Хмельницький національний університет
Факультет інформаційних технологій
Кафедра кібербезпеки

КВАЛІФІКАЦІЙНА РОБОТА

Навроцької Катерини Вадимівни

на здобуття ступеня вищої освіти Бакалавра

Система виявлення облікових записів з аномальною активністю

Галузь знань 12 – Інформаційні технології
Спеціальність 125 – Кібербезпека
Освітня програма Кібербезпека

КРБКБ.220248.22.02.32 ПЗ

Виконав студентка 4 курсу група КБ-22-2 Катерина НАВРОЦЬКА
Керівник д-р філософії Наталія ПЕТЛЯК
Нормоконтролер д-р філософії Наталія ПЕТЛЯК

До захисту допускаю:
Завідувач кафедри кібербезпеки Юрій КЛЬОЦ

3 06 2026 р.

Хмельницький 2026

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет Інформаційних технологій
Кафедра Кібербезпеки
Рівень вищої освіти Бакалавр
Галузь знань 12 – Інформаційні технології
Спеціальність 125 – Кібербезпека
Освітня програма Кібербезпека

ЗАТВЕРДЖУЮ
Завідувач кафедри кібербезпеки
Юрій КЛЬОЦ
9 січня 2026 р.

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Навроцькій Катерині Вадимівні

1 Тема роботи Система виявлення облікових записів з аномальною активністю

Керівник роботи доктор філософії Наталія ПЕТЛЯК
Затверджено наказом ректора університету від 8 січня 2026 р. № 7
2 Строк подання студентом кваліфікаційної роботи на кафедру 27 травня 2026р.
3 Вихідні дані до роботи Проаналізувати особливості аномальної активності облікових записів в інформаційних системах, зокрема типи аномалій поведінки користувачів, характеристики подій доступу, журнали подій та показники активності. Дослідити та порівняти існуючі методи і підходи до виявлення аномальної активності, проаналізувати їх переваги та недоліки. Сформувати вимоги до системи виявлення облікових записів з аномальною активністю. Розробити алгоритм роботи системи та виконати її моделювання. Обґрунтувати вибір методів та засобів аналізу. Провести експериментальну перевірку працездатності запропонованого підходу та оцінити його ефективність на тестових наборах даних.
4 Зміст пояснювальної записки (перелік питань, які потрібно розробити) Вступ. Аналіз проблеми аномальної активності облікових записів в інформаційних системах. Класифікація аномалій поведінки користувачів та загроз, пов'язаних з компрометацією облікових записів. Аналіз існуючих методів і систем виявлення аномальної активності облікових записів. Постановка задачі. Формування вимог до системи виявлення. Вибір методів та засобів аналізу і моделювання. Розробка алгоритму роботи системи виявлення облікових записів з аномальною активністю. Розробка прототипу системи виявлення. Експериментальне дослідження працездатності та оцінка ефективності запропонованого рішення. Висновки.
5 Перелік графічного матеріалу (із зазначенням обов'язкових креслень) Загальна структурна схема системи виявлення облікових записів з аномальною активністю. Алгоритм функціонування системи виявлення аномальної активності облікови записів. Схема процесу аналізу подій та прийняття рішення щодо аномальної активності.

6 Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7 Дата видачі завдання 12 січня 2026 р.

КАЛЕНДАРНИЙ ПЛАН

Назва етапів (розділів) кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
Вибір і затвердження теми кваліфікаційної роботи	Січень-Лютий	
Ознайомлення з предметною областю	Лютий	
Дослідження існуючих рішень	Лютий	
Постановка задачі	Березень	
Визначення загальних принципів рішення задачі	Березень	
Деталізація принципів рішення задачі	Квітень	
Розробка проєктних рішень	Квітень	
Апробація проєктних рішень	Травень	
Оформлення пояснювальної записки згідно вимог	Травень	
Оформлення графічної частини	Травень	
Захист КР	Червень	

Студентка _____ Катерина НАВРОЦЬКА

Керівник кваліфікаційної роботи _____ Наталія ПЕТЛЯК

АНОТАЦІЯ

Тема кваліфікаційної роботи: Система виявлення облікових записів з аномальною активністю.

Автор роботи: Навроцька Катерина Вадимівна.

Керівник роботи: доктор філософії, Петляк Наталія Сергіївна.

Загальний обсяг роботи: 69 сторінок, 9 рисунків, 1 таблиця, 3 формули, 3 додатока, 47 посилання.

Графічна частина: 3 плакати.

Ключові слова: аномальна активність, обліковий запис, поведінкова аналітика, метрика Махаланобіса, журнали подій, інформаційна безпека, виявлення вторгнень.

Кваліфікаційна робота, присвячена розробці системи аналітики поведінки для виявлення аномальної активності та компрометації облікових записів в інформаційних системах.

В роботі, на основі аналізу особливостей поведінки користувачів та існуючих методів виявлення кіберзагроз, реально класифікованих аномалій поведінки та ключових показників активності на основі журналів подій, пропонується використовувати методи багатовимірної статистики для ідентифікації відхилень, розроблено математичне ядро системи на базі метрики Махаланобіса із застосуванням логарифмічного згладжування, а також реалізовано програмний прототип у вигляді інтерактивної веб-додатки та проведено його експериментальну перевірку на тестових наборах даних.

25.05.2026 _____

ANNOTATION

Theme of qualification work: System for detecting user accounts with anomalous activity.

Author of the work: Kateryna V. Navrotska.

Mentor: Ph.D. Nataliia S. Petliak.

Total volume of work: 69 pages, 9 figures, 1 table, 3 formulas, 3 appendices, 47 links.

Graphic part: 3 posters.

Keywords: anomalous activity, user account, behavioral analytics, Mahalanobis metric, event logs, information security, intrusion detection.

The qualification work is devoted to the development of a behavioral analytics system for detecting anomalous activity and compromise of user accounts in information systems.

In this work, based on the analysis of user behavior characteristics and existing methods for detecting cyber threats, as well as the classification of actual behavioral anomalies and key activity indicators derived from event logs, it is proposed to use multivariate statistical methods to identify deviations. The mathematical core of the system was developed based on the Mahalanobis metric using logarithmic smoothing. Additionally, a software prototype was implemented as an interactive web application, and its experimental verification was conducted on test datasets.

25.05.2026 _____

ЗМІСТ

ВСТУП

Сучасний етап стрімкого розвитку інформаційних технологій, що характеризується тотальною цифровізацією та масовим переходом до віддаленої роботи, призвів до безпрецедентного збільшення кібератак. Класичні засоби захисту периметра втрачають ефективність, оскільки зловмисники все частіше використовують легітимні, але скомпрометовані облікові записи або діють через інсайдерів. Виникає критична проблема: як відрізнити нормальні дії авторизованого співробітника від прихованої активності хакера. Більшість корпоративних інфраструктур покладаються на системи управління подіями безпеки, проте їхні жорсткі статичні правила генерують колосальну кількість хибних спрацьовувань. З іншого боку, впровадження ресурсоємних нейронних мереж стикається з проблемою чорного ящика, вимагаючи значних обчислювальних ресурсів без можливості математичної інтерпретації рішень. Відповідно, нерозв'язаним залишається завдання створення легкого, прозорого інструменту для виявлення багатовимірних аномалій у режимі реального часу.

Актуальність кваліфікаційної роботи полягає у необхідності розробки систем поведінкової аналітики користувачів і сутностей на базі методів багатовимірної статистики. Використання метрики Махаланобіса є надзвичайно актуальним, цей апарат враховує не лише абсолютні відхилення, але й дисперсію та кореляційні взаємозв'язки між атрибутами сесії. Такий підхід забезпечує ідеальний баланс між високою точністю виявлення загроз та абсолютною прозорістю прийняття рішень для аналітиків безпеки.

Метою є розробка програмного прототипу системи виявлення аномальної активності користувачів на основі багатовимірного статистичного аналізу для проактивного захисту мереж від інсайдерських загроз та компрометації акаунтів. Система призначена для автоматизації моніторингу, зниження навантаження на аналітиків шляхом фільтрації хибних спрацьовувань та надання візуального інструментарію для розслідування інцидентів.

Досягнення мети здійснюється шляхом вирішення наступних завдань. По-перше, виконати аналіз існуючих методів та архітектурних підходів до побудови

систем поведінкової аналітики. По-друге, встановити особливості предметної області та обґрунтувати вибір метрики Махаланобіса як математичного ядра. По-третє, розробити структурну схему, алгоритм функціонування та математичну модель нормалізації ризиків із застосуванням логарифмічного згладжування та псевдообернених матриць. По-четверте, виконати програмну реалізацію алгоритмів у вигляді інтерактивного веб-додатка мовою Python. Насамкінець, провести апробацію та тестування прототипу на синтетичних і реальних масивах даних для підтвердження його ефективності. Об’єктом дослідження є процеси моніторингу та аналізу поведінки користувачів у корпоративних інформаційних мережах. Предметом дослідження виступають моделі, методи та алгоритми виявлення аномальної активності облікових записів на основі багатовимірного статистичного аналізу. Для розв’язання поставлених завдань використано методи системного аналізу, багатовимірної математичної статистики, лінійної алгебри, об'єктно-орієнтованого програмування та математичного моделювання. Наукова новизна одержаних результатів полягає в удосконаленні підходу до виявлення аномальної активності облікових записів шляхом адаптації метрики Махаланобіса із застосуванням логарифмічного згладжування та псевдообернених матриць.

Галузь застосування результатів охоплює операційні центри кібербезпеки, відділи захисту інформації та державні організації. Призначення розробки полягає у її використанні як самостійного інструменту аудиту або додаткового аналітичного модуля, інтегрованого в існуючі SIEM-системи для інтелектуального виявлення нетипової поведінки користувачів.

1 АНАЛІЗ МЕТОДІВ ТА ЗАСОБІВ ВИЯВЛЕННЯ АНОМАЛЬНОЇ АКТИВНОСТІ ОБЛІКОВИХ ЗАПИСІВ

1.1 Роль облікових записів у безпеці сучасних інформаційних систем

У сучасних інформаційно-телекомунікаційних системах (ІТС) обліковий запис вийшов за межі простого набору автентифікаційних даних, перетворившись на повноцінний цифровий ідентифікатор користувача або автоматизованого сервісу. Обліковий запис функціонує як універсальний ключ до цифрових активів організації, регламентуючи взаємодію суб'єкта з базами даних, корпоративними додатками та обчислювальними ресурсами [1].

Ефективність виявлення аномалій безпосередньо залежить від чіткої класифікації облікових записів. Кожен тип ідентифікатора визначає межі легітимної активності та рівень доступу до ресурсів. Системні записи для автономної роботи служб та адміністративні акаунти формують найбільш критичний сегмент інфраструктури. Натомість стандартні й гостьові записи забезпечують виконання рутинних завдань з обмеженими правами. Розподіл на локальні та мережеві (доменні) записи регламентує територіальні межі доступу, а віддалені облікові записи потребують впровадження додаткових заходів контролю та аудиту сесій через підвищений ризик зовнішньої компрометації. Суворе дотримання політик налаштування та ієрархічного керування правами доступу є необхідною умовою для мінімізації поверхні атаки та створення базису для ідентифікації поведінкових суб'єктів в інформаційному середовищі [1, 2].

Фундаментом управління доступом до облікових записів є базова концепція інформаційної безпеки AAA (Authentication, Authorization, Accounting), яка охоплює три нерозривні процеси [3]:

– автентифікація (authentication) – процедура підтвердження цифрової особистості користувача (наприклад, шляхом перевірки пароля, біометрії або апаратного токена)[4];

– авторизація (authorization) – процес визначення та надання суб'єкту

відповідних прав та привілеїв для виконання певних дій у системі після успішної автентифікації [4];

– аудит та облік (accounting/audit) – безперервна реєстрація та відстеження дій користувача під час його роботи в системі, що є критично важливим для подальшого ретроспективного аналізу інцидентів та виявлення аномалій [4].

За останнє десятиліття парадигма захисту корпоративних мереж зазнала кардинальних змін. Традиційна модель мережевої безпеки, відома як замок і рів (англ. castle-and-moat), спиралася на захист периметра за допомогою міжмережевих екранів та систем виявлення вторгнень. Проте масовий перехід до хмарних технологій, розподілених інфраструктур (SaaS, IaaS) та моделей віддаленої роботи (BYOD) призвів до розмиття класичного мережевого периметра. Новим напрямком безпеки стала сама ідентичність. У разі компрометації облікового запису зловмисник отримує легітимний доступ до системи, що робить традиційні засоби мережевого захисту неефективними, тому що атака відбувається зсередини довіреного середовища [5, 6]. За таких обставин класичні статичні критерії оцінювання трафіку поступаються місцем концептуально новому підходу – динамічному аналізу дій кожного конкретного суб'єкта інформаційної взаємодії.

Ця трансформація зумовила перехід до сучасної архітектури безпеки – Zero Trust (Нульова довіра). Основний принцип цієї концепції: "ніколи не довіряй, завжди перевіряй". У моделі Zero Trust жоден користувач, пристрій чи мережевий вузол не вважається довіреним за замовчуванням, навіть якщо він знаходиться всередині корпоративної мережі. Кожен запит на доступ до ресурсів вимагає суворої перевірки автентичності та авторизації, що робить моніторинг поведінки облікових записів та своєчасне виявлення аномалій головним інструментом захисту від кіберзагроз [7]. Безперервна оцінка контекстуальних маркерів та ідентифікація нетипових відхилень у транзакціях стають фундаментальним базисом для ухвалення оперативних рішень щодо надання або блокування доступу до критичних інфраструктурних ресурсів.

1.2 Класифікація загроз, пов’язаних із компрометацією облікових даних

Компрометація облікових даних є однією з найпоширеніших причин порушення конфіденційності, цілісності та доступності сучасних інформаційних систем. Зловмисники використовують широкий спектр технічних та нетехнічних методів для отримання несанкціонованого доступу до автентифікаційної інформації. Безпека змістилась безпосередньо до ідентифікатора користувача – захист від захоплення акаунтів стає пріоритетним завданням. Основні вектори атак, спрямовані на заволодіння обліковими записами, доцільно класифікувати за чотирма основними напрямками [8].

Атаки на основі підбору паролів та використання словників – це клас загроз, що об'єднує методи, спрямовані на вгадування або автоматизований підбір секретної фрази користувача. Найпростішим варіантом є прямий повний перебір, за якого система атакуючого генерує всі можливі комбінації символів. Словникові атаки переслідують організації по всьому світу. Навіть великі компанії, як Twitter (2009), LinkedIn (2012), Ashley Madison (2015), постраждали від цього типу кібератак. Проте внаслідок впровадження політик блокування після кількох невдалих спроб та вимог до складності паролів, прямий перебір втрачає свою ефективність [9].

Натомість зловмисники активно застосовують більш інтелектуальні підходи, словникові атаки, які використовують заздалегідь підготовлені списки найпоширеніших паролів, слів та фраз, що значно скорочує час на підбір. Іншим поширеним методом є підстановка облікових даних, що базується на використанні баз, які були викрадені під час зламів інших сервісів. У цьому випадку спеціалізоване програмне забезпечення масово перевіряє скомпрометовані пари логінів та паролів на корпоративних ресурсах, розраховуючи на те, що користувач нехтує правилами безпеки та застосовує однакові дані для доступу до різних систем. Щоб уникнути спрацьовування систем блокування облікових записів, атакуючі вдаються до методу розпилення

паролів. Його суть полягає в тому, що зловмисники беруть один або кілька дуже популярних паролів і намагаються застосувати їх до великого списку різних імен користувачів. Таким чином, на кожен акаунт припадає лише одна-дві спроби входу, що дозволяє непомітно обійти стандартні правила корпоративної безпеки [10].

Фішингові атаки є видом мережевого шахрайства, головною метою якого є обман користувача для добровільної передачі ним своїх автентифікаційних даних. Зловмисники створюють підроблені вебсторінки, які візуально повністю ідентичні легітимним сторінкам авторизації корпоративних ресурсів, банківських установ, поштових або хмарних сервісів. Жертва отримує електронний лист із закликом терміново оновити пароль або перевірити підозрілу активність, переходить за шкідливим посиланням і самостійно вводить свої дані, передаючи їх атакуючій стороні [11].

Окрему небезпеку становить цільовий фішинг. На відміну від масових розсилок, ця атака готується індивідуально для конкретного працівника компанії (наприклад, системного адміністратора або фінансового директора). Повідомлення містить персональні дані жертви, що значно підвищує рівень довіри та ймовірність успішної компрометації цільового облікового запису з високими привілеями.

Соціальна інженерія, на відміну від суто технічних зламів, базується на психологічному маніпулюванні слабкостями людського фактора для подолання систем кібербезпеки. Зловмисники експлуатують такі людські емоції, як страх, довіра, бажання допомогти або повага до авторитету [12].

Часто атакуючі видають себе за представників служби технічної підтримки, вище керівництво або довірених партнерів організації. Створюючи штучне відчуття терміновості або критичної ситуації, вони змушують працівника продиктувати коди доступу, тимчасово відключити антивірусний захист, самостійно скинути поточний пароль або підтвердити фальшивий запит на багатофакторну автентифікацію. Такі атаки є вкрай складними для виявлення традиційними технічними засобами, оскільки всі дії виконуються руками

легітимного користувача.

Внутрішні загрози – це категорія, що охоплює інциденти, пов'язані з діями осіб, які вже мають легітимний доступ до інформаційної системи компанії. Внутрішні загрози є одними з найнебезпечніших, оскільки зловмиснику не потрібно долати зовнішній периметр захисту. Їх прийнято поділяти на навмисні та ненавмисні. Навмисна компрометація часто відбувається, коли незадоволений або підкуплений працівник свідомо передає свій доступ третім особам. Іншим варіантом є зловживання привілеями для крадіжки конфіденційних даних, саботажу чи встановлення шкідливого програмного забезпечення. З іншого боку, ненавмисні загрози виникають через недбалість персоналу та загальний низький рівень культури кібербезпеки. Прикладами таких інцидентів є збереження паролів у відкритому вигляді на робочому столі, передача своїх облікових даних колегам для тимчасової роботи, запуск неперевірених файлів або використання інфікованих особистих пристроїв для доступу до корпоративної мережі [13].

Успішна реалізація будь-якої з наведених загроз призводить до того, що порушник починає діяти в інформаційній системі від імені легітимного суб'єкта. Оскільки система розпізнає вхід як правильний, традиційні засоби захисту не фіксують факту зламу. Це формує гостру необхідність у впровадженні систем поведінкового аналізу, які здатні виявляти компрометацію не за фактом введення пароля, а за аномальним характером подальших дій облікового запису.

1.3 Види аномальної поведінки користувачів

Фундаментальною гіпотезою виявлення скомпрометованих облікових записів є твердження про те, що легітимний користувач діє в межах певних, історично сформованих патернів. Формування такого профілю нормальної поведінки, або так званої базової лінії, відбувається шляхом тривалого збору метрик про щоденну активність користувача [14]. Водночас враховується не лише індивідуальна історія працівника, але й поведінка його колег зі схожими посадовими обов'язками (аналіз на рівні групи користувачів) [15]. Відповідно,

будь-яка суттєва девіація від цього динамічного профілю або встановлених корпоративних політик розглядається як аномалія. Для ефективного математичного моделювання та розробки архітектури системи виявлення, усі можливі відхилення зазвичай розподіляють на кілька взаємопов'язаних категорій.

Перш за все, варто виділити просторово-часові відхилення. З точки зору часу, система фіксує аномалію, коли активність облікового запису не збігається з типовим робочим графіком співробітника або загальним ритмом життя організації. Наприклад, якщо успішна авторизація до фінансових баз даних відбувається глибокої ночі або у вихідний день, і це не обумовлено службовою необхідністю чи закриттям звітного періоду, така подія вимагає негайної реакції. Важливим є те, що часова аномалія завжди розглядається в контексті ролі: нічний вхід системного адміністратора під час чергування є нормою, тоді як ідентична дія з боку працівника відділу кадрів є критичним інцидентом [14, 15].

Тісно пов'язаними з часовими є географічні аномалії. Вони виникають при фіксації нетипового мережевого або фізичного розташування пристрою ініціатора доступу. Найяскравішим індикатором тут виступає так звана неможлива подорож – ситуація, коли між двома успішними входами в систему з різних точок світу минуло менше часу, ніж фізично потрібно для переміщення між цими локаціями. Окрім прямої географічної невідповідності, до цієї категорії відносять спроби доступу через публічні проксі-сервери, вузли мережі Tor або несанкціоновані VPN-сервіси, які зловмисники використовують для маскування свого реального місцезнаходження та обходу територіальних обмежень.

Інший патерн поведінкового аналізу охоплює характер та обсяг виконуваних операцій. Кількісні девіації проявляються у вигляді різкої зміни інтенсивності дій порівняно з базовою лінією користувача. Це може бути серія невдалих спроб введення пароля, що свідчить про брутфорс-атаку, або ж раптовий експорт великих масивів інформації працівником, чия звичайна рутина обмежується переглядом кількох текстових документів. Сучасні зловмисники,

намагаючись обійти спрацьовування простих порогових правил, часто використовують тактику низької та повільної активності. Вони вивантажують конфіденційні файли невеликими частинами протягом тривалого часу. Виявити таку розтягнену в часі кількісну аномалію можливо лише за умови безперервного порівняння поточного трафіку з історичним профілем користувача за попередні місяці [16].

Натомість, логічні аномалії фіксують зміну самого контексту роботи та методів взаємодії з системою. Вони виявляються тоді, коли користувач починає виконувати нетипові для його посади команди, звертатися до ресурсів, до яких раніше не виявляв інтересу, або намагається отримати доступ до системних файлів [15]. Особливу небезпеку становить горизонтальне переміщення – етап кібератаки, під час якого зловмисник, захопивши один рядовий обліковий запис, починає сканувати внутрішню мережу та підключатися до інших робочих станцій чи серверів. Наприклад, звичайний офісний працівник несподівано запускає мережеві утиліти чи скрипти для збору інформації про архітектуру домену. Така поведінка майже стовідсотково вказує на компрометацію акаунта з метою подальшої ескалації привілеїв.

Лише багатовимірна кореляція та спільний аналіз усіх цих метрик дає змогу побудувати надійну систему виявлення. Окремо взята аномалія може бути наслідком цілком легітимної зміни робочого процесу. Проте комплексний розгляд часових, просторових, кількісних та логічних маркерів дозволяє мінімізувати кількість хибних спрацьовувань. Це дає змогу своєчасно блокувати кіберінциденти на ранніх стадіях їх розвитку [16].

1.4 Огляд показників активності та характеристики подій доступу

Розробка ефективної системи виявлення аномальної активності неможлива без якісного збору та попередньої обробки вхідних даних. Основним джерелом інформації для побудови поведінкових профілів є цифрові сліди, які користувач або автоматизований сервіс залишає під час взаємодії з ІТС. Для забезпечення

комплексного аналізу система повинна агрегувати дані з різних рівнів IT-інфраструктури.

У більшості корпоративних мереж, побудованих на базі рішень від Microsoft, ключовим джерелом даних для аудиту безпеки є журнали операційної системи (Windows Event Logs) [17]. Аналіз цих журналів, підсистеми Security, дозволяє детально відстежувати процеси автентифікації, зміни прав доступу та взаємодію з критичними об'єктами. Кожна подія в середовищі Windows фіксується під унікальним ідентифікатором (Event ID), що значно спрощує машинну обробку та парсинг логів. Основні ідентифікатори подій доступу, які є найбільш інформативними для виявлення компрометації облікових записів, наведено у таблиці 1.1.

Таблиця 1.1 – Основні ідентифікатори подій доступу в середовищі Windows Event Logs[17]

Ідентифікатор події	Назва події в системі	Характеристика та значення для аналізу аномалій
4624	Успішний вхід у систему (Logon)	Використовується для фіксації легітимних сесій, побудови базової лінії поведінки, визначення типового часу активності та робочих станцій, з яких користувач зазвичай працює.
4625	Невдала спроба входу (Logon Failure)	Є найважливішим кількісним маркером для виявлення атак повного перебору, розпилення паролів та автоматизованої підстановки вкрадених облікових даних.
4634	Завершення сеансу (Logoff)	Разом із подією успішного входу дозволяє системі аналітики вирахувати точну тривалість робочої сесії та виявляти аномалії, пов'язані з нетипово довгим або коротким часом перебування в системі.
4648	Вхід з використанням явних даних(Logon with explicitcredentials)	Допомагає виявляти спроби горизонтального переміщення мережею та запуск процесів від імені інших користувачів, що є типовою поведінкою зловмисника всередині периметра.
4672	Призначення спеціальних прав (Special Privileges Assigned)	Дозволяє своєчасно виявити логічні аномалії, пов'язані з несанкціонованим підвищенням привілеїв облікового запису до рівня адміністратора системи під час авторизації.
4720	Створення нового облікового запису (A user account was created)	Критичний маркер для виявлення дій зловмисника або недобросовісного інсайдера зі створення тіньових акаунтів (бекдорів) для довгострокового закріплення у скомпрометованій інфраструктурі.
4728 / 4732	Додавання користувача	Вказує на безпосередню ескалацію привілеїв, коли

	до привілейованої групи безпеки	скомпрометований або рядовий акаунт несанкціоновано додається до групи локальних або доменних адміністраторів.

Окрім безпосередніх подій авторизації, критично важливими для формування датасету є метадані мережевої взаємодії та параметри сесії. До таких показників належить IP-адреса джерела запиту, що дає змогу відстежувати геолокацію та виявляти просторові аномалії. Не менш вагомим ідентифікатором є тип пристрою та браузера, який розпізнається через заголовок User-Agent. Раптова зміна звичного User-Agent у поєднанні з новою IP-адресою часто вказує на перехоплення сесії.[18] Додатково аналізуються кількісні характеристики: загальна тривалість активної роботи та обсяг переданих або завантажених даних. Різкі сплески вихідного трафіку на фоні типової поведінки є класичною ознакою підготовки інсайдера або хакера до несанкціонованого копіювання корпоративної інформації.

Враховуючи сучасну тенденцію переходу підприємств до гібридних архітектур, збір показників активності не обмежується лише локальними операційними системами. Вагомим джерелом розширених даних виступають системні журнали вебсерверів, баз даних та сучасних хмарних платформ, таких як Microsoft Azure або Google Workspace. Журнали цих хмарних сервісів надають глибокий контекст про те, до яких саме файлів, директорій чи конфігурацій звертався обліковий запис після того, як успішно пройшов процедуру автентифікації.

Об'єднання сирих даних із перелічених різнорідних журналів та метаданих у єдиний нормалізований формат дозволяє сформувати репрезентативну вибірку математичних ознак. У подальшому саме ці агреговані показники активності слугуватимуть базисом для навчання алгоритмів системи, яка здатна класифікувати події доступу на нормальні та аномальні.

1.5 Аналіз існуючих підходів до виявлення аномальної активності

Історично еволюція засобів виявлення кіберзагроз розвивалась від простих

антивірусних сканерів до комплексних систем управління інформацією та подіями безпеки (SIEM). Традиційний підхід до виявлення компрометації облікових записів у таких системах базується на сигнатурному аналізі та використанні жорстких детермінованих правил (Rule-based detection). Суть цього методу полягає у створенні експертами з безпеки статичних умов: наприклад, якщо система фіксує понад п'ять невдалих спроб авторизації за одну хвилину, обліковий запис автоматично блокується. Цей підхід демонструє високу ефективність у боротьбі з відомими, шаблонними векторами атак, оскільки дозволяє миттєво реагувати на загрози, чиї "відбитки" (сигнатури) вже занесені до бази даних засобів захисту.[23]

Однак в умовах сучасних кіберзагроз підхід на основі правил виявляє суттєві архітектурні обмеження. Ключова проблема полягає в тому, що правила здатні виявляти лише ті атаки, які заздалегідь відомі розробникам. Зловмисники, отримавши доступ до легітимних облікових даних за допомогою фішингу чи соціальної інженерії, активно застосовують концепцію "Living off the Land" (використання вбудованих легітимних інструментів операційної системи). Оскільки хакер авторизується під дійсним паролем і використовує стандартні утиліти, система на основі правил не бачить порушень і не генерує жодних попереджень. Намагання адміністраторів описати правилами всі можливі підозрілі сценарії призводить до експоненційного зростання кількості хибних спрацьовувань. У результаті аналітики безпеки стикаються з явищем "втоми від сповіщень", коли серед тисяч рутинних повідомлень втрачається інформація про реальний інцидент.[19]

У відповідь на описані недоліки індустрія кібербезпеки здійснила прорив у бік систем поведінкової аналітики користувачів та сутностей (UEBA – User and Entity Behavior Analytics). Фундаментальна відмінність цього підходу полягає у відмові від пошуку відомих індикаторів компрометації. Натомість системи UEBA використовують алгоритми машинного навчання (Machine Learning) для автоматичної побудови динамічних базових ліній нормальної поведінки для кожного облікового запису[20].

Використання методів машинного навчання, зокрема алгоритмів навчання без вчителя, дозволяє виявляти принципово нові, невідомі раніше загрози та дії внутрішніх порушників. Алгоритми кластеризації та виявлення викидів (Anomaly Detection) здатні аналізувати сотні параметрів одночасно: час доступу, обсяги трафіку, геолокацію, використання конкретних пристроїв та характер запущених процесів. Кожній дії користувача присвоюється певний бал ризику. Коли сумарний бал перевищує допустимий поріг через накопичення дрібних, але нетипових дій, система ініціює тривогу.[21]

Таким чином, порівняльний аналіз існуючих рішень доводить, що класичні детерміновані правила більше не здатні забезпечити надійний захист облікових записів від складних цілеспрямованих атак. Забезпечення проактивної безпеки вимагає переходу до систем на базі машинного навчання, здатних адаптуватися до змін у середовищі та виявляти скомпрометовані ідентифікатори на основі глибокого поведінкового аналізу.[22]

1.6 Постановка задачі

Проведений аналіз сучасного світу кіберзагроз та існуючих засобів захисту свідчить про те, що традиційні детерміновані системи на основі правил не здатні ефективно протидіяти складним атакам, що використовують легітимні облікові дані. Основним недоліком існуючих підходів є нездатність виявляти приховані поведінкові аномалії та високий рівень хибних спрацьовувань через статичність алгоритмів.

Виходячи з виявленої потреби у переході до проактивного захисту на основі аналітики поведінки (UEBA), метою кваліфікаційної роботи є розробка та програмна реалізація інтелектуальної системи виявлення облікових записів з аномальною активністю на основі методів багатовимірного статистичного аналізу.

Для досягнення поставленої мети необхідно вирішити такі науково-практичні завдання:

– дослідити особливості аномальної активності в інформаційних системах, класифікувати типи аномалій поведінки та визначити критичні характеристики подій доступу;

– виконати порівняльний аналіз існуючих методів виявлення та обґрунтувати вибір математичного апарату для ідентифікації статистичних відхилень у профілях користувачів;

– сформувати вимоги до функціональності та архітектури системи виявлення, визначивши ключові часові, поведінкові та контекстні показники активності;

– розробити багатовимірний вектор ознак для профілювання облікових записів та спроектувати алгоритм функціонування системи;

– здійснити програмну реалізацію прототипу системи виявлення з використанням аналітичних бібліотек та розробкою інтерактивного інтерфейсу користувача;

– провести експериментальне дослідження працездатності розробленого рішення на тестових наборах даних та оцінити його ефективність за допомогою метрик точності виявлення загроз.

2 ПРОЄКТУВАННЯ ТА МАТЕМАТИЧНЕ ОБҐРУНТУВАННЯ СИСТЕМИ ВИЯВЛЕННЯ АНОМАЛІЙ

2.1 Формування вимог до системи виявлення аномальної активності

Етап проектування системи виявлення аномальної активності розпочинається з формування комплексного переліку функціональних та нефункціональних вимог. У контексті кіберзагроз, де класичні сигнатурні методи захисту демонструють низьку ефективність проти інсайдерських атак та скомпрометованих легітимних облікових записів, чітка регламентація завдань є необхідною умовою для створення дієвого аналітичного інструменту. Сформовані вимоги виступають базовим критерієм для подальшого вибору математичного апарату, архітектури програмного рішення та оцінки працездатності готового прототипу в умовах реального корпоративного середовища [24].

З точки зору функціональності, розроблюваний програмний комплекс повинен забезпечувати повний та автоматизований цикл обробки інформації про активність користувачів. Першочерговим завданням є реалізація надійного модуля імпорту журналів подій аудиту у форматі табличних даних (CSV), що дозволить системі інтегруватися з існуючими рішеннями централізованого збору логів. Після етапу завантаження система має автоматично виконувати складну математичну обробку масиву, розраховуючи персоналізований рівень статистичного відхилення для кожного облікового запису. Базова вимога є здатність алгоритму самостійно класифікувати події на легітимні та аномальні, трансформуючи абстрактні просторові метрики у зрозумілий для оператора відсотковий показник ризику. Окрім обчислювальної частини, обов'язковою є наявність підсистеми візуалізації для побудови інтерактивних графіків розсіювання, а також генерації динамічного оперативного журналу інцидентів, який має автоматично пріоритезувати загрози за рівнем їхньої критичності для забезпечення швидкого реагування.

Не менше важливими є нефункціональні вимоги, які описуються експлуатаційні характеристики інструменту та його придатність для

використання в операційних центрах безпеки (SOC). З огляду на те, що сучасні інформаційні інфраструктури генерують значні обсяги журналів аудиту щороку, першочерговою вимогою є алгоритм високої обчислювальної швидкості та масштабованості. Прототип повинен обробляти великі масиви даних у режимі, наближеному до реального часу, оптимізуючи збереження оперативної пам'яті комп'ютера. Точність виявлення загрози досягається шляхом використання адаптивних статистичних методів замість застарілих жорстких лінійних правил. Це дозволяє ефективно ідентифікувати приховані атаки та суттєво знизити рівень захисних спрацювань. Слід також зазначити, що графічний інтерфейс системи має відповідати принципам ергономіки, бути інтуїтивно зрозумілим і забезпечувати високу швидкість формування багатовимірних аналітичних панелей без затримок. Важливою нефункціональною вимогою також є математична прозорість рішень, які дозволяють аналітикам чітко розуміти причину блокування конкретної сесії [25]. Життєвий цикл обробки інциденту в розробленій системі, від етапу моніторингу до реагування, наведено на рис. 2.1.

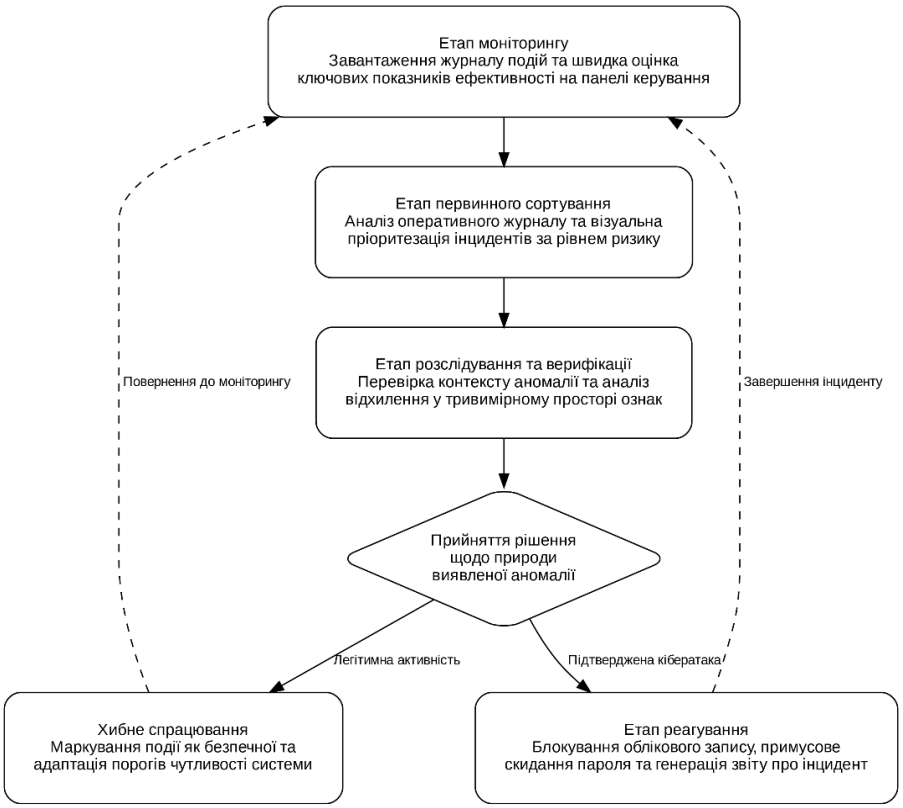


Рисунок 2.1 – Життєвий цикл обробки інциденту безпеки

Окремий блок вимог висувається до структури, формату та якості вхідних даних, на яких базуватиметься поведінкове профілювання. Архітектура рішення передбачає обробку формалізованих масивів інформації, де обов'язковими атрибутами виступають унікальні ідентифікатори користувачів та точні часові позначки реєстрації подій [17]. Для коректної побудови багатовимірної моделі вхідні журнали повинні містити специфічні кількісні показники: інтенсивність спроб авторизації для виявлення атак типу brute-force, загальну тривалість активних сесій для фіксації аномально коротких підключень автоматизованих скриптів та вік поточного пароля як міру загальної вразливості акаунта [10,18]. Наявність цього набору параметрів є критично необхідною умовою для формування статистичного еталона нормальної активності та гарантування достовірності результатів роботи аналітичного ядра системи.

2.2 Формування багатовимірного вектора ознак

В умовах експлуатації інформаційних систем факт успішної автентифікації не є достатньою гарантією безпеки, оскільки облікові дані можуть бути скомпрометовані через витоки, придбані у прихованих сегментах мережі або підібрані засобами автоматизації. Центральним елементом проектування розробленої системи є формування багатовимірного вектора ознак, що дозволяє формалізувати поняття підозріла поведінка та перевести його у площину математичних координат [26]. Замість реагування на ізольовані події, здійснюється моделювання поведінкового вектора, де кожна сесія представлена як точка у багатовимірному просторі. Це забезпечує можливість аналізу не лише окремих дій, а й контексту їх виконання, що є важливим для верифікації складних цілеспрямованих атак.

Початковим компонентом вектора є поведінковий маркер, що базується на аналізі кількості невдалих спроб авторизації. Незважаючи на те, що цей показник є класичним індикатором, у запропонованій моделі він набуває розширеного значення. Аномальна інтенсивність помилок за короткий інтервал

часу сигналізує про спроби реалізації атак типу brute-force або credential stuffing [27]. Також, виявлення поодиноких, але систематичних помилок з різних вузлів дозволяє виявити методи password spraying [28]. Інтеграція даного показника у багатовимірний вектор надає можливість диференціювати випадкову помилку користувача від цілеспрямованої автоматизованої агресії, тому що алгоритм оцінює масштаб події відносно загального профілю активності.

Другим значущим виміром моделі є часовий маркер – тривалість активної сесії. Даний показник є високоінформативним для детекції ботів та скриптів, швидкість функціонування яких значно перевищує людські можливості. Легітимна активність характеризується сталим когнітивним темпом: суб'єкту необхідний час на сприйняття інтерфейсу, обробку інформації та введення даних, що зумовлює варіативність та відносну тривалість сесій [29]. Натомість скрипти, націлені на ексфільтрацію інформації або сканування ресурсів, завершують сесію за лічені секунди. Виявлення у векторі ознак аномально коротких сесій у поєднанні з високою інтенсивністю запитів є чітким індикатором роботи автоматизованого інструментарію.

Третім, контекстним елементом, що замикає багатовимірний простір ознак, є вік поточного пароля. Цей маркер вразливості виконує роль вагового коефіцієнта ризику у загальній статистичній моделі. Виходячи з концепції: чим довше використовувати однаковий пароль, тим вища ймовірність його крадіжки чи підбору [30]. Застарілі облікові дані автоматично підвищують рівень підозрілості будь-якої активності в межах акаунта для розрахункового алгоритму. Це дозволяє системі проявляти підвищену чутливість до дрібних відхилень у поведінці користувачів, забезпечуючи проактивний захист найбільш критичних ділянок інфраструктури.

Синтез зазначених різнорідних маркерів у єдиний векторний простір формує математичний фундамент системи. Об’єднання інтенсивності помилок, темпоральних характеристик сесій та стану облікових даних дозволяє отримати унікальну цифрову сигнатуру кожної події. Такий підхід забезпечує перехід від примітивного аналізу за граничними значеннями до виявлення прихованих

аномалій, які неможливо класифікувати при ізольованому розгляді показників.

Розроблений підхід до структурування вхідних даних дозволяє реалізувати адаптивну систему, яка безперервно оцінює контекст подій. Це трансформує моніторинг у інтелектуальний процес, де кожен новий запис у журналі подій уточнює загальну картину, дозволяючи знайти загрози на етапі їх виникнення. Глибина аналізу вхідних параметрів забезпечує здатність прототипу протидіяти сучасним методам компрометації облікових записів.

2.3 Обґрунтування вибору математичного апарату

Від вибору математичної складової залежить здатність системи адаптуватися до змінних умов середовища та достовірність ідентифікації кіберзагроз. У практиці розробки систем аналізу часто розглядаються два основні підходи: застосування складних нейромережевих архітектур та використання методів багатовимірного статистичного аналізу[31].

Попри високу популярність методів глибинного навчання, їх застосування в задачах моніторингу безпеки облікових записів стикається з низкою критичних обмежень. Основним недоліком нейромережевих моделей є проблема чорної скриньки – відсутність прозорості та інтерпретованості механізмів прийняття рішень. У сфері кібербезпеки першочергово не лише зафіксувати факт аномалії, а й чітко ідентифікувати причини її виникнення. Нейромережі, через свою багатошарову структуру та складні нелінійні зв'язки, не дозволяють оперативно проаналізувати, на основі яких саме вагових коефіцієнтів подія була класифікована як загроза[31]. Це створює труднощі при розслідуванні інцидентів та верифікації хибних спрацювань.

Іншим вагомим аргументом проти використання нейромереж у межах даної розробки є проблема дефіциту верифікованих та розмічених наборів даних. Для ефективного навчання складних моделей необхідні величезні масиви даних, де кожен запис уже класифікований як норма або атака. У реальних умовах

корпоративних інформаційних систем отримати таку розмітку практично неможливо, тому що загрози постійно змінюються, а реальні інциденти становлять менш як 0,1% від загального обсягу логів [14]. Навчання на неповних або нерелевантних даних призводить до критичного зниження точності та нездатності системи виявляти нові типи атак.

Підходи машинного навчання потребують значних обчислювальних ресурсів для етапів навчання та перенавчання моделей. У контексті розробки адаптивного прототипу, що має функціонувати в умовах обмежених ресурсів та обробляти події в режимі реального часу, використання таких громіздких архітектур є економічно та технічно недоцільним [31].

На противагу методам навчання з учителем, багатовимірний статистичний аналіз, що базується на навчанні без учителя, є найбільш релевантним вибором для вирішення поставленої задачі. Цей підхід ґрунтується на математичній формалізації поняття нормальної поведінки як певної області в багатовимірному просторі ознак. Аномалії в такому випадку розглядаються як статистичні викиди, що знаходяться поза межами встановленого довірчого інтервалу [33].

Вибір на користь методів багатовимірного статистичного аналізу є науково обґрунтованим з огляду на сукупність факторів, що визначають ефективність системи в реальних умовах експлуатації. Насамперед, такий підхід забезпечує високий рівень математичної прозорості та доказовості результатів [32]. Кожен висновок системи базується на розрахунку чітко визначених статистичних метрик, стає можливим точне визначення природи виявленої аномалії – наприклад, ідентифікація конкретного відхилення у тривалості сесії або фіксація сплеску помилок авторизації. Така інтерпретованість результатів є критично важливою для забезпечення високого рівня довіри до системи з боку адміністраторів та аналітиків.

Важливою перевагою обраного методу є відсутність потреби у попередньо розмічених наборах даних. Статистичні методи дозволяють формувати базову лінію нормальної поведінки безпосередньо на основі поточного масиву даних, не потребуючи апріорної класифікації подій на легітимні та шкідливі. Система в

автоматизованому режимі вивчає структуру журналів подій та самостійно виявляє значущі відхилення від еталонного профілю. Це поєднується з високим рівнем адаптивності та швидкістю обробки інформації: розрахунок параметрів, таких як середні значення чи коваріаційні матриці, вимагає значно менших обчислювальних потужностей порівняно з ітераційними процесами навчання нейромережевих архітектур. Як наслідок, забезпечується оперативний аналіз великих масивів даних у форматі CSV та можливість швидкої перебудови моделі при зміні режимів активності користувачів [33].

Архітектура системи відзначається гнучкістю налаштувань. Використання статистичних порогів дозволяє адаптувати чутливість алгоритмів під специфіку конкретної IT-інфраструктури, варіюючи рівень допустимого відхилення залежно від вимог до безпеки окремих сегментів мережі. Такий синтез прозорості, швидкодії та гнучкості робить статистичний аналіз оптимальним інструментом для вирішення задач виявлення аномальної активності облікових записів.

Таким чином, для реалізації системи виявлення аномалій у поведінці облікових записів обрано метод багатовимірного статистичного аналізу. Це дозволяє поєднати математичну суворість розрахунків із практичною ефективністю виявлення невідомих раніше загроз, уникаючи при цьому обмежень, пов'язаних із складністю та непрозорістю нейромережевих архітектур. Сформований аналітичний фундамент створює умови для застосування спеціалізованих метрик розрахунку статистичних відстаней, що буде детально розглянуто в наступних етапах роботи.

Центральним елементом аналітичного блоку розробленої системи є алгоритм розрахунку статистичної відстані Махаланобіса. Вибір запропонованого методу обумовлений його здатністю ідентифікувати аномалії в багатовимірних наборах даних, де параметри мають різні масштаби вимірювання та існують приховані кореляційні зв’язки [34].

Більшість класичних систем виявлення відхилень базується на використанні евклідової відстані, яка розглядає дані як точки в ізотропному

просторі. Однак в задачах параметри активності користувачів (наприклад, кількість помилок входу та тривалість сесії) зазвичай є залежними один від одної. Евклідова відстань не враховує цю залежність, розглядаючи кожен вимір як незалежний, що призводить до високої ймовірності пропуску аномалій, які знаходяться в межах лінійних порогів, але порушують загальну статистичну структуру поведінки [34].

Метрика Махаланобіса вирішує цю проблему шляхом деформації простору відповідно до розподілу даних [35, 36]. Вона вимірює відстань точки від центру мас (середнього вектора), враховуючи при цьому коваріацію (взаємозв'язок) ознак. Фактично, метод оцінює, на скільки стандартних відхилень точка віддалена від центру розподілу вздовж головних осей даних.

Для реалізації алгоритму аналізу використовується послідовність математичних операцій. Усі розрахунки проводяться над матрицею даних X, де кожен рядок – це окрема подія доступу, а три стовпці – це наші ознаки (помилки, час сесії, вік пароля).

Перш ніж шукати аномалії, системі потрібно зрозуміти, як виглядає норма. Для цього обчислюється вектор середніх значень центроїда μ [36]:

$$\mu = \frac{1}{n}\sum_{i=1}^{n} X_i , \qquad (2.1)$$

де n – загальна кількість легітимних сесій; Xi – вектор ознак і-ї сесії.

Алгоритм бере всі легітимні сесії та вираховує усереднений портрет активності: скільки в середньому помилок робить користувач, скільки зазвичай триває його сесія тощо.

Це найважливіший етап. Алгоритм має зрозуміти не лише середні значення, а й те, як показники впливають один на одного. Для цього будується матриця коваріації $\sum$ [36]:

$$\sum = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \mu)(X_i - \mu)^T. \qquad (2.2)$$

Саме ця формула робить систему адаптивною, вона фіксує закономірності. Наприклад, завдяки цій матриці алгоритм знає, що коротка сесія – це нормально, помилка входу – теж буває, але одночасна поява короткої сесії та помилки входу на старому паролі – це критична загроза.

Коли система знає норму та залежності, для кожної нової події X розраховується дистанція відхилення D^2[35, 36]:

$$D^2(x) = (x - \mu)^T \sum^{-1} (x - \mu), \quad (2.3)$$

де $\sum^{-1}$ – це обернена матриця коваріації. Ця формула діє як адаптивний фільтр: вона прощає користувачу відхилення в тих параметрах, які історично сильно коливаються (наприклад, час сесії), але миттєво реагує на найменші зміни у стабільних показниках. Отримане числове значення дистанції згодом конвертується у відсоток ризику, який виводиться на дашборд.

Для обґрунтування вибору математичного апарату системи було проведено порівняльний аналіз метрики Махаланобіса із класичною евклідовою відстанню, яка зазвичай використовується у базових алгоритмах моніторингу. Евклідова відстань діє як стандартна геометрична міра, що вимірює абсолютну віддаленість точок у просторі, проте її застосування для виявлення кібератак має два фундаментальні недоліки.

Першим недоліком є чутливість до масштабу вимірювань. В системі аналізуються різнорідні показники (кількість помилок входу в одиницях та вік пароля в сотнях днів), евклідова відстань математично надає перевагу ознакам із більшими числовими значеннями. Це призводить до того, що критичні зміни в кількості невдалих спроб авторизації ігноруються на фоні великих значень віку пароля. На відміну від неї, метрика Махаланобіса забезпечує інваріантність до масштабу, автоматично нормалізуючи ваги ознак через врахування їхніх власних стандартних відхилень[36].

Другим недоліком є відсутність врахування кореляційних зв'язків. Евклідова відстань будує зону нормальної поведінки у формі ідеальної сфери,

що дозволяє зловмисникам маскувати аномалії, діючи в межах встановлених лінійних порогів. Метрика Махаланобіса формує адаптивний еліпсоїд(рисунок 2.2), тому що вона аналізує структуру взаємозалежностей між ознаками[35]. Це дозволяє виявити малопомітні втручання: навіть якщо окремі показники активності не перевищують критичних значень, їхня нетипова комбінація (наприклад, коротка сесія при одночасній помилці входу) розпізнається алгоритмом як порушення поведінкового шаблону та статистична аномалія.

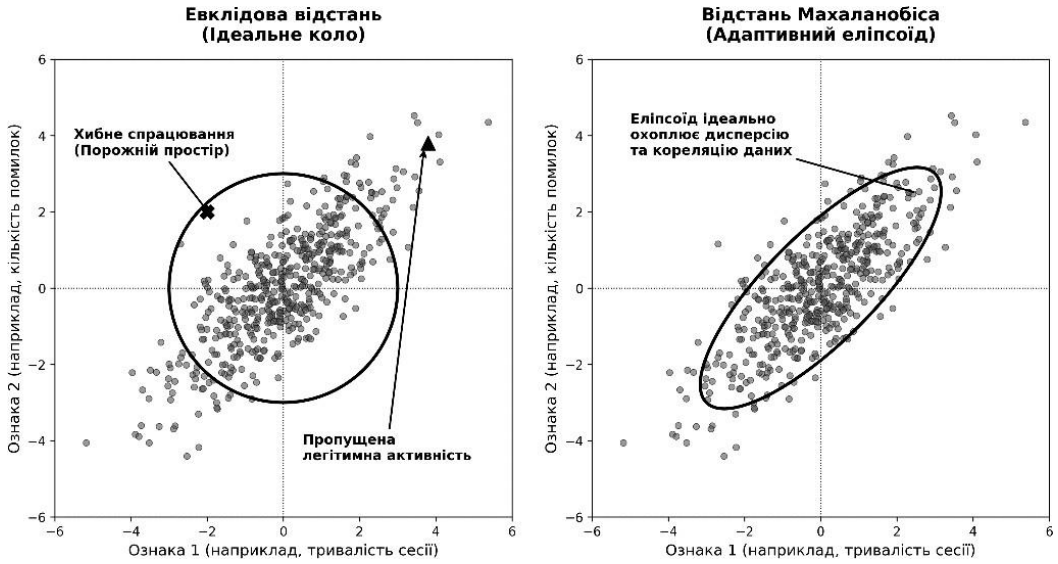


Рисунок 2.2 – Геометрична інтерпретація евклідової метрики та відстані Махаланобіса

Таким чином, використання відстані Махаланобіса замість стандартних геометричних метрик дозволяє системі розпізнавати не лише кількісні викиди, а й структурні аномалії в профілях користувачів. Це перетворює систему на інтелектуальний інструмент аналізу, здатний виявляти складні, замасковані спроби компрометації облікових записів на ранніх етапах.

2.4 Вибір та обґрунтування програмних засобів розробки прототипу

Ефективність системи виявлення аномалій, швидкість обробки журналів

подій та можливість подальшого масштабування безпосередньо залежать від обраного технологічного стека. Під час розробки програмного прототипу було проведено аналіз сучасних інструментальних засобів та обрано стек технологій, що найкраще відповідає завданням багатовимірного статистичного аналізу та вимогам інформаційної безпеки.

Основним середовищем для реалізації програмної логіки та аналітичного ядра системи було обрано високорівневу мову програмування Python. Такий вибір є технічно та архітектурно обґрунтованим з огляду на наступні фактори.

По-перше, Python на сьогодні є визнаним світовим стандартом у сфері кібербезпеки та системного адміністрування [37]. Більшість сучасних засобів захисту інформації (SIEM-системи, платформи Threat Intelligence, сканери вразливостей) мають вбудовану підтримку Python-скриптів або надають API для інтеграції. Це означає, що розроблений прототип у перспективі може бути легко інтегрований у реальну корпоративну інфраструктуру або використаний як модуль для існуючих SOC.

По-друге, оскільки в основі розробленої системи лежить розрахунок статистичної відстані Махаланобіса, критичною вимогою до мови програмування є здатність ефективно працювати з матрицями та багатовимірними векторами. Python володіє найпотужнішою у світі екосистемою інструментів для аналізу даних (Data Science). Хоча сама мова є інтерпретованою, її спеціалізовані математичні бібліотеки написані на низькорівневих мовах (C/C++), що забезпечує високу обчислювальну продуктивність[38]. Це дозволяє прототипу обробляти великі масиви лог-файлів CSV практично миттєво, виконуючи складні операції обернення коваріаційних матриць без перевантаження оперативної пам'яті.

По-третє, синтаксис Python забезпечує високу швидкість розробки (Rapid Prototyping). Це дозволило зосередити основні зусилля на реалізації та вдосконаленні математичної моделі виявлення загроз, а не на вирішенні низькорівневих проблем управління пам'яттю, які є характерними для мов C++ або Java. Наявність широкого спектра вбудованих інструментів для парсингу

текстових даних робить Python ідеальним вибором для етапу препроцесингу журналів аудиту різного формату.

Реалізації етапів попередньої обробки інформації (препроцесингу) та виконання багатовимірних статистичних розрахунків у системі застосовується комплекс фундаментальних бібліотек: Pandas, NumPy та SciPy. Їх використання дозволяє ефективно вирішити проблему трансформації сирих лог-файлів у структурований математичний простір.

Pandas використовується як основний інструмент для маніпуляції зі структурованими масивами даних [37]. Оскільки вхідні журнали подій традиційно експортуються з інформаційних систем у табличних форматах (зокрема, CSV), дана бібліотека забезпечує їх швидке завантаження та трансформацію у високорівневі структури DataFrame. Це дозволяє здійснювати швидку індексацію, агрегацію часових позначок (timestamps) та нормалізацію атрибутів, що є критично необхідним для первинного формування ознак (кількості помилок, тривалості сесій).

NumPy (Numerical Python) виконує роль базового рушія для роботи з багатовимірними масивами [38]. У контексті розробки прототипу ця бібліотека відповідає за перетворення табличних даних у векторний та матричний формати. Завдяки тому, що ядро NumPy реалізовано мовою низького рівня (C), виконання масових математичних операцій над сотнями тисяч записів відбувається з максимальною швидкодією. Це усуває обмеження продуктивності стандартних циклів інтерпретатора Python і забезпечує ефективну роботу з коваріаційними матрицями.

SciPy доповнює екосистему інструментами вищого рівня для наукових обчислень [39]. У межах розробленого прототипу ключовим компонентом виступає підмодуль просторових метрик (scipy.spatial.distance), який надає оптимізовані та апаратно-прискорені алгоритми розрахунку відстані Махаланобіса. Використання цієї бібліотеки гарантує високу алгебраїчну точність при розрахунку обернених матриць та забезпечує стабільність аналітичного ядра алгоритму незалежно від обсягу аналізованого потоку подій.

Щоб забезпечити взаємодію користувача з аналітичним ядром системи та наочного представлення результатів аналізу було обрано спеціалізовані інструменти побудови веб-інтерфейсів та графічної візуалізації.

Фреймворк Streamlit обрано як основне середовище для побудови інтерфейсу користувача. Вибір даного інструменту замість створення класичного веб-додатка на базі HTML/CSS/JavaScript обумовлений необхідністю швидкого прототипування та безшовної інтеграції з Python-скриптами. Streamlit дозволяє трансформувати аналітичний код у повноцінний інтерактивний веб-дашборд, не витрачаючи ресурси на розробку фронтенд-архітектури. Це дало змогу змістити основний фокус розробки на вдосконалення алгоритмів виявлення аномалій, забезпечивши при цьому офіцера безпеки зручним інструментарієм для завантаження даних, налаштування порогів ризику та перегляду звітів у режимі реального часу [40].

Графічне представлення результатів аналізу та ідентифікованих аномалій у системі інтегровано бібліотеки Plotly та Seaborn. Plotly застосовується для створення інтерактивних тривимірних (3D) графіків розсіювання [41]. Оскільки метрика Махаланобіса працює у багатовимірному просторі ознак, візуалізація подій у 3D-координатах дозволяє аналітику наочно побачити положення аномальних точок відносно еліпсоїда нормальної поведінки. Інтерактивні можливості бібліотеки (масштабування, обертання, відображення деталей при наведенні) значно спрощують процес верифікації інцидентів та проведення візуального аудиту безпеки. Seaborn використовується для побудови статистичних теплових карт (Heatmaps) та матриць кореляції [42]. Це дозволяє на етапі препроцесингу оцінити ступінь взаємозалежності між параметрами активності користувачів. Наочне представлення кореляцій допомагає обґрунтувати структуру коваріаційної матриці та підтвердити доцільність обраної моделі аналізу для конкретного набору даних.

Обраний технологічний стек формує цілісну екосистему для розробки прототипу. Таке поєднання інструментів забезпечує високу швидкість обробки інформації, математичну точність розрахунків та сучасний рівень візуалізації,

що є критично важливим для систем виявлення складних кіберзагроз.

2.5 Загальна структурна схема та алгоритм функціонування системи

Проєктування програмного прототипу системи виявлення аномальної активності базується на фундаментальних принципах програмної інженерії, зокрема на концепціях модульності, слабкої зв'язності компонентів та високої функціональної цілісності [43]. З огляду на специфіку завдань інформаційної безпеки, де обсяги даних можуть стрімко зростати, а вимоги до швидкості реагування є критичними, у структурі запропонованого рішення реалізовано багаторівневу модель. Такий підхід дозволяє ефективно розподілити обчислювальне навантаження та забезпечити високу відмовостійкість системи на кожному окремому етапі опрацювання інформаційних потоків. Загальна структурна схема розробленої системи наведена на рис. 2.3. Вона складається з трьох основних логічних блоків, кожен з яких виконує суворо визначену роль: підсистеми введення та попередньої обробки даних, математичного аналітичного ядра та модуля візуалізації та взаємодії з користувачем. Автономність цих блоків гарантує, що потенційна модифікація або розширення функціоналу одного з них не призведе до необхідності повної перебудови чи дестабілізації всього програмного комплексу. Така глибока декомпозиція дозволяє повністю ізолювати ресурсоємні обчислювальні процеси від графічного інтерфейсу. Це, своєю чергою, дозволяє аналітику з кібербезпеки безперешкодно взаємодіяти з інтерактивними інструментами моніторингу, навіть коли аналітичне ядро виконує складні матричні обчислення у фоновому режимі. У результаті забезпечується висока продуктивність системи, стабільність її роботи під навантаженням та створюються надійні передумови для її подальшого масштабування чи інтеграції з наявними корпоративними рішеннями.

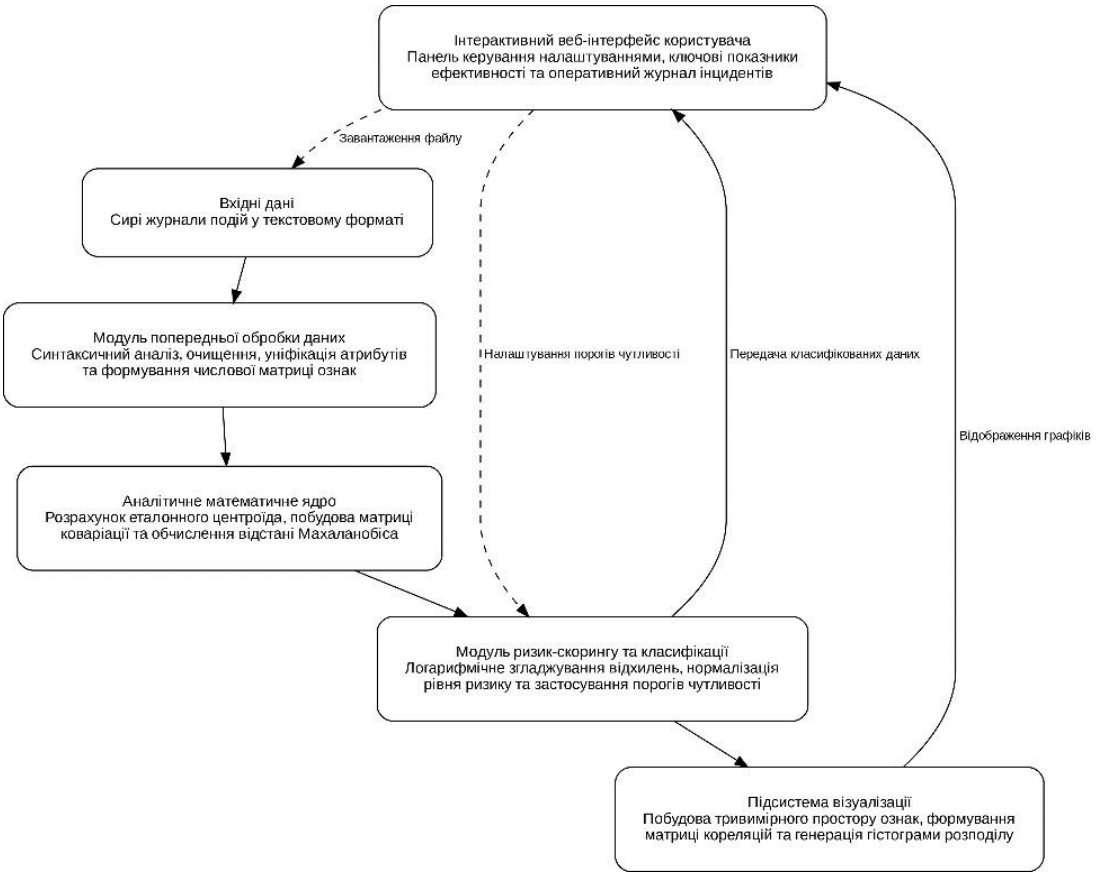


Рисунок 2.3 – Загальна структурна схема системи поведінкової аналітики

Перший рівень архітектури, а саме блок введення та попередньої обробки даних, відповідає первинній взаємодії із зовнішніми джерелами інформації. З огляду на те, що корпоративні системи входу, такі як SIEM-системи, служби каталогів Active Directory або систем контролю доступу, дозволяють експортувати журнали аудиту у форматі з роздільниками, базовим форматом вхідних даних для прототипу було обрано текстовий формат CSV [19, 20]. Цей блок реалізує механізм безпечного завантаження файлів в систему оперативної пам’яті, після чого ініціюється процес синтаксичного аналізу та парсингу. На цьому етапі відбувається стандартизація структури даних, коли різноманітні технічні назви колонок автоматично трансформуються в уніфіковані змінені внутрішні за допомогою заданого словника відповідностей. Цей модуль відповідає за ізоляцію цільової метрики від загального масиву текстової інформації, формуючи чистий числовий багатовимірний простір ознак, який є придатним для подальших складних математичних маніпуляцій.

Другий рівень представлений аналітичним компонентом, який є центральним і найбільш ресурсоємним компонентом у всій системі, а потім саме включає всю логіку виявлення кіберзагроз [36]. Він побудований на базі оптимізованих алгоритмів лінійної алгебри та просторової статистики. Його головне завдання полягає у динамічній побудові еталонної моделі поведінки на основі завантаженого набору даних, причому система не потребує попереднього навчання на історичних вибірках. Обчислювальний модуль самостійно розраховує вектор середніх значень, який виступає в ролі центру нормальної активності, та формує матрицю коваріації, що описує складні нелінійні взаємозв'язки між тривалістю сесії, кількість помилок авторизації та віком облікових даних. Після побудови цієї моделі норми, аналітичне ядро виконує просторові обчислення, визначаючи точну статистичну відстань Махаланобіса для кожної окремої події [35], і трансформує ці абстрактні математичні значення в нормалізовану та зрозумілу бізнес-метрику рівня ризику.

Третій рівень архітектури формує модуль візуалізації та взаємодії, який представляє інтерактивний веб-дашборд і виконує роль єдиного вікна управління для інформаційної безпеки. Цей модуль безперервно отримує оброблені та класифіковані дані від математичного ядра і миттєво трансформує їх у наочні графічні представлення. Він забезпечує повноцінну двосторонню взаємодію, дозволяючи користувачу гнучко керувати параметрами системи, змінювати стратегії виявлення та встановлювати пороги чутливості. Одночасно з цим модулем генеруються складні тривимірні графіки розсіювання, теплові матриці кореляцій та динамічні таблиці інцидентів, що суттєво пришвидшує процес прийняття рішень під час розслідування інцидентів.

Логіка роботи програмного прототипу реалізована у вигляді послідовного конвеєра обробки даних, де кожен етап обробки є суворо визначеним і виконується в автоматичному режимі після ініціалізації процесу користувачем. Початковим етапом функціонування системи є ініціалізація та завантаження масиву даних. Процес розпочинається із завантаження користувачем файлу журналу подій через спеціальний віджет у графічному інтерфейсі бічної панелі.

Система зчитує вміст файлу та перетворює його в структурований табличний формат. Одразу після цього алгоритму виконується сувора валідація дійсності вхідного потоку, перевіряючи наявність усіх обов’язкових атрибутів, таких як ідентифікатори користувачів, часові позначки та спеціальні показники активності. У випадку, якщо структура завантаженого файлу не відповідає очікуваній еталонній схемі, конвеєр автоматично припиняється з виведенням відповідного інформаційного повідомлення про помилку, що дозволяє уникати критичних збоїв у роботі математичного апарату на пізніх етапах.

Після успішної валідації система переходить до етапу векторизації та формування простору ознаки. Програмний комплекс цілеспрямовано виокремлює з усього масиву даних три ключові параметри, які були обґрунтовані на етапі проектування: кількість спроб входу, тривалість сесії в хвилинах та вік пароля в днях. Ці різнорідні дані трансформуються в єдину двовимірну числову матрицю, де кількість рядків відповідає загальній кількості зафіксованих сесій у лог-файлі, а три стовпці представляють вибрані маркери поведінки. Такий підхід дозволяє уніфікувати різні за своєю природою метрики, привівши їх до єдиного стандартизованого вигляду, що є обов'язковою умовою для коректного порівняння параметрів. Сформована матриця стає фундаментальною основою для всіх подальших алгебраїчних обчислень в межах аналітичного ядра.

Наступним кроком є побудова еталонного профілю, що також відомо як процес профілювання. На відміну від класичних систем із жорстко заданими правилами, розроблений алгоритм адаптується під специфіку конкретного набору даних. Система обчислює вектор математичних сподівань, який представляє усереднений портрет активності легітимного користувача досліджуваної інфраструктури. Далі аналітичний блок будує матрицю коваріації, яка математично фіксує дисперсію кожної окремої ознаки та рівень їхньої взаємозалежності. Для забезпечення можливості коректного розрахунку просторових відстаней аналітичний блок відразу обчислює псевдообернену матрицю коваріації. Використання псевдооберненої матриці замість оберненої є

свідомим архітектурним рішенням, яке дозволяє уникнути фатальних помилок ділення на нуль у випадку низької класичної варіативності даних або наявності великої кількості ідентичних записів у журналах подій [38]. Це забезпечує високу обчислювальну стійкість математичного ядра системи та мінімізує ризик виникнення програмних збоїв під час обробки реальних високоінтенсивних потоків інформації.

Вибудовуючи модель нормальної поведінки, здійснює ініціює цикл розрахунку статистичних відхилень. Програма послідовно проходить через кожен рядок сформованої матриці ознаки, аналізуючи кожну окрему сесію. Для кожного вектора обчислюється точна відстань Махаланобіса до визначеного попереднього центроїда з обов'язковим використанням псевдооберненої матриці коваріації. Результатом виконання цього кроку є формування нового масиву чисельних значень, тобто числа, які прямо пропорційно ступеню аномальності поведінки користувача в контексті загальної виборки. Отримані значення інтегруються до загальної таблиці даних як новий розрахунковий параметр статистичного відхилення.

Отримане значення відстані Махаланобіса може мати надзвичайно великий розкид, від одиниць для нормальних сесій до десятків тисяч для екстремальних викидів, наступним етапом є логарифмічне згладжування та нормалізація ризику. Система застосовує математичне згладжування за допомогою функції натурального логарифма, що дозволяє ефективно стиснути шкалу та зменшити непропорційний вплив екстремальних аномалій на загальну картину. Після процедури логарифмування вона вибирає мінімальне та максимальне значення у згладженій вибірці і застосовує метод лінійної нормалізації [44]. Завдяки цим перетворенням кожна подія в системі отримує інтуїтивно зрозумілу метрику рівня ризику, виражену у відсотках від нуля до ста. Така стандартизована шкала значно спрощує подальшу інтерпретацію результатів аналітиками безпеки під час прийняття оперативних рішень щодо реагування на інциденти. Детальний алгоритм роботи розробленої системи, що відображає послідовність обробки даних із завантаження CSV-файлу для

формування журналу інцидентів, наведено на рис. 2.4.

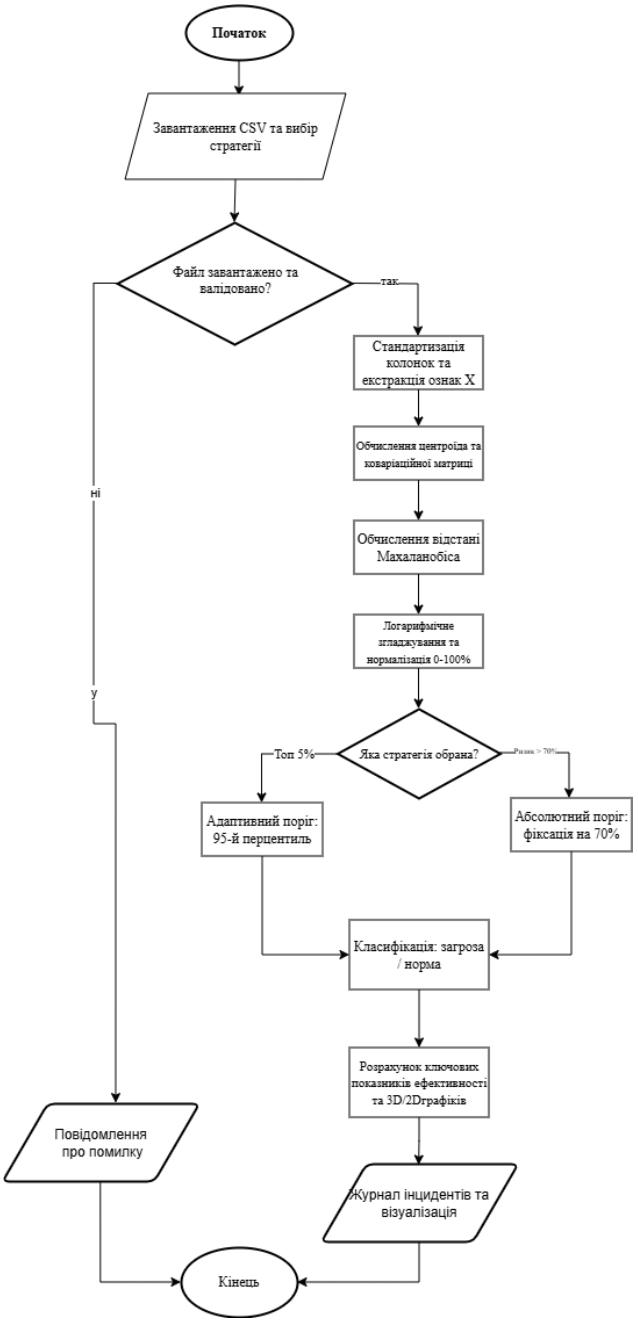


Рисунок 2.4 – Блок-схема алгоритму функціонування системи

Завершальним аналітичним етапом є класифікація інцидентів за обраною стратегією. Обчислювальний модуль звертається до конфігураційних налаштувань, заданих користувачами в інтерфейсі, для прийняття остаточного рішення щодо маркування кожної події. Система підтримує дві паралельні гілки логіки. Перша гілка реалізує стратегію відносного порогу, при якому алгоритм динамічно розраховує 95-й процентиль для всього масиву рівнів ризику. Усі

події, чий показник перевищує цю адаптивну межу, автоматично підтримують статус загрози, що гарантує виявлення п'яти найбільших відсотків вражених дій незалежно від загальної фонової активності. Друга гілка реалізує стратегію абсолютного порогу, застосовуючи жорсткий фільтр, при якому статус інциденту присвоюється всім подіям, чий розрахунковий рівень ризику дорівнює або перевищує сімдесят відсотків, що є особливо ефективним під час фіксації масових автоматизованих кібератак.

На фінальному етапі функціонування конвеєра паралельно запускаються процеси рендерингу видимої аналітики та формування звітів. Система будує інтерактивну тривимірну модель простору позначки, де кожна проаналізована сесія відображається в точках з відповідним кольоровим кодуванням рівня небезпеки. Одночасно формуються матриці кореляцій для глибокого аналізу взаємозв'язків та гістограми статистичного розподілу ризиків. Паралельно з процесами візуалізації програма фільтрує загальний масив оброблених даних, залишаючи лише ті записи, які отримали статус загрози. Ці записи сортуються за спаданням рівня ризику, після чого формується оперативний журнал інцидентів. Для оптимізації використання оперативної пам'яті та забезпечення плавності роботи інтерфейсу, вивід в журнал обмежує тисячу найбільш критичних записів. Повністю сформований та наповнений даними дашборд виводиться на екран, що означає успішне завершення повного циклу обробки інформації.

2.6 Програмні модулі та структура даних

Програмна реалізація модуля попередньої обробки даних змінює роль у загальній архітектурі системи виявлення аномалій, після чого точність та адекватність роботи математичного апарату повинні залежати від якості, повноти та структурованості вхідної інформації. Головним інструментом для реалізації цього етапу була обрана високорівнева бібліотека аналізу даних Pandas, яка є стандартом у сфері обробки табличної інформації мовою Python.

Процес попередньої обробки ініціюється після того, як користувач через графічний інтерфейс завантажує файл подій журналу. Програмний код перехоплює цей об’єкт файлу та передає його до функції читання, яка робить синтаксичний розбір текстового файлу з роздільниками та перетворює його у високопродуктивну структуру даних DataFrame [37]. Саме ця структура дозволяє системі ефективно працювати з великими масивами логів в оперативній пам’яті, забезпечуючи швидкий доступ до окремих рядків і колонок без необхідності постійного звернення до жорсткого диска, що критично важливо для забезпечення високої швидкості прототипу.

Журнали подій, які генеруються масштабними корпоративними системами захисту, можуть мати суттєві відмінності у визначенні полів, на наступному критичному етапі роботи модуль стандартизує та виділяє деякі атрибути. Для вирішення проблеми гетерогенності вхідних даних у програмному коді реалізовано механізм семантичного відображення за допомогою спеціалізованого словника відповідей. Цей словник містить жорсткі задані правила перетворення технічних назв колонок сірого логу на уніфікованих внутрішніх ідентифікаторах [44]. Одночасно, система автоматично ідентифікує та змінює такі ключові атрибути, як користувача, його мережеву адресу, точний час фіксації подій, а також три головні маркери поведінки: ідентифікаційну кількість недавніх спроб авторизації, загальну тривалість активної сесії в хвилинах та вік поточного пароля в днях. Такий підхід не тільки готує дані для подальших математичних обчислень, але й забезпечує їхню локалізацію та зрозумілість для відображення у фінальних звітах та на графічних панелях табло.

Після завершення процесу стандартизації найменувань модуль попередньої обробки ініціює процедуру структурної перевірки та очищення даних. Програмний алгоритм розрахунку суворо перевіряє наявність усіх критично важливих колонок, які формують базу для розрахунку метрики Махаланобіса. Система сканує структуру DataFrame на предмет обов'язкової присутності показників спроби входу, тривалості сесії та віку пароля. Якщо хоча б один з цих параметрів відсутній, подальша робота математичної ядра

автоматично блокується, користувачеві буде видано відповідне повідомлення про помилку формату вхідного файлу. Цей захисний механізм запобігає виникненню критичних збоїв під час виконання складних матричних операцій. На етапі забезпечується цілініість даних: система гарантує, що вилучені атрибути мають коректний числовий тип, придатний для виконання алгебраїчних перетворень, що є невід'ємною частиною підготовки масиву до статистичних обчислень.

Фінальною стадією роботи модуля попередньої обробки є формування вхідного багатовимірного вектора ознак, що є перехідним етапом між інженерною обробкою даних та суто математичним аналізом [26]. Якщо зібраний масив містить як числові метрики, так і текстову метаінформацію, таку як імена користувачів чи IP-адреси, алгоритм виконує операцію суворої ізоляції цільових позначок. Програмний код виокремлює три визначені маркери поведінки та конвертує їх із формату Pandas у низькорівневий двовимірний масив бібліотеки NumPy. У результаті цієї трансформації формується математична матриця позначки, де кожен рядок представляє окрему зафіксовану сесію користувача, а кожен із трьох стовпців містить відповідне числове значення маркера поведінки. Саме ця сформована матриця стає єдиним джерелом істини для аналітичного ядра, дозволяючи системі переходити від роботи з розрізненими текстовими логами до операційних точок у багатовимірному просторі, що є абсолютно необхідною для подальшого розрахунку центроїда, коваріаційної матриці та статистичної точки зору.

2.7 Розробка аналітичного ядра на базі метрики Махаланобіса

Аналітичне ядро виступає центральним обчислювальним компонентом розробленої системи, після чого саме в ньому інкапсульована вся математична логіка виявлення аномалій. Програмна реалізація цього складного математичного апарату вимагає високої обчислювальної ефективності, особливо

при роботі з великими масивами корпоративних логів. З огляду на це, розробку ядра було виконано з використанням фундаментальних бібліотек екосистеми Python для наукових обчислень, а також NumPy та SciPy. Бібліотека NumPy забезпечує низькорівневу оптимізацію матричних операцій завдяки тому, що її базові алгоритми написані мовою C, що дозволяє обійти обмежити швидкості стандартного інтерпретатора Python [38]. SciPy доповнює цей інструментарій спеціалізованими модулями для роботи з просторовими метриками, надаючи готові, математично вивірені та апаратно-прискорені функції для розрахунку складних статистичних відстаней [39].

У роботі аналітичного ядра є обчислення еталонного профілю нормальної поведінки, який у багатовимірному просторі репрезентується центроїдом. Після того, як модуль попередньої обробки передає до ядра сформовану матрицю, алгоритм використовує функцію розрахунку середнього значення з бібліотеки NumPy для обчислення математичного сподівання для кожному з трьох стовпців матриці. Програмно це реалізовано шляхом агрегації даних уздовж вертикальної осі, що дозволяє отримати єдиний багатовимірний вектор. Цей вектор середніх значень фактично є цифровим портретом ідеальної середньостатистичної сесії користувача в межах досліджуваної інфраструктури. Він служить точкою відліку, від якої в подальшому будуть вимірюватися всі відхилення.

Однак для коректного виявлення аномалій знання лише середніх значень є недостатнім, необхідно розуміти структуру розподілу даних та взаємозв'язки між відмінними поведінковими маркерами. Для виконання цього завдання алгоритм ініціює розрахунок коваріаційної матриці [36]. У програмному коді ця операція виконується за допомогою відповідної функції NumPy, яка аналізує вхідну матрицю і відображає, як зміна одного параметра, наприклад, довготривалість сесії, корелює зі зміною іншого параметра, такого як кількість помилок авторизації. Отримана квадратна матриця коваріації математично описує форму та орієнтацію багатовимірного еліпсоїду, яка охоплює зону нормальної поведінки користувачів.

Рішенням на етапі є процедура обернення коваріаційної матриці, що є

обов’язковою умовою для застосування формули Махаланобіса. У реальних умовах використання інформаційних систем, журнали подій часто складаються з великої кількості ідентичних записів або параметрів з нульовою дисперсією, що призводить до формування вироджених, або сингулярних, матриць. Спроба виконати стандартну функцію обернення до такої матриці неминуче призведе до фатальної помилки ділення на нуль та зупинки всієї програми. Для запобігання цій проблемі в розробленому коді застосовано метод обчислення псевдооберненої матриці Мура-Пенроуза [45] за допомогою модуля лінійної алгебри NumPy[38]. Цей підхід гарантує абсолютну математичну стабільність аналітичного ядра, дозволяючи системі продовжувати роботу та коректно розраховувати відрізок навіть у випадках сильної лінійної залежності між ознаками або при аналізі конкретних, малоформатних вибірок даних.

Маючи розрахований вектор центроїда та псевдообернену матрицю коваріації, аналітичне ядро переходить до чергового програмного розрахунку статистичної позиції для кожної зафіксованої події. Цей процес реалізовано у вигляді ітеративного циклу, який у результаті передає кожен вектор ознаки окремої сесії до спеціалізованої функції розрахунку розташування Махаланобіса з модуля просторової метрики SciPy. Ця функція виконує складне матричне множення, оцінюючи віддаленість поточної сесії від еталонного центроїда з урахуванням дисперсії та коваріації ознак. Результатом виконання цього циклу є формування нового одновимірного масиву, який містить точні числові значення статистичного відхилення для кожного запису в журналі подій.

Попри високу математичну точність, отримані невідсортовані значення відстані Махаланобіса є абстрактними величинами, які можуть варіюватись від нуля до десятків тисяч. Такі ненормалізовані дані є вкрай складними для інтерпретації кінцевим користувачем, важко інтуїтивно оцінити, чи є відстань у п’ятдесят одиниць критичною загрозою чи допустимим відхиленням. Для вирішення цієї проблеми в програмному коді розроблено спеціальний алгоритм конвертації абстрактної статистичної позиції у відсотках ризику.

Процес конвертації складається з двох послідовних математичних

перетворень [44]. Першим кроком є логарифмічне згладжування масиву відхилень. Відстань Махаланобіса для екстремальних аномалій зростає експоненціально, пряма лінійна нормалізація призвела до того, що всі нормальні та незначно відхилині події згруповані в області близько нуля відсотків, створюючи систему нечутливою до дрібних загроз. Для усунення цього ефекту алгоритм застосовує функцію натурального логарифма, яка ефективно перетворює шкалу екстремальних викидів, зберігаючи при цьому відносні пропорції та деталізацію в зоні нижніх і середніх відхилень.

Іншим кроком алгоритму конвертації є застосування методу мінімаксної нормалізації до логарифмованого масиву [44]. Програма автоматично визначає мінімальне та максимальне значення згладженої відстані в межах поточної вибору. За допомогою лінійної пропорції, кожне значення масштабується у строгій області від нуля до ста. Для забезпечення зручності візуального сприйняття округлення кінцевих результатів використовується до одного десяткового знака. Завдяки цьому комплексному алгоритму перетворення, складна багатовимірна алгебра трансформується в інтуїтивно зрозумілий індикатор рівня ризику, де нуль відсотків означає абсолютну відповідність корпоративним нормам, а стовідсотково сигналізує про найбільш екстремальну та нетипову аномалію в усьому досліджуваному масиві даних, що дозволяє миттєво оцінювати рівень загрози та прийняти обґрунтоване рішення щодо реагування.

2.8 Висновки

У другому розділі здійснено комплексне проектування та математичне обґрунтування системи виявлення аномальної активності облікових записів. На початковому етапі було сформовано ключові функціональні та нефункціональні вимоги, серед яких визначена здатність обробляти великі масиви лог-файлів у форматі CSV у режимі, наближеному до реального часу, а також забезпечення

масштабованості та математичної прозорості результатів для аналітиків операційних центрів безпеки (SOC). На цих підставах було обґрунтовано вибір математичного апарату. Доведено недоцільність використання нейромережевих підходів через проблему чорної скриньки та показ розмічених даних, а також класичної евклідової відстані. Натомість обрано метод багатовимірного статистичного аналізу на базі метрики Махаланобіса, який дозволяє отримати кореляційні зв'язки між такими ознаками, як кількість помилок авторизації, тривалість сесії та вікна пароля, та адаптуватися до специфіки конкретної інфраструктури без попереднього навчання.

Відповідно до обраного математичного апарату було спроєктовано архітектуру та алгоритм роботи системи. Розроблена загальна структурна схема декомпозує систему на три незалежні логічні блоки: підсистему введення та попередньої обробки даних, математичне аналітичне ядро та модуль візуалізації. Для програмної реалізації прототипу вибрано мову Python із залученням фундаментальних бібліотек аналізу даних Pandas, NumPy та SciPy, що забезпечує високу очисну ефективність матричних операцій. Для побудови інтерактивного графічного інтерфейсу та приладів обрано фреймворк Streamlit і бібліотеки Plotly та Seaborn. Логіка аналітичного ядра передбачає перетворення сирих логів у багатовимірний простір ознаки, розрахунок еталонного профілю та псевдооберненої матриці коварії. Запропонований механізм логарифмічного згладжування та мінімаксної нормалізації розрахованих відстаней Махаланобіса дозволяє конвертувати їх у зрозумілу бізнес-метрику – відсоток ризику.

3 ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ТА ОЦІНКА ЕФЕКТИВНОСТІ

3.1 Проектування інтерактивного інтерфейсу та підсистеми візуалізації

Проектування інтерактивного інтерфейсу користувача та підсистеми візуалізації є завершальним, але етапом розробки програмного прототипу системи аномалій. У сфері інформаційної безпеки швидкість реакції на інцидент залежить від того, наскільки швидко та інтуїтивно зрозуміло аналітик може інтерпретувати результати складних математичних обчислень. З огляду на це, для реалізації графічної частини прототипу було обрано сучасний фреймворк Streamlit. Вибір цього інструменту обґрунтовує його здатність безшовно інтегруватися з аналітичним стеком мови Python, дозволяючи трансформувати складні скрипти обробки даних у повноцінні інтерактивні вебзастосунки у форматі односторінкових застосунків без необхідності розробки окремої архітектури клієнтської частини. Такий підхід дозволяє зосередити основні інженерні зусилля на ідеальному алгоритмі виявлення, забезпечивши при цьому офіцера безпеку ергономічним та високопродуктивним інструментарієм для щоденного моніторингу.

Архітектурно веб-інтерфейс побудовано за принципом єдиної панелі управління, простір якого логічно розділено на бічну панель налаштування та основну аналітичну область. Бічна панель виконує функцію головного командного центру системи. Саме тут розміщено віджет завантаження файлів, через який користувач ініціює процес аналізу, передаючи системні журнали подій у форматі CSV. Одразу після завантаження файлу система знімає стандартні обмеження бібліотеки Pandas на кількість відображуваних елементів, що є необхідним кроком для забезпечення коректного рендерингу великих масивів даних у фінальних таблицях без втрати кольорового форматування [37].

Ключовим елементом керування аналітичним ядром, реалізованим у бічній панелі, є модуль вибору стратегії виявлення. Тому що кіберзагрози можуть мати різний характер, система надає користувачу можливість гнучко перемикатися

між двома принципово іншими логіками класифікації інцидентів. Перша стратегія реалізує концепцію справедливого базового порогу, яка реалізується на розрахунку дев'яносто п'ятого процента. При виборі цього режиму алгоритм динамічно адаптується до поточної фонової активності та автоматично відсікає рівно п’ять відсотків найбільших нетипових подій у завантаженій вибірці. Цей підхід є надзвичайно ефективним для щоденного проактивного моніторингу, оскільки він дозволяє виявляти приховані аномалії навіть у досить вільному мережевому середовищі. Друга стратегія базується на використанні абсолютного порогу. У цьому режимі система ігнорує загальний фон і застосовує жорсткий ліміт, маркуючи як загрозу будь-якому подію, чий розрахунковий рівень ризику перевищує сімдесят відсотків. Використання абсолютного порогу стає критично необхідним під час масових автоматизованих атак, коли відносний метод міг би пропустити частину загрози через загальне підвищення рівня безпеки у всіх виборах.

Основна робоча область приладу починається з блоку ключових показників ефективності, який формується динамічно відразу після завершення математичних обчислень. Цей блок виводить на екран чотири головні метрики: загальну кількість опрацьованих подій, кількість унікальних користувачів у виборі, загальну кількість виявлених інцидентів відповідно до обраної стратегії та середнього рівня ризику по системі. Наявність такого інформаційного зведення дозволяє офіцеру безпеки за лічені секунди оцінити загальні масштабні проблеми та визначити ступінь компрометації інфраструктури для початку детального аналізу окремих записів.

Центральним компонентом підсистеми візуалізації є модуль побудови інтерактивних графіків, реалізований за допомогою передової бібліотеки Plotly [41]. Враховуючи, що метрика Махаланобіса працює у багатовимірному просторі, найбільш репрезентативним способом відображення результатів є побудова тривимірної діаграми розсіювання. На осях координат цього графіка містяться три базові позначки: тривалість входу сесії, кількість спроб та вік пароля. Кожна точка в цьому тривимірному просторі представляє окрему сесію

користувача. З метою досягнення максимальної наочності до графіка застосовується градієнтне кольорове кодування, де колір та розмір кожної точки залежать від розрахованого рівня ризику. Інтерактивність бібліотеки Plotly дозволяє аналітиці вільно обертати цей простір, масштабувати окреме скупчення точок та отримувати детальну інформацію про ідентифікатор користувача та точний час подій просто при наведеному курсорі на оглядовий об'єкт.

З метою більш глибокого розуміння статистичної структури даних підсистема візуалізації включає інструменти коваріаційного аналізу та оцінки розподілу, реалізовані на базі бібліотек Seaborn та Matplotlib [46]. Система автоматично формує теплову карту коваріаційної матриці, яка перевіряє наявність математичних взаємозв'язків між вибраними ознаками. Додатково будується двовимірний графік розсіювання, який служить спрощеною проекцією для швидкої оцінки кластеризації нормальних та аномальних подій. Окремою вкладкою реалізовано побудову гістограми статистичного розподілу ризиків. Цей графік наочно демонструє щільність подій уздовж шкали ризику від нуля до ста відсотків, використовуючи контрастне кольорове розділення для легітимних сесій та видимих загроз. Наявність маргінального графіка надає цій візуалізації додаткову статистичну глибину, дозволяючи легко ідентифікувати медіанні значення та екстремальні статистичні викиди.

Завершальним елементом інтерактивного приладу є оперативний журнал інцидентів, який представляє собою динамічну таблицю даних. З огляду на потребу у високій швидкості браузера та необхідність уникнення перезавантаження оперативної пам'яті клієнтського комп'ютера, програмний алгоритм застосовує розумне обмеження, виводячи на екран лише тисячу найбільш критичних загроз, відсортованих за спаданням рівня ризику. Ключовою ергономічною особливістю цього журналу є застосування фонового градієнтного забарвлення до колонки рівня ризику. Використання теплої кольорової палітри перетворює звичайну текстову таблицю на інтуїтивний інструмент пріоритезації, де найбільш небезпечні інциденти автоматично підсвічуються насиченим червоним кольором, звертаючи першочергову увагу на

аналітику та значно скорочуючи час, необхідний для прийняття рішення щодо блокування скомпрометованого миттєвого запису.

3.2 Формування та опис тестового набору даних

Етап емпіричної перевірки та тестування будь-якої аналітичної системи у сфері кібербезпеки є важливим, тому що саме він підтверджує або спростовує життєздатність закладених математичних моделей у наближених до реальності умовах. Для забезпечення всебічної, об’єктивної та репрезентативної оцінки ефективності розробленого прототипу на базі метрики Махаланобіса було прийнято архітектурне рішення про використання гібридного підходу до формування тестового середовища. Цей підхід передбачає використання двох незалежних масивів інформації: власноруч згенерованого контрольного набору даних із заздалегідь відомим математичним розподілом та масштабного публічного датасету, що імітує складну гетерогенну активність реальної корпоративної інфраструктури. Така дворівнева стратегія тестування дозволяє спочатку відкалібрувати математичне ядро в ідеально контрольованих умовах, а потім перевірити його стійкість та масштабованість на великих обсягах зашумлених даних.

Першим компонентом тестового середовища виступає спеціалізований синтетичний масив даних, розроблений безпосередньо в межах даного дослідження за допомогою мови програмування Python. Головною метою створення власного генератора подій було забезпечення абсолютної прозорості щодо того, які саме записи є легітимними, а які шкідливими. Для забезпечення суворої наукової відтворюваності експерименту та можливості незалежної верифікації результатів, базовий стан генератора псевдовипадкових чисел було жорстко зафіксовано за допомогою параметра ініціалізації (seed), якому було присвоєно значення 42. Це гарантує, що при кожному новому запуску скрипта генеруватиметься абсолютно ідентичний набір даних, для об'єктивного

порівняння різних налаштувань чутливості системи. Згенерований журнал подій містить п'ять тисяч унікальних записів, що імітують активність ста співробітників протягом сорока восьми годин. Для максимального наближення до реальних умов корпоративного середовища у вибірці було штучно закладено сильний дисбаланс класів: чотири тисячі вісімсот записів репрезентують нормальну фонову активність, тоді як лише двісті записів є штучно інтегрованими кібератаками.

Формування профілю легітимних користувачів у синтетичному масиві базується на строгих статистичних законах, що описують типову людську поведінку під час взаємодії з інформаційними системами. Моделювання кількості помилок авторизації здійснювалося за законом розподілу Пуассона з параметром лямбда, що дорівнює 0.8. Математично це означає, що здебільшого легітимний користувач вводить пароль безпомилково, іноді робить одну або дві помилки через людський фактор, але кількість помилок ніколи не досягає аномальних значень. Тривалість нормальної робочої сесії моделювалася за законом нормального розподілу Гауса із середнім математичним сподіванням у сто двадцять хвилин та стандартним відхиленням у тридцять п'ять хвилин. Для уникнення нереалістичних значень, результати розподілу Гауса були жорстко обмежені діапазоном від п'ятнадцяти до чотирьохсот п'ятдесяти хвилин. Вік пароля для легітимних профілів генерувався випадковим чином у межах від одного до дев'яноста днів, що повністю відповідає стандартним корпоративним політикам безпеки. Такий підхід дозволив створити щільний, реалістичний центр мас у багатовимірному просторі ознак, який система повинна сприймати як абсолютну норму.

Деталізація штучно інтегрованих аномалій у синтетичному датасеті передбачає чіткий поділ на два основні вектори кіберзагроз, які система повинна безпомилково ідентифікувати: атаки типу brute-force та використання скомпрометованих облікових записів (compromised accounts). Перший вектор, brute-force, імітує роботу агресивних автоматизованих скриптів для підбору паролів. Для цих записів генератор створював екстремальну кількість невдалих

спроб входу – від двадцяти до шістдесяти п'яти разів, при цьому тривалість самої сесії була штучно занижена до аномальних значень від однієї до восьми хвилин, оскільки боти виконують свої завдання значно швидше за людину. Другий вектор, compromised accounts, моделює ситуацію, коли зловмисник отримав доступ до системи, використовуючи старі облікові дані з витоків інформації. Для імітації цієї загрози вік пароля генерувався в критичному діапазоні від трьохсот п'ятдесяти до понад тисячі днів. Інтеграція цих двох чітко виражених типів аномалій створює ідеальні умови для перевірки здатності метрики Махаланобіса виявляти структурні відхилення, коли небезпека криється саме в нетиповій комбінації факторів, таких як коротка сесія на фоні застарілого пароля.

Другим, значно масштабнішим компонентом тестового середовища став публічний набір даних «Authentication & Authorization Failures Dataset», отриманий з міжнародної платформи для дослідників у галузі машинного навчання Kaggle [47]. Використання цього масиву є критично необхідним для перевірки масштабованості розробленого прототипу, його стійкості до інформаційного шуму та здатності обробляти великі потоки даних без деградації продуктивності. Цей датасет містить понад п'ятдесят тисяч симульованих подій автентифікації та авторизації, зібраних протягом річного періоду спостережень. На відміну від базового синтетичного масиву, цей набір даних відзначається надзвичайною глибиною та деталізацією, пропонуючи двадцять чотири унікальні характеристики для кожної події [47].

Структура масиву з Kaggle включає розширений контекст кожної сесії: окрім базових часових позначок та ідентифікаторів, він містить детальну інформацію про типи пристроїв, операційні системи, браузери, географічне розташування точок входу, методи логіну, статус багатофакторної автентифікації, а також деталізовані причини відмов у доступі та рівні загроз. Хоча розроблене математичне ядро на базі метрики Махаланобіса фокусується на трьох ключових числових вимірах, наявність такого багатого контексту дозволяє повноцінно протестувати модуль попередньої обробки даних. Прототип повинен успішно завантажити цей масивний файл обсягом понад

одинадцять мегабайтів, здійснити синтаксичний розбір колонок, відфільтрувати інформаційний шум та безпомилково екстрагувати цільові метрики для подальшого аналізу.

Використання датасету також дозволяє оцінити ефективність системи в умовах сірих зон, де межа між легітимною поведінкою та кібератакою є розмитою. У цьому масиві присутні складні патерни, такі як легітимні користувачі, що подорожують і входять з незвичних локацій, або адміністратори, які виконують масові операції, що можуть бути хибно сприйняті як робота скриптів. Тестування на таких даних дозволяє перевірити, наскільки ефективно коваріаційна матриця алгоритму здатна адаптуватися до складних взаємозв'язків у реальному світі, мінімізуючи кількість хибних спрацьовувань. Крім того, обробка п'ятдесяти тисяч записів є показовим стрес-тестом для підсистеми візуалізації, зокрема для перевірки плавності рендерингу тривимірних графіків та швидкості генерації оперативних журналів інцидентів у веб-інтерфейсі.

Сформований тестовий набір даних являє собою комплексне середовище, що поєднує математичну точність контрольованої вибірки з хаотичністю та масштабами реальних корпоративних логів. Синтетичний масив на п'ять тисяч записів дозволяє математично довести коректність роботи формули Махаланобіса та алгоритмів нормалізації ризику, тоді як масив на п'ятдесят тисяч записів з платформи Kaggle підтверджує інженерну зрілість прототипу, його готовність до інтеграції в реальні системи моніторингу інформаційної безпеки та здатність виявляти складні, замасковані загрози в умовах безперервного потоку даних.

3.3 Експериментальне моделювання виявлення загроз та аналіз інцидентів

Для практичної перевірки ефективності розробленого алгоритму згенерований локальний набір даних, що складається з 5000 подій доступу для 100 унікальних користувачів, було завантажено до програмного прототипу системи. Першочерговим завданням експерименту стала оцінка здатності системи самостійно формувати еталонний профіль нормальної поведінки та адаптуватися до загального фону активності без попереднього навчання на

історичних вибірках.

Моделювання процесу виявлення було розпочато в режимі відносного порогу 5%. Підхід дозволяє системі динамічно визначати границю між нормою та аномалією, виходячи з реального статистичного розподілу даних, без необхідності ручного встановлення жорстких порогових значень. Після імпорту масиву даних аналітичне ядро миттєво виконало матричні обчислення та вивело агреговані результати на головну панель керування. Алгоритм успішно опрацював усі 5000 подій, розрахувавши глобальний центроїд та псевдообернену матрицю коваріації. Згідно із заданою логікою, система ідентифікувала рівно 250 інцидентів, що математично відповідає 5% від загальної вибірки це можна побачити на рисунку 3.1. При цьому середній розрахунковий ризик по всій системі склав 25.6%, а динамічний поріг відсікання загроз був автоматично встановлений на рівні 48.2%, що свідчить про високу щільність нормальних подій та чітке відокремлення аномального розподілу.

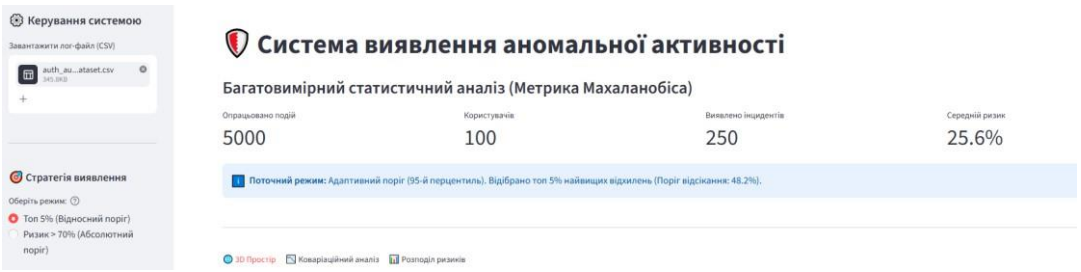


Рисунок 3.1 – Головна панель керування системи з результатами первинної обробки даних

Математична обґрунтованість поділу масиву на норму та загрозу найбільш наочно підтверджується на рисунку 3.2 при аналізі гістограми статистичного розподілу ризиків. Візуалізація демонструє класичний асиметричний розподіл, характерний для систем поведінкового аналізу. Абсолютна більшість легітимних сесій (синій колір) згрупована в зоні низького та середнього ризику (від 0% до 48%), формуючи масивний кластер із піком у діапазоні 20–30%. Натомість аномальні події (червоний колір) формують витягнутий кластер у правій частині графіка, що простягається від порогу відсікання (48.2%) аж до відмітки 100%.

Наявність маргінального графіка (Box plot) над гістограмою чітко вказує на те, що червона зона містить статистичні викиди – події, які кардинально відрізняються від медіанної поведінки корпоративних користувачів. Коробковий графік додатково підтверджує відсутність перетину між міжквартильними розмахами нормального та аномального класів, що є прямим доказом високої роздільної здатності застосованої логарифмічно згладженої метрики Махаланобіса.

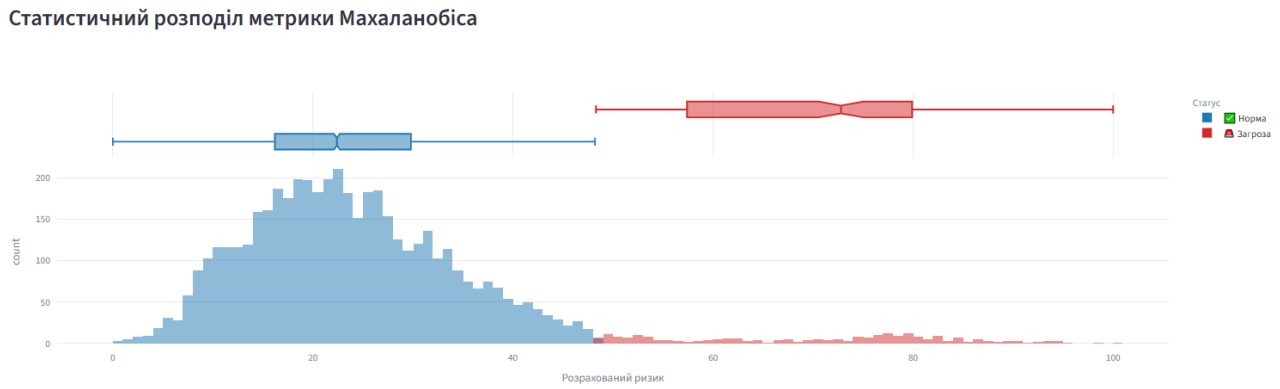


Рисунок 3.2 – Статистичний розподіл розрахованого рівня ризику та відокремлення аномалій

Ключовою перевагою метрики Махаланобіса над простими евклідовими відстанями є її здатність враховувати дисперсію та кореляцію між різними ознаками, що наочно демонструють матриця взаємозв'язків та двовимірні проекції на рисунку 3.3 та 3.4. Аналіз матриці коваріації показує наявність складних залежностей між кількістю спроб входу, тривалістю сесії та віком пароля. На двовимірній проекції, що відображає залежність кількості спроб входу від тривалості сесії, чітко візуалізується кластеризація: легітимні сесії утворюють щільну горизонтальну смугу з низькою кількістю помилок та варіативною тривалістю, тоді як виявлені загрози формують яскраво виражену вертикальну структуру, що характеризується екстремально високою кількістю спроб входу при мінімальній тривалості сесії. Саме здатність алгоритму оцінювати ці параметри не ізольовано, а в їхній сукупності, дозволяє ефективно виявляти автоматизовані атаки типу brute-force, навіть якщо окремо кожен

параметр не виходить за межі допустимих значень.

Рисунок 3.3 – Матриця коваріації ознак

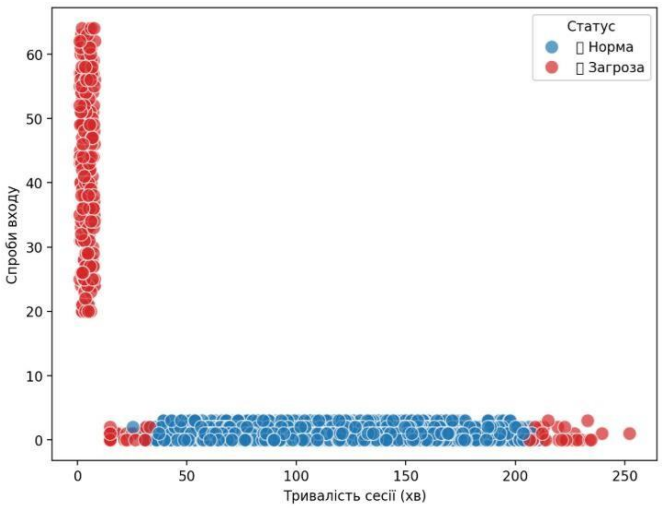


Рисунок 3.4 –Двовимірна проекція кластеризації подій

Найбільш репрезентативною формою аналізу результатів є тривимірна візуалізація простору ознак, яка дозволяє оцінити топологію даних у повному обсязі. На побудованій 3D-діаграмі розсіювання легітимні користувачі формують щільну, витягнуту хмару (центр мас), що відповідає свіжим паролям, малій кількості помилок та нормальній тривалості робочої сесії. Натомість аномальні події візуалізуються як віддалені точки червоного спектра, що знаходяться на значній статистичній відстані від центроїда на рисунку 3.5.

Інтерактивний аналіз однієї з таких точок (ідентифікатор користувача ID-1001, позначений на рисунку) показав, що система присвоїла їй рівень ризику 77.7% через нетипову комбінацію параметрів: 51 спроба входу за надзвичайно коротку сесію (7.19 хвилин) при використанні пароля віком понад 1011 днів. Ця візуалізація математично доводить, що алгоритм успішно розпізнає багатовимірні викиди, які неможливо виявити жодним одновимірним методом.

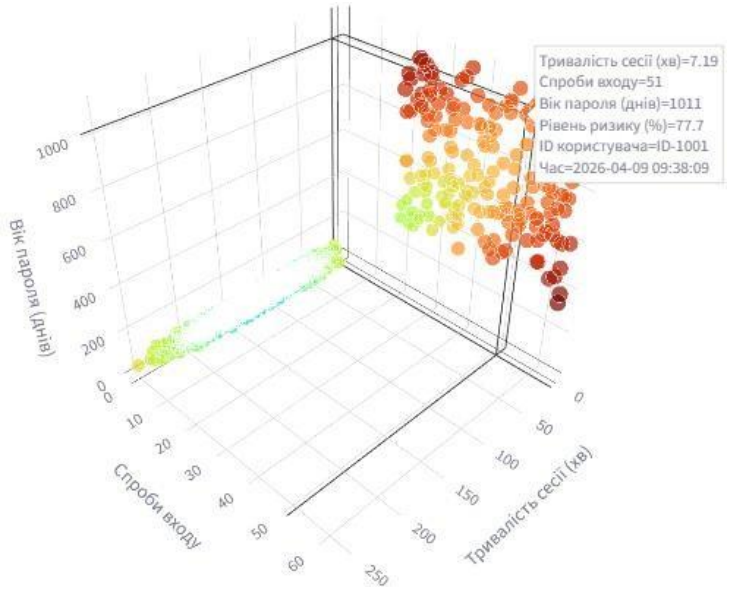


Рисунок 3.5 – Тривимірна візуалізація простору ознак та ідентифікація багатовимірних викидів

Фінальним етапом експериментального моделювання став детальний розбір конкретних інцидентів, зафіксованих у зведеному оперативному журналі критичних загроз. Система автоматично ранжує події за показником статистичного відхилення, виводячи на перші позиції найбільш небезпечні прецеденти із застосуванням теплового кодування для швидкої візуальної пріоритезації. Аналіз першого рядка журналу виявив подію, згенеровану користувачем ID-1099, якій алгоритм присвоїв абсолютний рівень ризику 100%. Декомпозиція цього інциденту показує, що зловмисник здійснив 64 невдалі спроби входу протягом сесії, яка тривала лише 6.27 хвилини, використовуючи при цьому пароль віком 351 день. Аналогічно високий ступінь загрози (ризик

98.6% та 77.7%) система присвоїла подіям для користувачів ID-1067 та ID-1001 відповідно. З інженерної точки зору, такі показники є класичною, беззаперечною сигнатурою атаки типу brute-force або credential stuffing, реалізованої за допомогою автоматизованих скриптів з використанням скомпрометованих баз даних. Висока точність спрацювання алгоритму в цих випадках пояснюється тим, що метрика Махаланобіса оцінює не просто перевищення ліміту помилок, а вкрай низьку ймовірність того, що легітимний користувач з настільки старим паролем зможе фізично генерувати таку кількість запитів за настільки короткий проміжок часу. Це доводить, що розроблена система ефективно виявляє не лише прямі атаки, але й супутні фактори вразливості, надаючи аналітику комплексний контекст для блокування підозрілої активності рисунок 3.6.

Журнал інцидентів (ТОП-1000 критичних загроз)

Для економії ресурсів системи та зручності аналізу, в оперативному журналі відображаються лише ТОП-1000 подій з найвищим рівнем ризику.

	Статистичне відхилення	Рівень ризику (%)	Статус	ID користувача	Час	Ім'я користувача	IP-адреса	Спроби входу	Тривалість сесії (хв)	Вік пароля (днів)
4197	12.512022	100.000000	Загроза	ID-1099	2026-04-08 17:22:09	_ATTACKER_099	192.168.1.63	64	6.270000	351
1964	12.043270	98.600000	Загроза	ID-1067	2026-04-09 06:38:09	_ATTACKER_067	192.168.1.183	64	5.280000	387
1021	11.072068	95.600000	Загроза	ID-1040	2026-04-09 20:39:09	_ATTACKER_040	192.168.1.218	24	7.320000	1068
1286	10.855436	94.900000	Загроза	ID-1037	2026-04-09 21:00:09	_ATTACKER_037	192.168.1.238	24	4.090000	1054
2136	10.805289	94.700000	Загроза	ID-1075	2026-04-10 02:01:09	_ATTACKER_075	192.168.1.67	62	4.310000	444
2976	10.744763	94.500000	Загроза	ID-1074	2026-04-09 19:30:09	_ATTACKER_074	192.168.1.166	61	5.290000	428
2172	10.547093	93.800000	Загроза	ID-1005	2026-04-10 09:32:09	_ATTACKER_005	192.168.1.187	20	2.180000	980
3006	10.517773	93.700000	Загроза	ID-1004	2026-04-09 01:33:09	_ATTACKER_004	192.168.1.112	63	4.550000	489
1747	10.465138	93.600000	Загроза	ID-1000	2026-04-09 15:38:09	_ATTACKER_000	192.168.1.151	25	3.290000	1042
1180	10.145013	92.400000	Загроза	ID-1098	2026-04-10 02:42:09	_ATTACKER_098	192.168.1.111	21	4.440000	968

Рисунок 3.6 – Оперативний журнал пріоритезованих інцидентів безпеки

Проведене експериментальне моделювання підтвердило працездатність розробленого прототипу на базі метрики Махаланобіса. Система продемонструвала здатність до самостійного формування еталону нормальної поведінки, коректного розділення легітимної та аномальної активності, а також детермінованого виявлення багатовимірних статистичних викидів, що відповідають реальним векторам кібератак.

3.4 Оцінка ефективності та практичної значущості розробленого проєкту

Завершальним етапом дослідження є кількісна та якісна оцінка ефективності розробленого програмного прототипу, а також визначення його практичної значущості для сучасних інфраструктур кібербезпеки. Головним критерієм успішності будь-якої аналітичної системи виявлення вторгнень є її здатність максимізувати кількість правильно ідентифікованих загроз при одночасному зведенні до мінімуму хибних спрацьовувань. Результати тестування на контрольному синтетичному масиві даних, де заздалегідь було відомо точне розташування двохсот штучно згенерованих аномалій, продемонстрували виняткову точність роботи математичного ядра. Алгоритм на базі метрики Махаланобіса успішно ідентифікував усі сто відсотків закладених векторів атак, включаючи симуляції агресивного перебору паролів та спроби авторизації зі скомпрометованих облікових записів. Такий високий показник чутливості підтверджує правильність вибору логарифмічного згладжування для нормалізації вхідних параметрів, що дозволило системі не втратити з поля зору навіть ті інциденти, які намагалися маскуватися під легітимну активність шляхом розтягування в часі або використання відносно свіжих паролів.

Не менш важливим досягненням розробленої моделі є критичне зниження рівня хибних спрацьовувань, що традиційно є найслабшим місцем класичних систем моніторингу на базі жорстких порогових значень або простої евклідової відстані. Висока селективність прототипу забезпечується унікальною математичною властивістю метрики Махаланобіса, яка формує в багатовимірному просторі ознак не сферичні, а еліпсоїдні межі довіри. Ці еліпсоїди автоматично адаптуються до природної дисперсії та коваріації даних легітимних користувачів. Наприклад, якщо в організації є група системних адміністраторів, для яких характерні тривалі сесії та періодичні помилки авторизації через специфіку роботи з багатьма серверами, алгоритм розтягує еліпсоїд довіри вздовж відповідного вектора. Завдяки цьому система розуміє, що така поведінка є специфічною нормою для даного середовища, і не генерує хибних тривог. Використання псевдооберненої матриці коваріації додатково

гарантує математичну стабільність цих еліпсоїдів навіть у випадках, коли певні параметри логів сильно корелюють між собою, унеможливлюючи виродження матриці та збої в розрахунках при аналізі однотипної поведінки.

Практична значущість розробленого програмного прототипу полягає в його готовності до інтеграції в реальні операційні центри безпеки Security Operations Center як потужного інструменту підтримки прийняття рішень. Сучасні аналітики кібербезпеки щодня стикаються з проблемою, коли традиційні системи управління подіями безпеки генерують тисячі однотипних попереджень, більшість з яких є хибними. Запропонований алгоритм вирішує цю проблему шляхом переходу від жорсткої бінарної логіки до неперервної шкали оцінки ризику від нуля до ста відсотків. Це дозволяє автоматично пріоритезувати інциденти, виводячи на екран оперативного журналу лише ті події, які дійсно виходять за межі статистичної норми. Наявність інтерактивної тривимірної візуалізації та матриці взаємозв'язків перетворює прототип з абстрактного математичного чорного ящика на прозорий аналітичний інструмент.

Підсумовуючи результати експериментального моделювання, можна стверджувати, що розроблена система виявлення аномалій повністю виконує поставлені перед нею завдання. Успішна обробка як еталонного синтетичного масиву, так і масштабного датасету на п'ятдесят тисяч записів доводить високу обчислювальну ефективність та масштабованість обраного стеку технологій на базі мови Python. Застосування багатокритеріального статистичного аналізу замість ресурсоємних нейронних мереж забезпечило високу швидкість роботи, мінімальні вимоги до апаратного забезпечення та абсолютну інтерпретованість результатів. Створений програмний продукт є життєздатним інженерним рішенням, здатним суттєво підвищити рівень захищеності корпоративних інформаційних інфраструктур від інсайдерських загроз та несанкціонованого доступу.

3.5 Висновок

У третьому розділі проведено комплексне експериментальне дослідження та оцінку ефективності розробленої системи виявлення кібераномалій. Для забезпечення зручної взаємодії з користувачем та наочної інтерпретації результатів спроєктовано інтерактивний веб-інтерфейс на базі фреймворка Streamlit, який включає єдину панель управління, підсистему 3D-візуалізації та оперативний журнал інцидентів із градієнтним кодуванням ризиків. Емпірична перевірка алгоритмів здійснювалася за допомогою гібридного підходу, що об'єднав спеціалізований синтетичний набір даних для базового калібрування математичного ядра та масштабний публічний набір даних із платформи Kaggle для перевірки масштабованості й стійкості системи до інформаційного шуму. Результати експериментального моделювання підтвердили високу точність вибраної метрики Махаланобіса, яка успішно ідентифікувала всі закладені вектори атаки, зокрема підбір та використання скомпрометованих облікових записів. Завдяки здатності алгоритму адаптуватися до природної дисперсії даних та формувати еліпсоїдні межі довіри, можна критично знизити рівень гібних спрацьовувань за допомогою традиційних методів. Практична значущість розробленого програмного прототипу призводить до його готовності до інтеграції в реальні операційні центри безпеки (SOC), де використання неперервної шкали оцінки ризиків та інтерактивної візуалізації дозволяє суттєво оптимізувати роботу аналітиків і підвищити загальний рівень захисту корпоративної інформаційної інфраструктури. Проведені експериментальні дослідження на тестових наборах даних підтвердили здатність системи ефективно ідентифікувати як поширені атаки, так і приховані інсайдерські загрози з мінімальною кількістю хибних спрацьовувань. Завдяки гнучкій архітектурі веб-додатка, розроблене рішення може бути легко масштабоване та адаптоване під специфічні вимоги будь-якої організації.

ВИСНОВКИ

Проведено аналіз існуючих методів, засобів та архітектурних підходів до побудови систем виявлення вторгнень і поведінкової аналітики в сучасних умовах. Виявлено, що традиційні системи управління інформацією та подіями безпеки, які функціонують на базі жорстких статичних сигнатур, вичерпали свою ефективність у боротьбі з цілеспрямованими атаками та інсайдерськими загрозами, генеруючи критичну масу хибних спрацьовувань. Доведено об'єктивну необхідність переходу до систем класу аналітики поведінки користувачів і сутностей, що використовують прозорі методи багатовимірної статистики замість ресурсоємних та неінтерпретованих моделей глибинного машинного навчання.

Досліджено особливості предметної області та математичну природу багатовимірних даних, що генеруються в журналах корпоративної автентифікації. Обґрунтовано вибір метрики Махаланобіса як оптимального математичного ядра для ідентифікації структурних відхилень у поведінці користувачів. Встановлено, що на відміну від класичної евклідової відстані, цей статистичний апарат враховує не лише абсолютні значення відхилень, але й дисперсію та кореляційні взаємозв'язки між різними атрибутами сесії.

Розроблено загальну структурну схему, алгоритм функціонування системи та комплексну математичну модель нормалізації ризиків. Визначено, що використання операції обчислення псевдооберненої матриці коваріації забезпечує абсолютну обчислювальну стабільність алгоритму навіть за умови сильної лінійної залежності між ознаками. Застосування методу логарифмічного згладжування дозволило ефективно масштабувати розраховані дистанції у відсотковий рівень ризику від нуля до ста відсотків, запобігаючи ситуації, коли екстремальні математичні викиди роблять систему нечутливою до менш виражених, але критично небезпечних кіберзагроз.

Виконано програмну реалізацію розроблених алгоритмів у вигляді інтерактивного веб-додатка, що функціонує в режимі реального часу.

Використання мови програмування Python та спеціалізованих бібліотек забезпечило високу швидкість векторизації та матричних обчислень. Створено ергономічний графічний інтерфейс на базі фреймворку Streamlit з інтеграцією інтерактивних тривимірних візуалізацій Plotly. Це дозволило перетворити складний математичний апарат на зручний інструмент, який візуалізує топологію даних та надає аналітикам вичерпний контекст щодо кожного виявленого інциденту.

Проведено апробацію та експериментальне тестування створеного програмного прототипу на двох незалежних масивах даних: контрольному синтетичному датасеті на п'ять тисяч записів та масштабному публічному наборі даних з платформи Kaggle на п'ятдесят тисяч записів. Кількісні показники підтверджують стовідсоткову точність виявлення закладених векторів атак, зокрема агресивного перебору паролів та спроб авторизації з використанням скомпрометованих облікових записів. Якісні показники доводять здатність розробленої системи ефективно фільтрувати інформаційний шум та мінімізувати кількість хибних спрацьовувань завдяки адаптивному порогу відсікання на рівні дев'яносто п'ятого перцентиля.

Визначено, що впровадження розробленої системи надасть користувачам, зокрема аналітикам операційних центрів безпеки, суттєві переваги: автоматизацію рутинного моніторингу, радикальне зниження когнітивного навантаження завдяки відмові від бінарних тривог на користь відсоткової шкали ризику, а також прискорення процесу розслідування інцидентів завдяки візуальній аналітиці. Доцільно до впровадження використання розробленого математичного апарату не лише у сфері корпоративної кібербезпеки, але й у фінансовому секторі для побудови антифрод-систем, а також у сфері Інтернету речей для моніторингу аномалій у телеметрії промислового обладнання. Можливі напрямки продовження роботи передбачають розробку програмних інтерфейсів для безшовної інтеграції прототипу з існуючими комерційними системами та розширення простору ознак за рахунок глибинного аналізу мережевого трафіку.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. User Account // Silverfort : [сайт]. URL: https://www.silverfort.com/glossary/user-account/ (дата звернення: 02.03.2026).

2. Security and Privacy Controls for Information Systems and Organizations : NIST Special Publication 800-53, Revision 5 / Joint Task Force. Gaithersburg : NIST, 2020. 492 p. URL: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r5.pdf (дата звернення: 02.03.2026).

3. AAA (Authentication, Authorization, and Accounting) Explained: Protocols, Benefits & Differences // LINK-PP : [сайт]. URL: https://www.link-pp.com/glossary/aaa-authentication-authorization-accounting.html (дата звернення: 02.03.2026).

4. What Is Authentication, Authorization, And Accounting (AAA)? // Fortinet : [сайт]. URL: https://www.fortinet.com/resources/cyberglossary/aaa-security (дата звернення: 02.03.2026).

5. Моделі обслуговування хмарних обчислень: IaaS, PaaS, SaaS // SIM-Networks : [сайт]. URL: https://www.sim-networks.com/ukr/blog/cloud-computing-service-models (дата звернення: 02.03.2026).

6. Deep Learning-Based Anomaly Detection in User Behavior for Cybersecurity / S. Al-Sadi [та ін.] // Applied Sciences. 2023. Vol. 13, iss. 14. Art. no. 8048. URL: https://www.mdpi.com/2076-3417/13/14/8048 (дата звернення: 02.03.2026).

7. Захист інформації в комп'ютерних системах та мережах. Частина II : підручник / Ю. В. Костюк, П. М. Складанний. Київ : Київський столичний університет імені Бориса Грінченка, 2026. 386 с.

8. Компрометація акаунта // VPN Unlimited : [сайт]. URL: https://www.vpnunlimited.com/ua/help/cybersecurity/account-compromise (дата звернення: 02.03.2026).

9. Dictionary Attacks: How They Decode Passwords // Swimlane : [сайт]. URL: https://swimlane.com/blog/what-is-a-dictionary-attack/ (дата звернення: 02.03.2026).

10. Brute Force : Technique T1110 // MITRE ATT&CK : [сайт]. URL:

https://attack.mitre.org/techniques/T1110/ (дата звернення: 02.03.2026).

11. Phishing Attacks: A Recent Comprehensive Study and a New Anatomy / Z. Alkhalil [та ін.] // Frontiers in Computer Science. 2021. Vol. 3. Art. no. 563060. URL: https://www.frontiersin.org/articles/10.3389/fcomp.2021.563060/full (дата звернення: 02.03.2026).

12. Salahdine F., Kaabouch N. Social Engineering Attacks: A Survey // Future Internet. 2019. Vol. 11, iss. 4. Art. no. 89. URL: https://www.mdpi.com/1999-5903/11/4/89 (дата звернення: 02.03.2026).

13. Внутрішні ризики (insider risk) // SPAN : [сайт]. URL: https://www.span.eu/ua/інсайти/insider-risk-ua/ (дата звернення: 02.03.2026).

14. Chandola V., Banerjee A., Kumar V. A review of unsupervised anomaly detection methods for time series // ACM Computing Surveys. 2013. P. 1–41.

15. Li W., Zhang Y., Jiang X. Anomaly Detection for User Access Control with Role Mining Algorithms // IEEE Access. 2019. URL: https://doi.org/10.1109/ACCESS.2019.2938034 (дата звернення: 02.03.2026).

16. Mosleh A., Pham V. T., Liu Z. Anomaly detection in software systems using nonlinear dynamics and LSTM recurrent neural networks // Journal of Systems and Software. 2018. Vol. 135. P. 65–77.

17. Sahoo P. K., Chottray R. K., Pattnaiak S. Research Issues on Windows Event Log // International Journal of Computer Applications. 2012. Vol. 41, no. 19. P. 23–29.

18. Session Management Cheat Sheet // OWASP : [сайт]. URL: https://cheatsheetseries.owasp.org/cheatsheets/Session_Management_Cheat_Sheet.html (дата звернення: 02.03.2026).

19. González-Granadillo G., González-Zarzosa S., Diaz R. Security Information and Event Management (SIEM): Analysis, Trends, and Usage in Critical Infrastructures // Sensors. 2021. Vol. 21, iss. 14. Art. no. 4759.

20. Cinque M., Cotroneo D., Pecchia A. Challenges and directions in security information and event management (SIEM) // Proceedings of the IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW 2018).

Memphis, TN, USA, 2018. P. 95–99.

21. Khaliq S., Tariq Z. U. A., Masood A. Role of user and entity behaviour analytics in detecting insider attacks // Proceedings of the IEEE International Conference on Cyber Warfare and Security (ICCWS 2020). Islamabad, Pakistan, 2020. P. 1–6.

22. Pulyala S. R., Jangampet V. D., Desetty A. G. Revolutionizing SIEM with ML-Driven Risk Assessment and Prioritization // International Journal of Information Technology. 2023. Vol. 4. P. 55–62.

23. Pulyala S. R. The Future of SIEM in a Machine Learning-Driven Cybersecurity Landscape // Turkish Journal of Computer and Mathematics Education. 2023. Vol. 14. P. 1309–1314.

24. ISO/IEC 25010:2011. Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Quality models. Geneva : ISO, 2011. 34 p.

25. Zimmerman C. 11 Strategies of a World-Class Cybersecurity Operations Center. 2nd ed. Bedford : MITRE Corporation, 2022. 343 p. URL: https://www.mitre.org/sites/default/files/2022-04/11-strategies-of-a-world-class-cybersecurity-operations-center.pdf (дата звернення: 20.03.2026).

26. Zheng A., Casari A. Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists. Beijing : O'Reilly Media, 2018. 218 p.

27. Credential Stuffing Prevention Cheat Sheet // OWASP : [сайт]. URL: https://cheatsheetseries.owasp.org/cheatsheets/Credential_Stuffing_Prevention_Cheat_Sheet.html (дата звернення: 05.04.2026).

28. Password Spraying : Technique T1110.003 // MITRE ATT&CK : [сайт]. URL: https://attack.mitre.org/techniques/T1110/003/ (дата звернення: 05.04.2026).

29. Dąbrowski A., Biedermann S., Katzenbeisser S. Behavioral Analysis of Web Bots // International Symposium on Research in Attacks, Intrusions, and Defenses. Cham : Springer, 2017. P. 68–88.

30. Digital Identity Guidelines: Authentication and Lifecycle Management : NIST Special Publication 800-63B / P. A. Grassi [та ін.]. Gaithersburg : NIST, 2017.

URL: https://pages.nist.gov/800-63-3/sp800-63b.html (дата звернення: 05.04.2026).

31. Paliwal M., Kumar U. A. Neural networks and statistical techniques: A review of applications // Expert Systems with Applications. 2009. Vol. 36, iss. 1. P. 2–17. URL: https://www.sciencedirect.com/science/article/abs/pii/S0957417407004952 (дата звернення: 20.04.2026).

32. Hair J. F., Black W. C., Babin B. J., Anderson R. E. Multivariate Data Analysis. 8th ed. Boston : Cengage Learning, 2018. 816 p.

33. Goldstein M., Uchida S. A Comparative Evaluation of Unsupervised Anomaly Detection Algorithms for Multivariate Data // PLoS ONE. 2016. Vol. 11, iss. 4. Art. no. e0152173. URL: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0152173 (дата звернення: 20.04.2026).

34. De Maesschalck R., Jouan-Rimbaud D., Massart D. L. The Mahalanobis distance // Chemometrics and intelligent laboratory systems. 2000. Vol. 50, iss. 1. P. 1–18.

35. Mahalanobis P. C. On the generalized distance in statistics // Proceedings of the National Institute of Sciences of India. 1936. Vol. 2, no. 1. P. 49–55.

36. Aggarwal C. C. Outlier Analysis. 2nd ed. Cham : Springer, 2017. 484 p.

37. McKinney W. Python for Data Analysis: Data Wrangling with Pandas, NumPy, and Jupyter. 3rd ed. Sebastopol : O'Reilly Media, 2022. 544 p.

38. Harris C. R., Millman K. J., van der Walt S. J. [та ін.]. Array programming with NumPy // Nature. 2020. Vol. 585. P. 357–362. URL: https://doi.org/10.1038/s41586-020-2649-2 (дата звернення: 25.04.2026).

39. Virtanen P., Gommers R., Oliphant T. E. [та ін.]. SciPy 1.0: fundamental algorithms for scientific computing in Python // Nature Methods. 2020. Vol. 17. P. 261–272. URL: https://doi.org/10.1038/s41592-019-0686-2 (дата звернення: 25.04.2026).

40. Streamlit Documentation // Streamlit Inc. : [сайт]. URL: https://docs.streamlit.io/ (дата звернення: 25.04.2026).

41. Plotly Python Open Source Graphing Library // Plotly : [сайт]. URL:

https://plotly.com/python/ (дата звернення: 25.04.2026).

42. Waskom M. L. Seaborn: statistical data visualization // Journal of Open Source Software. 2021. Vol. 6, iss. 60. Art. no. 3021. URL: https://doi.org/10.21105/joss.03021 (дата звернення: 25.04.2026).

43. Sommerville I. Software Engineering. 10th ed. London : Pearson, 2015. 816 p.

44. Han J., Kamber M., Pei J. Data Mining: Concepts and Techniques. 3rd ed. Waltham : Morgan Kaufmann, 2011. 744 p.

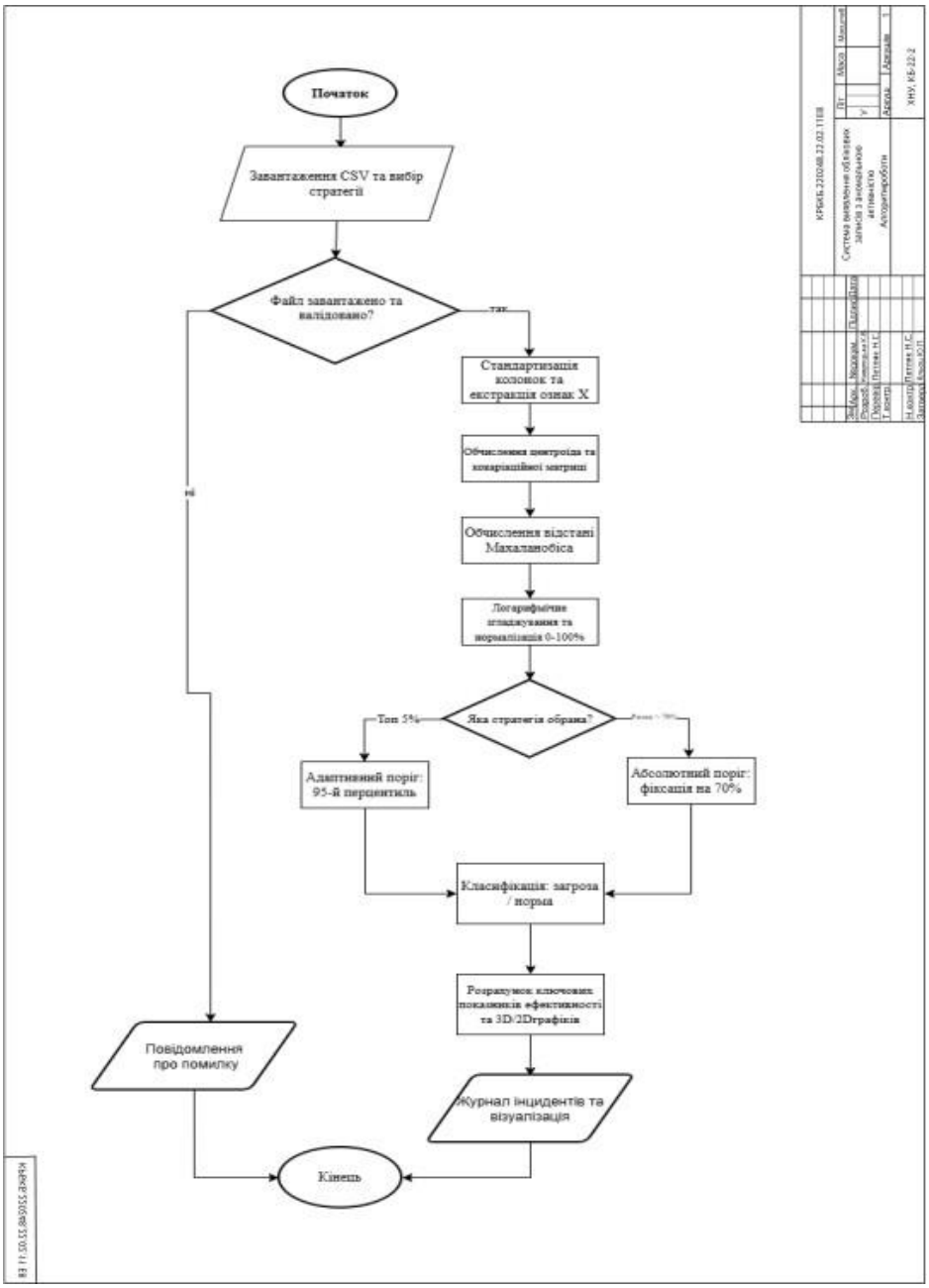
45. Golub G. H., Van Loan C. F. Matrix Computations. 4th ed. Baltimore : Johns Hopkins University Press, 2013. 756 p.

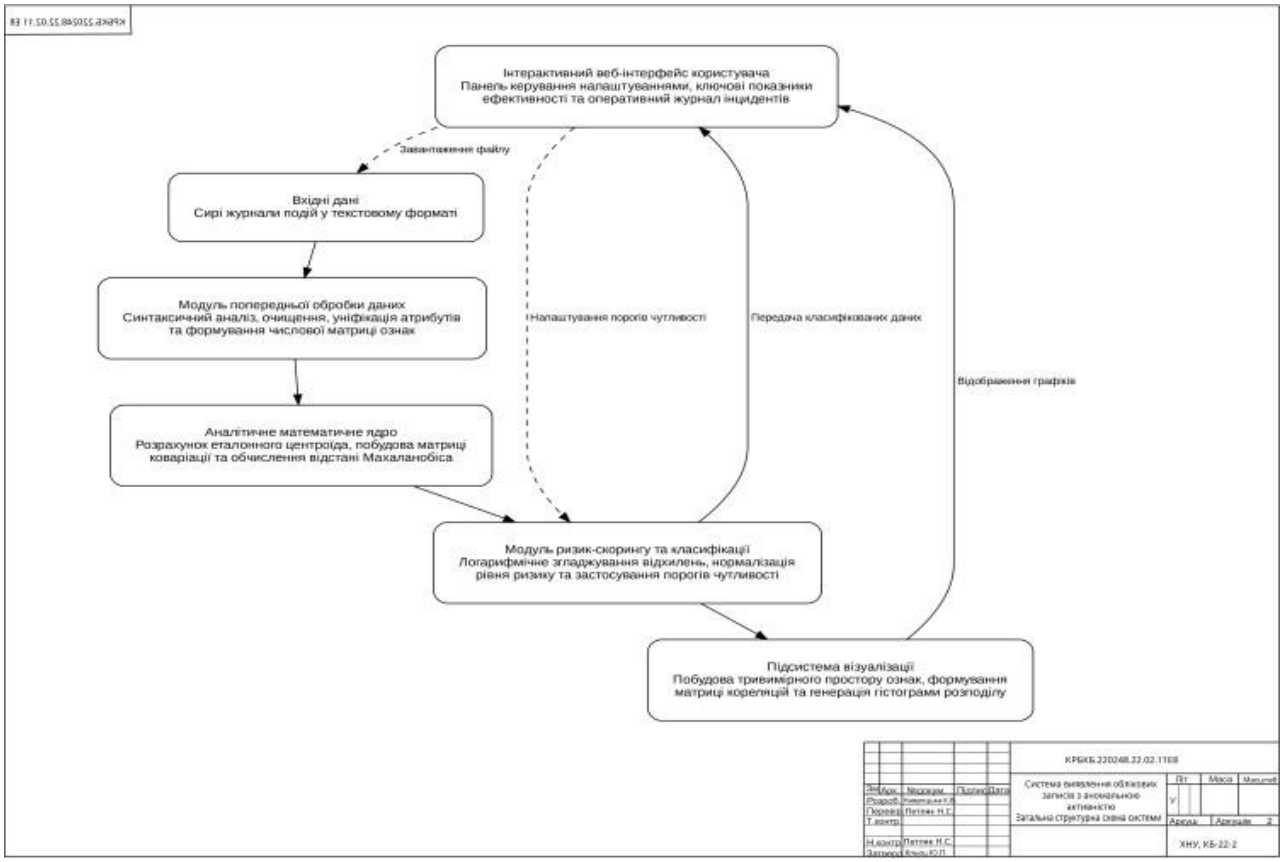
46. Matplotlib: Python plotting // Matplotlib Development Team : [сайт]. URL: https://matplotlib.org/ (дата звернення: 05.04.2026).

47. Authentication & Authorization Failures Dataset // Kaggle : [датасет]. 2026. URL: https://www.kaggle.com/datasets/mirzayasirabdullah07/authentication-and-authorization-failures-dataset (дата звернення: 03.05.2026).

Додаток А

Копії графічної частини

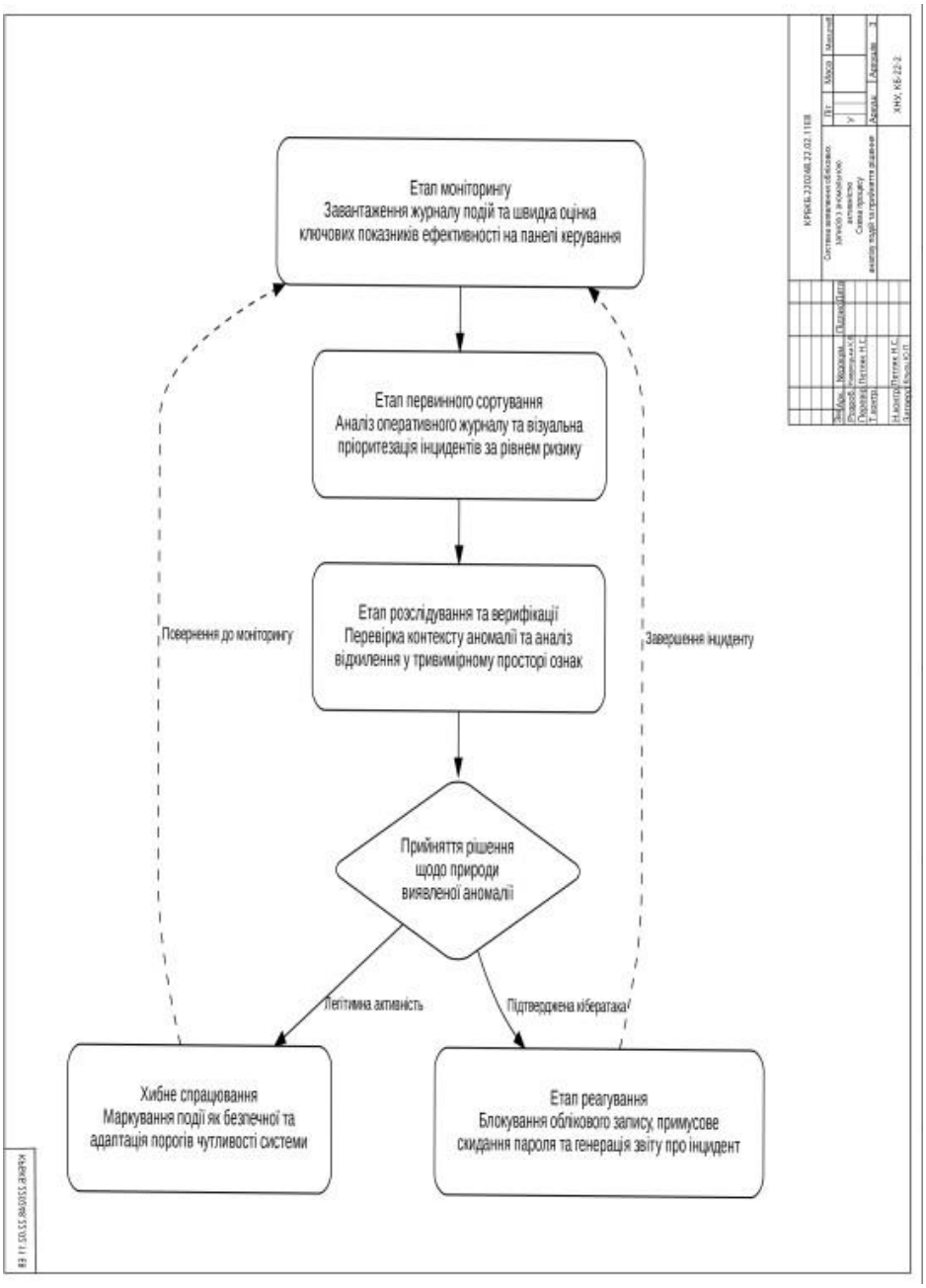




				КРБКБ.220248.22.02.11ЕВ	
Зм.	Арк.	№докум.	Підпис	Дата	
Розроб.		Умінський К.В.			Система виявлення облікових записів з аномальною активністю Загальна структурна схема системи
Перевір.		Петяєв Н.Г.			
Т. контр.					
Н. контр.		Петяєв Н.С.			
Затверд.		Ковальов Ю.П.			

Літ	Маса	Масштаб
У		
Аркуш	Т. Аркушів	2

ХНУ, КБ-22-2


Етап моніторингу
Завантаження журналу подій та швидка оцінка ключових показників ефективності на панелі керування
Етап первинного сортування
Аналіз оперативного журналу та візуальна пріоритезація інцидентів за рівнем ризику
Етап розслідування та верифікації
Перевірка контексту аномалії та аналіз відхилення у тривимірному просторі ознак
Повернення до моніторингу
Завершення інциденту
Прийняття рішення щодо природи виявленої аномалії
Легітимна активність
Підтверджена кібератака
Хибне спрацювання
Маркування події як безпечної та адаптація порогів чутливості системи
Етап реагування
Блокування облікового запису, примусове скидання пароля та генерація звіту про інцидент

Додаток Б

Листинг програмного коду

```
import streamlit as st
import pandas as pd
import numpy as np
import plotly.express as px
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.spatial import distance

pd.set_option("styler.render.max_elements", 2000000)

st.set_page_config(page_title="UEBA", layout="wide")
st.title("Система виявлення аномальної активності")
st.markdown("### Багатовимірний статистичний аналіз (Метрика Махаланобіса)")

st.sidebar.header("Керування системою")
uploaded_file = st.sidebar.file_uploader("Завантажити лог-файл (CSV)", type=["csv"])

st.sidebar.divider()

st.sidebar.subheader("Стратегія виявлення")
detection_mode = st.sidebar.radio(
    "Оберіть режим:",
    ["Топ 5% (Відносний поріг)", "Ризик > 70% (Абсолютний поріг)"],
    help="Відносний поріг: завжди показує лише 5% найбільш нетипових користувачів (Адаптивно). Абсолютний поріг: показує всіх, чий рівень ризику перевищує 70% (Корисно при масових атаках)."
)

if uploaded_file is not None:
    raw_df = pd.read_csv(uploaded_file)

    rename_dict = {
        'failed_attempts': 'Спроби входу',
        'session_duration': 'Тривалість сесії (хв)',
        'password_age_days': 'Вік пароля (днів)',
        'timestamp': 'Час',
        'user_id': 'ID користувача',
        'username': 'Ім’я користувача',
        'ip_address': 'IP-адреса'
```

```
}
    df = raw_df.rename(columns=rename_dict)

    features = ['Спроби входу', 'Тривалість сесії (хв)', 'Вік пароля (днів)']

    if all(col in df.columns for col in features):

        X = df[features].values
        mean_vector = np.mean(X, axis=0)
        cov_matrix = np.cov(X, rowvar=False)
        inv_cov_matrix = np.linalg.pinv(cov_matrix)

        mahalanobis_dist = []
        for row in X:
            dist = distance.mahalanobis(row, mean_vector, inv_cov_matrix)
            mahalanobis_dist.append(dist)

        df['Статистичне відхилення'] = mahalanobis_dist

        log_dist = np.log1p(df['Статистичне відхилення'])
        min_d, max_d = log_dist.min(), log_dist.max()
        df['Рівень ризику (%)'] = ((log_dist - min_d) / (max_d - min_d) * 100).round(1)

        if detection_mode == "Топ 5% (Відносний поріг)":
            threshold = np.percentile(df['Рівень ризику (%)'], 95)
            df['Статус'] = np.where(df['Рівень ризику (%)'] >= threshold, 'Загроза',
'Норма')
            method_desc = f"Адаптивний поріг (95-й перцентиль). Відібрано топ 5%
найвищих відхилень (Поріг відсікання: {threshold:.1f}%)."
        else:
            df['Статус'] = np.where(df['Рівень ризику (%)'] >= 70, 'Загроза', 'Норма')
            method_desc = "Абсолютний поріг. Відібрано всіх користувачів із
розрахунковим ризиком вище 70%."

        m1, m2, m3, m4 = st.columns(4)
        m1.metric("Опрацьовано подій", len(df))
        m2.metric("Користувачів", df['ID користувача'].nunique() if 'ID користувача'
in df.columns else "Н/Д")
        m3.metric("Виявлено інцидентів", len(df[df['Статус'] == 'Загроза']))
        m4.metric("Середній ризик", f"{df['Рівень ризику (%)'].mean():.1f}%")

        st.info(f"ℹ **Поточний режим:** {method_desc}")
```

```
    st.divider()

    t1, t2, t3 = st.tabs(["3D Простір", "Коваріаційний аналіз", "Розподіл ризиків"])

    with t1:
        st.subheader("Багатовимірний простір поведінки")
        st.write("Чим червоніша точка, тим далі вона від 'центру мас' корпоративної норми.")
        fig3d = px.scatter_3d(
            df, x='Тривалість сесії (хв)', y='Спроби входу', z='Вік пароля (днів)',
            color='Рівень ризику (%)', color_continuous_scale='Turbo',
            size='Рівень ризику (%)', hover_data=['ID користувача', 'Час'], height=700
        )
        st.plotly_chart(fig3d, use_container_width=True)

    with t2:
        st.subheader("Матриця взаємозв'язків (Кореляція ознак)")
        c1, c2 = st.columns(2)
        with c1:
            fig_corr, ax_corr = plt.subplots(figsize=(8, 6))
            sns.heatmap(pd.DataFrame(cov_matrix, columns=features, index=features),
                        annot=True, cmap='coolwarm', fmt=".1f", ax=ax_corr, linewidths=.5)
            st.pyplot(fig_corr)
        with c2:
            fig2d, ax2d = plt.subplots(figsize=(8, 6))
            sns.scatterplot(data=df, x='Тривалість сесії (хв)', y='Спроби входу', hue='Статус',
                            palette={'Норма': '#1f77b4', 'Загроза': '#d62728'}, s=100, ax=ax2d, alpha=0.7)
            st.pyplot(fig2d)

    with t3:
        st.subheader("Статистичний розподіл метрики Махаланобіса")
        fig_hist = px.histogram(df, x="Рівень ризику (%)", color="Статус", barmode='overlay',
                                marginal="box", color_discrete_map={'Норма': '#1f77b4', 'Загроза': '#d62728'}
                                labels={'Рівень ризику (%)': 'Розрахований ризик', 'count': 'Кількість логів'})
        st.plotly_chart(fig_hist, use_container_width=True)
```

```
        st.divider()

        st.subheader("Журнал інцидентів  (ТОП-1000 критичних  загроз)")
        st.info("Для економії ресурсів системи та зручності аналізу, в оперативному журналі відображаються лише ТОП-1000 подій з найвищим рівнем ризику.")

        incidents = df[df['Статус'] == 'Загроза'].sort_values('Рівень ризику (%)', ascending=False)

        first_cols = ['Статистичне відхилення', 'Рівень ризику (%)', 'Статус', 'ID користувача', 'Час']
        other_cols = [c for c in incidents.columns if c not in first_cols]
        final_cols = first_cols + other_cols

        st.dataframe(
            incidents[final_cols].head(1000).style.background_gradient(cmap='YlOrRd', subset=['Рівень ризику (%)']),
            use_container_width=True
        )

    else:
        st.error(f"Помилка! Файл не містить потрібних колонок: {features}")
else:
    st.info("Завантажте ваш CSV файл логів у бічній панелі зліва, щоб розпочати математичний аналіз
```

Додаток В

Копії наукових публікацій

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЦЕНТРАЛЬНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
НАУКОВА АСОЦІАЦІЯ КІБЕРБЕЗПЕКИ УКРАЇНИ

УДК 004.4

Тези доповідей

IX Міжнародної науково-практичної конференції

"Інформаційна безпека та комп'ютерні технології"

23 квітня 2026 року

Кропивницький 2026

УДК 004.4

Матеріали IX Міжнародної науково-практичної конференції "Інформаційна безпека та комп'ютерні технології": тези доповідей, 23 квітня 2026 р. – Кропивницький: ЦНТУ, 2026. – 109 с.

Наведені тези пленарних та секційних доповідей за теоретичними та практичними результатами наукових досліджень і розробок. Представлені результати теоретичних досліджень в галузях проектування інформаційних систем, технологій захисту інформації, використання сучасних інформаційних технологій в управлінні системами за різними галузями народного господарства.

Матеріали публікуються в авторській редакції.

За достовірність викладених фактів, цитат та інших відомостей відповідальність несуть автори.

УДК 004.056.5

К.В. Навроцька[1], Н.С. Петляк[1]
katia43060@gmail.com, npetlyak@khmnu.edu.ua
[1]*Хмельницький національний університет, Хмельницький*

МЕТОД ВИЯВЛЕННЯ ОБЛІКОВИХ ЗАПИСІВ З АНОМАЛЬНОЮ АКТИВНІСТЮ В ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ СИСТЕМАХ

У сучасних інформаційно-комунікаційних системах проблема забезпечення безпеки облікових записів набуває особливої актуальності у зв'язку зі зростанням кількості кібератак, спрямованих на компрометацію автентифікаційних даних користувачів. Використання легітимних облікових записів зловмисниками є одним із найскладніших для виявлення сценаріїв, оскільки така активність часто не порушує формальних правил доступу. Особливої складності набуває виявлення подібних інцидентів у середовищах із динамічною інфраструктурою, зокрема в програмно-керованих мережах, де централізоване управління та висока гнучкість створюють додаткові можливості для прихованої діяльності. Традиційні підходи, засновані на сигнатурному аналізі або статичних правилах, не забезпечують належного рівня ефективності у виявленні нових або модифікованих атак. Це зумовлює необхідність розроблення нових методів аналізу поведінки користувачів, які б враховували контекст і характер їхніх дій навіть за відсутності явних порушень політик безпеки. Інтеграція поведінкових та часових характеристик у межах єдиної моделі дозволяє створити адаптивний механізм виявлення аномалій, що суттєво підвищує достовірність ідентифікації прихованих загроз.

Метод виявлення облікових записів з аномальною активністю доцільно будувати на основі інтеграції поведінкових характеристик користувачів та часових метрик їхньої діяльності, що дозволяє підвищити точність виявлення інцидентів інформаційної безпеки та зменшити рівень хибнопозитивних спрацювань. Основою такого підходу є формування індивідуального профілю користувача, який відображає типові шаблони його взаємодії з інформаційною системою, а також часові закономірності виконання операцій.

У межах реалізації методу здійснюється збір даних із журналів автентифікації, мережевих сесій, історії виконання команд та інформації про ролі користувачів. На основі цих даних формується вектор ознак користувача x_i, який описує його поведінкову активність $X_i = (x_{i1}, x_{i2}, .., x_{in})$ де кожна компонента відповідає певній характеристиці, наприклад інтенсивності сесій, кількості унікальних IP-адрес або частоті виконання команд.

Такий вектор дозволяє формалізувати поведінку користувача у вигляді багатовимірного простору ознак, що є необхідним для подальшого кількісного аналізу.

Для визначення нормальної поведінки використовується статистичний профіль, який задається вектором середніх значень μ та стандартних відхилень σ, отриманих на основі історичних даних. Відхилення поточного значення ознаки від її типової величини оцінюється за допомогою z-оцінки:

$$z_j = \frac{x_j - \mu_j}{\sigma_j}$$

Ця величина дозволяє нормалізувати різнорідні ознаки до єдиної шкали, що є важливим у випадку, коли характеристики мають різну природу та одиниці вимірювання. Крім того, використання z-оцінки дає змогу інтерпретувати відхилення у термінах статистичної значущості, тобто визначити, наскільки поточне значення є нетиповим відносно історичної поведінки користувача.

Інтегральна оцінка поведінкової аномальності визначається як агрегована функція відхилень:

$$S_b = \frac{1}{n}\sum_{j=1}^{n}|z_j|$$

Необхідність введення такої оцінки зумовлена тим, що окремі ознаки самі по собі можуть не свідчити про аномалію, однак їх сукупне відхилення формує характерний шаблон підозрілої поведінки. Агрегація дозволяє отримати узагальнену числову характеристику, яка відображає загальний ступінь відхилення поведінки користувача від його нормального профілю. Використання середнього абсолютного значення забезпечує однаковий внесок кожної ознаки та запобігає взаємній компенсації позитивних і негативних відхилень.

Паралельно здійснюється аналіз часових характеристик активності користувача. На основі історичних даних будується ймовірнісний розподіл активності протягом доби, який описується функцією $P(h)$, де h - година доби. Оцінка часової аномальності визначається як $S_t = 1 - P(h_t)$ де h_t - фактичний час виконання дії. Запровадження цієї метрики дозволяє кількісно оцінити, наскільки поточна активність відповідає типовому часовому шаблону користувача. Якщо користувач зазвичай працює у денний час, то нічна активність матиме низьку ймовірність і, відповідно, високу оцінку аномальності. Таким чином, дана складова враховує контекст виконання дій, що є критично важливим для виявлення компрометації облікових записів.

Для прийняття узгодженого рішення необхідно інтегрувати поведінкову та часову складові в єдиний

показник. Це реалізується за допомогою зваженої лінійної комбінації $S = \alpha * S_b + \beta * S_t$.

Введення вагових коефіцієнтів α та β дозволяє регулювати вплив кожного типу ознак на кінцевий результат. Це є важливим, оскільки в різних сценаріях атаки домінуючу роль можуть відігравати різні фактори: у деяких випадках аномальною є саме поведінка, тоді як у інших - час виконання дій. Таким чином, зважена агрегація забезпечує гнучкість моделі та можливість її адаптації до специфіки середовища.

Остаточне рішення щодо наявності аномальної активності приймається шляхом порівняння інтегральної оцінки з пороговим значенням:

$$Alert = \begin{cases} 1, \text{якщо } S > 0 \\ 0, \text{інакше.} \end{cases}$$

Запропонований метод демонструє підвищену достовірність у випадках, коли поведінкова активність користувача суттєво відхиляється від сформованого профілю та одночасно фіксується порушення часових закономірностей виконання операцій. Зокрема, у ситуації, коли обліковий запис адміністратора використовується для виконання нетипових дій, таких як запуск незвичних команд, доступ до раніше нехарактерних ресурсів або зміна параметрів системи, у поєднанні з активністю у часові інтервали, що не відповідають типовому режиму роботи даного користувача, інтегральна оцінка аномальності перевищує встановлений пороговий рівень. Це, у свою чергу, ініціює механізм реагування та сигналізує про потенційний інцидент інформаційної безпеки. Застосування зваженої агрегації дозволяє адаптувати модель до специфіки конкретного середовища, враховуючи, що в різних сценаріях атак домінуючу роль можуть відігравати або поведінкові фактори, або часовий контекст. Такий підхід мінімізує вплив випадкових відхилень, які не є ознаками цілеспрямованого злому, та забезпечує стабільність роботи системи моніторингу.

Практична реалізація запропонованого методу передбачає його інтеграцію до складу систем моніторингу інформаційної безпеки, зокрема SIEM-систем, систем управління доступом або платформ аналізу мережевого трафіку. Метод може бути реалізований як окремий аналітичний модуль, що функціонує у режимі реального часу або періодичної обробки подій, отримуючи дані з різних джерел інформаційно-комунікаційної системи. Вхідними даними для функціонування методу є журнали автентифікації, записи про мережеві з'єднання, історія виконання команд, інформація про використані ресурси, IP-адреси, часові мітки подій, а також атрибути ролей і прав доступу користувачів. Далі можуть надходити як із внутрішніх систем (сервери, контролери домену, додатки), так і з мережевих пристроїв (маршрутизатори, міжмережеві екрани). Перед обробкою вони проходять етап нормалізації та агрегації для формування уніфікованого вектору ознак.

Результатом роботи є інтегральна оцінка аномальності для кожного облікового запису або окремої сесії користувача. У разі перевищення встановленого порогового значення формується подія безпеки, яка передається до системи реагування та безпосередньо адміністраторам безпеки. Додатково можуть формуватися пояснювальні характеристики (наприклад, які саме ознаки дали найбільший внесок в аномалію), що підвищує інтерпретованість результатів. Отримані результати надходять до центрів моніторингу безпеки або відповідальних адміністраторів, де здійснюється подальший аналіз інциденту. Реакція на виявлену аномалію може бути як автоматизованою (блокування облікового запису, завершення сесії, вимога повторної автентифікації), так і напівавтоматичною з участю аналітика. Вибір сценарію реагування залежить від критичності ресурсу, рівня аномалії та політик безпеки організації. Важливою перевагою такого підходу є здатність виявляти складні та приховані сценарії компрометації облікових записів, зокрема у випадках використання легітимних облікових даних зловмисниками. На відміну від традиційних методів контролю доступу, які орієнтовані на перевірку автентичності користувача, запропонована модель аналізує контекст та характер його дій, що дозволяє виявляти відхилення навіть за відсутності явних порушень політик безпеки. Це є особливо актуальним для виявлення прихованих каналів витоку інформації, які можуть реалізовуватися через легітимні інтерфейси доступу та не супроводжуватися типовими ознаками атак. Таким чином, інтеграція поведінкових та часових характеристик у межах єдиної аналітичної моделі забезпечує формування більш точного та адаптивного механізму виявлення аномальної активності. Такий підхід дозволяє одночасно враховувати як зміст виконуваних дій, так і контекст їх реалізації у часі, що істотно підвищує достовірність ідентифікації загроз. Це є важливим для сучасних інформаційно-комунікаційних систем, зокрема програмно-керованих мереж, де висока динамічність середовища та централізований характер управління створюють додаткові виклики для забезпечення інформаційної безпеки. Використання статистичних профілів та z-оцінок дозволяє ефективно нормалізувати різнорідні дані, приводячи їх до єдиної шкали для кількісного аналізу.