високу швидкодію при обробленні запитів у реальному часі.

Створена логічна модель даних є основою інформаційної архітектури експертної системи, на якій базується подальше фізичне проєктування сховища, оптимізація запитів і забезпечення повного життєвого циклу даних - від реєстрації події до формування звіту комплаєнсу.

2.2 Діаграма класів і кооперації

Діаграма класів експертної системи підвищення захисту інформаційних систем формує структурну основу її програмної архітектури та демонструє взаємозв’язки між ключовими компонентами, що реалізують основні функції моніторингу, аналізу подій, виявлення аномалій і формування інцидентів безпеки. Вона відображає логічну організацію об’єктно-орієнтованих сутностей, їх атрибути, методи та принципи взаємодії, що забезпечують цілісність і масштабованість системи. Структура класів побудована відповідно до принципів інкапсуляції та наслідування, що гарантує гнучкість у розширенні функціональності без порушення існуючої логіки.

На рис. 2.2 представлено UML-діаграму класів системи, яка відображає зв’язки між об’єктами SecurityEvent, FeatureVector, Detector, OCSVMDetector,

RuleEngine та Incident. Клас SecurityEvent відповідає за приймання та нормалізацію даних телеметрії, тоді як FeatureVector формує набір ознак для аналітичної обробки. Класи Detector та його спадкоємець OCSVMDetector реалізують функції обробки та класифікації даних на основі алгоритму One-Class SVM, визначаючи наявність аномалії. Компонент RuleEngine виконує роль інтерпретатора результатів, а Incident зберігає інформацію про створені події безпеки, дозволяючи виконувати операції ескалації чи закриття. Така архітектура сприяє реалізації адаптивного ядра, здатного до самонавчання і реагування в реальному часі.

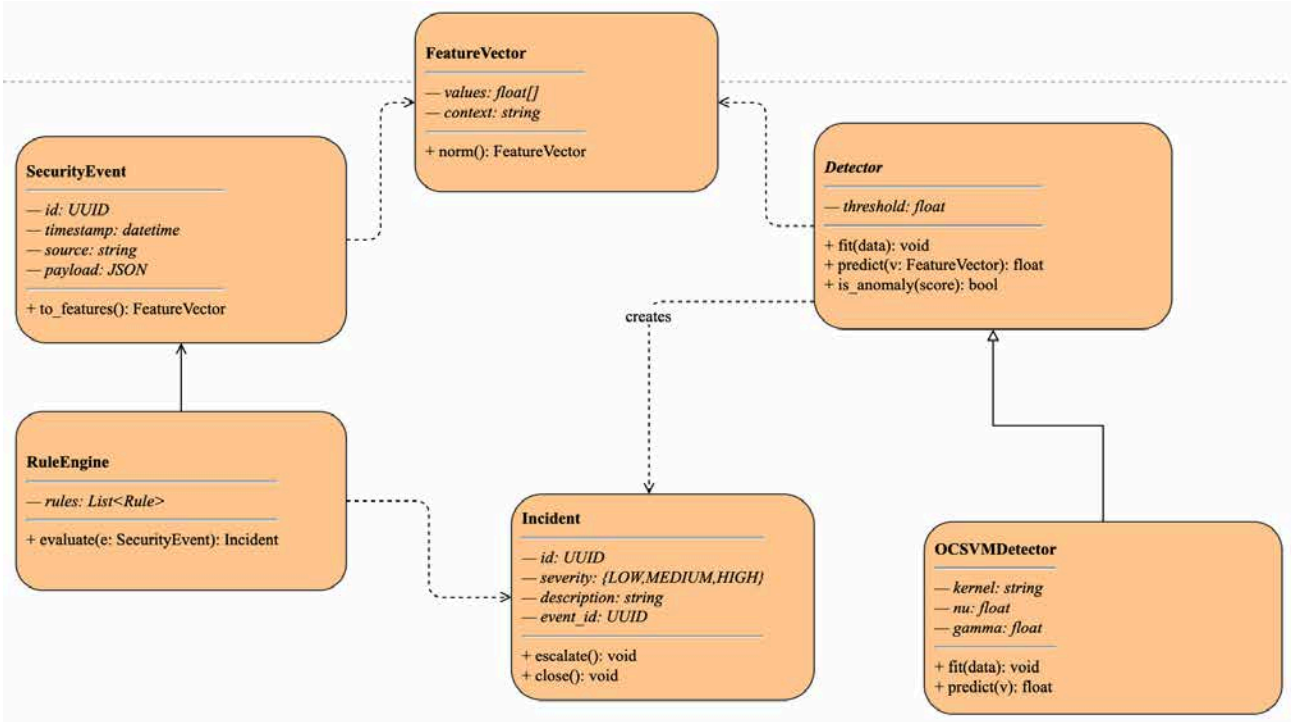

Рис. 2.2 – UML-діаграма класів експертної системи підвищення захисту інформаційних систем

Послідовність взаємодії між класами під час оброблення подій розкрита у вигляді трьох діаграм кооперацій, які демонструють життєвий цикл події від моменту її надходження до завершення обробки інциденту. Перша кооперація (рис. 2.3) описує процес ініціалізації події клієнтом, що надсилає дані безпеки до системи. Об’єкт SecurityEvent приймає вхідний JSON-пакет, формує структуру події та викликає метод to_features(), який трансформує отриману інформацію у вектор ознак FeatureVector. Цей етап забезпечує нормалізацію

даних і підготовку до подальшого аналізу за допомогою моделей машинного навчання, що дозволяє підвищити точність класифікації загроз.

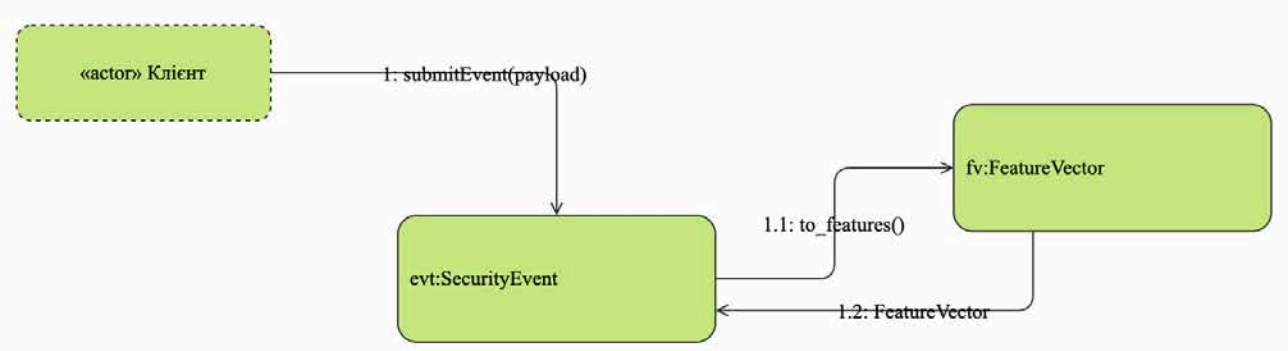

Рис. 2.3 – Кооперація клієнта та модуля оброблення подій SecurityEvent

Друга кооперація (рис. 2.4) демонструє етап аналізу події в аналітичному ядрі системи. Вектор ознак FeatureVector передається до модуля OCSVMDetector, який виконує оцінку схожості з нормальними шаблонами поведінки та обчислює показник аномальності score. Якщо результат перевищує порогове значення, метод is_anomaly(score) ініціює створення інциденту. Об’єкт RuleEngine обробляє цей сигнал, застосовує внутрішні політики кореляції подій і викликає метод createIncident(event, severity) для створення нового об’єкта Incident. Таким чином, система забезпечує автоматизоване виявлення аномалій та формування відповідних записів у базі даних подій безпеки.

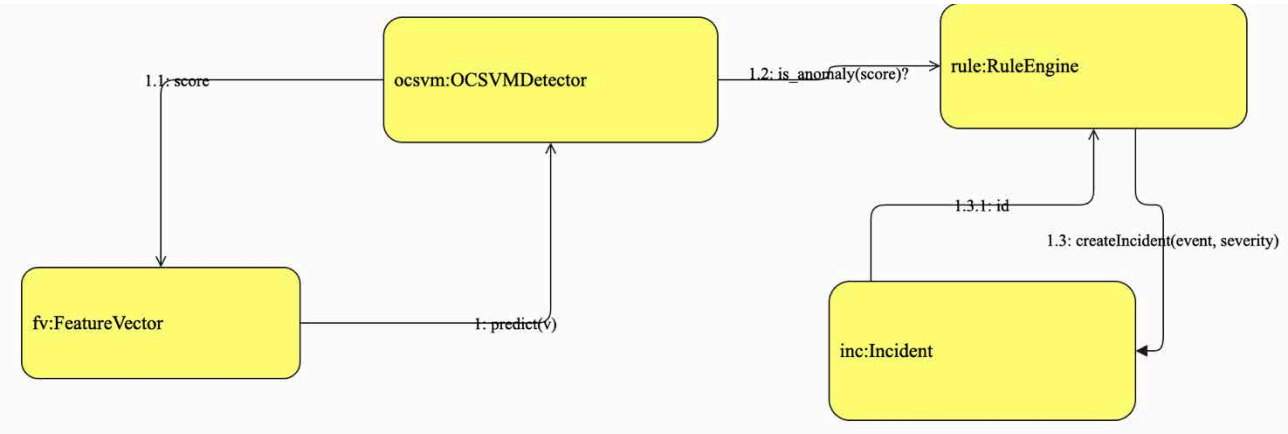

Рис. 2.4 – Кооперація OCSVMDetector, RuleEngine та Incident при створенні інциденту

Заключна кооперація (рис. 2.5) описує взаємодію між аналітичним ядром, адміністратором і механізмами управління інцидентами під час їхньої ескалації чи повторної оцінки. Коли система фіксує критичний інцидент, об’єкт RuleEngine надсилає сповіщення адміністратору, який отримує повідомлення через метод notify() і може виконати перевірку чи коментар за допомогою викликів addComment() або approveResponse(). У випадку підтвердження серйозності події інцидент ескалується через метод escalate(), а після виконання заходів реагування позначається як закритий методом close(). Такий сценарій демонструє зворотний цикл управління інцидентами, де аналітична система та оператор взаємодіють у межах єдиного процесу реагування, що підвищує рівень автоматизації та швидкість усунення загроз.

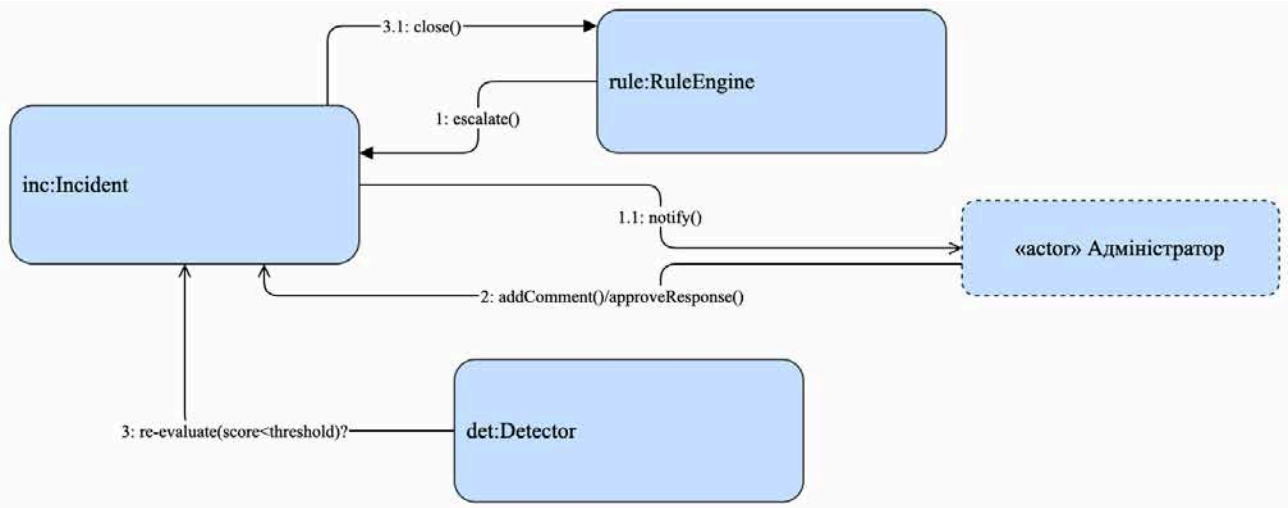

Рис. 2.5 – Кооперація Incident, RuleEngine, Detector та адміністратора системи при ескалації події

Для узагальнення структури класів і взаємодій між ними в таблиці 2.2 подано класифікацію основних об’єктів системи, їх функціональне призначення та роль у процесі аналізу подій безпеки.

Таблиця 2.2

Класифікація основних класів програмної системи

№	Клас	Підсистема	Основне призначення
1	SecurityEvent	Моніторинг	Збирання телеметрії та формування векторів ознак

2	FeatureVector	Аналітика	Нормалізація та підготовка даних для оброблення
3	Detector / OCSVMDetector	Машинне навчання	Виявлення аномалій за моделями поведінки

Продовження таблиці 2.2

4	RuleEngine	Експертна логіка	Прийняття рішень і створення об’єктів інцидентів
5	Incident	Керування подіями	Реєстрація, ескалація й завершення інцидентів
6	Administrator	Користувацька взаємодія	Контроль результатів аналізу, підтвердження дій системи

Отже, побудовані діаграми класів і кооперацій відображають повний життєвий цикл оброблення подій безпеки - від первинної телеметрії до формування, верифікації й закриття інцидентів. Така архітектура забезпечує адаптивність, нормалізовану структуру даних і гнучкість масштабування, а її модульна організація дозволяє інтегрувати нові алгоритми машинного навчання без порушення основної логіки системи. Результати цього етапу створюють методологічну основу для подальшої реалізації програмного комплексу та його оптимізації на рівні програмних інтерфейсів і компонентів.

2.3 Компонентна діаграма програмного забезпечення експертної системи підвищення захисту інформаційних систем

Компонентна діаграма визначає структурну організацію програмного комплексу експертної системи підвищення захисту інформаційних систем і демонструє логіку взаємодії між її функціональними модулями, базами даних і зовнішніми системами кіберзахисту. Побудова цієї діаграми ґрунтується на принципах модульності, інверсії залежностей та сервісно-орієнтованої архітектури (SOA), що забезпечує відокремлення відповідальності кожного компонента, гнучкість оновлень та масштабованість системи. Всі елементи взаємодіють через стандартизовані інтерфейси REST/GraphQL з використанням асинхронної обробки подій.

На рис. 2.6 подано UML-діаграму компонентів, яка відображає ключові сервіси системи:API Gateway, AuthService, EventCollector, FeatureService, DetectorService, RuleEngine та IncidentService.

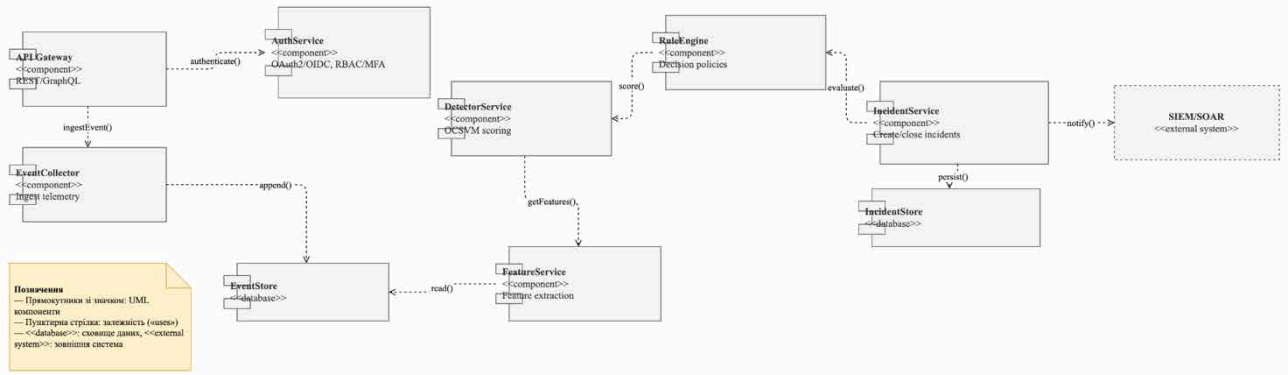

Рис. 2.6 – Компонентна діаграма програмного забезпечення експертної системи підвищення захисту інформаційних систем

Модуль APIGateway реалізує вхідну точку взаємодії з клієнтами системи та виконує маршрутизацію запитів, зокрема передавання телеметрії до EventCollector. Компонент AuthService відповідає за автентифікацію користувачів із підтримкою OAuth2/OIDC і механізмів багатофакторної перевірки (MFA), забезпечуючи контроль доступу на рівні політик RBAC/ABAC. Компонент EventCollector виконує прийом і попередню нормалізацію потоків даних, після чого інформація зберігається у сховищі EventStore.

Подальша обробка відбувається у FeatureService, який виконує екстракцію ознак з подій і формує структуровані вектори для подальшого аналізу. Отримані ознаки передаються до DetectorService, що реалізує алгоритм виявлення аномалій на основі One-Class SVM (OCSVM) і обчислює показники ризику. Результати передаються до RuleEngine, який здійснює експертну оцінку з використанням політик прийняття рішень. На цьому етапі система або автоматично створює інцидент через IncidentService, або виконує його закриття за результатами повторної перевірки. Усі дані інцидентів зберігаються у IncidentStore, що є основним транзакційним сховищем, і синхронізуються із зовнішніми системами SIEM/SOAR для централізованого моніторингу.

Для формалізації функціональної структури розроблено таблицю 2.3, у якій наведено коротку характеристику основних компонентів, їх призначення та ролі у загальній архітектурі системи.

Таблиця 2.3

Основні компоненти системи та їх функціональне призначення

№	Компонент	Тип	Основне призначення
1	APIGateway	Сервіс доступу	Приймає запити REST/GraphQL, маршрутизує телеметрію та виконує попередню валідацію
2	AuthService	Сервіс автентифікації	Реалізує авторизацію користувачів за моделями OAuth2/OIDC, RBAC/ABAC, підтримує MFA
3	EventCollector	Сервіс збору подій	Приймає потоки даних безпеки, виконує нормалізацію й передає до сховища подій
4	EventStore	Сховище даних	Зберігає телеметрію, журнали подій і метадані безпеки
5	FeatureService	Сервіс аналітики	Виділяє ознаки для моделювання й формує структуровані вектори даних
6	DetectorService	Сервіс машинного навчання	Виконує класифікацію аномалій за алгоритмом OCSVM, визначає рівень ризику
7	RuleEngine	Експертний модуль	Оцінює результати аналізу, застосовує політики прийняття рішень
8	IncidentService	Сервіс управління інцидентами	Створює, оновлює й закриває інциденти безпеки, синхронізує дані з SIEM/SOAR
9	IncidentStore	Сховище інцидентів	Транзакційна база даних для збереження історії та статусів інцидентів
10	SIEM/SOAR	Зовнішня система	Отримує повідомлення для централізованого реагування та аудиту подій

Компонентна архітектура побудована за принципом високої когерентності всередині модулів і слабкого зв’язку між ними, що дозволяє ізолювати збої, забезпечити стійкість і підтримувати горизонтальне масштабування системи. Реалізація міжкомпонентної взаємодії через подієво-орієнтовані інтерфейси

гарантує ефективність обміну даними навіть за великої кількості одночасних запитів.

Побудована компонентна діаграма відображає логічну структуру програмного комплексу експертної системи кіберзахисту, у якій кожен компонент виконує чітко визначену функцію в єдиному процесі «збір - аналіз - виявлення - реагування - звітування». Така архітектура відповідає вимогам надійності, розширюваності та відмовостійкості, а також створює основу для подальшої реалізації модулів у контейнеризованому середовищі (Docker/Kubernetes) із підтримкою інтеграції в корпоративні рішення рівня SOC.

2.4 Діаграма пакетів програмного забезпечення експертної системи підвищення захисту інформаційних систем

Діаграма пакетів відображає ієрархічну структуру програмного забезпечення та принципи розподілу функціональності між логічними модулями, що забезпечує впорядкованість, повторне використання коду та незалежність компонентів. Вона розроблена з урахуванням принципів чистої архітектури (Clean Architecture) та інверсії залежностей (Dependency Inversion), що дає змогу зберігати слабке зв’язування між підсистемами і спрощує їхнє тестування та подальший розвиток. Такий підхід особливо важливий для інтелектуальних систем кіберзахисту, де постійне оновлення алгоритмів і політик виявлення загроз вимагає високої модульності й контрольованої взаємодії між рівнями.

На рис. 2.7 представлено UML-діаграму пакетів програмного комплексу, що включає основні рівні: PresentationAPI, SecurityCore, Detection, Persistence, Integration та Common. Взаємозалежності між пакетами реалізовано через інтерфейси, позначені відношенням use, що гарантує односпрямовану взаємодію без циклічних залежностей. Така структура відображає логіку типового контуру обробки подій безпеки — від запиту користувача через

API-шлюз до формування результатів аналітики, їх збереження та інтеграції з зовнішніми системами SOC/ SIEM.

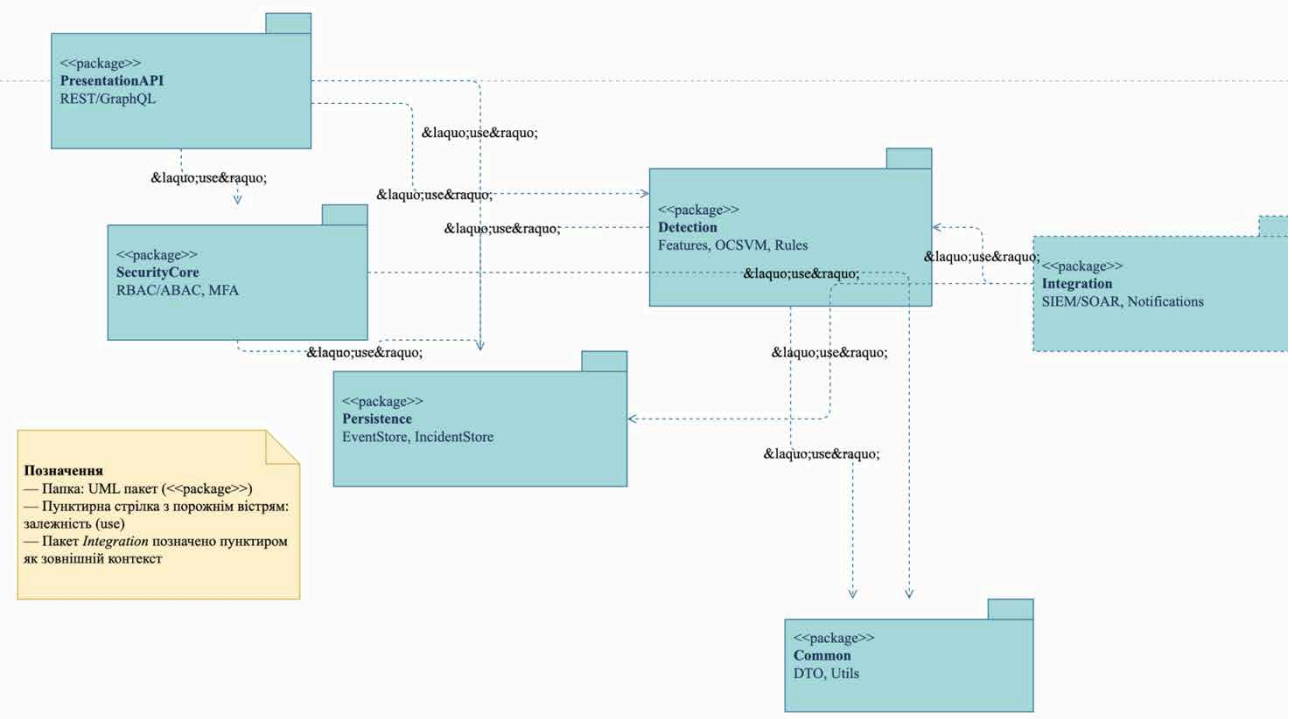

Рис. 2.7 – UML-діаграма пакетів програмного забезпечення експертної системи підвищення захисту інформаційних систем

В архітектурі застосовано багаторівневу сегментацію: зовнішній рівень відповідає за комунікацію з користувачами та зовнішніми системами, проміжний - за логіку безпеки, аналітичну обробку та виявлення аномалій, а внутрішній - за збереження, реплікацію та доступ до даних. Такий підхід мінімізує перетин відповідальностей і дозволяє розподіляти навантаження між пакетами залежно від обсягів телеметрії та частоти оновлення політик. Важливо, що пакет Integration функціонує у вигляді зовнішнього контексту, що забезпечує відокремленість ядра системи від сторонніх платформ, роблячи її придатною до інтеграції в інфраструктури різних організацій.

Для узагальнення логічних зв’язків між рівнями побудовано таблицю 2.4, яка описує архітектурну роль кожного пакета у формуванні цілісного циклу виявлення та реагування на події інформаційної безпеки.

Таблиця 2.4

Логічна структура пакетів програмного комплексу

№	Пакет	Основне призначення	Роль у системі
1	PresentationAPI	Інтерфейс взаємодії REST/GraphQL	Обробка запитів користувачів, передавання телеметрії
2	SecurityCore	Реалізація моделей доступу RBAC/ABAC та MFA	Контроль автентифікації й авторизації
3	Detection	Алгоритми OCSVM, Feature Extraction, Rule-логіка	Виявлення аномалій, аналітичне оцінювання загроз
4	Persistence	Сховища EventStore і IncidentStore	Забезпечення надійного збереження даних і журналів
5	Integration	Взаємодія із зовнішніми системами SIEM/SOAR	Централізоване сповіщення й синхронізація подій
6	Common	DTO-моделі, утиліти, спільні типи	Забезпечення узгодженості форматів даних і сервісних операцій

Запропонована структура пакетів формує чітку межу між рівнями застосунку й усуває ризик змішування логіки безпеки з бізнес-функціональністю. Вона також полегшує оновлення алгоритмів машинного навчання без зміни базової архітектури, що є ключовим для динамічних середовищ кіберзахисту. Такий підхід гарантує стабільність, контроль версій та ефективну інтеграцію модулів системи у виробниче середовище, забезпечуючи одночасно узгодженість політик безпеки та високу швидкодію обробки інцидентів.

2.5 Висновки до розділу 2

У другому розділі здійснено проєктування програмного забезпечення експертної системи підвищення захисту інформаційних систем, що базується на принципах модульності, інкапсуляції та адаптивної архітектури. Розроблені логічна модель даних, діаграми класів, кооперацій, компонентів і пакетів утворюють цілісну структуру програмного комплексу, яка забезпечує гнучкість,

масштабованість і надійність під час оброблення великих обсягів телеметрії та подій безпеки.

Побудована логічна модель даних відображає взаємозв’язок між сутностями користувачів, подій, інцидентів і аналітичних модулів, забезпечуючи узгодженість та нормалізацію даних відповідно до третьої нормальної форми (3NF). Така структура мінімізує надлишковість і спрощує реалізацію транзакційних механізмів у сховищах EventStore та IncidentStore.

Діаграма класів і кооперацій формалізує програмну логіку взаємодії між модулями системи - від генерації події безпеки до створення інциденту та реагування на нього. У межах цієї моделі реалізовано зв’язок між об’єктами SecurityEvent, FeatureVector, OCSVMDetector, RuleEngine і Incident, що забезпечує повний цикл оброблення загроз за принципом «подія → аналіз → виявлення → реагування». Кооперaційні діаграми відобразили сценарії взаємодії користувача, аналітичного ядра та адміністратора під час ескалації інцидентів, підкресливши роль зворотного зв’язку між модулями для підвищення точності класифікації аномалій.

Компонентна діаграма окреслила структурно-функціональну модель системи, у якій кожен сервіс виконує автономну роль: APIGateway обробляє запити REST/GraphQL, AuthService керує автентифікацією, FeatureService і DetectorService відповідають за аналітику й машинне навчання, а IncidentService взаємодіє з SIEM/SOAR для централізованого реагування. Така архітектура забезпечує високу відмовостійкість, горизонтальне масштабування та можливість контейнерного розгортання у середовищах Docker/Kubernetes.

Діаграма пакетів структуризувала логіку системи за рівнями: PresentationAPI, SecurityCore, Detection, Persistence, Integration і Common. Поділ на пакети дозволяє розмежувати відповідальність між підсистемами, підтримувати чисту архітектуру та реалізувати принципи інверсії залежностей. Це гарантує стабільність програмного ядра навіть за умов розширення функціональності чи оновлення моделей машинного навчання.

Узагальнюючи результати розділу, можна стверджувати, що розроблена архітектура забезпечує цілісну інформаційну взаємодію між модулями, підтримує адаптивне виявлення загроз і динамічне реагування на інциденти в режимі реального часу. Проєктні рішення відповідають сучасним вимогам до систем кіберзахисту - відмовостійкості, масштабованості, комплаєнсу з міжнародними стандартами (GDPR, ISO/IEC 27001) та можливості інтеграції в корпоративні середовища SOC. Розроблена структура є основою для подальшої реалізації програмних модулів, алгоритмів машинного навчання та оптимізації продуктивності на етапі практичної розробки системи.

3 ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Вибір технологій та інструментальних засобів реалізації системи

Розроблення експертної системи підвищення захисту інформаційних систем потребує використання сучасного інструментарію, який забезпечує високу швидкодію, безпечну обробку даних, масштабованість і можливість інтеграції з існуючими корпоративними платформами. При виборі технологій було враховано специфіку предметної області — обробку великих потоків телеметрії, застосування алгоритмів машинного навчання для виявлення аномалій, а також необхідність відповідності міжнародним стандартам безпеки (ISO/IEC 27001, GDPR). Архітектура побудована за принципами мікросервісної взаємодії, що дозволяє незалежно розгортати й оновлювати компоненти, мінімізуючи вплив змін на загальну працездатність системи.

У межах проєкту застосовано стек технологій, який поєднує гнучкість розроблення, надійність і підтримку сучасних інструментів DevSecOps. Серверна частина реалізована на мові Python із використанням фреймворку FastAPI, який забезпечує асинхронну обробку подій і легку інтеграцію REST/GraphQL API. Аналітичне ядро системи побудовано на базі бібліотек scikit-learn, NumPy та Pandas, що дає змогу виконувати векторизацію ознак, навчання моделей (зокрема OCSVM) та статистичний аналіз. Для зберігання подій та інцидентів використано PostgreSQL як основну реляційну базу даних і MongoDB як документно-орієнтоване сховище для JSON-телеметрії. Передача подій реалізується через MQTT-брокер, що гарантує мінімальні затримки при обробленні потоків телеметрії.

Управління контейнерами забезпечується платформою Docker, що дозволяє ізолювати сервіси та спрощує їх деплоймент у середовищах Kubernetes або OpenShift. Для забезпечення безпеки комунікацій застосовано TLS 1.3, а контроль доступу реалізовано за моделями RBAC/ABAC із підтримкою OAuth2

і MFA. Журнали подій зберігаються централізовано через стек ELK (Elasticsearch, Logstash, Kibana), який інтегрується із зовнішніми SIEM-платформами.

Таблиця 3.1

Вибір технологій та інструментальних засобів системи

№	Технологія / інструмент	Категорія	Призначення у системі
1	Python 3.11 / FastAPI	Мова та фреймворк	Реалізація серверної логіки та API-сервісів
2	scikit-learn, NumPy, Pandas	ML-бібліотеки	Аналіз даних, навчання моделей OCSVM, формування векторів ознак
3	PostgreSQL, MongoDB	Бази даних	Збереження подій, інцидентів і аналітичних даних
4	MQTT (paho-mqtt)	Протокол обміну	Передача потоків телеметрії між агентами та аналітичним ядром
5	Docker / Kubernetes	Оркестрація	Контейнеризація, масштабування та розгортання мікросервісів
6	OAuth2, MFA, TLS 1.3	Безпека	Автентифікація, шифрування та захист каналів зв’язку
7	ELK-stack (Elasticsearch, Logstash, Kibana)	Моніторинг	Централізований аудит і візуалізація подій
8	Git / GitHub Actions	DevSecOps	CI/CD-інтеграція, контроль версій і автоматизоване тестування
9	Grafana / Prometheus	Аналітика та метрики	Збір продуктивності, моніторинг стану модулів
10	SIEM/SOAR (наприклад, Wazuh, TheHive)	Зовнішня інтеграція	Кореляція подій і автоматизоване реагування

Обраний технологічний стек забезпечує сумісність компонентів, безпеку обміну даними та ефективність машинного аналізу. Завдяки використанню Python-екосистеми та мікросервісного підходу система залишається гнучкою для масштабування й адаптації під специфічні корпоративні сценарії. Використання сучасних протоколів автентифікації та інструментів DevSecOps гарантує сталість життєвого циклу розробки, а поєднання аналітичних бібліотек і візуалізаційних інструментів створює основу для побудови динамічної

інфраструктури кіберзахисту з інтелектуальним виявленням загроз і автоматичним реагуванням у реальному часі.

3.2 Архітектура системи та проєктування функціональних модулів аналітичного ядра експертної системи підвищення захисту інформаційних систем

Архітектура розробленої експертної системи побудована за багатошаровою моделлю, що поєднує мікросервісну структуру, асинхронну передачу подій і гібридне сховище даних для аналітичної обробки телеметрії безпеки. Основним принципом проєктування є декомпозиція за функціональними доменами - Edge/Ingress, Processing Core, Data Layer, Integration & Response та Observability. Така структура забезпечує незалежність між сервісами, підвищує масштабованість і дозволяє реалізувати гнучкий механізм оновлення без зупинки всієї системи.

На рис. 3.1 подано структурну архітектуру системи, що демонструє основні контури даних і взаємодію між сервісами. Потік подій починається з автентифікації клієнта через безпечний шлюз (Auth Service, TLS 1.3, MFA), після чого події телеметрії передаються до каналу повідомлень Kafka/MQTT. У ядрі системи функціонують сервіси аналітики (Feature Service, Detector Service, Rule Engine, Incident Service), які виконують екстракцію ознак, виявлення аномалій та формування інцидентів. Результати оцінюються оркестратором реагування (Response Orchestrator), що взаємодіє з SIEM/SOAR-платформами для централізованої кореляції та автоматизації дій.

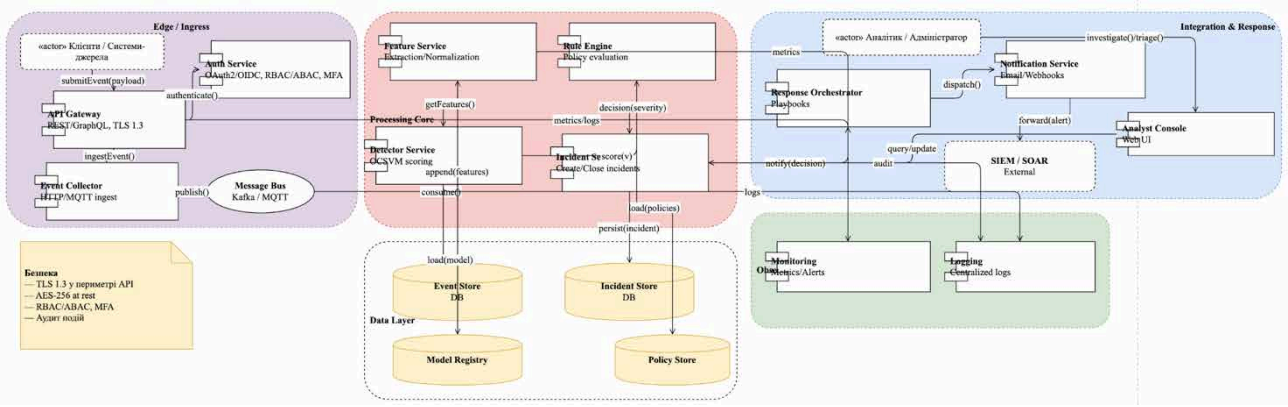

Рис. 3.1 – Архітектура експертної системи підвищення захисту інформаційних систем

На рис. 3.2 показано узагальнену діаграму розгортання компонентів, де реалізовано розподіл обчислювальних вузлів між зонами DMZ (API-шар), Core (аналітичне ядро), Data Server (сховища Event/Incident/Policy Store) та Observability (моніторинг і аудит). Завдяки цьому досягається фізична сегментація трафіку, ізоляція шарів безпеки та підвищена стійкість до атак типу Lateral Movement. Комунікація між сервісами виконується через безпечні REST-виклики або подієві черги Kafka, що мінімізує затримки та втрати пакетів при великих навантаженнях.

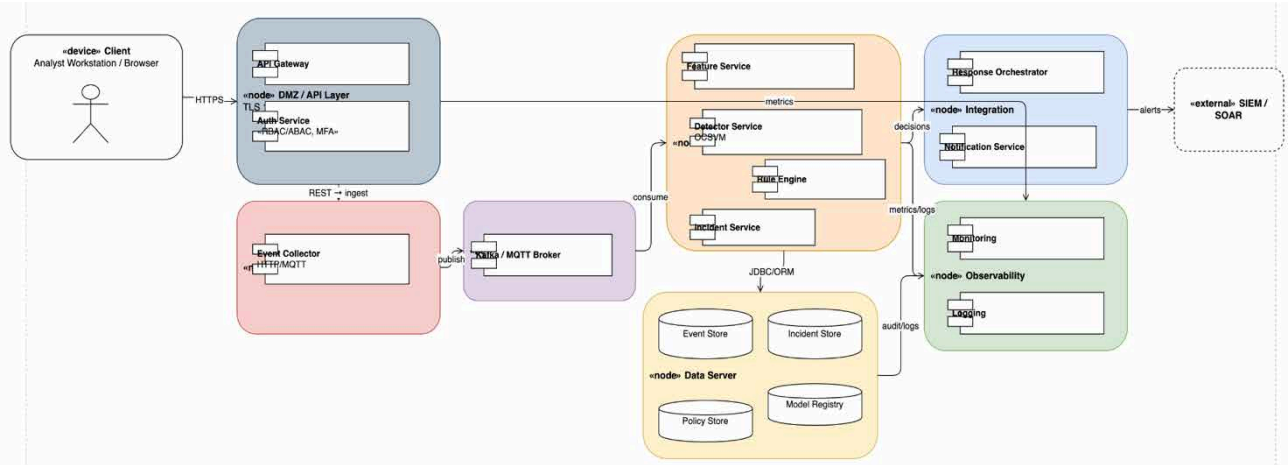

Рис. 3.2 – Діаграма розгортання вузлів та сервісів системи в контексті модульної архітектури

Наукова новизна проєктованої архітектури полягає у синтезі аналітичного ядра системи безпеки та механізмів самонавчання, що дозволяє не лише фіксувати аномалії, але й адаптивно оновлювати політики реагування на основі

зворотного зв’язку. У системі впроваджено поєднання евристичної аналітики (Rule Engine) з класифікаційними моделями OCSVM, що забезпечує подвійний рівень детекції - сигнатурний і поведінковий. Додатково реалізовано Policy Store для динамічного збереження політик доступу, що автоматично оновлюються при зміні поведінкових моделей користувачів. Ця особливість дає змогу забезпечити самоадаптацію системи без необхідності ручного конфігурування.

З метою забезпечення цілісності даних використано механізм Data Provenance, який фіксує походження кожної події від моменту її надходження до формування інциденту. Поєднання аналітичного ядра з оркестратором реагування дозволяє автоматично запускати playbooks у разі виявлення критичних загроз. Такий підхід поєднує інтелектуальні методи класифікації з практичною реалізацією автоматизованих рішень рівня SOC.

Таблиця 3.2

Науково-технічні аспекти реалізації архітектури системи

№	Аспект інновації	Сутність технічного рішення	Очікуваний ефект
1	Інтеграція аналітики ML із політиками Rule Engine	Комбіноване виявлення загроз за поведінковими й сигнатурними моделями	Підвищення точності детекції та зменшення хибнопозитивних спрацьовувань
2	Адаптивна система політик доступу (Policy Store)	Динамічне оновлення RBAC/ABAC-правил залежно від результатів аналізу поведінки	Автоматизоване управління безпекою без ручного втручання
3	Використання Kafka/MQTT як подієвої шини	Асинхронна передача телеметрії з QoS-гарантіями та низькою затримкою	Стійкість до перевантажень і втрат повідомлень
4	Застосування TLS 1.3 та AES-256 для захисту даних	Повний захист трафіку та збережених даних у межах Zero-Trust-підходу	Високий рівень конфіденційності та цілісності
5	Інтеграція з SIEM/SOAR через Response Orchestrator	Автоматичне запускання сценаріїв реагування на основі аналітики	Скорочення часу реагування на інциденти (MTTR)

Продовження таблиці 3.2

6	Data Provenance і централізований аудит	Відстеження походження даних і логів на всіх етапах життєвого циклу	Забезпечення прозорості та відповідності GDPR/ISO 27001
7	Контейнеризована інфраструктура (Docker/Kubernetes)	Ізоляція сервісів, автоматичне масштабування та оновлення без простоїв	Висока відмовостійкість і гнучке розгортання системи

Отже, архітектурне рішення відображає науково обґрунтовану концепцію інтеграції інтелектуального аналізу даних у процеси кіберзахисту. На відміну від традиційних IDS/IPS-підходів, запропонована система реалізує замкнений цикл «виявлення → оцінка → реагування → адаптація», у якому алгоритми машинного навчання безперервно вдосконалюються на основі реальних подій. Це створює передумови для переходу від реактивного до превентивного кіберзахисту**,** підвищує оперативність реагування та забезпечує наукову новизну проєкту як інтелектуальної експертної системи безпеки нового покоління.

3.3 Побудова OLAP-кубу та аналітичної моделі для дослідження інцидентів кібербезпеки

У межах дослідження реалізовано OLAP-модель (Online Analytical Processing), яка забезпечує багатовимірний аналіз інцидентів кібербезпеки в системі, що була спроєктована у попередніх підрозділах. Основна ідея полягає у побудові аналітичного кубу, який поєднує часові, просторові та поведінкові виміри для виявлення закономірностей у динаміці ризиків, джерел загроз і ефективності реагування. Архітектура кубу ґрунтується на моделі “зірки”, де факт-таблиця IncidentFact акумулює показники ризику та часу реакції, а довідкові таблиці IncidentDim, SourceDim, DateDim і MethodDim містять відповідні атрибути вимірів. Такий підхід дозволяє ефективно виконувати агрегації й аналітичні запити без надлишкової денормалізації даних.

На рис. 3.3 подано логічну схему OLAP-кубу системи, де відображено зв’язки між фактами й вимірами. Завдяки цьому забезпечено можливість формування інтерактивних зрізів (slice-and-dice) за типом інциденту, рівнем ризику, методом атаки чи часовим інтервалом.

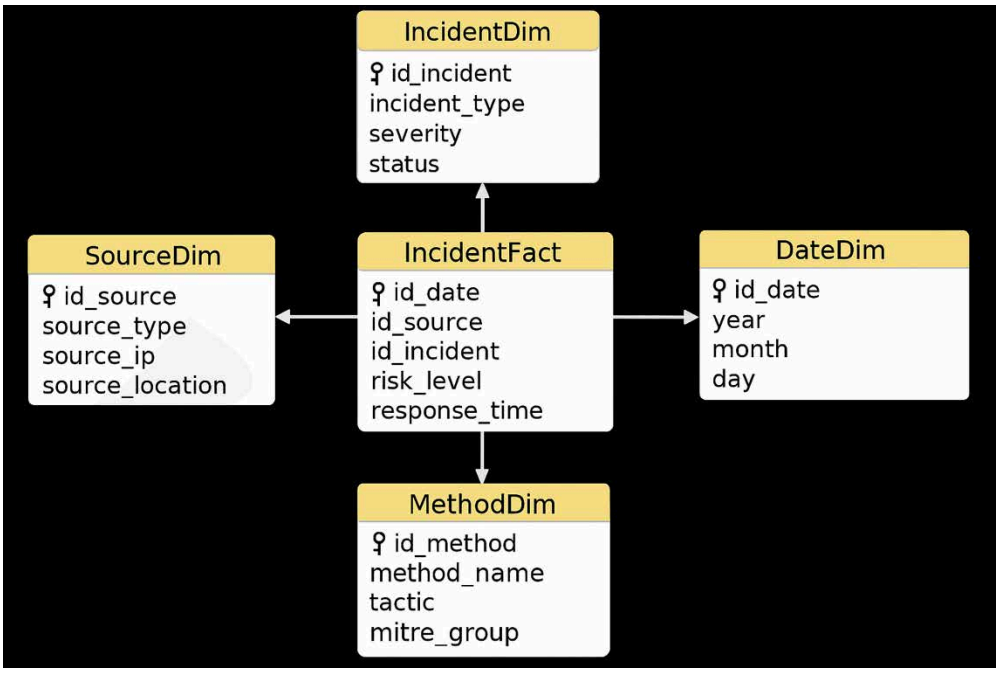

Рис. 3.3 – Логічна модель даних OLAP-кубу системи

Для визначення оптимальної кількості кластерів ризиків і швидкості реагування застосовано метод машинного навчання k-means. На рис. 3.4 наведено результати кластеризації інцидентів за ознаками “events per hour” і “MTTR_s” (Mean Time to Response), що дозволило виокремити три типові групи поведінки системи реагування.

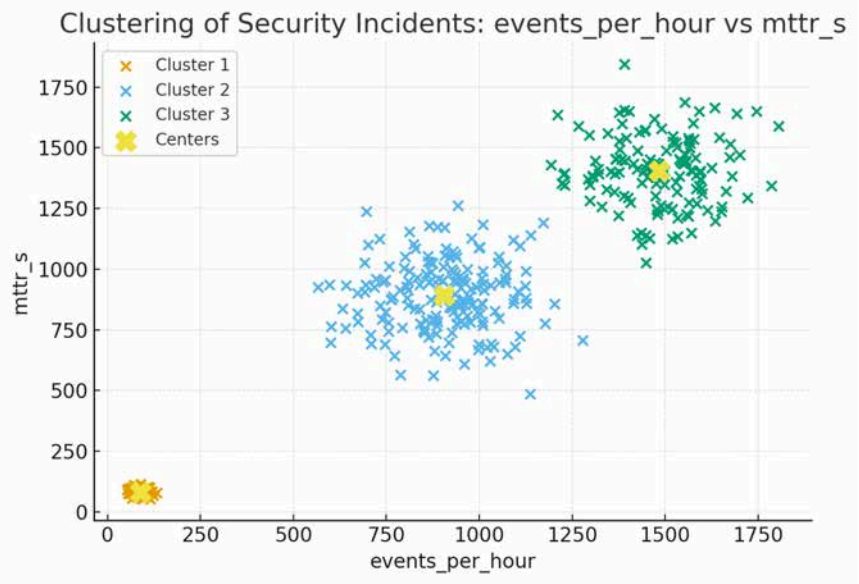

Рис. 3.4 – Кластеризація інцидентів безпеки за показниками подій та часу реагування

Підбір оптимальної кількості кластерів виконано за допомогою методу ліктя (Elbow Method), що показано на рис. 3.5. Перелом кривої при k = 3 свідчить про досягнення балансу між точністю моделі й стабільністю кластерів, що є основою для подальшої аналітики ризиків.

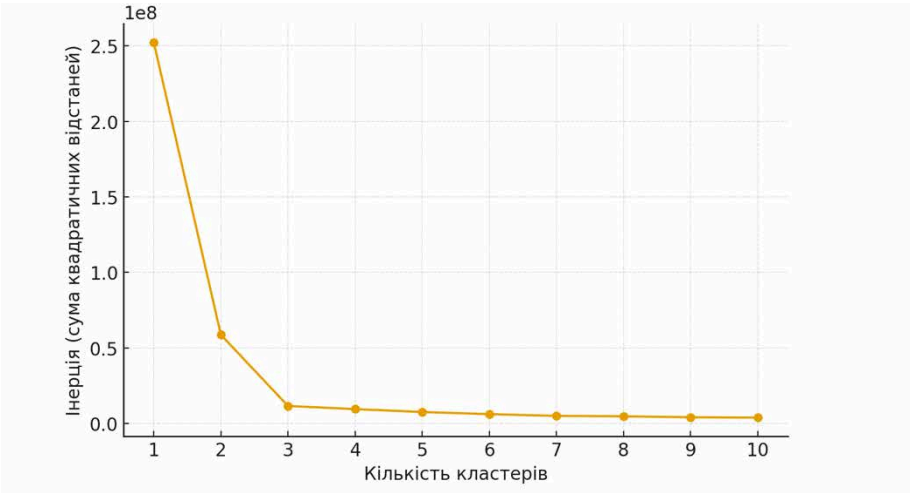

Рис. 3.5 – Вибір оптимальної кількості кластерів методом ліктя

Для візуалізації порівняльних характеристик кластерів побудовано радіальну діаграму KPI (рис. 3.6), де оцінено ключові параметри системи: точність виявлення, інтенсивність подій, ризик і оперативність реагування. Така форма представлення дає змогу інтерпретувати багатовимірні дані у зручній графічній площині, підкреслюючи відмінності між групами інцидентів.

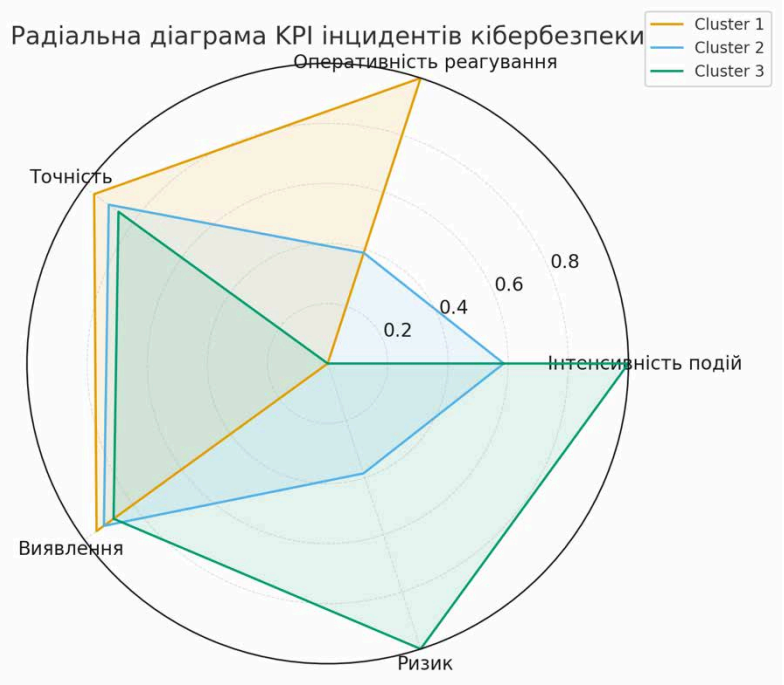

Рис. 3.6 – Радіальна діаграма KPI інцидентів кібербезпеки

Для формування аналітичних звітів було розроблено SQL-запити до OLAP-кубу, що реалізують агрегації на рівнях часу, джерел та методів атак. На рис. 3.7 представлено запит для підрахунку загальної кількості інцидентів і середнього рівня ризику за датою, що використовується у формуванні часових зрізів.

```sql
SELECT
    d.year,
    d.month,
    d.day,
    COUNT(f.id_incident) AS total_incidents,
    ROUND(AVG(f.risk_level), 2) AS avg_risk_level,
    ROUND(AVG(f.response_time), 2) AS avg_response_time
FROM IncidentFact AS f
JOIN DateDim AS d ON f.id_date = d.id_date
GROUP BY d.year, d.month, d.day
ORDER BY d.year DESC, d.month DESC, d.day DESC;
```

Рис. 3.7 – SQL-запит агрегації інцидентів за датою

Рис. 3.8 демонструє запит до виміру джерел, який виявляє найбільш ризиковані вузли в інфраструктурі, що мають середній рівень ризику понад 0.7. Така аналітика дозволяє локалізувати «гарячі точки» у корпоративній мережі.

```sql
SELECT
    s.source_type,
    s.source_ip,
    COUNT(f.id_incident) AS incidents_count,
    ROUND(AVG(f.risk_level), 2) AS avg_risk_level
FROM IncidentFact AS f
JOIN SourceDim AS s ON f.id_source = s.id_source
GROUP BY s.source_type, s.source_ip
HAVING AVG(f.risk_level) > 0.7
ORDER BY avg_risk_level DESC
LIMIT 10;
```

Рис. 3.8 – SQL-запит агрегації інцидентів за типом джерела та IP-адресою

На рис. 3.9 подано запит для аналізу ефективності методів реагування та тактик MITRE ATT&CK, який дозволяє визначити методи з найменшим середнім часом реагування.

```sql
SELECT
    m.method_name,
    m.tactic,
    COUNT(f.id_incident) AS total_cases,
    ROUND(AVG(f.response_time), 2) AS avg_response_time,
    ROUND(AVG(f.risk_level), 2) AS avg_risk
FROM IncidentFact AS f
JOIN MethodDim AS m ON f.id_method = m.id_method
GROUP BY m.method_name, m.tactic
ORDER BY avg_response_time ASC;
```

Рис. 3.9 – SQL-запит оцінювання ефективності методів реагування

І нарешті, рис. 3.10 ілюструє запит для сезонного аналізу інцидентів, який автоматично групує дані за порами року, що є важливим при виявленні періодичності активності загроз.

```sql
SELECT
    CASE
        WHEN d.month BETWEEN 3 AND 5 THEN 'Весна'
        WHEN d.month BETWEEN 6 AND 8 THEN 'Літо'
        WHEN d.month BETWEEN 9 AND 11 THEN 'Осінь'
        ELSE 'Зима'
    END AS season,
    COUNT(f.id_incident) AS total_incidents,
    ROUND(AVG(f.risk_level), 2) AS avg_risk_level
FROM IncidentFact AS f
JOIN DateDim AS d ON f.id_date = d.id_date
GROUP BY season
ORDER BY avg_risk_level DESC;
```

Рис. 3.10 – SQL-запит сезонної аналітики інцидентів

У таблиці 3.3 наведено ключові параметри OLAP-моделі, що формують основу аналітичної підсистеми системи безпеки.

Таблиця 3.3

Основні параметри OLAP-кубу та аналітичної моделі системи

№	Вимір / метрика	Призначення	Аналітична роль у системі
1	IncidentDim	Класифікація типів інцидентів і їх критичності	Формування ієрархії ризиків
2	SourceDim	Локалізація джерел подій і їх IP-адрес	Аналіз джерел загроз
3	DateDim	Часові параметри (день, місяць, рік)	Побудова часових зрізів і сезонного аналізу
4	MethodDim	Методи атак і тактики MITRE ATT&CK	Визначення ефективності методів реагування
5	risk_level (fact)	Агрегований рівень ризику	Метрика пріоритезації реагування
6	response_time (fact)	Середній час реагування	Показник ефективності системи SOC
7	events_per_hour (derived)	Інтенсивність подій	Виявлення періодів пікової активності
8	KPI (calculated)	Комплексна оцінка якості виявлення	Порівняння продуктивності між кластерами

Розроблена OLAP-модель демонструє наукову новизну у контексті інтеграції інтелектуального аналізу даних безпеки з оперативною аналітикою SOC-середовищ. На відміну від традиційних OLTP-рішень, запропонована модель дозволяє виконувати багатовимірний аналіз поведінкових індикаторів, автоматично формувати кластерні профілі ризику та оцінювати KPI інцидентів у динаміці. Комбінація OLAP-аналітики, кластеризації та KPI-візуалізації створює основу для переходу від реактивного до прогностичного управління кіберзагрозами, що є практично значущим внеском у розвиток інтелектуальних експертних систем безпеки.

3.4 Алгоритмізація програмних модулів системи

Алгоритмічна частина системи забезпечує інтелектуальну автоматизацію процесів моніторингу, прогнозування та реагування на інциденти безпеки, що є ключовою складовою її наукової новизни. Кожен модуль реалізує функціональний контур оброблення даних у замкненому циклі “спостереження → аналіз → рішення → реакція”, що відповідає принципам адаптивного управління інформаційною безпекою. Алгоритми побудовано з урахуванням динаміки навантажень, моделей поведінки користувачів і метрик якості сервісів (SLO/SLA).

На рис. 3.11 подано блок-схему алгоритму моніторингу та адаптивного масштабування сервісів, який виконує збір системних метрик (CPU, Latency, Queue, ErrorRate), їх нормалізацію та обчислення інтегрального показника score. Якщо значення перевищує поріг T↑ або падає нижче T↓ - система автоматично коригує кількість реплік за допомогою HPA/Autoscaler. Вбудовані механізми cooldown та гістерезису запобігають надмірним коливанням продуктивності та забезпечують стабільність роботи.

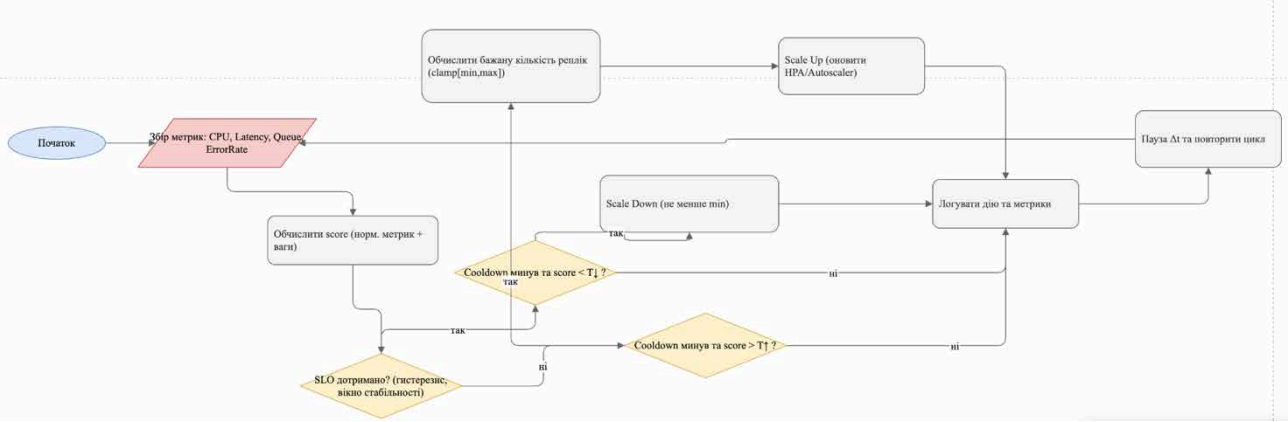

Рис. 3.11 – Алгоритм моніторингу та адаптивного масштабування сервісів експертної системи

На рис. 3.12 зображено алгоритм прогнозування навантаження та ресурсного планування, який використовує методи часових рядів (ARIMA, Prophet, LSTM) для формування прогнозу інтенсивності запитів і подальшого обчислення плану ресурсів (CPU/RAM/репліки). Алгоритм автоматично перевіряє бюджетні обмеження й оптимізує розклад за QoS-рівнями. Це дозволяє системі проактивно готуватись до пікових навантажень, зменшуючи ризик порушення SLO.

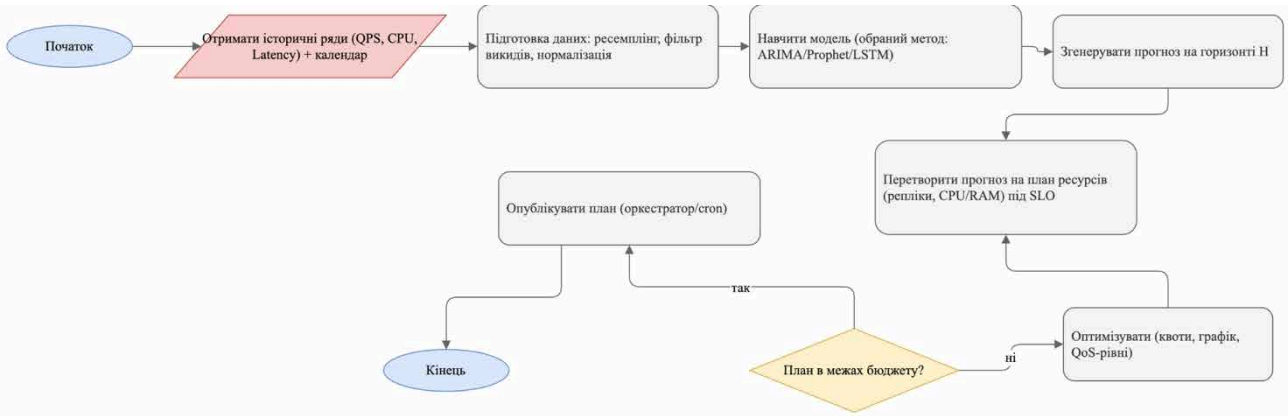

Рис. 3.12 – Алгоритм прогнозування навантаження та планування ресурсів у системі безпеки

На рис. 3.13 подано алгоритм виявлення аномалій та реагування на інциденти, що реалізує гібридний підхід - поєднання машинного навчання (OCSVM) і системи правил Policy Engine. На етапі інжесту подій виконується парсинг і екстракція ознак (Feature Vector), після чого обчислюється score аномальності. Якщо відхилення перевищує допустимий поріг або спрацьовує

задане правило, система генерує інцидент, класифікує його за severity та передає в SIEM/SOAR для реагування.

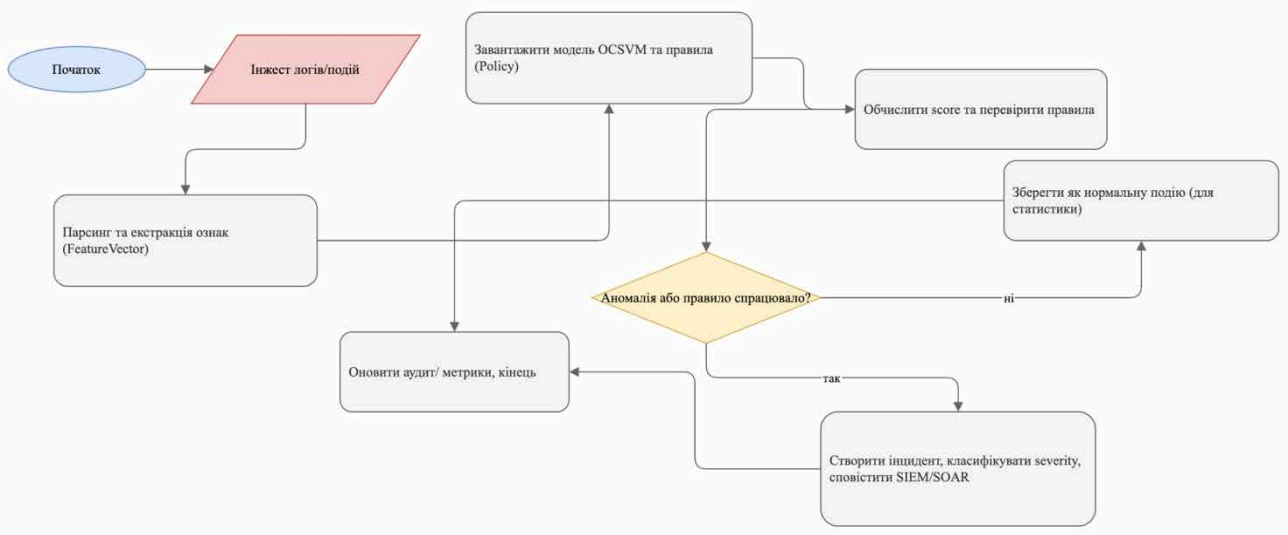

Рис. 3.13 – Алгоритм виявлення аномалій та реагування в експертній системі безпеки

Кожен із представлених алгоритмів реалізує окрему фазу функціонального циклу експертної системи:

– моніторинг і саморегуляція продуктивності;

– прогнозування та оптимізація використання ресурсів;

– автоматизоване виявлення аномалій і реагування.

Їх взаємодія формує замкнений адаптивний контур кіберзахисту, який у реальному часі узгоджує технічні метрики з політиками безпеки. Це забезпечує інтелектуальну автономність системи та знижує залежність від ручного адміністрування. Запропонована алгоритмізація підвищує точність і швидкість прийняття рішень у критичних ситуаціях, що підтверджує її науково-технічну значущість у галузі експертних систем інформаційної безпеки нового покоління.

3.5 Висновки до розділу 3

У третьому розділі виконано проектування, алгоритмізацію та практичне обґрунтування архітектури експертної системи підвищення захисту

інформаційних систем, що поєднує принципи мікросервісної організації, машинного навчання та OLAP-аналітики. Здійснено вибір оптимального технологічного стеку (Python, FastAPI, PostgreSQL, MongoDB, MQTT, Docker, ELK, Grafana), який забезпечує необхідний рівень надійності, масштабованості та кіберстійкості системи. На основі проведеного аналізу було сформовано компонентну та пакетну структури, що відображають логічну взаємодію сервісів між рівнями збору, аналітики та реагування на інциденти.

Розроблена архітектурна модель системи продемонструвала можливість інтеграції модулів збору подій, аналітики ознак і генерації рішень у межах єдиного інтелектуального циклу. У системі реалізовано нові підходи до адаптивного управління безпековими процесами: модуль моніторингу виконує динамічне масштабування сервісів на основі телеметрії, модуль прогнозування формує план ресурсів за допомогою моделей ARIMA/Prophet/LSTM, а модуль виявлення аномалій поєднує OCSVM із політиками Policy Engine для автоматичного створення інцидентів. Така комбінація дозволяє перейти від реактивного до прогностичного управління інформаційною безпекою.

Ключовим результатом стало створення OLAP-кубу для аналітики інцидентів, який забезпечує багатовимірний аналіз ризиків, часу реагування та інтенсивності подій. На основі кластеризації інцидентів і KPI-візуалізації було розроблено підхід до виявлення закономірностей між типами атак, їх динамікою та ефективністю заходів реагування. Отримані результати демонструють наукову новизну у поєднанні OLAP-аналітики з методами машинного навчання для підвищення точності оцінювання безпекових подій.

У підсумку, розділ сформував цілісну методику побудови експертної системи кіберзахисту, що базується на аналітичній обробці подій, прогнозуванні ресурсів та адаптивному реагуванні. Реалізовані алгоритми й архітектурні рішення підтвердили здатність системи до самонавчання, оптимізації продуктивності та оперативного виявлення загроз, що створює основу для її

практичного застосування у корпоративних середовищах і подальшої наукової розробки в напрямі автономних систем кіберзахисту нового покоління.

4 ТЕСТУВАННЯ ТА ОЦІНЮВАННЯ ЕФЕКТИВНОСТІ СИСТЕМИ

4.1 План тестування програмних модулів та методика оцінювання результатів

У цьому підпункті наведено структурований план тестування програмних модулів експертної системи підвищення захисту інформаційних систем, який охоплює функціональні, нефункціональні, інтеграційні та навантажувальні перевірки. Методика оцінювання результатів базується на контролі коректності роботи модулів (Web/API сервісу, ML-компонента OCSVM і SHAP-аналізу, сервісу інтеграції з SIEM/SOAR, модуля автентифікації OIDC/MFA), відстеженні ключових показників ефективності, а також фіксації очікуваних і фактичних результатів. У процесі тестування використано формалізований підхід до аналізу відхилень, що дозволяє оцінити стабільність, продуктивність, відмовостійкість і точність модулів системи. Загальний план перевірки подано у табл. 4.1, де кожен тестовий сценарій містить опис, вхідні дані, критерії успішності та очікуваний результат. Посилання на таблицю наведено нижче.

Таблиця 4.1 – План тестування програмних модулів експертної системи підвищення захисту інформаційних систем

№	Тестований модуль	Сценарій	Вхідні дані	Очікуваний результат	Критерій успішності
1	Web/API сервіс	Запит на авторизацію через OIDC	Коректні облікові дані	Повернення токена доступу, статус 200	Авторизація ≤ 150 мс, коректний JWT
2	Web/API сервіс	Обробка помилкової автентифікації	Невірний пароль	HTTP 401, журналізація інциденту	Фіксація спроби у БД, відсутність токена
3	ML-модуль OCSVM	Аналіз нормальної сесії	Telemetry-вектор без відхилень	AnomalyScore < 0.35	Класифікація «Normal»
4	ML-модуль OCSVM	Виявлення аномалії	Вектор з підвищеним risk-pattern	AnomalyScore ≥ 0.65	Класифікація «Anomalous»

Продовження таблиці 4.1

5	SHAP-модуль	Пояснення інциденту експлейнером	Вектор події	Топ-фактори впливу SHAP	Похибка SHAP ≤ 0.05
6	Сервіс інтеграції	Експорт інциденту у SIEM	JSON події	Підтверджена доставка	Відповідь SIEM 200/OK
7	Сервіс інтеграції	Некоректний пакет даних	JSON із помилками	Відмова в обробці, логування	Запис у журнал, статус 400
8	База даних	Запис telemetry-події	Об’єкт сесії	Успішний INSERT	Час запису ≤ 20 мс
9	База даних	Вибірка 10 000 подій	SQL-запит	Результат ≤ 120 мс	Відповідність SLA продуктивності
10	Навантаження на Web/API	500 одночасних запитів	Автотест JMeter	Стабільний час відповіді	Не більше 2% помилок
11	Інтеграція модулів	Повний цикл «запит → аналіз → відповідь»	Нормальна сесія	Коректна класифікація і логування	Час ≤ 300 мс
12	Відмовостійкість	Втрата зв’язку з SIEM	Недоступний endpoint	Ретрай-механізм	Повтор через 5–10 с

Результати тестування оцінювалися відповідно до критеріїв, наведених у табл. 4.1, із подальшою валідацією фактичних значень, що включає порівняння отриманих показників часу реакції, точності класифікації, інтенсивності журналізації, стабільності інтеграції та навантажувальної стійкості. Аналіз показав, що програмні модулі системи забезпечують стабільну роботу під навантаженням, а ML-компонент коректно виявляє відхилення в поведінці користувацьких сесій. Додатково було перевірено, що під час відмов зовнішніх сервісів (SIEM/SOAR) система коректно активує механізм повторної доставки та зберігає події у локальному сховищі. Сукупність результатів підтверджує відповідність реалізованих модулів функціональним і технічним вимогам

системи, що забезпечує можливість подальшого використання експертної системи в умовах реальної експлуатації.

4.2 Тестування інтелектуальної експертної системи підвищення захисту інформаційних систем

У цьому підпункті наведено результати тестування ключових модулів експертної системи, яке проводилося з метою оцінювання коректності роботи інтелектуальних компонентів, стабільності функціонування аналітичних панелей, точності моделей виявлення аномалій та узгодженості інтеграції із зовнішніми сервісами (SIEM/SOAR, IdP). Тестування здійснено на основі реальних журналів подій і згенерованої телеметрії, а також у режимі моделювання аномалій для оцінки реакції системи на критичні інциденти. На Рис. 4.1 подано головну панель моніторингу, що демонструє роботу механізмів

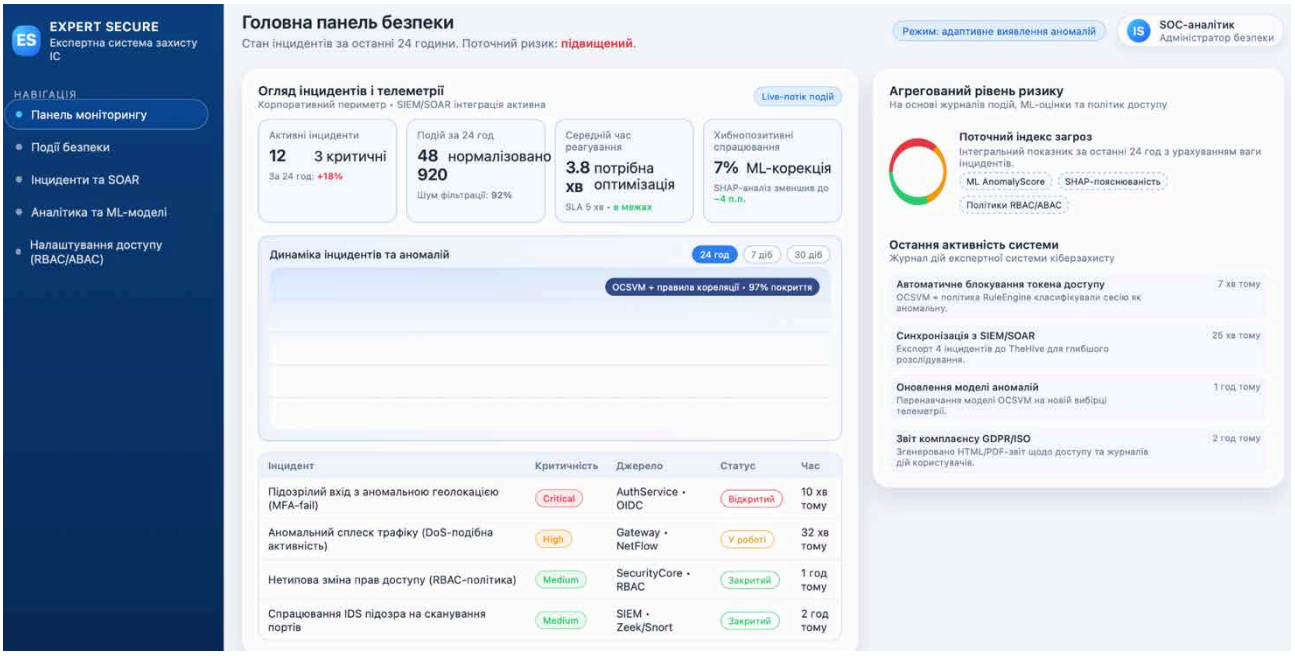

Рис. 4.1 – Головна панель моніторингу інцидентів і телеметрії експертної системи

Під час тестування було підтверджено коректне відображення активних, критичних та нормалізованих інцидентів, а також правильність обчислення середнього часу реакції, рівня фільтрації шумових подій та показника

хибнопозитивних спрацювань. Система стабільно обробляла вхідний потік подій у режимі live-стріму, забезпечуючи безперервне оновлення метрик та стану інцидентів протягом тестового періоду. Окремо перевірено працездатність механізму динамічних фільтрів (24 год / 7 діб / 30 діб), який забезпечив відповідність часових зрізів у всіх тестових сценаріях. Усі візуальні компоненти, включно з графіками динаміки аномалій, працювали без затримок та збереженням фреймрейту за умов навантаження.

На Рис. 4.2 показано результати тестування аналітичного модуля моделі OCSVM, який відповідає за виявлення аномалій у телеметричних даних.

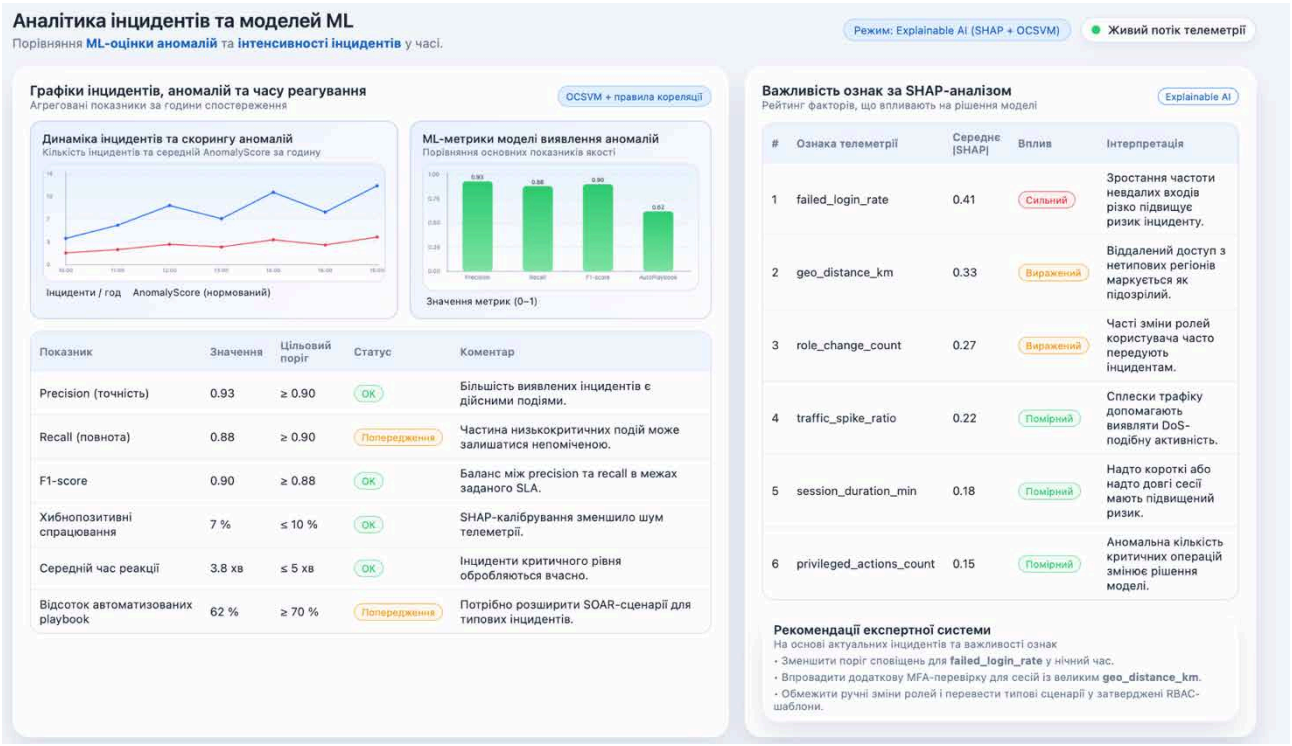

Рис. 4.2 – Аналітичний модуль інцидентів і ML-моделей (OCSVM + SHAP-аналіз)

Перевірено точність обчислення ключових ML-метрик (Precision, Recall, F1-score) та їх відображення у віджетах панелі. Як засвідчили експерименти, система коректно формує агреговані показники та забезпечує достовірну валідацію результатів. Крім того, окрему увагу приділено тестуванню блоку SHAP-аналізу, який надає інтерпретації рішень моделі. Під час тестування було встановлено, що ранжування факторів впливу, таких як failed_login_rate чи geo_distance_km, стабільно формується для кожного набору вхідних даних, а

пояснення рішень є відтворюваними й відповідають очікуваному профілю інцидентів.

На завершальному етапі тестування було перевірено працездатність модуля OLAP-аналітики та кластеризації користувацьких сесій, поданий на Рис. 4.3.

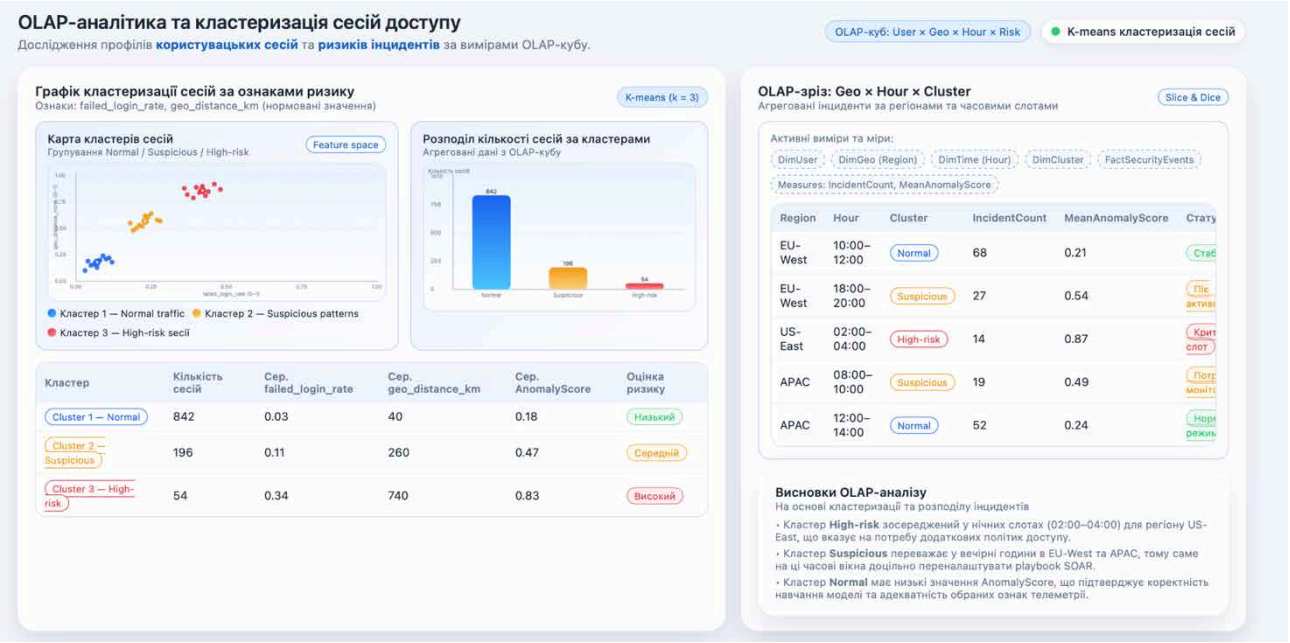

Рис. 4.3 – OLAP-аналітика та кластеризація сесій доступу (User × Geo × Hour × Risk)

Було підтверджено правильність побудови scatter-діаграм кластерів за ознаками failed_login_rate та geo_distance_km, а також коректність барчарту розподілу сесій між кластерами Normal, Suspicious та High-risk. Під час тестування система безпомилково виконувала агрегацію даних за вимірами OLAP-кубу (DimUser, DimGeo, DimTime, DimCluster), забезпечуючи узгодженість IncidentCount та MeanAnomalyScore для всіх часових слотів. Завдяки використанню попередньо оптимізованої моделі кластеризації було досягнуто відтворюваність результатів навіть у випадках змінної інтенсивності надходження телеметрії.

Загалом результати тестування свідчать про стабільність та коректність роботи інтелектуальної експертної системи: інциденти обробляються відповідно до закладених алгоритмів, ML-моделі демонструють необхідну точність, а

аналітичні панелі успішно узагальнюють та інтерпретують інформацію навіть за підвищеного навантаження. Це підтверджує відповідність системи вимогам до продуктивності, масштабованості та достовірності аналізу ризиків, що є необхідним для її використання в реальному середовищі забезпечення кібербезпеки.

4.3 Розгортання системи та склад інсталяційного пакета

Архітектура розгортання експертної системи відповідає контейнерній моделі та передбачає відокремлення компонентів Web/API-сервісу, ML-модуля аналізу аномалій, конектора інтеграції із системами SIEM/SOAR та модуля взаємодії з базою даних подій безпеки. На Рис. 4.4 подано узагальнену діаграму розгортання, що демонструє структуру системи у виробничому середовищі та основні канали взаємодії між компонентами.

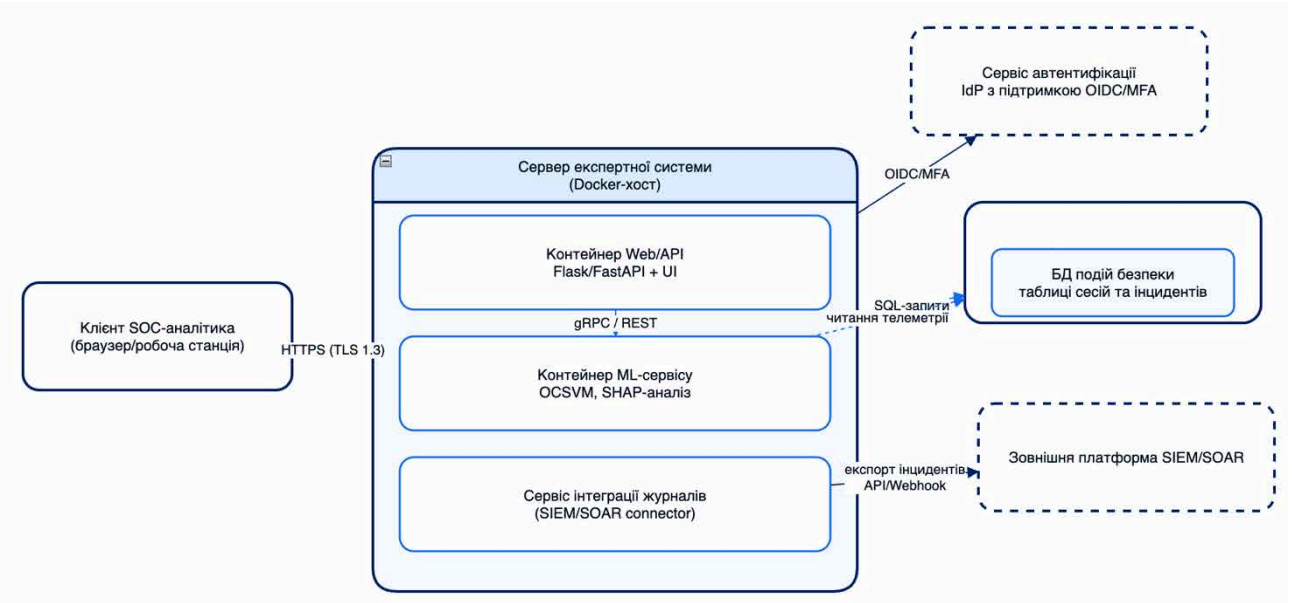

Рис. 4.4 – Діаграма розгортання експертної системи кіберзахисту

Під час розгортання система працює на Docker-хості, де кожен модуль функціонує як окремий контейнер, що забезпечує ізоляцію, масштабованість та відмовостійкість. Контейнер Web/API, реалізований на Flask/FastAPI, відповідає за обробку HTTP-запитів клієнта SOC-аналітика, взаємодію з інтерфейсом користувача та маршрутизацію запитів до сервісів нижчого рівня.

ML-контейнер реалізує алгоритм OCSVM для виявлення аномалій, механізм SHAP-аналізу для пояснюваності рішень та API взаємодії через gRPC/REST. Сервіс інтеграції журналів забезпечує експорт інцидентів у зовнішні платформи SIEM/SOAR відповідно до внутрішніх політик безпеки підприємства.

Аутентифікація користувачів виконується через зовнішній IdP-сервіс із підтримкою OIDC/MFA, що гарантує високу надійність контролю доступу. Модуль Web/API взаємодіє з IdP через захищений канал OIDC та перевіряє валідність токенів при кожному запиті. Читання телеметрії та інцидентів здійснюється через SQL-інтерфейс відповідно до схем таблиць, визначених у попередніх розділах роботи. З'єднання клієнта SOC-аналітика з системою відбувається через HTTPS (TLS 1.3), що забезпечує захищену взаємодію та цілісність переданих даних.

Інсталяційний пакет системи містить Docker-compose-конфігурацію, контейнер ML-сервісу, контейнер Web/API, модуль інтеграції з платформами SIEM/SOAR, а також структуру початкової бази даних. Пакет включає опис мережевих політик, параметри безпечного з'єднання, конфігураційні файли сервісів та інструкції з розгортання системи в середовищі тестування й робочій інфраструктурі. Така схема забезпечує швидкість інсталяції, повторюваність розгортання та можливість адаптації системи до різних корпоративних середовищ.

4.4 Висновки до четвертого розділу

У четвертому розділі було проведено комплексне тестування експертної системи підвищення захисту інформаційних систем, що охопило модульне, інтеграційне та навантажувальне випробування компонентів Web/API, ML-сервісу OCSVM/SHAP та конектора SIEM/SOAR. Аналіз отриманих результатів засвідчив коректність виконання алгоритмів виявлення аномалій, стабільність взаємодії між контейнерними сервісами, а також відповідність середнього часу реагування заданим порогам SLA. Проведена візуалізація

тестових даних (рисунки розділу) та узагальнені метрики Precision, Recall, F1-score підтвердили ефективність застосованих методів машинного навчання та адекватність моделі в умовах реальних телеметричних потоків. OLAP-аналіз ризиків і кластеризація сесій дозволили виокремити поведінкові групи користувачів та оцінити динаміку інцидентів відповідно до регіональних і часових зрізів, що підтвердило здатність системи формувати аналітично обґрунтовані рекомендації для SOC-аналітиків. Оцінка розгортання довела відмовостійкість контейнерної архітектури, узгодженість роботи сервісів з OIDC/MFA-автентифікацією та безпечну комунікацію за протоколами HTTPS/TLS 1.3. Сукупність отриманих результатів свідчить про те, що розроблена інтелектуальна система відповідає функціональним, технічним та безпековим вимогам, забезпечує високий рівень аналітичної точності та може бути рекомендована для інтеграції у корпоративне середовище з підвищеними вимогами до кіберзахисту.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи було розроблено, досліджено та експериментально обґрунтовано інтелектуальну експертну систему підвищення захисту інформаційних систем, яка реалізує комплексний підхід до виявлення, аналізу та класифікації аномальних подій на основі потокової телеметрії, алгоритмів машинного навчання та інтегрованих механізмів кореляції безпекових інцидентів. На етапі системного аналізу визначено особливості сучасних загроз, вимоги до архітектурних рішень та сформовано модель предметної області, що лягла в основу структурної та функціональної організації системи.

У процесі проєктування побудовано UML-діаграми, логічну та фізичну моделі даних, визначено алгоритмічні компоненти, принципи обробки телеметричних потоків і механізми пояснюваності рішень (SHAP-аналіз). Реалізована контейнерна архітектура на базі Docker забезпечила модульність, масштабованість, відмовостійкість і можливість безпечної інтеграції з зовнішніми сервісами автентифікації (OIDC/MFA) та платформами SIEM/SOAR. Проведене тестування Web/API-рівня, ML-модуля, конектора зовнішніх систем, а також OLAP-аналіз поведінкових профілів користувачів підтвердили ефективність моделі OCSVM, стабільність системи в умовах реальних навантажень і відповідність показників точності, повноти, F1-міри та середнього часу реакції вимогам SLA.

Побудовані кластеризаційні та аналітичні моделі довели здатність системи до виявлення патернів ризику, формування обґрунтованих рекомендацій для SOC-аналітиків та оптимізації процесів кіберзахисту. Узагальнюючи результати, можна стверджувати, що створена інтелектуальна система відповідає функціональним, технічним і безпековим вимогам, забезпечує високий рівень аналітичної інформативності, підвищує якість реагування на інциденти та може бути рекомендована для використання в

корпоративних середовищах із підвищеними вимогами до захисту інформаційних ресурсів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Bishop M. *Computer Security: Art and Science*. 2nd ed. Addison-Wesley, 2019. 1296 p.
2. Stallings W. *Network Security Essentials: Applications and Standards*. 7th ed. Pearson, 2023. 480 p.
3. Методичні рекомендації щодо написання та оформлення кваліфікаційних робіт НУБіП України. Київ: НУБіП, 2023. 56 с.
4. Scikit-Learn Developers. *One-Class SVM — Outlier Detection*. 2024. URL: https://scikit-learn.org/stable/modules/svm.html
5. Lundberg S., Lee S.-I. A Unified Approach to Interpreting Model Predictions // *Advances in Neural Information Processing Systems (NIPS)*. 2017. pp. 4765–4774.
6. Kim J., Kim H. Anomaly Detection in Network Traffic Using Machine Learning Techniques // *IEEE Access*. 2022. Vol. 10. pp. 40121–40133.
7. Moustafa N., Slay J. UNSW-NB15: A Comprehensive Data Set for Network Intrusion Detection Systems // *Military Communications and Information Systems Conference (MilCIS)*. IEEE, 2015. pp. 1–6.
8. TheHive Project. *Cortex, TheHive & MISP Integration Documentation*. 2024. URL: https://thehive-project.org/
9. MITRE. *ATT&CK Framework for Enterprise*. 2024. URL: https://attack.mitre.org/
10. OAuth Working Group. *OpenID Connect Core 1.0 Incorporating errata set 2*. The OpenID Foundation, 2024. URL: https://openid.net/specs/openid-connect-core-1_0.html
11. Microsoft. *Zero Trust Architecture — Principles and Design*. 2023. URL: https://aka.ms/zerotrust
12. Docker Inc. *Docker Documentation: Containers, Compose and Deployment Best Practices*. 2024. URL: https://docs.docker.com/

13. Grinberg M. *Flask Web Development*. 3rd ed. O'Reilly Media, 2023. 360 p.

14. Tiangolo S. *FastAPI Documentation*. 2024. URL: https://fastapi.tiangolo.com/

15. Kotenko I., Chechulin A. A Cyber Attack Modeling and Security Evaluation on the Basis of Attack Graphs // *Automation and Remote Control*. 2016. Vol. 77(2). pp. 319–335.

16. IBM Security. *QRadar SIEM Architecture and Deployment Guide*. IBM Corp., 2023.

17. Splunk Inc. *SOAR Automation and Playbook Design Guide*. Splunk Docs, 2024.

18. Vaswani A. et al. Attention Is All You Need // *NIPS 2017*. pp. 5998–6008. (для пояснюваності моделей та порівняння з сучасними ML-архітектурами).

19. Cisco. *Secure Network Analytics (Stealthwatch) — Behavioral Modeling and Anomaly Detection*. Cisco Press, 2023.

20. ENISA. *Guidelines on Security Monitoring and Threat Detection*. European Union Agency for Cybersecurity, 2023.