

ЗМІСТ

Стор.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	9
ВСТУП	12
1 Дослідження можливостей GAN у виявленні інсайдерських загроз . . .	14
1.1. Технологія застосування GAN	14
1.2. Проблеми і виклики інсайдерських загроз	20
1.3. Огляд використання машинного навчання для виявлення інсайдерських загроз	29
2 Аналіз методики використання GAN для виявлення інсайдерських загроз	35
2.1. Підготовка та використання датасету CERT r4.2	35
2.2. Методика використання GAN для виявлення інсайдерських загроз .	46
2.3. Методика збагачення даних за допомогою GAN	50
3 Модель виявлення інсайдерських загроз на базі GAN	55
3.1. Результати навчання GAN для самостійного виявлення інсайдерських загроз	55
3.2. Збагачення даних за допомогою GAN	60
3.3. Тестування інших моделей на збагачених за допомогою GAN даних	65
ВИСНОВКИ	70
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	72

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

AnoGAN	Anomaly Detection Generative Adversarial Network — модифікація GAN для виявлення аномалій у даних.
AUC	Area Under the ROC Curve — інтегральний показник якості бінарного класифікатора за площею під ROC-кривою.
AUPRC	Area Under the Precision–Recall Curve — інтегральний показник якості класифікатора за площею під кривою «точність–повнота».
BiGAN	Bidirectional Generative Adversarial Network — різновид GAN, у якому додатково навчається кодувальник з простору даних у латентний простір.
CERT	Computer Emergency Response Team — команда реагування на комп'ютерні надзвичайні події; у роботі — також назва відкритого набору даних CERT університету Карнегі-Меллон.
CSV	Comma-Separated Values — текстовий формат зберігання табличних даних, у якому значення розділено комами.
CTGAN	Conditional Tabular Generative Adversarial Network — умовна GAN для моделювання табличних даних з числовими й категоріальними ознаками.
DOC	Document (Microsoft Word Document) — формат текстових документів текстового процесора Microsoft Word.
GAN	Generative Adversarial Network — генеративно-змагальна нейронна мережа для моделювання розподілу даних та генерації синтетичних зразків.
GRU	Gated Recurrent Unit — тип рекурентної нейронної мережі з керованими (gated) станами, спрощена альтернатива LSTM.
HTTP	HyperText Transfer Protocol — протокол передавання гіпертексту в мережі Інтернет.
ISO	International Organization for Standardization — Міжнародна організація зі стандартизації.
ITD	Insider Threat Detection — виявлення інсайдерських загроз (шкідливої активності користувачів усередині організації).

LDAP	Lightweight Directory Access Protocol — полегшений протокол доступу до служб каталогів (наприклад, корпоративних довідників користувачів).
LSTM	Long Short-Term Memory — тип рекурентної нейронної мережі з комірками пам'яті для моделювання довгострокових залежностей.
ML	Machine Learning — машинне навчання; сукупність методів побудови моделей, що навчаються на даних.
PDF	Portable Document Format — формат електронних документів із фіксованим макетом сторінки.
POSIX	Portable Operating System Interface — набір стандартів сумісності для операційних систем, насамперед сімейства Unix-подібних.
ReLU	Rectified Linear Unit — нелінійна активаційна функція нейронних мереж типу $f(x) = \max(0, x)$.
RNN	Recurrent Neural Network — рекурентна нейронна мережа для оброблення послідовностей (часові ряди, текст тощо).
ROC	Receiver Operating Characteristic — ROC-крива; графік, що відображає залежність між часткою хибних спрацювань і чутливістю класифікатора.
SGD	Stochastic Gradient Descent — стохастичний градієнтний спуск, ітеративний метод оптимізації параметрів моделей машинного навчання.
SIEM	Security Information and Event Management — система централізованого збирання та аналізу подій інформаційної безпеки.
SMOTE	Synthetic Minority Over-sampling Technique — метод синтетичного збільшення вибірки меншого класу для розв'язання проблеми дисбалансу класів.
SVM	Support Vector Machine — метод опорних векторів для класифікації та регресії.
t-SNE	t-Distributed Stochastic Neighbor Embedding — нелінійний метод зниження розмірності для візуалізації високовимірних даних.
UEBA	User and Entity Behavior Analytics — аналітика поведінки користувачів та інших сутностей (пристроїв, облікових записів) з метою виявлення аномалій.

URL	Uniform Resource Locator — уніфікований покажчик ресурсу, стандартний запис веб-адреси ресурсу в мережі Інтернет.
USB	Universal Serial Bus — універсальна послідовна шина для підключення периферійних пристроїв.
UTC	Coordinated Universal Time — всесвітній координований час, базова шкала часу для часових поясів.
VGM	Variational Gaussian Mixture — варіаційна суміш гауссових розподілів для моделювання складних розподілів даних.
WGAN-GP	Wasserstein Generative Adversarial Network with Gradient Penalty — різновид GAN із метрикою Вассерштейна та штрафом за норму градієнта для стабілізації навчання.
XLS	Excel Spreadsheet — формат файлів електронних таблиць Microsoft Excel.

ВСТУП

В умовах стрімкої цифровізації та зростання складності інформаційних систем традиційні підходи до кібербезпеки, що базуються переважно на захисті периметра та сигнатурному аналізі, демонструють зниження ефективності. Особливо гостро ця проблема постає в контексті інсайдерських загроз — зловмисних дій співробітників, які мають легітимний доступ до інформаційних ресурсів. На відміну від зовнішніх атак, дії інсайдера часто технічно не відрізняються від штатної роботи, маскуючись під рутинні операції, що робить їх невидимими для класичних засобів захисту.

Існуючі рішення класу UEBA намагаються вирішити цю проблему шляхом аналізу поведінки. Проте критичною, досі не розв’язаною проблемою залишається значний дисбаланс даних. У реальних журналах подій (логах) кількість записів про нормальну активність у тисячі разів перевищує кількість зафіксованих інцидентів. Класичні алгоритми машинного навчання в таких умовах схильні ігнорувати рідкісні аномалії, оптимізуючись під домінуючий мажоритарний клас, який становить стандартна поведінка користувачів. Відомі методи балансування даних, такі як SMOTE (лінійна інтерполяція), виявляються неефективними для складних поведінкових логів, оскільки вони руйнують часові та логічні зв’язки між подіями, створюючи нереалістичні синтетичні зразки.

Отже, актуальність роботи зумовлена необхідністю розробки нових методів генерації та аналізу поведінкових даних, які дозволять компенсувати дефіцит реальних прикладів атак. Перспективним напрямом є використання генеративно-змагальних мереж (GAN), зокрема архітектур WGAN-GP та CTGAN. Вони здатні моделювати складні нелінійні розподіли гетерогенних даних, зберігаючи семантичну цілісність поведінкових патернів. Це дозволяє створювати якісні синтетичні сценарії атак для навчання систем захисту, що є критично важливим для підвищення кіберстійкості інформаційних систем в умовах обмеженої статистики реальних інцидентів.

Для досягнення мети покращеного виявлення інсайдерів, необхідно виконати такі завдання:

- 1) Проаналізувати сучасні методи виявлення інсайдерських загроз та виявити недоліки існуючих підходів до балансування даних (зокрема SMOTE) при роботі з гетерогенними журналами подій.

- 2) Обґрунтувати вибір архітектур генеративно-змагальних мереж

(зокрема AnoGAN, WGAN-GP, CTGAN) для моделювання табличних даних та послідовностей подій користувачів.

3) Розробити методику попередньої обробки та векторизації поведінкових логів (на прикладі датасету CERT r4.2), що забезпечує збереження контекстних та часових ознак для навчання нейронних мереж.

4) Експериментально дослідити вплив синтетичного збагачення даних за допомогою GAN на метрики якості класифікації (Precision, Recall, F1-score) алгоритмів машинного навчання.

5) Здійснити порівняльний аналіз ефективності запропонованого генеративного підходу з класичними методами балансування та визначити умови його доцільного використання у системах виявлення загроз.

Практичні значення одержаних результатів полягатимуть у розробці алгоритмічного конвеєра підготовки та нормалізації поведінкових журналів, який дозволяє трансформувати дані в структуровані вектори ознак для навчання нейронних мереж.

Запропоновану методику генерації синтетичних сценаріїв інсайдерських атак на основі GAN, що дозволяє суттєво зменшити дисбаланс класів у навчальних вибірках без втрати логічної структури подій.

Залежно від отриманих результатів проведені дослідження можуть дати покращення повноти виявлення загроз (Recall) у системах моніторингу (SIEM/UEBA) та можуть бути використані для налаштування детекторів безпеки в організаціях, де відсутня достатня база історичних інцидентів.

За матеріалами кваліфікаційного дослідження було опубліковано одну статтю (Верби́ненко В.О., Зибін С.В. (2025). Проблеми кібербезпеки в організаціях з дистанційною формою праці. *Сучасний захист інформації*, 2(62), 22-30. <https://doi.org/10.31673/2409-7292.2025.020383>), а також подано одну статтю для публікації (Верби́ненко В.О., Зибін С.В. Метод підвищення ефективності детектування інсайдерських загроз із застосуванням GAN-аугментації) у науково-технічному журналі «Сучасний захист інформації».

1 ДОСЛІДЖЕННЯ МОЖЛИВОСТЕЙ GAN У ВИЯВЛЕННІ ІНСАЙДЕРСЬКИХ ЗАГРОЗ

1.1. Технологія застосування GAN

Генеративно-змагальні мережі, вперше запропоновані Яном Гудфелоу та співавторами у 2014 році, належать до класу алгоритмів навчання без учителя, що ґрунтуються на ідеях теорії ігор. Ключова особливість GAN полягає в тому, що процес навчання формулюється як антагоністична гра між двома нейронними мережами, які, змагаючись, навчаються одночасно: генератором G та дискримінатором D [1].

Генератор перетворює вхідний випадковий шум на синтетичні зразки даних таким чином, щоб якомога точніше імітувати розподіл реальних даних і пройти перевірку дискримінатором. У свою чергу, дискримінатор працює як бінарний класифікатор, що оцінює ймовірність того, що поданий йому на перевірку зразок походить з реального набору даних, а не був створений генератором.

Важливою рисою GAN є відсутність явно заданих правильних виходів для генератора: він не має безпосередньо визначеного цільового значення і отримує зворотний зв'язок лише у вигляді оцінок від дискримінатора. Завдяки цьому після успішного навчання вдається отримувати синтетичні дані, що максимально наближені до реальних.

Класичним прикладом застосування генеративно-змагальних мереж є генерування фотографій або відеофрагментів. Водночас цей підхід можна застосовувати й до інших форматів даних, зокрема для створення табличних даних, аудіозаписів чи текстів. Таким чином, GAN поступово еволюціонували від інструмента генерації зображень до універсальної технології моделювання різнотипної інформації.

Розглянемо базові принципи роботи генеративно-змагальної мережі, незалежно від конкретного типу даних. Для тренування обох моделей (генератора та дискримінатора) використовують два джерела даних: реальний розподіл та латентний розподіл. Реальний розподіл відповідає наявному набору справжніх даних і використовується під час навчання дискримінатора, який намагається відрізнити реальні приклади з набору даних від синтетично згенерованих. Натомість генератор працює з латентним розподілом, тобто випадковим шумом, який подається на вхід у вигляді випадкових векторів z , що перетворюються на

елементи, подібні до реальних даних [2].

Під реальним розподілом $p_{\text{data}}(x)$ розуміють заранне невідому функцію, яка описує принцип поширення і існування об'єктів (наприклад, усі можливі зображення облич людей). Безпосередньо ця функція нам не відома, тому на практиці працюють з вибіркою реальних прикладів $x \sim p_{\text{data}}$ — підмножиною даних, отриманою з цього розподілу. Саме ця вибірка використовується під час тренування мереж, щоб генератор поступово навчився відтворювати приклади, подібні до тих, що надходять із реального розподілу.

Зі свого боку, латентний розподіл $p(z)$ є відомою простою функцією, яка задає спосіб генерування випадкового шуму. Зазвичай обирають розподіл, з якого легко брати вибірки, як от багатовимірний нормальний розподіл $N(0, I)$ або рівномірний розподіл на відрізку $[-1, 1]$. Випадковий вектор $z \sim p(z)$ визначає початкову точку, з якої генератор створює новий приклад, поступово перетворюючи шум у структури, подібні до реальних.

Таким чином, генератор можна розглядати як диференційовану функцію

$$G_{\theta} : Z \rightarrow X,$$

що залежить від параметрів θ і відображає простір латентних змінних Z у простір даних X . Вона перетворює випадковий шум z на синтетичний приклад даних

$$x' = G(z; \theta),$$

а отримані зразки задають розподіл моделі $p_g(x)$, який визначається параметрами θ генератора.

Дискримінатор, у свою чергу, можна формально подати як відображення

$$D_{\phi} : X \rightarrow [0, 1]$$

з параметрами ϕ . Ця модель зіставляє кожен вхідний приклад x зі скалярним значенням $D(x)$, що розглядається як ймовірність того, що x походить із реального розподілу даних, а не є синтетично згенерованим зразком. Дискримінатор виконує роль класифікатора-експерта: він повертає значення, близьке до 1, якщо приклад класифіковано як автентичний, і наближене до 0, якщо він розпізнається як штучно створений. Його завдання полягає в якомога точнішому розрізненні реальних

зразків $x \sim p_{\text{data}}$ та синтетичних прикладів $x' = G(z) \sim p_g$ [3].

Генератор і дискримінатор навчаються одночасно в рамках змагання в оптимізації, яке можна інтерпретувати як гру з нульовою сумою. Це означає, що виграш однієї сторони є програшем іншої: коли дискримінатор підвищує точність класифікації, генератор умовно програє, і навпаки. Формально навчання GAN описується як мінімакс-задача оптимізації, у якій дискримінатор намагається максимізувати функціонал правильності класифікації, а генератор — мінімізувати той самий функціонал:

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{\text{data}}(x)} [\log D(x)] + \mathbb{E}_{z \sim p(z)} [\log(1 - D(G(z)))] .$$

У цій формулі дискримінатор оптимізує свої параметри так, щоб підвищити ймовірність правильного розпізнавання справжніх і синтетичних прикладів, тоді як генератор оновлюється з метою мінімізувати значення цільової функції, генеруючи такі дані, які проходять перевірки дискримінатора [1].

З точки зору дискримінатора, перший доданок

$$\mathbb{E}_{x \sim p_{\text{data}}(x)} [\log D(x)]$$

відображає середнє значення $\log D(x)$ на реальних даних. Модель D прагне максимізувати цей компонент, що досягається, коли $D(x) \rightarrow 1$ для всіх автентичних прикладів (оскільки $\log 1 = 0$, і це є максимальним значенням логарифмічної функції на відрізку $(0, 1]$ за умов упевненої класифікації). Отже, оптимальною стратегією для дискримінатора є призначення високої ймовірності (наближеної до 1) справжнім зразкам.

Другий доданок

$$\mathbb{E}_{z \sim p(z)} [\log(1 - D(G(z)))]$$

є очікуваним значенням $\log(1 - D(x'))$ для синтетично згенерованих даних $x' = G(z)$. Дискримінатор також намагається максимізувати цей термін; це виконується, коли $D(G(z)) \rightarrow 0$ для всіх згенерованих прикладів (оскільки тоді $\log(1 - 0) = 0$). Це також відповідає ситуації, коли дискримінатор упевнено класифікує всі штучно створені зразки як несправжні [4].

Таким чином, тренування D фактично зводиться до стандартної задачі бінарної класифікації: відрізнити автентичні зразки від згенерованих.

Навчання моделі полягає в максимізації ймовірності правильних рішень, або, що еквівалентно, у мінімізації функції втрат, яка вимірює відмінність між передбаченими ймовірностями та правильними відповідями. Така функція відома як «крос-ентропія» і є стандартним критерієм для класифікаційних моделей.

З точки зору генератора, його мета є протилежною до мети дискримінатора. Генератор впливає на другий доданок цільової функції $\log(1 - D(G(z)))$, який він прагне мінімізувати. Щоб зменшити цей показник, модель G намагається досягти того, щоб $D(G(z)) \rightarrow 1$, тобто щоб дискримінатор сприймав згенеровані приклади як справжні. Ідеальна для генератора ситуація відповідає випадку, коли дискримінатор помиляється на всіх синтетичних зразках ($D(G(z)) \approx 1$), адже тоді $\log(1 - D(G(z))) \rightarrow \log(0)$, що прямує до $-\infty$, і очікуване значення другого доданку різко зменшується. Таким чином, генератор оновлює свої параметри так, щоб обійти дискримінатор, змушуючи його класифікувати згенеровані дані як автентичні, тобто G намагається досягти мети, прямо протилежної до мети D .

Як показано в у [3], у результаті формується змагальний процес: дискримінатор постійно вдосконалюється, намагаючись дедалі точніше розпізнавати зразки, тоді як генератор адаптується, роблячи свої підробки все більш правдоподібними. З часом система в ідеальному випадку має наблизитися до стану рівноваги, коли згенеровані дані настільки реалістичні, що дискримінатор уже не здатний впевнено відрізнити їх від справжніх.

Теоретично можна показати, що для будь-якого фіксованого генератора існує оптимальний дискримінатор

$$D^*(x) = \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_g(x)},$$

який оцінює ймовірність належності x до реальних даних шляхом порівняння внеску обох розподілів — справжнього p_{data} та згенерованого p_g . Із цієї формули випливає, що якщо генератор досягає досконалого відтворення вибірки (тобто $p_g(x) = p_{\text{data}}(x)$ для всіх x), то оптимальний дискримінатор даватиме вихід

$$D^*(x) = 0.5$$

для кожного прикладу. Інакше кажучи, коли розподіл згенерованих даних збігається зі справжнім розподілом, дискримінатор повністю втрачає можливість відрізнити

штучні зразки від реальних — і саме це є бажаним станом рівноваги для GAN.

На практиці досягти такого балансу надзвичайно складно через низку проблем, притаманних процесу навчання GAN. Наприклад, при одночасному навчанні двох мереж градієнтним спуском — методом, який крок за кроком змінює параметри в напрямку найшвидшого зменшення помилки — дискримінатор ніколи не є ідеально оптимальним: його параметри постійно змінюються в процесі власного навчання. Тому генератор змушений оптимізуватися відносно поточної версії D , яка завжди є певною мірою недосконалою. Ця невідповідність між теоретичною задачею (максимізація D за фіксованого G) та реальним одночасним навчанням методом стохастичного градієнтного спуску (SGD), тобто коли параметри оновлюються за випадково вибраними підмножинами даних, є фундаментальною причиною нестабільності GAN.

Внаслідок цього виникають типові проблеми тренування: насичення градієнтів, колапс мод (коли модель генерує лише підмножину можливих вихідних зразків) та навіть повне розходження процесу. Подолання цих труднощів вимагає використання додаткових методів і технік. За останні роки дослідниками було запропоновано низку варіацій архітектур і функцій втрат, які підвищують стабільність змагального навчання.

Так, умовні GAN, що вводять додаткові ознаки чи мітки класів у процес генерації, дають змогу моделювати розподіл під конкретні умови; Wasserstein-GAN та його варіант WGAN-GP [5, 6] використовують іншу міру відстані між розподілами та регуляризацію градієнта для запобігання розходженню. Ще одна технологія — двонапрямлені GAN (BiGAN) [7] додають до генератора енкодер — мережу, яка відображає дані назад у латентний простір. Генератор і енкодер навчаються одночасно, завдяки чому модель уміє як генерувати дані, так і кодувати їх у зручне приховане представлення.

Сучасні роботи розглядають GAN і їхні модифікації як потужний механізм моделювання складних розподілів даних, зокрема табличних і мережевих, характерних для задач виявлення інсайдерських загроз та навіть систем виявлення вторгнень.

Особливо перспективним може бути застосування GAN у ситуаціях, коли бракує даних з мітками або спостерігається значний дисбаланс класів. Виявлення інсайдерських загроз належить саме до таких задач, оскільки шкідливі дії зловмисників-інсайдерів трапляються дуже рідко порівняно з нормальними діями працівників, і відповідні вибірки є вкрай незбалансованими. Генеративні моделі

можуть допомогти розв'язати цю проблему двома шляхами: шляхом аугментації даних, тобто генерування додаткових штучних прикладів рідкісної шкідливої поведінки для поповнення навчальних наборів, або шляхом безпосереднього виявлення аномалій, коли сама GAN-модель слугує детектором нетипових дій [8].

У першому підході GAN використовується як інструмент синтезу даних. До прикладу, у [9] показано, що застосування WCGAN-GP для генерування додаткових сценаріїв атак дає змогу ефективно розширити датасет інсайдерських інцидентів і зменшити дисбаланс класів. Згенеровані вибірки інтегруються у навчальний набір, після чого на доповнених даних тренуються моделі класифікаторів для виявлення загроз. Така стратегія аугментації демонструє істотне підвищення рівня детекції інсайдерів, зберігаючи при цьому низький рівень хибнопозитивних спрацювань порівняно з базовим підходом без використання GAN.

Інше сучасне дослідження [10] запропонувало GAN-архітектуру під назвою SPCAGAN, що поєднує генеративну модель з методом головних компонент. Це дає змогу навчатись на гетерогенних даних про дії користувачів — тобто на даних, які містять різні типи подій, як-от перегляди, кліки, покупки, тощо. Запропонований підхід показав, що генерація синтетичних даних із використанням GAN може конкурувати з типовими методами: модель SPCAGAN змогла уникнути колапсу мод і швидше збігалася, генеруючи більш якісні та різноманітні зразки інсайдерської активності порівняно з попередніми моделями.

Також GAN можна безпосередньо залучати до пошуку аномалій. У такому випадку генеративна мережа навчається лише на нормальних даних, після чого аномальні дії виявляються як ті, що погано відтворюються генератором або ж викликають велику невпевненість дискримінатора. Існують різні реалізації цього підходу. Зокрема, згаданий вище BiGAN надає значно багатший інструментарій для оцінювання аномальності, оскільки, крім генератора G , містить енкодер E , що відображає дані x у латентний простір, та спеціальний дискримінатор, який розрізняє пари $(x, E(x))$ і $(G(z), z)$. Це дозволяє обчислювати для кожного реального прикладу міру «відхилення», наприклад шляхом оцінювання помилки реконструкції або невідповідності між x та $G(E(x))$. Таким чином, BiGAN може слугувати основою для детектора: нормальні об'єкти добре відновлюються мережею ($x \approx G(E(x))$), тоді як аномалії — ні, що вказує на їхню присутність [7].

У цілому інтеграція генеративних моделей у процес аналізу поведінки користувачів є дієвим способом підвищити ефективність систем виявлення інсайдерських загроз. Здатність GAN моделювати складні багатовимірні розподіли

дозволяє частково компенсувати дефіцит реальних даних про інциденти та перевести задачу пошуку аномалій у більш керовану площину контрольованої класифікації з високим рівнем надійності. Відтак використання GAN постає логічним і перспективним напрямом розвитку засобів кібербезпеки для протидії внутрішнім загрозам.

1.2. Проблеми і виклики інсайдерських загроз

Інсайдерська загроза зазвичай визначається як зловмисна дія з боку поточного або колишнього співробітника, підрядника чи бізнес-партнера, який має (або мав) авторизований доступ до мереж, систем чи даних організації та зловживає цим доступом у спосіб, що шкодить конфіденційності, цілісності або доступності інформаційних активів. Такі загрози належать до найбільш підступних, оскільки ґрунтуються не на зовнішньому зламі, а на зловживанні довірою, наданою особі в межах її службових повноважень [11].

Складність виявлення інсайдера зумовлена тим, що його шкідливі дії майже не відрізняються від повсякденних легітимних операцій користувачів. Інсайдер, маючи актуальні облікові дані та права доступу, може фактично обходити традиційні периметрові засоби захисту, маскуючи зловмисну активність під рутинну роботу. Прихований і динамічний характер поведінки зловмисників в середині організації робить їх виявлення надзвичайно складним.

Згідно з дослідженнями, значна частка кіберінцидентів припадає саме на інсайдерів. Зокрема, опитування Ponemon Institute (2018) показало, що близько 25% атак здійснюються зсередини, причому інциденти, спричинені інсайдерами, часто є більш затратними та руйнівними, ніж зовнішні атаки [12]. Це підкреслює критичну необхідність упровадження ефективних підходів до їх виявлення.

Типові прояви інсайдерських загроз охоплюють широкий спектр шкідливої діяльності. До них, зокрема, належать несанкціоноване винесення конфіденційних даних за межі організації (ексфільтрація інформації), умисний саботаж чи модифікація ІТ-систем, фінансове та інше шахрайство, а також ескалація привілеїв з метою розширення власних можливостей у системі. Подібні дії можуть здійснювати як актори, що свідомо зловживають наданими їм привілеями, так і особи, які діють від імені легітимних користувачів, а також недобросовісні чи недбалі співробітники, котрі ненавмисно спричиняють шкоду. Незалежно від конкретного сценарію,

така активність відбувається під прикриттям формально легальних операцій — копіювання файлів, доступу до баз даних, надсилання електронних листів, тощо — що самі по собі є звичайною частиною робочих процесів. Відтак ключова відмінність між нормальною та шкідливою поведінкою полягає не стільки в самих діях, скільки в їх контексті, намірах інсайдера та відхиленнях від усталених патернів поведінки.

Основним джерелом даних для виявлення інсайдерської активності в сучасних організаціях є журнали (логи) дій користувачів, які генеруються різними інформаційними системами. До таких логів належать, зокрема, записи із сервісів віддаленого доступу, систем контролю доступу до файлових ресурсів, веб-серверів, засобів аутентифікації, корпоративних поштових та офісних платформ. Різні компоненти інфраструктури формують набори подій, які у сукупності утворюють детальний цифровий слід діяльності кожного співробітника. Аналіз поведінки користувачів на основі таких аудиторських даних (хост-логи, мережеві логи, контекстна інформація про облікові записи, тощо) становить основу більшості підходів до виявлення інсайдерів. Водночас робота з цими даними пов'язана з низкою істотних труднощів, подолання яких потребує використання спеціалізованих методів попередньої обробки, моделювання та аналізу [11].

Кожна подія в системі може містити десятки різномірних атрибутів: від мережевих параметрів (IP-адреси, порти, геолокація, тощо) до специфічних параметрів (імена змінених файлів, типи виконаних команд, використані протоколи, характеристики сеансів автентифікації, тощо). У межах однієї організації різні системи генерують журнали у власних форматах, зі своєю структурою та рівнем деталізації. Об'єднання таких різномірних записів в єдиний корпус даних часто перетворюється на складний процес нормалізації та уніфікації. Логи зазвичай є високовимірними, комплексними, гетерогенними й до того ж розрідженими (значна частина атрибутів у конкретних прикладах може виявитись непрактичною). Це створює серйозні виклики для автоматизованого аналізу, оскільки моделі машинного навчання потребують подання інформації в спеціально обробленому, узгодженому та структурованому вигляді.

Без належної трансформації та агрегування журналів до форми, яка, з одного боку, зберігає істотну поведінкову інформацію, а з іншого — зменшує надлишкову варіативність атрибутів, побудова ефективних методів виявлення аномалій або шкідливої активності практично неможлива. На практиці кількість потенційно корисних ознак, що можуть бути виділені з детальних логів, зростає

майже експоненційно зі збагаченням даних. Наприклад, в [13] з використанням CERT-дасету для виявлення інсайдерів було автоматично сформовано близько 70 тисяч поведінкових ознак на одного користувача з журналів активності, після чого довелося застосувати метод головних компонент для зниження розмірності простору ознак. Цей приклад демонструє масштаб проблеми: надто велика кількість характеристик не лише ускладнює побудову моделей, а й загрожує надмірним перенавчанням.

Отже, ситуація протистояння між легітимними користувачами та потенційними інсайдерами в контексті оцінювання ризиків є складною конфліктною ситуацією: на аналіз впливають численні другорядні та випадкові фактори. Тому для можливості математичного аналізу такої ситуації необхідно абстрагуватися від випадкових чинників і побудувати спрощену формалізовану модель процесу та факторів керування ризиками [14]. У цій моделі мають бути явно відображені можливі способи поведінки всіх учасників і результати, до яких призводять усі можливі комбінації їхніх дій. На практичному рівні це означає, що зменшення розмірності та відбір найінформативніших атрибутів, а також ретельна підготовка даних — очищення, фільтрація, нормалізація та приведення до спільного стандарту — є обов'язковими етапами перед побудовою моделей виявлення інсайдерів, без яких алгоритми навчання не здатні ефективно виявляти тонкі поведінкові аномалії у таких журналах.

Дії користувачів формують логічно пов'язані послідовності, тому значущість окремої події часто неможливо коректно оцінити без урахування контексту попередніх та наступних кроків. Наприклад, одноразовий доступ співробітника до конфіденційної директорії може бути цілком легітимним, тоді як багаторазовий доступ у нетиповий час доби або специфічна комбінація операцій (до прикладу, спроба ескалації привілеїв, а потім завантаження великого обсягу даних) є серйозними індикаторами потенційно шкідливої активності. Тому аналіз журналів потребує методів, здатних враховувати часову структуру даних і залежності між подіями, а не лише їх ізольовані характеристики.

У відповідь на цю потребу в сучасних дослідженнях [15] активно застосовуються моделі глибинного навчання для послідовностей: рекурентні нейронні мережі (на кожному кроці беруть до уваги не лише поточний вхід, а й свій попередній стан, як RNN/LSTM), трансформери, а також графові нейромережі для урахування зв'язків між об'єктами. Такі моделі дають змогу відтворювати довгострокові залежності та поведінковий контекст користувача, підвищуючи

здатність алгоритмів розпізнавати складні патерни й виявляти відхилення, непомітні при розгляді одиничних подій. Ключову роль відіграє саме аналіз взаємозв'язків між подіями з урахуванням повної історії активності: це дозволяє фіксувати аномалії, що не проявляються як окремі явно шкідливі дії, але стають видимими як відхилення від усталеної поведінкової норми. Такий підхід особливо важливий з огляду на багатоетапний характер багатьох інсайдерських загроз: атаки зсередини часто розгортаються впродовж тривалого часу малими кроками, з низькою частотою подій і фрагментованими ознаками, тож їх виявлення потребує тонкого темпорального моделювання та високого рівня деталізації аналізу дій.

Журнали активності містять величезні обсяги технічної інформації, значна частина якої не має безпосереднього стосунку до задачі виявлення інсайдерських загроз. Формат і рівень деталізації записів залежать від налаштувань конкретних систем, тому одна й та сама подія може дублюватися в різних журналах або, навпаки, фіксуватися з неповним набором полів. Додаткові складнощі створюють некоректні часові мітки (через проблеми синхронізації), пропущені записи або записи з хибною інформацією. Усе це зумовлює потребу в ретельній попередній обробці даних, яка охоплює очищення логів від шуму (видалення дублікатів і нерелевантних подій), агрегацію дрібних подій у більш значущі дії, корекцію часових полів, а також відбір лише тих атрибутів, що можуть бути корисними для подальшого моделювання.

На цьому етапі ключову роль відіграє інженерія ознак — процес перетворення даних з журналів на інформативні вектори, що відображають релевантні характеристики поведінки користувача. До таких ознак можуть належати статистичні показники активності (наприклад, кількість доступів до певних ресурсів за одиницю часу), метрики частотності та регулярності дій, векторні подання типів операцій, інтервали між подіями, індикатори відхилення поточної активності від власної норми дій користувача. Якісно спроектовані ознаки дають стисле й водночас інформативне узагальнення поведінки та є критичними для успішного навчання моделей. Без такого кроку алгоритми, як правило, реєструватимуть переважно шум і буде генеруватись надто багато хибних спрацьовувань. Отже, зменшення зашумленості даних і виокремлення значущих сигналів є обов'язковою передумовою побудови надійної системи виявлення інсайдерських загроз.

У реальних корпоративних інфраструктурах обсяги журналів активності можуть сягати терабайтів за добу. Системи моніторингу формують потоки подій у реальному часі, що вимагає масштабованих обчислювальних рішень і засобів потокової обробки даних. Моделі, призначені для аналізу такої інформації, повинні,

з одного боку, забезпечувати достатню точність, а з іншого — залишатися працездатними за умов високої варіативності даних та інтенсивних навантажень.

Окремим викликом є сильна незбалансованість класів: лише незначна частина записів відповідає реальним інцидентам, тоді як переважна більшість відображає легітимну повсякденну активність користувачів. Така ситуація, коли шкідливі приклади губляться серед сотень мільйонів нормальних подій, є крайнім випадком задачі з незбалансованими даними. У цих умовах традиційні алгоритми класифікації схильні орієнтуватися на домінуючий клас і пропускати рідкісні аномалії. Застосування методів збільшення вибірки даних, зокрема SMOTE [16], частково пом'якшує проблему: штучне доповнення зазвичай покращує показники виявлення та загальну чутливість детектора, але може знижувати точність через збільшення кількості хибних спрацювань.

Тому в задачах виявлення інсайдерських загроз широко застосовуються неконтрольовані методи та підходи аномального аналізу, які не вимагають великого масиву мічених прикладів атак. Багато сучасних рішень поєднують моделювання нормальної поведінки з кластеризацією та адаптивним навчанням. Основна ідея полягає в тому, щоб навчити систему розуміти, що є нормальним для кожного користувача чи ролі, та виявляти статистично значущі відхилення від цієї норми. При цьому модель поступово оновлює свої уявлення про звичайну поведінку в міру надходження нових даних, враховуючи природну еволюцію робочих патернів.

Найбільш ефективними вважаються гібридні підходи, які поєднують навчання без учителя для виявлення нових, раніше невідомих аномалій, контрольовані методи (якщо доступна хоча б обмежена кількість підтверджених інцидентів), синтетичне збагачення даних та безперервну адаптацію моделі. Саме така комбінація дозволяє краще долати проблему дисбалансу та різномірності даних у задачах детектування інсайдерських загроз.

Ще один важливий виклик полягає в тому, що система має виявляти не лише явно шкідливу активність (на кшталт запуску відомого шкідливого ПЗ або масового копіювання заборонених даних), а й приховані відхилення, які залежать від контексту й проявляються не відразу. Інсайдер, що готує зловмисні дії, може тривалий час поводитися цілком у межах формально дозволеного, поступово збираючи інформацію або підвищуючи власні права доступу, фактично готуючи підґрунтя для фінальної атаки. У таких сценаріях типовий інцидент може взагалі не супроводжуватися разовим різким сплеском аномальної активності, через що традиційні методи виявлення аномалій, орієнтовані на різкі відхилення,

виявляються малоефективними.

Необхідні моделі, здатні будувати довгостроковий профіль поведінки кожного користувача [11] й відстежувати глобальні патерни його активності. По суті, йдеться про аналіз поведінки в розширеному часовому вікні — не в межах годин чи днів, а протягом місяців, упродовж яких загроза може формуватися поступово. Методи побудови поведінкових профілів набувають особливої актуальності, оскільки дають змогу порівнювати поточні дії користувача з його типовим історичним контекстом і виявляти відхилення, що не вписуються у багаторівневу модель нормальності для конкретної особи. Наприклад, якщо співробітник протягом тривалого часу ніколи не звертався до певних конфіденційних ресурсів, а потім раптово починає регулярно їх запитувати, причому у нетиповий час, це відхилення від власного профілю може сигналізувати про приховану підготовку до порушення.

Важливо також, що інсайдери нерідко адаптують свою поведінку, намагаючись не викликати підозр: вони можуть чергувати періоди цілком звичайної активності з епізодами поступового накопичення прав або критичної інформації. У таких умовах успішне виявлення загроз залежить від здатності моделі фіксувати навіть тонкі тренди та зміни у поведінці, які виходять за межі статистичної норми для конкретного користувача або ролі.

Окрім суто технічних аспектів, важливо враховувати й організаційні особливості роботи з такими даними. Журнали активності є не лише інструментом аналітики, а й критично важливим джерелом аудиту, що використовується для розслідування інцидентів та ретроспективного аналізу порушень. Відповідно, процеси їх збирання та обробки мають відповідати вимогам цілісності, безперервності та відтворюваності. Будь-які зміни у способі логування або форматі записів можуть суттєво вплинути на якість роботи моделей машинного навчання. Тому під час розгортання системи виявлення інсайдерів необхідно забезпечити стабільність процесів логування (щоб дані які надходять мали повноту, узгодженість і коректність), а також передбачити регулярне перенавчання чи оновлення моделей відповідно до змін у цифровій інфраструктурі організації. Наприклад, запровадження нового ІТ-сервісу або модифікація політик доступу здатні істотно змінити характер даних у журналах; щоб модель не почала продукувати некоректні спрацювання або, навпаки, пропускати загрози, її необхідно адаптувати до нових умов. Врахування таких організаційних моментів є невід'ємною складовою успішного розгортання систем моніторингу поведінки.

У сукупності наведені фактори формують складне середовище для

дослідження та протидії інсайдерським загрозам. З одного боку, журнали активності містять величезні масиви інформації, які потенційно дають змогу з високою точністю ідентифікувати аномальні чи шкідливі дії. З іншого боку, саме багатовимірність, послідовний характер та значна зашумленість цих даних створюють серйозні бар'єри на шляху до такої точності. Подолати їх можливо лише на основі комплексних підходів, що поєднують продуману інженерію ознак, сучасні алгоритми моделювання поведінки та ретельно організовану інтеграцію процесів моніторингу й реагування. Іншими словами, ефективна система виявлення інсайдера має спиратися на синтез технологій різноманітних технологій для обробки значних масивів даних, засобів збору цих даних та машинного навчання для виявлення прихованих патернів. Журнали активності користувачів є не лише фундаментом для побудови детекторів інсайдерських загроз, а й водночас демонструють усю багатовимірність проблеми, що потребує постійних наукових досліджень і вдосконалення підходів до аналізу поведінкових даних.

Традиційні засоби кібербезпеки, зокрема антивірусні програми та класичні системи виявлення вторгнень, історично спираються на жорстко визначені правила й сигнатури, що описують характеристики уже відомих типів атак. Такі інструменти ефективні лише тоді, коли зловмисна активність має чітко окреслені технічні маркери: відомі шаблони мережевого трафіку, маркери шкідливого коду або типові ознаки експлуатації певних вразливостей. Підхід, заснований на таких сигнатурах, виходить з припущення, що загрозу можна однозначно описати й розпізнати за унікальним відбитком — набором ознак, який відтворюється у різних інцидентах одного типу.

У контексті інсайдерських загроз цей принцип суттєво втрачає ефективність. Дії інсайдера погано вкладаються в парадигму сигнатурного аналізу, оскільки часто не мають заздалегідь відомої технічної сигнатури. Порушник може використовувати на вигляд легітимні інструменти та операції, комбінуючи їх у нетиповий спосіб. Однією з ключових причин низької результативності стандартних технік проти інсайдерів є феномен атак нульового дня та невідомих моделей поведінки: якщо зловмисник застосовує нову техніку, що раніше не спостерігалася в інцидентах і не описана в базах знань про кіберзагрози, сигнатур для такої поведінки просто не існує, а отже, зафіксувати її за допомогою правил неможливо.

До того ж інсайдерські атаки часто проявляються через семантично складні й контекстуальні відхилення, а не через прості технічні показники компрометації. Тому цілком очікувано, що суто сигнатурні системи пропускають значну частину

внутрішніх загроз. Це спричинило зміщення акценту у сфері кібербезпеки в бік аналізу поведінки користувачів та застосування сучасних методів машинного й глибокого навчання для виявлення аномалій. Замість пошуку наперед визначених шкідливих ознак такі підходи намагаються виявляти відхилення від типової поведінки, які можуть свідчити про атаку.

На відміну від сигнатурних методів, які реагують лише на заздалегідь відомі шаблони, поведінкові моделі здатні виявляти нові, раніше не реєстровані загрози. Генеративні моделі, зокрема GAN, можуть бути навчені в режимі без вчителя — виключно на даних, що вважаються відображенням нормальної, легітимної активності користувачів. Їх архітектура дає змогу моделювати складні багатовимірні розподіли без необхідності явно й наперед визначати ознаки, які характеризують аномалії.

Завдяки цьому використання GAN у системах виявлення інсайдерських загроз дає змогу подолати ключові обмеження традиційних підходів. Замість того щоб шукати прямі технічні ознаки атаки, такі моделі будують багатовимірний опис типової поведінки користувача й оцінюють усі подальші відхилення від неї. Це добре відповідає природі інсайдерських атак, які рідко мають чіткі повторювані сигнатури, але проявляються як непрямі, залежні від контексту зміни в поведінці. Іншими словами, GAN допомагають відтворити приховану структуру нормальної активності й виявити випадки, коли нові дії виходять за її межі.

Це має особливе значення в умовах відсутності великого масиву даних для навчання з учителем: система, орієнтована на модель стандартної поведінки, здатна сигналізувати про аномалію, навіть якщо вона не відповідає жодному відомому шаблону зловживання. У підсумку впровадження генеративних підходів (таких як мережі GAN) до аналізу користувацьких логів дозволяє виявляти інсайдерів за рахунок фіксації статистично значущих відхилень у поведінці, а не шляхом пошуку наперед заданих сигнатурних відбитків атаки [17].

Розглянуті принципи роботи генеративно-змагальних мереж демонструють, що ця технологія вийшла далеко за межі початкової сфери створення зображень і стала потужним інструментом для моделювання складних розподілів даних. Ключовою перевагою архітектури є здатність генератора навчатися відтворювати структуру реальної інформації без прямого доступу до розмічених прикладів, спираючись лише на зворотний зв'язок від дискримінатора. Це дозволяє моделі вловлювати приховані закономірності та кореляції, які важко формалізувати за допомогою жорстких правил або простих статистичних методів. Саме

ця властивість робить змагальні мережі перспективним вибором для задач інформаційної безпеки, де доводиться працювати з великими масивами неструктурованих та різнорідних журналів подій.

Специфіка виявлення інсайдерських загроз полягає у тому, що дії зловмисника технічно є легітимними, оскільки виконуються в межах наданих йому повноважень. Відсутність явних ознак зламу робить традиційні сигнатурні методи малоефективними, адже вони розраховані на пошук відомих шаблонів атак. Натомість виникає потреба в аналізі поведінки користувача, де головним індикатором загрози стає відхилення від усталеної норми. Побудова таких поведінкових профілів ускладнюється природою вхідних даних, які зазвичай є багатовимірними, зашумленими та містять складні часові залежності між окремими діями співробітника.

Однією з найгостріших проблем у цій сфері залишається критичний дисбаланс даних. Реальні інциденти трапляються вкрай рідко порівняно з мільйонами записів нормальної щоденної активності. Стандартні алгоритми класифікації в таких умовах схильні ігнорувати приклади атак, що призводить до низької ефективності систем захисту. Використання генеративно-змагальних мереж дозволяє вирішити цю проблему шляхом створення синтетичних даних. На відміну від простих методів інтерполяції, які лише механічно розмножують наявні точки, генератор здатен створювати нові, варіативні сценарії поведінки, що зберігають логічну структуру та часову послідовність реальних дій. Це збагачує навчальну вибірку та допомагає класифікаторам точніше розпізнавати загрози.

Окрім генерації даних, розглянута архітектура пропонує ефективний механізм пошуку аномалій у режимі навчання без учителя. Модель, натренована виключно на легітимній активності, формує детальне уявлення про нормальну поведінку користувачів. У такому разі будь-які дії, які генератор не може якісно реконструювати або які дискримінатор визначає як неправдоподібні, автоматично розглядаються як підозрілі. Такий підхід усуває необхідність у розмічених наборах атак і дозволяє виявляти навіть ті види зловживань, які раніше не були відомі системі безпеки.

Важливим аспектом є врахування часового контексту, оскільки інсайдерські атаки часто розтягнуті в часі й складаються з ланцюжка подій, кожна з яких окремо може виглядати безпечною. Сучасні модифікації генеративних мереж, що використовують рекурентні блоку або механізми уваги, здатні аналізувати цілі послідовності дій. Це дає змогу виявляти складні патерни підготовки до порушення,

такі як повільне накопичення прав доступу чи нетипова активність у позаробочий час, які неможливо помітити при аналізі окремих ізольованих подій.

Водночас практичне застосування цих методів вимагає ретельної підготовки даних. Журнали активності потребують очищення, агрегації та перетворення в числовий формат, придатний для обробки нейронними мережами. Від якості інженерії ознак безпосередньо залежить здатність моделі розрізняти тонкі нюанси поведінки. Також варто враховувати, що навчання змагальних мереж є складним обчислювальним процесом, який потребує налаштування гіперпараметрів для досягнення стабільної рівноваги між генератором і дискримінатором.

Підсумовуючи, поєднання генеративних підходів із методами глибокого навчання створює основу для нового покоління систем виявлення внутрішніх загроз. Вони здатні адаптуватися до змін у поведінці співробітників і забезпечують вищий рівень детекції порівняно з класичними статистичними методами. Перехід від жорстких правил до гнучкого моделювання розподілів даних дозволяє нівелювати розрив між технічними параметрами подій та їх реальним змістом, що є необхідною умовою для побудови надійного захисту корпоративних інформаційних систем.

1.3. Огляд використання машинного навчання для виявлення інсайдерських загроз

Фундаментальна проблема класичних систем виявлення вторгнень у тому, що вони аналізують лише низькорівневі технічні події (наприклад, системні виклики чи мережеві пакети), але майже не розуміють, який реальний намір стоїть за діями користувача. Цей розрив між технічними сигналами та змістовною поведінкою користувача й називають семантичним розривом. Оскільки інсайдерські загрози реалізуються через послідовність легітимних команд, прості сигнатурні методи виявляються безсилими через відсутність статичних ознак атаки [18]. На відміну від зовнішніх вторгнень, де вразливості мають фіксовані закономірності, дії інсайдера варіативні та залежать від контексту. Це зумовлює високий рівень хибно-позитивних спрацювань у системах, які не здатні моделювати природу поведінки людини та адаптуватися до поведінкових ознак, характерних для живих корпоративних середовищ.

Через це традиційні системи виявлення вторгнень, демонструють низьку ефективність: інсайдерська активність рідко має чіткі та повторювані фрагменти

поведінки. Крім того, класичні методи виявлення, які моделюють нормальну активність користувачів, схильні продукувати велику кількість хибних спрацювань: щойно базовий профіль поведінки формалізовано, будь-яке відхилення від нього інтерпретується як потенційна загроза. Такі підходи часто виявляються недостатніми через сильний дисбаланс даних та прихований характер шкідливих дій, коли зловмисник може тривалий час маскувати свої наміри. У результаті фокус сучасних досліджень змістився в бік аналізу поведінки користувачів та застосування методів машинного навчання і глибокого навчання для виявлення аномальних дій співробітників [11].

Виклики в цій сфері зумовлені, зокрема, дисбалансом даних, складною структурою логів та мінливістю поведінки користувачів. У типових корпоративних журналах подій частка зловмисних інцидентів мізерно мала порівняно з масивом легітимної активності (часто менше ніж 1 %). Це ускладнює навчання моделей, тому що навіть передові алгоритми можуть демонструвати обмежену ефективність, коли шкідливі події становлять лише незначну частку всіх записів [19].

Крім того, поведінка користувачів є динамічною: змінюються ролі, обов'язки, типові робочі процеси. Моделі, які не враховують поступову еволюцію поведінкових патернів, швидко втрачають актуальність і деградують за показниками точності прогнозування. Потрібне або регулярне перенавчання, або ж впровадження адаптивних механізмів оновлення моделей, інакше зростає кількість хибних негативів і пропущених атак.

Ще одним критичним фактором є необхідність пояснень: результати автоматичного детектування мають бути зрозумілими аналітикам безпеки. Відсутність прозорості у чорних скриньках рішень, які пропонує машинне навчання, ускладнює формування довіри до таких систем і їх широке впровадження в реальних організаціях. Це, у свою чергу, зумовило активний розвиток напрямку пояснювального штучного інтелекту, покликаного зробити роботу моделей більш прозорою та зрозумілою для людини-аналітика.

Попри згадані труднощі, сучасні методи машинного та глибокого навчання демонструють суттєві переваги у виявленні інсайдерської активності [20]. Розроблено низку підходів, що навчаються ідентифікувати аномалії в поведінці користувачів. Особливо активно використовуються методи навчання без учителя, які дозволяють виявляти відхилення без потреби у великій кількості мічених прикладів атак. До таких підходів належать, зокрема, алгоритми кластеризації та нейромережеві автоенкодера.

Алгоритми кластеризації групують схожі поведінкові сесії й виявляють точки, що не належать жодному з утворених кластерів; такі ізольовані точки інтерпретуються як потенційні інциденти (аномальні користувачі або дії). Наприклад, у [21] застосували методи кластеризації потокових даних для виявлення інсайдерських аномалій у реальному часі.

Нейромережеві автоенкодери, своєю чергою, навчаються відтворювати нормальну поведінку, мінімізуючи помилку реконструкції для типових шаблонів активності. На етапі детектування значне зростання помилки відновлення сигналізує про нетипову активність. У [22] представлено гібридний детектор LAGS, який поєднує локальний автоенкодер ознак із глобальною статистичною моделлю для пошуку аномалій.

Рекурентні мережі на кшталт LSTM та GRU тривалий час були стандартом для аналізу часових рядів логів. Проте за дуже довгих послідовностей подій, характерних для атак, що розгортаються годинами чи днями, у них виникають архітектурні обмеження: градієнт таких мереж поступово затухає, і модель «забуває» події, що сталися давно. У результаті їй важко пов'язати ранні підготовчі дії з пізнішими етапами атаки.

Деякі автори пропонують натомість використовувати механізми «самоуваги» та «трансформери» [23]. На відміну від послідовної обробки в RNN, ці моделі можуть паралельно оцінювати взаємний вплив будь-яких подій у послідовності, незалежно від часової відстані між ними, і таким чином будувати більш цілісний контекстний профіль поведінки користувача. Експерименти на наборах даних CERT показують, що такий підхід покращує якість виявлення атак.

Водночас довгі та нерівномірні часові ряди разом із сильним дисбалансом класів можуть негативно впливати на стабільність навчання RNN-моделей. Зокрема, у [24] відзначають зниження ефективності RNN у разі дуже довгих послідовностей із мінімальною часткою аномальних подій.

Паралельно розвиваються моделі на основі трансформерів, які стають популярними для аналізу послідовностей подій. Завдяки механізму детального навчання, такі моделі можуть для кожної події звернути увагу на всі інші події в історії користувача й визначити, які з них важливі. Це дає змогу добре враховувати довгострокові залежності та загальний контекст його дій. Дослідження показують, що трансформери можна ефективно поєднувати з іншими типами нейронних мереж. Наприклад, моделі, які комбінують трансформер з графовою мережею, демонструють кращу якість виявлення підозрілої активності на реальних

даних CERT порівняно з класичними підходами. Інші роботи показують успішне поєднання тимчасових згорткових мереж із трансформерами для щоденного моніторингу дій співробітників: така гібридна модель змогла виявляти більшість інсайдерських атак протягом приблизно 30 днів, тобто на рівні окремих робочих сесій [25, 26].

Також усе частіше використовують гібридні підходи, які поєднують експертні правила, поведінкову аналітику та машинне навчання, щоб знайти баланс між автоматизацією та контролем з боку людини. У таких системах прості правила спершу відсіюють підозрілі дії, а вже потім спеціалізовані моделі аномалій підтверджують або відхиляють ці спрацювання. Це допомагає зменшити кількість хибних тривог. Щоб аналітики довіряли автоматизованим рішенням, усе більшої ваги набувають пояснювальні методи III. Вони дають змогу показати, чому модель визнала певну активність підозрілою — наприклад, які саме характеристики поведінки найбільше вплинули на рішення. Окремі підходи описують аномалії через набір обмежень між властивостями дій користувача, що робить результат схожим на зрозумілу людині логіку. Загалом, інтеграція таких пояснювальних засобів робить моделі прозорішими для аналітиків і водночас підвищує і рівень безпеки, і довіру команди кібербезпеки до системи.

Окремо слід підкреслити проблему значного дисбалансу даних під час побудови моделей для виявлення інсайдерських загроз. Більшість загальнодоступних датасетів — зокрема CERT від CMU (версії r4.2, r5.2) — містять мільйони подій, з яких лише кілька десятків відповідають зловмисним діям. Така непропорційність (співвідношення нормальних та аномальних прикладів може досягати 1000:1 і більше) призводить до того, що стандартні підходи балансування даних нерідко виявляються неефективними. Наприклад, популярний алгоритм доповнення даних SMOTE генерує нові штучні точки класу меншості шляхом лінійної інтерполяції між наявними прикладами, однак цього може бути недостатньо для складних поведінкових даних.

Журнали дій співробітників мають високорозмірну, нелінійну структуру: вони включають часові залежності, раптові сплески активності, контекстуальні зв'язки між подіями та специфіку ролей користувачів. Відтворити всі ці особливості за допомогою простих методів на кшталт інтерполяції надзвичайно складно. Традиційні алгоритми навчання (як класифікації, так і виявлення аномалій), застосовані безпосередньо до настільки незбалансованих даних, схильні ігнорувати рідкісні класи даних, втрачаючи здатність до узагальнення. У підсумку модель може

демонструвати високі показники на навчальних вибірках, але пропускати реальні атаки або генерувати надто багато хибних сигналів тривоги в реальних умовах експлуатації.

Застосування генеративно-змагальних мереж у кібербезпеці має суттєві відмінності від роботи із зображеннями. Журнали подій є дискретними, категоріальними та залежними від часу даними, що ускладнює пряме використання градієнтного спуску для оновлення генератора. Для вирішення цього в сучасних дослідженнях використовуються адаптовані архітектури, такі як WGAN-GP або методи навчання з підкріпленням SeqGAN, де генератор розглядається як агент. Це дозволяє моделювати складні розподіли ймовірностей переходів між станами системи, зберігаючи логічну структуру сесій користувача, що є недосяжним для простих методів інтерполяції типу SMOTE, які часто створюють «мертві» або нереалістичні комбінації логів.

Важлива перевага генеративних моделей порівняно з класичними методами вирівнювання вибірки полягає в тому, що за правильного налаштування вони краще зберігають різноманіття можливих аномалій. Звичайні методи вибірки часто просто додають дуже схожі одна на одну атаки поруч із наявними прикладами. Натомість GAN навчається так, щоб згенеровані дані в цілому були схожі на реальні, а не тільки повторювали кілька уже відомих патернів. Для аналізу інсайдерських загроз це означає, що модель може створювати нові, раніше не спостережувані комбінації дій, які виглядають правдоподібно, але розширюють діапазон сценаріїв, які класифікатор здатний виявити.

Експериментальні результати застосування GAN для аугментації та доповнення даних демонструють значну перевагу над традиційними методами. Зокрема, використання архітектури SPCAGAN дозволило не просто збільшити кількість мінорних прикладів, а й розширити варіативність сценаріїв атак. Це критично важливо, оскільки реальні атаки часто еволюціонують. Дослідження показують, що класифікатори, навчені на змішаних (реальних + GAN-згенерованих) даних, демонструють підвищення метрики F1-оцінювання та зниження рівня помилкових тривог порівняно з моделями, що використовували ADASYN або SMOTE. Це пояснюється здатністю GAN генерувати валідні послідовності дій, що зберігають часові кореляції, тоді як геометричні методи доповнення даних часто руйнують часову логіку подій.

У режимі навчання без учителя модель GAN трансформується з генератора даних у інструмент вимірювання відхилень. Формально процедура виявлення

базується на обчисленні інтегрального значення аномальності, яке визначається як зважена сума втрат: $A(x) = (1 - \lambda)L_{rec}(x) + \lambda L_{discr}(x)$. Перший компонент, L_{rec} (residual loss), вимірює відстань між реальним вектором подій та його найкращою реконструкцією генератором, тоді як другий, L_{discr} , відображає впевненість дискримінатора у приналежності зразка до нормального розподілу. Високе значення $A(x)$ вказує на те, що поведінка користувача не вкладається у вивчений латентний простір нормальності, що є підставою для генерації інциденту безпеки.

У науковій літературі вже пропонуються подібні підходи. Зокрема, у роботі [10] запропоновано гібридну модель глибокого навчання, що інтегрує генеративно-змагальну архітектуру (SPCAGAN) для ефективного аналізу загроз. Основний акцент у цьому підході зроблено на вирішенні проблеми дисбалансу даних: генерація високоякісних синтетичних зразків поведінки дозволяє значно підвищити точність виявлення інсайдерів навіть за умов обмеженої кількості реальних інцидентів. Загалом застосування GAN суттєво посилює можливості поведінкової аналітики та забезпечує глибше розуміння прихованих ризиків. Генеративні моделі дають змогу подолати низку обмежень традиційних методів, формуючи більш стійкі та адаптивні границі прийняття рішень для розрізнення легітимної активності та складних інсайдерських атак.

Таким чином, машинне навчання сьогодні є ключовою складовою систем виявлення інсайдерських загроз. Проте успішне впровадження таких рішень вимагає врахування згаданих викликів – передусім дисбалансу даних, мінливості поведінки користувачів та необхідності інтерпретованості результатів. У поєднанні з класичними механізмами контролю (зокрема, системами управління доступом, журналюванням та реагуванням на інциденти) системи на основі ML формують перспективний напрям розвитку корпоративної кібербезпеки.

2 АНАЛІЗ МЕТОДИКИ ВИКОРИСТАННЯ GAN ДЛЯ ВИЯВЛЕННЯ ІНСАЙДЕРСЬКИХ ЗАГРОЗ

2.1. Підготовка та використання датасету CERT r4.2

Розробка ефективної методики виявлення інсайдерських загроз за допомогою GAN потребує ретельного підходу до вибору, аналізу та попередньої обробки емпіричних даних. Спираючись на теоретичні засади, розглянуті раніше, у даній роботі як експериментальну базу обрано синтетичний набір даних CERT Insider Threat Dataset версії r4.2. Розглянемо алгоритм трансформації необроблених журналів подій у структурований векторний простір, придатний для навчання нейромережових моделей.

Вибір синтетичного набору даних CERT r4.2 як основної емпіричної бази зумовлений поєднанням кількох чинників, що є критичними для дослідження інсайдерських загроз: наявність достовірної розмітки інцидентів, висока щільність та різноманітність змодельованих сценаріїв, присутність надлишкових даних у поведінці користувачів, а також багатоканальний характер журналів подій. Сукупність цих властивостей дає змогу максимально наблизити умови експерименту до реальних корпоративних середовищ при збереженні контрольованості й відтворюваності експериментальних результатів.

По-перше, використання CERT r4.2 дає можливість працювати з даними, де кожен інцидент інсайдерської активності супроводжується відомою наперед розміткою класу. У реальних корпоративних логах подібна інформація майже завжди недоступна через конфіденційність, юридичні обмеження та складність ретроспективної атрибуції дій окремих користувачів. Як показано в низці робіт, присвячених аналізу інсайдерських загроз, відсутність надійної основи суттєво ускладнює валідацію моделей виявлення аномалій і призводить до завищеної або, навпаки, заниженого оцінювання їхньої ефективності порівняно з реальними умовами експлуатації. У цьому сенсі CERT r4.2 забезпечує збалансований компроміс між реалізмом та експериментальною контрольованістю.

По-друге, версія r4.2 відрізняється підвищеною щільністю та різноманітністю змодельованих сценаріїв. На відміну від ранніх версій, вона описує активність 1000 віртуальних співробітників упродовж 500 днів і містить близько 32 мільйонів подій, у яких реалізовано щонайменше три ключові типи загроз: викрадення даних, саботаж ІТ-інфраструктури та крадіжка інтелектуальної власності. Така

масштабність дає змогу досліджувати рідкісні, контекстно залежні прояви зловмисної поведінки на тлі великого обсягу типових дій, що є принципово важливим для генеративно-змагальних моделей. Як демонструють сучасні роботи з використанням GAN у задачах виявлення аномалій, якісне навчання генератора та дискримінатора потребує багатого й різноманітного нормального профілю поведінки, а також репрезентативної вибірки аномальних патернів. CERT r4.2 задовольняє цю вимогу завдяки поєднанню нормального трафіку з кількома добре задокументованими сценаріями атак.

По-третє, важливою перевагою вибраного датасету є наявність так званого соціального шуму — природних варіацій у поведінці користувачів, пов'язаних з відпустками, зміною робочого графіка, ротацією посад, помилковими або нетиповими, але добросовісними діями. Дослідження у сфері виявлення інсайдерських загроз показують, що саме такі легітимні, але нетипові події часто стають джерелом помилкових спрацювань детекторів, особливо в системах, які базуються на простих порогових або суто статистичних моделях. Наявність соціального шуму в даних CERT r4.2 дозволяє більш реалістично оцінити стійкість запропонованих моделей до помилок першого роду та перевірити, наскільки ефективно GAN-підхід відрізняє шкідливі аномалії від безпечних відхилень у поведінці.

По-четверте, багатоканальна структура датасету, яка включає синхронізовані журнали подій з робочих станцій, мережевих сенсорів, поштових серверів та кадрових систем, створює передумови для комплексного поведінкового аналізу користувачів. Раніше в роботі було показано, що саме поєднання декількох джерел інформації є одним з найперспективніших напрямів розвитку систем виявлення інсайдерських загроз. Різноманітність даних дає змогу моделювати взаємозв'язок між діями користувача в різних підсистемах (наприклад, аномальне мережеве з'єднання, що передуює нетиповому експорту файлів чи змінам у кадрових записах), що суттєво збагачує вхідний вектор ознак для нейромережевих моделей та сприяє виявленню складних, багатокрокових атак.

Таким чином, CERT Insider Threat Dataset r4.2 є найбільш придатним для поставлених у роботі завдань: він одночасно відображає ключові властивості реальних інсайдерських загроз (низька частота, висока контекстуальна залежність, мультиканальна природа), надає повну інформацію про користувача та містить достатній обсяг «соціального шуму» для адекватного оцінювання якості моделей. Це дозволяє не лише валідувати запропонований підхід на репрезентативному

емпіричному матеріалі, а й порівняти отримані результати з іншими роботами, що використовують той самий датасету.

Методика дослідження передбачає інтеграцію різномірних потоків даних в єдиний семантичний простір, придатний для побудови узгодженого векторного подання поведінки користувачів. У межах набору даних CERT r4.2 основними каналами спостереження є журнали доступу (logon.csv), події роботи зі змінними носіями (device.csv), файлові операції (file.csv), веб-активність (http.csv), електронна пошта (email.csv), а також контекстні та психометричні характеристики користувачів (psychometric.csv, LDAP). Кожен із цих каналів має власну схему атрибутів, одиниці вимірювання та частоту реєстрації, тому інтеграція здійснюється через смислове узгодження їхніх атрибутів з уніфікованим набором показників ризику, які надалі використовуються під час нейромережевого моделювання, як це реалізується в сучасних UEBA-підходах [22, 27, 25].

Перший блок даних утворюють журнали доступу, представлені файлом logon.csv, що фіксує події автентифікації (LOGON) та завершення роботи (LOGOFF). На основі цих записів формується часовий профіль активності кожного співробітника: типові години початку та завершення роботи, тривалість сесій, частота повторних входів протягом дня, наявність нічних або вихідних сесій, тощо. У дослідженнях з поведінкового аналізу інсайдерів показано, що зсуви у розкладі роботи, поява нетипових нічних сесій чи різке збільшення тривалості безперервних сесій можуть бути показниками компрометації облікового запису і підготовки до шкідливих дій.

Другий важливий канал — події, пов'язані із зовнішніми носіями інформації, що фіксуються у файлі device.csv. Тут реєструються операції підключення (CONNECT) та відключення (DISCONNECT) USB-пристроїв до робочих станцій. У контексті сценаріїв вивантаження даних подібні події розглядаються у літературі як критичні індикатори потенційного фізичного винесення конфіденційних файлів. Семантична класифікація передбачає перехід від окремих подій до узагальнених характеристик: частоти використання зовнішніх носіїв у розрізі користувача, типових часових шаблонів їх підключення, співвіднесення цих подій із фазами робочого процесу (початок/кінець зміни) та з іншими каналами (наприклад, файловими операціями). Особливий інтерес становлять випадки частих або тривалих підключень USB-пристроїв поза робочим графіком у поєднанні з інтенсивною роботою з документами.

Третій канал — файлові операції, представлені у file.csv. Цей файл містить

інформацію про доступ до файлів, їхні імена, розширення, розташування у файловій системі та тип виконуваної операції (читання, запис, видалення). У сучасних роботах із виявлення інсайдерських загроз саме файловий рівень розглядається як ключове джерело знань про підготовку до ексфільтрації або саботажу, оскільки накопичення конфіденційних документів, масові операції копіювання чи видалення безпосередньо перед інцидентом формують характерні поведінкові патерни [27, 13]. У межах нашої методики особлива увага приділяється доступу до файлів із розширеннями, типовими для документів та звітів (.doc, .pdf, .xls, тощо), оцінці обсягу таких операцій у ковзних часових вікнах, а також їхньому співпадінню з іншими підозрілими подіями (зокрема, підключеннями USB-пристроїв або нетиповою веб-активністю). Важливим індикатором стає робота з конфіденційними файлами у неробочий час або на нетипових для користувача ресурсах.

Четвертий канал становить веб-активність, яка агрегується у файлі `http.csv`. Тут зберігаються URL-адреси, метадані HTTP-запитів та орієнтовні обсяги переданих даних. Веб-канал може розглядатися як один із базових для виявлення витоків через хмарні сервіси (Dropbox, Google Drive, OneDrive, тощо), використання анонімизаторів або підготовки до переходу в іншу організацію (відвідування сайтів пошуку роботи, порталів конкурентів). Класифікація у цьому випадку включає сортування відвідуваних ресурсів за категоріями (корпоративні, хмарні сховища, соціальні мережі, кар'єрні портали, технічні форуми), оцінювання інтенсивності та тривалості сесій для кожної категорії, а також аналіз напрямків та обсягів вихідного трафіку. Підвищену увагу приділено суттєвим зсувам профілю відвідуваних сайтів та незвично великим обсягам переданих даних на хмарні сервіси.

П'ятий канал — електронна пошта, що описується файлом `email.csv`. Цей набір містить інформацію про відправників і одержувачів листів, теми, наявність та розмір вкладень, а також часові мітки. У багатьох роботах інсайдерська активність виявляється через аномальні патерни електронної комунікації: раптове збільшення обсягу вихідної пошти, поява великої кількості листів із вкладеннями до зовнішніх контактів, зміна топології соціального графа користувача. У рамках інтеграційної методики ці події перетворюються на індикатори «аномального обсягу вихідних листів», «зростання частки зовнішніх доменів у комунікаціях», «нестандартних розмірів вкладень», тощо. Додатково аналізується кореляція між поштовою активністю та файловими операціями, що дозволяє виявляти сценарії, у яких документи готуються до відправлення заздалегідь, а сам інцидент ексфільтрації маскується під інтенсивну ділову переписку.

Останній блок становлять контекстні та психометричні дані, представлені у `psychometric.csv` та LDAP-записах. Вони містять інформацію про роль користувача в організації, його департамент, ієрархічне положення та проектну приналежність. У запропонованій архітектурі ці атрибути не є об'єктом генерації, а відіграють роль вектора умов (condition vector y). Це означає, що процес навчання спрямований на апроксимацію умовного розподілу $P_{\text{data}}(x|y)$, де x — вектор поведінкової активності, а y — контекст посади. Такий підхід дозволяє моделі формувати специфічні профілі стандартної поведінки для різних категорій співробітників (наприклад, розрізняти нічну активність системного адміністратора та рядового менеджера), суттєво знижуючи ймовірність помилкових спрацювань, викликаних професійною специфікою.

У результаті інтеграції всі перелічені канали приводяться до спільного простору ознак, структурованого за ідентифікатором користувача та часовими інтервалами (наприклад, денне або годинне вікно). Кожен вектор ознак поєднує часові, структурні та контекстні індикатори ризику, що утворюють багатовимірне подання поведінки, на основі якого у подальшому навчається модель виявлення інсайдерських загроз.

На першому етапі виконується уніфікація часових міток усіх журналів подій до єдиного формату (ISO 8601 або у вигляді POSIX timestamp). Оскільки вихідні CSV-файли можуть містити різні представлення часу (рядкові дати з різною локалізацією, окремі поля для дати та часу, відсутність часової зони), попередня нормалізація дає змогу уникнути двозначностей під час подальшої побудови хронології. На цьому кроці час усіх подій приводиться до єдиної часової зони (наприклад, UTC), вирівнюються можливі розбіжності між джерелами, а також перетворюються складені формати на стандартизовані числові представлення, зручні для подальшої агрегації у фіксованих часових вікнах. Така уніфікація є критичною для коректного злиття подій з різних каналів у єдину впорядковану послідовність, на основі якої будуються профілі поведінки користувачів і формуються вхідні вектори для моделей.

Паралельно із нормалізацією часу виконується очищення даних від технічних артефактів, що забезпечує цілісність та узгодженість вхідного масиву. Зокрема, видаляються дубльовані записи, які можуть виникати внаслідок повторного експорту логів або особливостей роботи систем збору подій; відфільтровуються пошкоджені рядки з неповними чи некоректними полями; уніфікуються кодування текстових полів для запобігання помилкам під час подальшої обробки.

За необхідності виконується базова валідація значень ключових атрибутів (ідентифікаторів користувачів, типів подій, IP-адрес), що дозволяє виявляти й вилучати записи, які суперечать схемі даних. Результатом цього етапу є очищений та узгоджений часовий ряд подій, який слугує надійною основою для подальшої семантичної інтеграції каналів та навчання генеративно-змагальних моделей.

Ключовим методичним рішенням у побудові моделі є вибір базової одиниці аналізу поведінки користувача в часі. У даній роботі, спираючись на огляд підходів, наведений раніше, як таку одиницю обрано агрегований інтервал типу «користувач-день». Іншими словами, кожен вектор ознак описує сукупну активність конкретного співробітника протягом однієї календарної доби, інтегруючи інформацію з усіх доступних каналів (логони, файлові операції, веб-активність, пошта, робота зі змінними носіями, тощо). Такий вибір є компромісом між надто дрібною (рівень окремих подій або сесій) та надто грубою (тиждень, місяць) часовою розбивкою та має низку переваг для навчання генеративно-змагальних моделей.

Добовий інтервал дає змогу будувати більш інформативні вектори ознак. Якщо брати надто короткі відрізки (сесію, годину, окремий запит), багато показників будуть нульовими, а корисні зв'язки між каналами майже не проявлятимуться. У результаті GAN отримує дані, з яких важко вивчити реалістичні патерни. Агрегація на рівні «користувач-день» накопичує достатньо подій кожного типу й формує щільніший опис поведінки, де видно і рутинні дії, і рідкісні, але важливі події.

Також, агрегація за добу зменшує шум у логах. Разові помилки або випадкові сплески активності можуть бути не очевидними в загальній статистиці, тоді як послідовні дні підвищеної активності в певних каналах чи повторювані нічні сесії починають виглядати як стійкі тренди. Для GAN це важливо: модель менше підлаштовується під випадкові особливості поодиноких подій і краще фокусується на накопичених сигналах ризику.

Додатковою перевагою вибору одиниці «користувач-день» є зручність побудови навчальної вибірки та подальшої інтерпретації результатів. Така розбивка дозволяє прямо співвідносити кожен вектор із конкретним користувачем та конкретною датою, що полегшує як аналіз виявлених аномалій (можна розглядати послідовність днів з підозрілою активністю для окремого співробітника), так і інтеграцію результатів із процедурою розслідування інцидентів. Водночас добовий масштаб є достатньо універсальним: він не прив'язується до конкретної

технологічної сесії або робочого місця й дає змогу узгоджувати події з різних систем (комп'ютер, мережа, пошта, кадрові записи) у спільному часовому вікні.

Таким чином, агрегація на рівні «користувач-день» поєднує статистичну повноту опису, чутливість до добових поведінкових патернів та здатність пригнічувати випадковий шум, що робить її добре придатною базовою одиницею аналізу для генеративно-змагальних моделей виявлення інсайдерських загроз.

Для кожного об'єкта типу «користувач-день» на першому етапі формується вектор динамічних поведінкових ознак $x \in \mathbb{R}^{74}$. Цей вектор агрегує змінну в часі активність співробітника в усіх розглянутих каналах і слугує безпосереднім об'єктом генерації для нейромережі. Процес вилучення цих ознак полягає у систематичному перетворенні сирих атрибутів журналів у набір кількісних показників, згрупованих за декількома семантичними напрямками.

Перший напрям становлять кількісні метрики, що відображають інтенсивність різних типів активності протягом доби. До цієї групи належать, зокрема, загальна кількість подій автентифікації та завершення сесій, число підключень змінних носіїв, кількість файлових операцій певних типів, обсяг вихідної поштової активності, тощо. Такі лічильники дозволяють фіксувати як загальний рівень «навантаження» на систему з боку користувача, так і специфічні сплески активності в окремих каналах, які можуть передувати інциденту або супроводжувати його.

Другий напрям охоплює часові характеристики. Тут необроблені часові мітки подій перетворюються на агреговані показники, що описують структуру робочого дня: час першої та останньої зафіксованої активності, приблизну тривалість робочого періоду, розподіл подій між робочими та неробочими годинами. Окремо оцінюється кількість дій, що відбулися в нічний час або у часових інтервалах, нетипових для даного співробітника (наприклад, з 20:00 до 06:00). Такі ознаки дають змогу моделі чутливо реагувати на порушення звичних патернів, які часто розглядаються як один із ранніх маркерів аномальної поведінки.

Третій напрям становлять статистичні показники, пов'язані з обсягами переданих даних та структурою комунікацій. Для каналів HTTP та електронної пошти обчислюються середні, максимальні та сумарні обсяги трафіку за добу, а також відносні показники на кшталт співвідношення відправлених листів, як відправлених стороннім особам, так і колегам, частки запитів на певні категорії ресурсів, тощо. Ці статистики дозволяють виявляти як аномально великі обсяги переданої інформації, так і поступові зміни профілю комунікації (наприклад,

зростання частки зовнішніх одержувачів), які можуть сигналізувати про підготовку до витоку.

Також, до вектора включаються специфічні індикатори загроз, що відображають окремі дії з підвищеним ризиком. Ідеться про цілеспрямовано сконструйовані лічильники, які відповідають відомим зі сценаріїв CERT та з наукових робіт патернам зловмисної активності: кількість операцій копіювання файлів певних типів (наприклад, .pdf, .doc, .xls) на USB-пристрої, частота відвідування сайтів категорії «Job Search» або хмарних сховищ, обсяги вкладених файлів у листах на зовнішні домени, тощо. Такі цільові ознаки безпосередньо інкапсулюють експертні знання про типові вектори інсайдерських атак і посилюють здатність моделі відрізнити потенційно шкідливу поведінку від нормальної.

У сукупності перелічені групи утворюють узгоджений набір вектор х розмірності 74, у якому кількісні, часові та статистичні характеристики поведінки користувача взаємодоповнюють одна одну. На наступному етапі цей блок х доповнюється 72-вимірним поданням одиничними векторами статичного контексту. Це багатовимірне подання забезпечує достатню виразність для моделювання складних патернів поведінки користувачів і слугує основою для подальшого застосування генеративно-змагальних мереж у задачі виявлення інсайдерських загроз.

Специфіка навчання генеративно-змагальних мереж висуває суворі вимоги до розподілу та масштабу вхідних даних. Для того щоб забезпечити стабільну збіжність процесу навчання та зменшити ризик згасання градієнтів, необхідно не лише підібрати архітектуру мережі й оптимізатор, а й відповідним чином підготувати вхідні ознаки. У даній роботі це реалізується через послідовне застосування логарифмічної трансформації до кількісних ознак та подальшого масштабування всіх компонентів векторів у фіксований діапазон значень, узгоджений з областю визначення активаційних функцій у вихідних шарах генератора та дискримінатора.

Поведінкові дані CERT характеризуються розподілами з вираженою асиметрією: для більшості користувачів переважна частина днів має порівняно низький рівень активності, але періодично трапляються доби з екстремально великою кількістю подій певного типу (наприклад, масові файлові операції чи значний обсяг HTTP-трафіку). Якщо до таких величин застосувати лише лінійну нормалізацію, основна маса спостережень буде стиснута у вузький інтервал поблизу нуля, тоді як кілька екстремальних значень визначатимуть масштаб ознаки. У результаті дискримінатор стає малочутливим до відмінностей між

звичайними днями, а генератору важко відтворювати тонкі варіації поведінки в області, де зосереджено більшість точок.

Щоб зменшити асиметрію розподілів, до всіх кількісних ознак v застосовується логарифмічна трансформація:

$$v' = \ln(v + 1).$$

Додавання одиниці гарантує коректність перетворення для нульових значень, а сам логарифм стискає великий діапазон величин, наближуючи екстремальні значення ближче до основної маси. У результаті розподіли окремих ознак стають більш плавними і наближаються до квазі-нормальних, що полегшує оптимізацію параметрів мережі й робить ландшафт функції втрат більш стабільним. З практичної точки зору це підвищує стабільність навчання GAN, зменшуючи ймовірність ситуацій, коли поодинокі великі значення домінують у градієнтах і спричиняють нестійкі оновлення ваг.

Після логарифмічної трансформації виконується друга стадія нормалізації — масштабування ознак до заданого числового інтервалу. Оскільки у вихідному шарі генератора використовується функція активації $\tanh$, область значень якої обмежена відрізком $[-1, 1]$, доцільно привести вхідні дані (а також цільові вектори для дискримінатора) до цього ж діапазону. Для кожної трансформованої ознаки v' виконується лінійне перетворення за схемою мінімаксного наближення:

$$\tilde{v} = 2 \cdot \frac{v' - \min(v')}{\max(v') - \min(v')} - 1,$$

де $\min(v')$ та $\max(v')$ обчислюються виключно на основі навчальної вибірки (training set). Такий підхід дозволяє уникнути витоку інформації з тестових даних у процес побудови моделі: параметри масштабування фіксуються після аналізу тренувального підмножини, а потім застосовуються до валідаційного та тестового наборів без переоцінювання.

Комбінація логарифмічної трансформації та масштабування в $[-1, 1]$ забезпечує для більшості ознак компактний, збалансований діапазон значень без домінування поодиноких екстремальних спостережень. У такому просторі GAN отримує більш керований вхід: градієнти не затухають через надто малі або надто великі аргументи активаційних функцій, а геометрія даних стає більш

придатною для відтворення генератором. У результаті процес навчання набуває більш стабільного характеру, а модель краще відтворює реалістичні профілі типової поведінки, на тлі яких аномалії інсайдерської активності проявляються більш контрастно.

Атрибути, що описують роль користувача, його підрозділ та інші дискретні характеристики, кодуються у векторній формі за допомогою методу одиничного кодування. Отриманий вектор умов u конкатенується з шумовим вектором z на вході генератора, а також із вектором реальних або згенерованих ознак x на вході дискримінатора. Оскільки ці дані є статичними умовами і не потребують генерації (зворотного поширення помилки через операцію вибору категорії), використання специфічних технік диференційовної дискретизації (на кшталт Gumbel-Softmax) не є необхідним. Використання одиничного кодування забезпечує чітке розділення категоріальних ознак у просторі умов, дозволяючи нейромережі ефективно вивчати залежності між організаційним контекстом та патернами поведінки.

Для кожного об'єкту «користувач-день» формується композитний вектор опису стану $S \in \mathbb{R}^{146}$, який об'єднує динамічні поведінкові ознаки та статичний контекст. Таким чином, добова активність кожного співробітника у всіх розглянутих каналах спостереження подається у вигляді єдиного 146-вимірного вектора $S = [x, y]$, де $x \in \mathbb{R}^{74}$ відповідає нормованим континуальним поведінковим ознакам, а $y \in \{0, 1\}^{72}$ — поданню контекстних атрибутів одиничним кодуванням. Таке узгоджене векторне представлення слугує базою для подальшого навчання GAN-моделі.

Після визначення схеми подання даних ключовим питанням стає стратегія формування вибірок для експерименту. Ця методика передбачає спеціальний підхід до поділу на навчальну, валідаційну та тестову множини. Навчальна вибірка формується таким чином, щоб містити виключно «чисті», нормальні (benign) дні активності користувачів. Для цього на основі доступної розмітки усі події, пов'язані з інсайдерською активністю, ідентифікуються та вилючаються з тренувального набору. Такий підхід дає змогу GAN наближено відтворити латентний розподіл нормальної поведінки P_{data} , не додаючи до нього аномальні приклади, які могли б викривити результати оцінювання та знизити чутливість до справжніх загроз.

Окремо виділяється валідаційна вибірка, що складається з частини нормальних даних, не використаних безпосередньо під час навчання. Вона слугує для налаштування гіперпараметрів моделі (розмір латентного простору, параметри регуляризації, швидкість навчання, тощо), вибору оптимальної кількості епох та

реалізації механізму ранньої зупинки. Оскільки валідаційний набір формально «невідомий» моделі, якість реконструкції та значення цільових критеріїв на ньому дають більш об'єктивне оцінювання ступеня узгодження генератора з розподілом нормальної поведінки, ніж показники, отримані на тренувальних даних.

Тестова вибірка містить суміш нормальних днів активності (які не з'являлися ні в навчальній, ні у валідаційній вибірках) та всіх доступних інсайдерських інцидентів. Саме на цьому наборі проводиться фінальне оцінювання ефективності детектора, для кожного об'єкта типу «користувач-день» обчислюється міра аномальності (наприклад, на основі помилки реконструкції або значення дискримінатора), після чого будуються метрики якості класифікації нормальних та інсайдерських днів (ROC-криві, AUC, показники чутливості та специфічності, тощо). Таким чином, тестовий набір відображає реалістичну для експлуатації ситуацію, у якій домінують нормальні спостереження, а інсайдерські епізоди становлять малу, але критично важливу частку.

Описана стратегія формування вибірок забезпечує базову чистоту експерименту: модель навчається виключно тому, що вважається нормальною поведінкою, і надалі інтерпретує загрози як статистичні відхилення від вивченого розподілу, не покладаючись на попередньо відомі сигнатури атак. Це повністю відповідає концепції виявлення аномалій, обґрунтованій раніше.

Таблиця 2 – Специфікація ознак для моделі cGAN

Група	К-ть	Опис	Тип
<i>Вектор поведінки x (генерується моделлю)</i>			
Logon	4	Час входу/виходу, тривалість сесії, нічна робота	Log + MinMax
Device	8	USB-активність (робочий/неробочий час)	Log + MinMax
File	32	Файлові операції (.doc, .pdf, .xls, .zip)	Log + MinMax
HTTP	18	Web-категорії, обсяги трафіку	Log + MinMax
Email	12	Зовнішні листи, вкладення	Log + MinMax
Підсумок x	74	Континуальні динамічні ознаки	
<i>Вектор умов y (подається на вхід)</i>			
Контекст	72	Роль, Підрозділ, Проект, тощо	Категоріальні
Всього	146	Вхідний простір дискримінатора [x, y]	

Представлена методика підготовки даних слугує фундаментом для побудови генеративної моделі. Її ключовими елементами є формування інформативного векторного подання «користувач-день», урахування організаційного та психометричного контексту, а також коректна математична нормалізація ознак

із урахуванням специфіки навчання GAN. Сукупність цих кроків дозволяє суттєво зменшити негативний вплив гетерогенності джерел, різнорідних масштабів вимірювання та наявного в даних шуму, перетворюючи журнали подій на узгоджений простір ознак зі стабільними статистичними властивостями, сприятливими для моделювання. На такій основі GAN отримують змогу ефективно вивчати латентну структуру нормальної поведінки користувачів і, відповідно, більш надійно виявляти приховані інсайдерські загрози як відхилення від цього розподілу.

2.2. Методика використання GAN для виявлення інсайдерських загроз

Спираючись на сформований векторний опис «користувач–день» (структура C), кожен день активності користувача описується композитним вектором стану

$$S_{u,d} = [x_{u,d}; y_{u,d}] \in \mathbb{R}^{146},$$

де $x \in [-1, 1]$ ⁷⁴ — нормовані динамічні поведінкові ознаки (кількості подій, обсяги даних, часові характеристики, тощо), а $y \in \{0, 1\}$ ⁷² — статичний та квазістатичний контекст (роль користувача, підрозділ, інші категоріальні атрибути). Аналогічний векторний опис будується і для інших варіантів структурування логів (A, B, D), але на рівні події, 30-хвилинного вікна або сесії користувач:комп'ютер.

Окремо з файлу answers.tar.bz2 формується розмічений набір інцидентів, який задає бінарні мітки «норма / інсайдерська загроза». Ці мітки потрібні для формування валідаційної та тестової вибірок і для обчислення метрик якості. Самі ж генеративні моделі навчаються у такому режимі: на входи подаються вектори S без явного використання міток під час оновлення параметрів, що відповідає типовому сценарію виявлення аномалій, коли заздалегідь відомо, що більшість спостережень є нормальними.

Усі розглянуті архітектури — AnoGAN, BiGAN та WGAN-GP — націлені на апроксимацію розподілу нормальної поведінки користувачів. Базова схема GAN включає генератор G та дискримінатор D . Генератор приймає на вхід латентний шум $z \sim p_z(z)$ (як правило, $\mathcal{N}(0, I)$) та контекст y і повертає синтетичний поведінковий вектор

$$\hat{x} = G(z, y),$$

а дискримінатор отримує пари (x, y) або $(\hat{x}, y)$ і намагається відрізнити реальні приклади від згенерованих.

У генераторі вектори z та y конкатенуються на вході першого прихованого шару багат шарового перцептрона, а в дискримінаторі / критику аналогічним чином конкатенуються поведінковий вектор і контекст:

$$[z; y] \xrightarrow{G} \hat{x}, \quad [x; y] \xrightarrow{D} D(x, y).$$

Це дозволяє моделі вивчати умовний розподіл $p(x \mid y)$, не генеруючи дискретні категоріальні ознаки.

Стандартна цільова функція GAN у такій постановці має вигляд

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{\text{data}}(x|y)} [\log D(x, y)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z, y), y))].$$

Дискримінатор D максимізує цю функцію, намагаючись правильно класифікувати реальні та фейкові зразки, тоді як генератор G мінімізує її, намагаючись «обдурити» дискримінатор.

У класичній схемі AnoGAN спочатку навчається звичайна GAN на (переважно) нормальних даних. Після навчання генератора та дискримінатора для кожного нового зразка x шукається такий латентний вектор z^* , що реконструкція $G(z^*, y)$ максимально близька до x . Це робиться шляхом ітераційної оптимізації z при фіксованих параметрах G :

$$z^* = \arg \min_z \left(\lambda_{\text{rec}} \cdot \|x - G(z, y)\|_1 + \lambda_{\text{feat}} \cdot \|f_D(x, y) - f_D(G(z, y), y)\|_1 \right),$$

де $f_D(\cdot)$ — вектор активацій одного з прихованих шарів дискримінатора. Після цього отриманий мінімум використовується для оцінювання аномальності: чим гірше реконструкція та узгодження ознак, тим вища ймовірність аномалії.

Архітектура BiGAN [7] розширює базову схему шляхом введення енкодера:

$$E : (x, y) \mapsto z,$$

який навчається одночасно з генератором. Дискримінатор у BiGAN оперує не лише

даними, а й латентними кодами, розрізняючи пари:

$$(x, E(x, y), y) \quad \text{та} \quad (G(z, y), z, y).$$

Оптимізаційна задача формулюється як:

$$\min_{G,E} \max_D V(D, G, E) = \mathbb{E}_{x \sim p_{\text{data}}} [\log D(x, E(x, y), y)] + \mathbb{E}_{z \sim p_z} [\log(1 - D(G(z, y), z, y))].$$

За оптимального навчання E наближається до оберненої функції для G , тобто виконується $x \approx G(E(x, y), y)$. Це дозволяє замінити повільний пошук по z (як в AnoGAN [28]) прямим кодуванням через E і виконувати реконструкцію в один прохід:

$$x \xrightarrow{E} z \xrightarrow{G} \hat{x} \approx x.$$

Оцінювання аномальності в цьому випадку природно базується на помилці реконструкції та різниці активацій дискримінатора між x та $\hat{x}$.

Класична GAN страждає від проблеми зникаючих градієнтів, коли дискримінатор стає надто сильним, а підтримки розподілів P_r та P_g майже не перетинаються. Wasserstein GAN замінює дивергенцію Дженсена–Шеннона на відстань Вассерштейна, що забезпечує змістовний градієнт навіть за малої області перетину розподілів.

Критик D (аналог дискримінатора) в WGAN навчається максимізувати різницю середніх значень своїх виходів для згенерованих і реальних даних:

$$L_{\text{WGAN}} = \mathbb{E}_{\tilde{x} \sim p_g} [D(\tilde{x}, y)] - \mathbb{E}_{x \sim p_{\text{data}}} [D(x, y)].$$

Щоб ця різниця коректно наближала відстань Вассерштейна, D повинен бути Ліпшицевою функцією з константою 1. У реалізації WGAN-GP це досягається додаванням штрафу градієнта:

$$L_{\text{critic}} = L_{\text{WGAN}} + \lambda \cdot \mathbb{E}_{\hat{x}} \left[\left(\|\nabla_{\hat{x}} D(\hat{x}, y)\|_2 - 1 \right)^2 \right],$$

де $\hat{x} = \epsilon x + (1 - \epsilon)\tilde{x}$, $\epsilon \sim U[0, 1]$. Параметр λ (у експериментах приймалося $\lambda = 10$) контролює жорсткість обмеження.

Для всіх трьох архітектур було використано єдиний набір ознак та однакові

схеми попередньої обробки даних (логарифмування та масштабування в $[-1, 1]$). Ключові гіперпараметри наведено в таблиці 3.

Таблиця 3 – Параметри архітектур GAN для векторів «Користувач-День»

Параметр	AnoGAN	BiGAN	WGAN-GP
Learning rate	$2 \cdot 10^{-4}$	$1 \cdot 10^{-4}$	$1 \cdot 10^{-4}$
Batch size	64	128	64
Latent dim d_z	100	64	100
Condition dim d_y	72	72	72

Для запобігання перенавчанню та пом'якшення колапсу мод застосовувався набір практичних евристик:

- Label smoothing для дискримінатора у GAN/BiGAN: замість жорстких міток 1 для реальних зразків використовувалися випадкові значення з інтервалу $[0.9, 1.0]$, що зменшує насичення градієнтів.
- У WGAN-GP на кожне оновлення генератора виконувалося $n_{\text{critic}} = 5$ кроків оновлення моделі критика, щоб розрахунок відстані Вассерштейна був достатньо точним.
- Для всіх моделей на входи дискримінатора / критика додавали слабкий гаусовий шум $\mathcal{N}(0, 0.01)$, амплітуда якого поступово зменшувалася протягом навчання.
- Навчальні пакети формувалися так, щоб у кожній групі були присутні як дні з майже нульовою активністю, так і дні з високим навантаженням, що запобігає тривіальному виродженню генератора в нульовий вектор.

Після завершення навчання параметри G , D та, за наявності, E фіксуються. Для кожного тестового зразка (x, y) обчислюється скалярне оцінювання аномальності $A(x, y)$. У загальному випадку вона будується як комбінація двох компонент:

- реконструкційної помилки $L_{\text{rec}}(x)$ (наприклад, L_1 -норма між x та $\hat{x}$),
- помилки співпадіння ознак дискримінатора $L_{\text{feat}}(x)$ (різниця активацій прихованого шару для x та $\hat{x}$).

Оскільки ці дві величини мають різні масштаби, вони стандартизуються (Z-score) за

статистиками, обчисленими на валідаційній підмножині нормальних даних:

$$A(x, y) = \alpha \cdot \frac{L_{\text{rec}}(x) - \mu_{\text{rec}}}{\sigma_{\text{rec}}} + (1 - \alpha) \cdot \frac{L_{\text{feat}}(x) - \mu_{\text{feat}}}{\sigma_{\text{feat}}},$$

де $\mu_{\text{rec}}, \sigma_{\text{rec}}$ та $\mu_{\text{feat}}, \sigma_{\text{feat}}$ — середні та стандартні відхилення відповідних втрат на валідаційній вибірці. У наведених експериментах коефіцієнт α було обрано рівним 0.9, що надає більшу вагу фізичній реконструкції.

Фінальне бінарне рішення приймається шляхом порівняння $A(x, y)$ з порогом τ :

$$\text{Label}(x) = \begin{cases} 1 \text{ (Anomaly)}, & \text{якщо } A(x, y) > \tau, \\ 0 \text{ (Normal)}, & \text{якщо } A(x, y) \leq \tau. \end{cases}$$

У роботі використовувалися два режими налаштування порогу. За наявності розмічених інцидентів поріг τ обирається за критерієм максимізації F_1 -оцінювання на валідаційній змішаній вибірці:

$$\tau^* = \arg \max_{\tau} \left(\frac{2 \cdot \text{Precision}(\tau) \cdot \text{Recall}(\tau)}{\text{Precision}(\tau) + \text{Recall}(\tau)} \right).$$

У справжньому сценарії без міток, коли розмічені атаки відсутні, поріг можна задавати як 99-й або 99.9-й перцентиль розподілу $A(x, y)$ на нормальних даних, тим самим фіксуючи очікуваний рівень хибних спрацювань.

2.3. Методика збагачення даних за допомогою GAN

Фундаментальною проблемою при навчанні моделей для виявлення інсайдерських загроз є дисбаланс класів. Нехай задано навчальну вибірку

$$\mathcal{L} = \{(x_i, y_i)\}_{i=1}^N,$$

де $y_i \in \{0, 1\}$, і $y_i = 1$ означає інцидент (шкідливу активність), а $y_i = 0$ — нормальну подію. Позначивши кількість позитивних та негативних прикладів як N_+ та N_- відповідно, ступінь дисбалансу визначається співвідношенням

$$\text{IR} = \frac{N_-}{N_+}.$$

У реальних наборах даних інсайдерських загроз (наприклад, сімейство наборів CERT) значення IR часто сягає 10^3 – 10^4 і більше. Тобто на один інцидент припадають тисячі чи десятки тисяч нормальних подій. За таких умов стандартні класифікатори, що оптимізують загальну точність, фактично ігнорують мінорний клас. Зокрема, модель, яка завжди прогнозує нормальність ($\hat{y} = 0$ для всіх спостережень), може досягати точності понад 99.9%, але при цьому не виявить жодного інциденту. Це робить метрику загальної точності непридатною як основний критерій. Натомість на перший план виходять метрики, чутливі до якості виявлення рідкісного класу: Recall, Precision, F_1 -міра, а також площі під ROC- та PR-кривими. У задачах із сильним дисбалансом доцільно аналізувати як AUC-ROC (якість ранжування загалом), так і AUPRC (баланс між Precision та Recall позитивного класу). Для детектора GAN основний акцент робиться на AUC-ROC.

Для подолання проблеми дисбалансу класів на рівні даних доцільно використати GAN, для синтезу нових прикладів мінорного класу. Підхід GAN принципово відрізняється від простих методів вибірки: замість дублювання чи лінійної інтерполяції існуючих зразків генератор GAN навчається породжувати штучні аномалії, що статистично не відрізняються від реальних інцидентів. У нашому випадку генеративна модель будується на основі даних CERT r4.2 — набору логів, який характеризується високою розмірністю та гетерогенністю ознак (містить суміш числових і категоріальних полів). Для ефективної генерації таких табличних даних із складним мультимодальним розподілом використовується архітектура CTGAN, що вирішує проблему моделювання суміші дискретних та неперервних колонок. Замість прямого породження всього вектора ознак CTGAN застосовує механізм нормалізації, специфічної для окремих мод розподілу: кожне неперервне значення v представляється у вигляді

$$v = \alpha_k \oplus \beta_k,$$

де α_k — це вектор одиничного кодування, що вказує на належність до k -ї компоненти розподілу (наприклад, компоненти гаусової суміші), а β_k — скаляр, що задає положення всередині цієї компоненти. Така декомпозиція дає змогу моделювати багатомодальні неперервні розподіли.

Для дискретних змінних, де стандартні GAN стикаються з нестабільністю через недиференційовану операцію вибору класу, в CTGAN інтегровано техніку Гамбел–Софтмакс [29]. Вона забезпечує диференційовану апроксимацію вибірки з

категоріального розподілу:

$$z_i = \frac{\exp((\log \pi_i + g_i)/\tau)}{\sum_{j=1}^K \exp((\log \pi_j + g_j)/\tau)},$$

де π_i — ймовірність i -ї категорії, g_i — шум із розподілу Гамбеля, а τ — температурний параметр. Використання Gumbel-Softmax дозволяє генератору навчатися методом зворотного поширення похибки навіть на змішаних даних, зберігаючи при цьому структурну цілісність атрибутів інсайдерської активності.

В рамках задач ITD на даних CERT r4.2 генеративні моделі застосовуються у табличній або умовній формах. Зокрема, CTGAN навчається моделювати умовний розподіл табличних ознак позитивного класу

$$p(x \mid y = 1, c),$$

де $y = 1$ відповідає інсайдерській атаці, а c — додатковий контекст (наприклад, роль користувача, підрозділ, день тижня, тип ресурсу, тощо). Врахування контекстних змінних дозволяє генератору зберігати різноманітні фактори, що впливають на поведінкові дані. Таким чином, генеративний підхід не тільки збалансовує вибірку, а й зберігає важливі властивості мінорного класу. Зокрема, він дозволяє:

- зберігати складні нелінійні залежності між ознаками. Наприклад, взаємозв'язки між часовими, мережевими та файловими подіями користувача зберігаються у синтетичних даних, чого важко досягти простими методами на кшталт лінійної інтерполяції існуючих точок. Такий підхід забезпечує семантичну узгодженість згенерованих аномалій;

- відтворювати багатомодальну структуру класу атак. Різні типи інсайдерської активності (витік даних, саботаж, зловживання привілеями, тощо) утворюють окремі кластери у просторі поведінкових патернів. GAN здатен відобразити ці декілька мод патернів і генерувати нові приклади в кожній з них, на відміну від методів, що усереднюють або генерують точки лише поблизу наявних зразків;

- генерувати цілі послідовності подій. У разі потреби GAN може бути розширений рекурентними або трансформерними компонентами у складі генератора для побудови синтетичних послідовностей дій. Це дозволяє зберегти хронологічний порядок і логічну узгодженість сценарію атаки, тобто відтворити

динамічні поведінкові патерни, характерні для інсайдерських інцидентів.

Практична реалізація такого збагачення даних полягає в тренуванні генеративної моделі на реальних даних мінорного класу, після чого генерується множина штучних прикладів атак. Зокрема, після навчання GAN на вихідних інцидентах CERT r4.2 отримують множину синтетичних атак

$$\tilde{\mathcal{D}}_+ = \{(\tilde{x}_j, \tilde{y}_j = 1)\}_{j=1}^M,$$

яка за розподілом наближено відповідає реальним атакам (враховуючи змішану природу неперервних та дискретних ознак, типову для логів CERT). Далі формується розширена навчальна вибірка шляхом об'єднання оригінальних даних із синтетичними:

$$\mathcal{D}_{\text{aug}} = \mathcal{D} \cup \tilde{\mathcal{D}}_+.$$

На цьому збалансованому наборі і тренується модель-класифікатор аномалій (наприклад, Random Forest або Isolation Forest). Завдяки появі великої кількості правдоподібних атак у $\mathcal{D}_{\text{aug}}$ класифікатор більше не ігнорує рідкісні сценарії — він отримує різноманітні приклади аномальної поведінки і вчиться розпізнавати узагальнені закономірності шкідливої активності, а не лише запам'ятовує окремі точки.

Застосування GAN як джерело генерації вибірки для рідкісних атак дає помітний виграш у якості детекції. Зокрема, класифікатори, навчені на аугментованих даних, значно перевершують варіанти без такого збагачення за показниками Recall, F_1 та AUC. Генератор синтезує різноманітні та статистично узгоджені приклади атак (у тому числі рідкісні патерни), що змушує модель ухвалення рішень будувати більш стійкі правила розділення. За рахунок покриття всього простору поведінкових ознак мінорного класу підвищується ймовірність того, що кожна композиція ознак атак буде розпізнана алгоритмом.

Таким чином, вибір синтетичного набору даних CERT версії r4.2 створює надійну емпіричну базу для дослідження, оскільки поєднує масштабність подій із точною розміткою інцидентів, що є недосяжним при роботі з реальними конфіденційними журналами. Розроблена процедура попередньої обробки вирішує проблему гетерогенності вхідної інформації шляхом інтеграції різнорідних каналів спостереження, таких як входи в систему, робота з файлами, веб-активність та електронна пошта, у єдиний векторний простір. Ключовим методичним

рішенням стало використання агрегованого інтервалу «користувач-день», що дозволяє сформувати інформативні вектори ознак розмірністю 146 компонентів. Такий підхід ефективно поєднує динамічні характеристики поведінки з незмінним організаційним контекстом, надаючи нейромережі цілісну картину діяльності співробітника.

Критично важливим етапом підготовки даних стало впровадження спеціалізованого конвеєра нормалізації, розробленого з урахуванням специфіки навчання генеративно-змагальних мереж. Застосування логарифмічної трансформації до кількісних показників дозволило нівелювати вплив асиметрії розподілів, характерної для дій користувачів, та збалансувати внесок рідкісних подій у процес навчання. Подальше масштабування ознак до фіксованого числового діапазону забезпечило узгодженість вхідних даних з активаційними функціями генератора, що є необхідною умовою для стабільної збіжності оптимізаційних алгоритмів та запобігання проблемам зникаючих градієнтів.

Запропонована методика виявлення загроз базується на використанні архітектур, здатних моделювати складні ймовірнісні розподіли нормальної поведінки. Порівняльний аналіз показав, що класична схема AnoGAN, хоч і забезпечує теоретичне підґрунтя для пошуку аномалій, є надто ресурсомісткою для реальних систем через необхідність ітераційного пошуку в латентному просторі. Тому перевагу віддано архітектурі BiGAN із додатковим кодувальником, що дозволяє виконувати реконструкцію даних за один прохід, та моделі WGAN-GP, яка завдяки використанню відстані Вассерштейна та штрафу градієнта демонструє високу стабільність навчання. Визначення аномальності здійснюється через комплексне оцінювання, що враховує як помилку фізичного відтворення вектора ознак, так і невпевненість дискримінатора, що дозволяє фіксувати найменші відхилення від профілю норми.

Окрім безпосереднього виявлення аномалій, методика передбачає використання генеративних моделей для вирішення проблеми критичного дисбалансу класів. Застосування архітектури CTGAN дозволяє синтезувати високоякісні зразки шкідливої активності, які зберігають статистичні властивості та логічні зв'язки реальних інцидентів. Збагачення навчальної вибірки такими даними дає змогу сформувати збалансований набір для тренування класифікаторів, суттєво підвищуючи їхню чутливість до рідкісних сценаріїв атак. Так, розроблений підхід формує замкнений технологічний цикл, який охоплює всі етапи від уніфікації сирих логів до побудови стійких детекторів інсайдерських загроз.

3 МОДЕЛЬ ВИЯВЛЕННЯ ІНСАЙДЕРСЬКИХ ЗАГРОЗ НА БАЗІ GAN

3.1. Результати навчання GAN для самостійного виявлення інсайдерських загроз

Першим кроком у проєктуванні системи є вибір рівня часової агрегації даних. На етапі підготовки всі журнали активності користувачів CERT були об'єднані в один датасет, доповнений бінарними мітками (нормальна активність позначена як 0, інцидент інсайдерської загрози — як 1). Ці мітки використовувалися лише для оцінювання якості моделей; навчання генеративних мереж виконувалося в напіваавтоматичному режимі, переважно на підмножині нормальних даних.

Для повноти дослідження дані були структуровані кількома різними способами, що відповідають чотирьом варіантам представлення вхідних зразків:

A) Неструктурована серія подій — модель бачить поодинокі записи журналів без додаткової агрегації.

B) 30-хвилинні вікна активності — всі події користувача усередині фіксованого часового інтервалу агрегуються в один вектор ознак.

C) 24-годинний робочий день — добова агрегація активності, коли всі події користувача за день об'єднуються в єдиний опис.

D) Сесії користувач:комп'ютер — сесія визначається як послідовність дій певного користувача на одному й тому самому комп'ютері між подіями входу та виходу із системи.

Роботи, зокрема [19], демонструють, що агрегація активності на рівні «користувач–день» забезпечує збалансований компроміс між деталізацією та здатністю моделі фіксувати довгострокові залежності в поведінці користувачів. Представлення на рівні окремих сесій часто виявляється шумним через випадкові короткі входи та технічні події, тоді як тижнева агрегація («користувач–тиждень») згладжує короточасні сплески аномальної активності й ускладнює виявлення контекстних загроз [19, 27]. У даній роботі добовим сесіям відповідає структура даних C, тоді як розбиття по сесіям використовується у структурі D. Це дозволяє прямо порівняти висновки попередніх досліджень з результатами на різних рівнях агрегації.

У варіанті C для кожного користувача u та кожного дня d формувалася композитний вектор опису стану $S_{u,d} \in \mathbb{R}^{146}$, який об'єднує динамічні поведінкові ознаки та статичний контекст. Формально цей вектор складається з

двох компонентів: $x \in \mathbb{R}^{74}$ (нормовані показники активності) та $y \in \{0,1\}^{72}$ (категоріальний контекст). Для інших структур (A, B, D) аналогічний набір ознак обчислювався не на рівні дня, а відповідно для окремих подій, 30-хвилинних інтервалів або наборів «користувач–комп’ютер». Процес вилучення ознак базувався на перетворенні часових міток та атрибутів журналів у кількісні показники, що є типовим підходом для роботи з журналами CERT.

Для логів входу в систему (logon.csv) розраховувалися не лише фактичні кількості входів та виходів, але й контекстні метрики: сумарна тривалість робочих сесій у межах обраного інтервалу (подія/вікно/день/сесія), час першого входу та останнього виходу, кількість переривань сесій, а також наявність активності у неробочий час (наприклад, між 20:00 та 06:00). Такі ознаки дозволяють моделі вивчити характерні циркадні ритми нормальних користувачів і виявляти відхилення від звичного профілю поведінки (нічні входи, аномально довгі сесії, тощо), що узгоджується з результатами робіт, орієнтованих на часово-залежні патерни поведінки [27].

Для файлової активності та використання зовнішніх пристроїв ключовими ознаками стали: кількість підключень USB-накопичувачів, зокрема у неробочий час, сумарний обсяг прочитаних та записаних даних у межах відповідної одиниці агрегації, кількість операцій із файлами певних типів. Оскільки інсайдерські загрози часто пов’язані з ексфільтрацією інформації, окремо виділялися лічильники для операцій з файлами чутливих розширень (наприклад, .doc, .pdf, .xls), а також частка таких операцій від загальної файлової активності. Подібні показники копіювання великих обсягів даних або доступу до чутливих документів у нетиповий час вважаються інформативними індикаторами сценаріїв викрадення даних.

Аналіз веб-активності включав агрегацію кількості відвіданих сторінок за високорівневими категоріями (соціальні мережі, хмарні сховища, сайти пошуку роботи, тощо), а також підрахунок сумарного обсягу завантажених та вивантажених даних. Додатково обчислювалися показники, пов’язані з потенційно ризикованими напрямками активності, наприклад інтенсивність доступу до публічних файлових хостингів та нетипова для конкретного користувача зміна структури відвідуваних сайтів.

У підсумку попередньої обробки було сформовано простір ознак, достатньо багатий для відображення багатоканальної поведінки, але придатний для глибокого навчання. Табличні дані журналів подій характеризуються високою гетерогенністю масштабів та важкими хвостами розподілів: бінарні індикатори сусідять з ознаками

обсягу даних, що можуть сягати мільйонів. Для забезпечення стабільної збіжності градієнтного спуску в GAN критично важливо привести розподіли до вигляду, близького до нормального.

Тому в даній роботі, на відміну від простої лінійної нормалізації, було застосовано двоетапну процедуру попередньої обробки. На першому етапі до всіх кількісних ознак v застосовувалася логарифмічна трансформація для зменшення асиметрії розподілів:

$$v' = \ln(v + 1).$$

Додавання одиниці гарантує коректність операції для нульових значень, типових для розріджених логів. На другому етапі виконувалося масштабування отриманих значень v' у діапазон $[-1, 1]$, який відповідає області значень гіперболічного тангенса $\tanh(\cdot)$, що використовується як активаційна функція на вихідному шарі генератора:

$$\tilde{v} = 2 \cdot \frac{v' - \min(v')}{\max(v') - \min(v')} - 1.$$

Такий підхід дозволяє уникнути насичення активації та зменшує ризик градієнтного згасання, що є критичним для навчання на даних CERT [2].

Категоріальні змінні, такі як роль користувача (наприклад, IT Admin, Sales) та підрозділ, були закодовані за допомогою представлення одиничним вектором і сформували вектор умов y . В архітектурах з умовами (Conditional GAN) цей вектор не генерується, а подається на вхід як додаткова умова для генератора та дискримінатора/критика, дозволяючи моделювати розподіл $P(x | y)$.

Дослідження проходило в кілька етапів, що відповідають трьом класичним підходам до генеративно-змагальних мереж:

1) AnoGAN (класична архітектура): модель навчалася на нормальних даних, а оцінювання аномалій виконувалася за помилкою реконструкції, отриманою шляхом пошуку латентного вектора z для кожного нового зразка (ітеративна оптимізація). Метод демонстрував коректну логіку, проте виявився надто повільним для практичного застосування — кожне оцінювання вимагало сотень ітерацій оптимізації, а підсумкові метрики виявилися близькими до випадкового вгадування [28].

2) BiGAN (двонаправлений підхід): введення енкодера E дозволило уникнути значних обчислювальних витрат на пошук латентного коду, але призвело до колапсу навчання через дисбаланс між потужністю генератора, енкодера та дискримінатора.

У більшості конфігурацій BiGAN показав нестабільну збіжність і низькі значення метрик [7].

3) WGAN / WGAN-GP: з метою підвищення стабільності було використано Wasserstein GAN із штрафом градієнта (WGAN-GP), що замінює дивергенцію Дженсена–Шеннона на відстань Вассерштейна і забезпечує змістовний градієнт навіть коли розподіли реальних та згенерованих даних не перетинаються [5, 6]. У реалізації для табличних даних застосовувалася умовна модифікація з подачею контексту y .

Для кожної архітектури (AnoGAN, BiGAN, WGAN-GP) було проведено повний цикл навчання (150 епох) на кожному з чотирьох варіантів структурування даних (A–D), що дозволило оцінити вплив способу агрегації логів на якість виявлення інсайдерських загроз.

У наведеній реалізації критично важливим є використання LayerNorm замість BatchNorm у критику, оскільки пакетна нормалізація порушує припущення про незалежність зразків при розрахунку штрафу градієнта. Параметр λ для члену штрафу був встановлений на рівні 10 [6].

Для вирішення проблеми відображення вхідного зразка у латентний простір (що необхідно для оцінювання аномальності в стилі AnoGAN) було також розглянуто підхід на базі BiGAN [30]. У цьому варіанті в систему додається енкодер E , який навчається відображати дані x у латентний код z ($E: (x, y) \mapsto z$), що теоретично дозволяє виконувати реконструкцію без додаткової оптимізації:

$$x \xrightarrow{E} z \xrightarrow{G} \hat{x} \approx x.$$

Однак у реалізованих експериментах BiGAN страждав від колапсу навчання та не забезпечив покращення метрик порівняно з іншими архітектурами (як показано в таблиці 4).

У загальному випадку оцінювання аномальності $A(x, y)$ для вхідного вектора розраховується не як проста сума помилок, а як стандартизована (Z-score) комбінація помилки реконструкції та невпевненості дискримінатора. Це необхідно, оскільки L_1 -помилка реконструкції та виходи дискримінатора мають різні фізичні масштаби. Формула такого оцінювання має вигляд:

$$A(x, y) = \alpha \cdot \frac{L_{\text{rec}}(x) - \mu_{\text{rec}}}{\sigma_{\text{rec}}} + (1 - \alpha) \cdot \frac{L_{\text{feat}}(x) - \mu_{\text{feat}}}{\sigma_{\text{feat}}},$$

де L_{rec} — L_1 -відстань між входом і реконструкцією, L_{feat} — різниця активацій прихованих шарів дискримінатора, а параметри μ та σ розраховуються на валідаційній вибірці нормальних даних. Ваговий коефіцієнт α (у наведеному прикладі 0.9) [28] надає пріоритет фізичній реконструкції. Наведена нижче реалізація демонструє узагальнену схему обчислення таких оцінок; в експериментах вона застосовувалася переважно як аналітичний інструмент для порівняння різних підходів.

Для оцінювання якості моделей використовувалися метрики AUC-ROC, Precision, Recall та F_1 -міра на тестовій вибірці, що містить суміш нормальних днів та інсайдерських інцидентів. В умовах сильного дисбалансу класів (співвідношення 1 : 1000) ключовим індикатором є здатність моделі ранжувати аномалії вище за норму (AUC-ROC) [31], однак для практичного застосування критично важливими є також Precision, Recall та AUPRC, які відображають поведінку моделі в зоні низьких порогів помилкових спрацьовувань.

Результати емпіричного тестування зведені у таблиці 4. Для кожної конфігурації моделі було проведено повний цикл навчання (150 епох) на кожному з чотирьох визначених способів представлення вхідних даних, що дозволяє оцінити їх ефективність у різних умовах структурування логів.

Таблиця 4 – Порівняння моделей за якістю виявлення інсайдерських загроз

Модель	Структура даних	Precision	Recall	AUC-ROC	AUPRC	F1-Score
Anogan	A	0.0435	0.0908	0.5056	0.0072	0.0589
Anogan	B	0.0754	0.0443	0.5075	0.0603	0.0558
Anogan	C	0.0493	0.0940	0.5049	0.0400	0.0647
Anogan	D	0.0479	0.0898	0.4003	0.0117	0.0625
BiGAN	A	0.0547	0.0258	0.5073	0.0296	0.0351
BiGAN	B	0.0033	0.0985	0.4028	0.0919	0.0065
BiGAN	C	0.0964	0.0396	0.5013	0.0863	0.0561
BiGAN	D	0.0230	0.0108	0.4033	0.0338	0.0147
WGAN	A	0.0232	0.0951	0.6096	0.0002	0.0373
WGAN	B	0.0126	0.0759	0.6084	0.0872	0.0217
WGAN	C	0.0572	0.0142	0.8037	0.0311	0.0228
WGAN	D	0.0261	0.0398	0.9413	0.0270	0.0315

Як видно з таблиці 4, класичні архітектури GAN (AnoGAN, BiGAN) характеризуються нестабільністю навчання та результатами, близькими до випадкового вгадування ($\text{AUROC} \approx 0.5$ при низьких Precision та Recall) практично для всіх способів структурування даних (A–D). Модель на базі WGAN/WGAN-GP,

що працювала з даними, агрегованими по сесіях (структура D), показала порівняно високий показник AUROC 0.9413, однак при цьому значення Precision, Recall, AUPRC та F_1 -міри залишаються низькими. Це означає, що, незважаючи на здатність моделі ранжувати частину аномалій вище за норму, вона не забезпечує достатньо надійного відокремлення нормальної поведінки користувача від дій потенційного зловмисника в режимі двокласової класифікації.

Проведений аналіз та серія експериментів із різними методами структурування даних призвели до однозначно негативного висновку щодо використання генеративно-змагальних мереж як самодостатнього інструменту для виявлення інсайдерських загроз. Навіть найкраща конфігурація (WGAN зі структурою D) скоріше свідчить про потенційну можливість аугментації та доповнення даних, ніж про здатність надійно відрізнити інсайдерську активність від нормальної.

Отже, для досягнення практично значущих результатів подальші дослідження мають бути зосереджені не на GAN як окремому рішенні, а на гібридних підходах. Ефективність може бути досягнута шляхом поєднання GAN з більш потужними методами проектування ознак або шляхом їх інтеграції в ансамблі з іншими класами моделей, здатними краще обробляти складну, високо-розріджену й гетерогенну природу поведінкових логів користувачів.

3.2. Збагачення даних за допомогою GAN

Однією з серйозних перешкод для побудови ефективних систем виявлення інсайдерських загроз (ITD) на базі наборів даних сімейства CERT (зокрема версії r4.2) є екстремальний дисбаланс класів. Співвідношення між кількістю нормальних подій та зафіксованими інцидентами часто перевищує показник 1000:1, що створює ситуацію, за якої стандартні алгоритми навчання (Random Forest, SVM, нейронні мережі) схильні ігнорувати міноритарний клас, оптимізуючи загальну точність (Accuracy) на користь домінуючого класу нормальної поведінки.

Традиційні підходи до вирішення цієї проблеми, такі як метод синтетичної передискретизації меншості (SMOTE), базуються на лінійній інтерполяції між сусідніми векторами у просторі ознак [16]:

$$x_{\text{new}} = x_i + \lambda(x_{\text{nn}} - x_i), \quad \lambda \sim U(0, 1).$$

Однак, застосування SMOTE до логів поведінкової активності має суттєві обмеження:

1) порушення семантичної цілісності. Вектори ознак у датасеті CERT є гетерогенними, тобто містять як неперервні змінні (час, обсяг трафіку), так і дискретні (роль користувача, ID департаменту). Лінійна інтерполяція між вектором IT-адміністратора та вектором бухгалтера призводить до створення синтетичного гібрида, що не відповідає жодній реальній ролі в організації;

2) ігнорування мультимодальності. Розподіли поведінкових метрик часто мають кілька локальних мод (наприклад, піки активності вранці та ввечері). SMOTE, заповнюючи простір між точками, може генерувати зразки в областях низької ймовірності (між модами), створюючи шум замість корисного сигналу;

3) втрата структурних залежностей. Логи подій мають внутрішню логіку (наприклад, обсяг переданих даних не може бути від'ємним, а кількість успішних логінів не може перевищувати кількість спроб), яку прості геометричні методи збагачення даних часто порушують [32].

Для подолання зазначених обмежень у даній роботі використано генеративний підхід на основі архітектури CTGAN, яка спеціально розроблена для моделювання складних табличних розподілів зі змішаними типами даних [33].

На відміну від методів, що використовують глобальну нормалізацію, CTGAN застосовує механізм специфічної для моди нормалізації для всіх неперервних змінних. Розподіл кожної числової колонки C_j розглядається як суміш нормальних компонент і моделюється за допомогою варіаційної моделі суміші нормальних розподілів Гауса (VGM):

$$P(c_j) = \sum_{k=1}^K w_k \mathcal{N}(c_j; \mu_k, \sigma_k),$$

де K — кількість мод (компонент), а w_k , μ_k , σ_k — вага, середнє та дисперсія k -ї моди відповідно. У процесі попередньої обробки кожне значення v трансформується у вектор, що складається з:

- одиничного вектора α , який вказує на приналежність значення до конкретної моди (компоненти GMM);

- скаляра β , що представляє нормалізоване значення в межах цієї моди.

Такий підхід дозволяє нейромережі ефективно вивчати не лише діапазон значень, а й складну форму розподілу (наприклад, наявність важких хвостів у розподілі

файлових операцій).

Генератор побудовано як глибоку нейронну мережу із залишковими зв'язками. На вхід подається конкатенація шумового вектора $z \sim \mathcal{N}(0, I)$ та умовного вектора c . Умовний вектор формується шляхом об'єднання кодованого одиничним вектором класу (наприклад, інсайдер/норма) та маски, що вказує на конкретну категорію в одній з дискретних колонок. Це дозволяє моделювати умовний розподіл $p(x | y, c)$ і цілеспрямовано генерувати приклади рідкісних класів, вирішуючи проблему дисбалансу [33].

Використання залишкових блоків, що містять пов'язані шари, нормалізація пакетів та ReLU-активації, дозволяє уникнути проблеми згасання градієнтів при навчанні глибокої мережі та краще відтворювати кореляції між віддаленими ознаками [34].

Дискримінатор (критик) реалізовано на основі багат шарового перцептрона. Для забезпечення стабільності навчання використано функцію втрат WGAN-GP. Замість класичної бінарної ентропії, яка схильна до насичення градієнтів, WGAN мінімізує відстань Вассерштейна між розподілом реальних даних P_r та згенерованих P_g [5]:

$$L_{\text{WGAN}} = \mathbb{E}_{\tilde{x} \sim P_g} [D(\tilde{x})] - \mathbb{E}_{x \sim P_r} [D(x)].$$

Для виконання умови Ліпшица з константою 1, необхідної для коректності WGAN, застосовано штраф за градієнт [6]:

$$\text{GP} = \lambda \cdot \mathbb{E}_{\hat{x} \sim P_{\hat{x}}} \left[\left(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1 \right)^2 \right],$$

де $\hat{x}$ — точки, отримані випадковою інтерполяцією між реальними та фейковими зразками. Коефіцієнт λ встановлено на рівні 10. Це запобігає колапсу мод і забезпечує стабільну збіжність.

Для оцінювання ефективності запропонованого підходу було проведено серію експериментів з використанням датасету CERT r4.2. Навчання генеративної моделі (CTGAN) проводилося протягом 500 епох з розміром пакету 64. Оцінювання якості синтетичних даних здійснювалося за двома напрямками: статистична достовірність та практична корисність для класифікації.

Основною метою генерації додаткових даних є покращення якості виявлення інсайдерів. Після успішного створення синтетичних даних та формування розширеного навчального набору, наступним кроком стало емпіричне

підтвердження гіпотези про те, що застосування GAN-аугментації призводить до статистично значущого підвищення ефективності детекції інсайдерських загроз. Для цього було проведено серію порівняльних експериментів із використанням сучасних методів ансамблевої класифікації, які зарекомендували себе найкраще на гетерогенних табличних даних.

Реалізація експериментів виконувалася мовою Python із використанням бібліотек pandas для попередньої обробки даних, scikit-learn для навчання моделей Random Forest та Isolation Forest, а також imbalanced-learn для застосування методу SMOTE. Весь код було організовано у вигляді відтворюваного конвеєра: етапи завантаження даних, попередньої обробки, балансування, навчання моделей та збереження налаштованих класифікаторів були винесені в окремі функції та класи.

На основі датасету CERT r4.2 було сформовано три варіанти тренувальних наборів, які відрізнялися стратегією роботи з дисбалансом класів, тоді як тестова вибірка залишалася спільною для всіх експериментів. На етапі попередньої обробки даних послідовно виконували фільтрацію технічних та дубльованих записів, агрегування подій до рівня користувача за фіксований часовий інтервал, нормалізацію числових ознак і кодування категоріальних атрибутів у формат, придатний для подальшого навчання моделей. Завершальним кроком було формування цільової змінної, що позначає наявність або відсутність інсайдерської активності.

Після виконання цих кроків було сформовано три тренувальні набори. Перший, базовий набір, зберіг природний дисбаланс класів зі співвідношенням інсайдерів до нормальних користувачів приблизно 1:1000 і використовувався як відправна точка для оцінювання впливу методів балансування. Другий, SMOTE-набір, передбачав балансування міноритарного класу «інсайдер» до співвідношення 1:1 за допомогою алгоритму SMOTE, який генерує синтетичні зразки у просторі ознак шляхом інтерполяції між найближчими сусідами цього класу. Третій, GAN-набір, також забезпечував баланс 1:1, але синтетичні приклади атак у ньому створювалися умовною табличною GAN (CTGAN/WGAN-GP); при цьому кількість згенерованих зразків для класу «інсайдер» узгоджувалася з відповідною кількістю у SMOTE-наборі, що забезпечувало коректність подальшого порівняння результатів.

У всіх випадках 20% вихідних даних відкладалися в окремий тестовий набір, який містив лише реальні історичні події та зберігав природний дисбаланс класів. Синтетичні приклади (SMOTE або GAN) використовувалися виключно на етапі

навчання моделей, тобто не потрапляли до відкладеної вибірки.

Для реалізації Random Forest використовувався клас RandomForestClassifier бібліотеки scikit-learn. Алгоритм базується на принципі бегінгу [35] та поєднує в собі побудову набору слабкозалежних дерев рішень на випадкових підвибірках навчальних даних і ознак. Для підвищення відтворюваності у всіх запусках фіксувався параметр `_state`, а також використовувався паралельний режим обчислень (`n_jobs = -1`).

Пошук оптимальних параметрів виконувався методом решітчастого пошуку з 5-кратною перехресною валідацією на тренувальній вибірці кожного з трьох варіантів даних (базовий, SMOTE, GAN). Для Random Forest розглядалася така решітка параметрів: кількість дерев у лісі $n_{est} \in \{100, 200, 300, 500\}$; максимальна глибина дерева `max_depth` $\in \{\text{None}, 10, 20, 30\}$; мінімальна кількість об'єктів у листі `min_samples_leaf` $\in \{1, 2, 4\}$; мінімальна кількість об'єктів для розбиття вузла `min_samples_split` $\in \{2, 4, 8\}$; стратегія вибору підмножини ознак при розбитті `max_features` $\in \{\text{«sqrt»}, \text{«log2»}\}$ [13].

Як цільову функцію решітчастого пошуку було обрано F_1 -міру для класу «інсайдер», що дозволяє збалансовано враховувати як точність (Precision), так і повноту (Recall) в умовах початкового сильного дисбалансу. Для підвищення надійності оцінювання, конвеєр навчання Random Forest повторювався кілька разів з різними розбиттями даних, після чого обиралася конфігурація, що демонструвала найстабільніші показники на валідаційних вибірках.

Алгоритм Isolation Forest реалізовано за допомогою класу IsolationForest бібліотеки scikit-learn. Це метод навчання без учителя, який формує ансамбль випадкових дерев для ізоляції аномалій [36]. На відміну від Random Forest, тут модель безпосередньо оцінює «аномальність» об'єкта за середньою довжиною шляху до ізоляції у дереві.

Для Isolation Forest застосовувався окремий решітчастий пошук гіперпараметрів, також із 5-кратною перехресною валідацією [36]. Налаштовувалася така решітка параметрів: кількість дерев у ансамблі $n_{est} \in \{100, 200, 300, 500\}$; частка вибірки, що використовується для побудови одного дерева, `max_samples` $\in \{0.5, 0.7, 1.0\}$; частка ознак, що випадково відбирається для кожного дерева, `max_features` $\in \{0.5, 0.7, 1.0\}$; очікувана частка аномалій у даних `contamination` $\in \{0.001, 0.005, 0.01\}$.

Оскільки Isolation Forest працює в режимі без учителя, для перетворення оцінок «аномальності» на бінарні передбачення (інсайдер/норма) під час

крос-валідації підбиралися порогові значення, які максимізують F_1 -міру на валідаційних вибірках. Надалі ці пороги фіксувалися для відповідних конфігурацій моделей.

Для обох алгоритмів (Random Forest та Isolation Forest) процес налаштування параметрів повністю виконувався на тренувальних вибірках (базовій, SMOTE та GAN). Відкладена тестова вибірка використовувалася лише на фінальному етапі для отримання підсумкових значень метрик якості й не брала участі ні в побудові, ні в налаштуванні моделей.

3.3. Тестування інших моделей на збагачених за допомогою GAN даних

Після успішної генерації синтетичних даних та формування розширеного навчального набору наступним кроком стало емпіричне підтвердження ефективності запропонованого підходу. Метою цього етапу було з'ясувати, чи призводить використання даних, згенерованих GAN, до статистично значущого покращення метрик виявлення інсайдерських загроз порівняно з навчанням на вихідних незбалансованих даних та класичними методами балансування. Для цього було проведено серію експериментів з використанням сучасних ансамблевих алгоритмів, які є поширеним стандартом у задачах кібербезпеки завдяки своїй стійкості та відносній інтерпретованості.

Практична цінність експерименту полягає у валідації запропонованих методів аугментації даних на архітектурах різної природи. Алгоритм Random Forest використано як основний інструмент класифікації завдяки його здатності ефективно обробляти багатовимірні дані та нівелювати вплив окремих викидів, що є критичним при роботі з синтетично згенерованими об'єктами (GAN/SMOTE) [37]. Водночас, застосування Isolation Forest дозволяє вирішити проблему виявлення прихованих загроз у нерозмічених потоках даних [38]. Оскільки цей метод ізолює аномалії на основі структури простору ознак, а не частотних характеристик, він слугує незалежним індикатором того, наскільки якісно синтетичні дані відтворюють структуру реальних інсайдерських атак, та дозволяє мінімізувати кількість хибно-позитивних спрацювань у реальних умовах моніторингу.

Фінальна параметризація моделей продемонструвала відмінності у вимогах алгоритмів до структури даних. Для Random Forest оптимальною стала конфігурація з високою ємністю ($n_{\text{est}} = 500$, $\text{max_depth} = \text{None}$) та низьким порогом

розбиття вузлів, що дозволило максимізувати повноту виявлення рідкісних подій [39, 40]. Для Isolation Forest налаштування підтвердило гіпотезу про швидку збіжність алгоритму: збільшення кількості дерев понад 100 не давало статистично значущого приросту якості. Визначальним фактором стало налаштування параметра *contamination*, який виступав динамічним порогом прийняття рішень: для базової вибірки він фіксувався на рівні 0.001, а для аугментованих наборів масштабувався до 0.5. Такий підхід дозволив трансформувати виходи моделі (*anomaly scores*) у бінарні передбачення, максимізуючи F_1 -міру специфічно для кожного типу навчальної вибірки.

Ключовою умовою експерименту було використання єдиної відкладеної тестової вибірки, спільної для всіх моделей. Вона становила 20% від початкового набору, містила лише реальні події та зберігала вихідний дисбаланс класів. Навчання виконувалося на решті 80% даних, до яких застосовувалися різні методи аугментації. Такий дизайн забезпечував, що отримані метрики якості відображають здатність моделей узагальнювати знання на нові реальні спостереження, а не на синтетичні приклади.

За результатами виконання 5-кратної перехресної валідації було визначено фінальні конфігурації моделей. Для Random Forest найвищу стійкість F_1 -міри продемонстрував ансамбль із максимальною кількістю дерев ($n_{est} = 500$) та необмеженою глибиною (*max_depth* = None). При цьому алгоритм надав перевагу детальному розбиттю вузлів (обрано мінімальне значення з решітки *min_samples_split* = 4), що свідчить про необхідність побудови складної розділяючої поверхні для ефективною класифікації інсайдерів. Для Isolation Forest експеримент показав швидку збіжність алгоритму: збільшення кількості дерев понад 100 не надавало суттєвого приросту якості ізоляції. Критичним етапом стало калібрування параметра *contamination*, який визначає поріг чутливості детектора. Для базового незбалансованого набору даних оптимальним виявилось консервативне значення $\nu = 0.001$, знайдене під час решітчастого пошуку. Водночас для збалансованих наборів (SMOTE/GAN) цей параметр було адаптовано відповідно до штучно створеної пропорції класів ($\nu \approx 0.5$). Такий підхід дозволив коректно трансформувати оцінювання аномальності у бінарні рішення, максимізуючи точність виявлення для кожного типу навчальної вибірки.

Результати тестування на відкладеній вибірці підтвердили критичну важливість балансування класів для задачі виявлення інсайдерів. Як видно з таблиці 5, використання Random Forest на вихідних (незбалансованих) даних

призводить до незадовільних результатів: модель демонструє низьку повноту ($\text{Recall} = 0.344$), пропускаючи майже дві третини атак. Це пояснюється тим, що алгоритм, оптимізуючи ентропію, схильний ігнорувати рідкісний клас на користь домінуючого нормального трафіку.

Isolation Forest у базовому варіанті виявився більш чутливим до аномалій $\text{Recall} = 0.624$, проте продемонстрував низьку точність $\text{Precision} = 0.566$, генеруючи значну кількість хибних спрацьовувань, що робить його малопридатним для автономної роботи без попереднього налаштування.

Найкращий приріст якості забезпечило застосування методу SMOTE. Для Random Forest це дозволило досягти найвищих показників у експерименті: F_1 -міра зросла з 0.443 до 0.877, при цьому вдалося збалансувати точність (0.895) та повноту (0.860). Це свідчить про те, що лінійна інтерполяція між векторами атак ефективно заповнила простір ознак, сформувавши щільні кластери класу «інсайдер», які Random Forest зміг легко відокремити від норми. Аналогічний ефект спостерігався і для Isolation Forest: на насичених SMOTE-даних модель краще сформувала границі нормальності, підвищивши F_1 -міру до 0.803.

Використання генеративно-змагальної мережі (GAN) продемонструвало змішані результати. Хоча навчання на синтетичних даних GAN покращило якість моделей порівняно з базовим варіантом (зокрема, F_1 -міра Random Forest зросла до 0.671), цей метод суттєво поступився SMOTE. Для Random Forest повнота склала лише 0.644 (проти 0.860 у SMOTE), а для Isolation Forest приріст повноти до 0.733 супроводжувався низькою точністю (0.596). Таке відставання GAN можна пояснити складністю навчання генератора на табличних даних CERT: ймовірно, згенеровані зразки, хоч і були валідними, не забезпечили достатнього покриття різноманітних сценаріїв атак (проблема колапсу мод), або ж внесли додатковий шум у граничні області, що ускладнило побудову чітких вирішальних правил.

Отже, експеримент показав, що для даного типу даних класичний метод SMOTE є найбільш ефективним інструментом боротьби з дисбалансом, перевершуючи складніші підходи на основі GAN.

Аналіз результатів, наведених у таблиці 5, дозволяє чітко ранжувати методи балансування за їхньою ефективністю. Безумовним лідером за комплексною F_1 -мірою виявився підхід SMOTE. Для алгоритму Random Forest застосування SMOTE дозволило підняти F_1 з базового рівня 0.443 до 0.877, забезпечивши майже симетричні показники точності $\text{Precision} = 0.895$ та повноти $\text{Recall} = 0.860$. Це свідчить про те, що лінійна інтерполяція успішно заповнила пропуски в просторі

Таблиця 5 – Порівняльні результати моделей на тестовій вибірці CERT r4.2

Модель	Аугментація даних	Precision	Recall	F1-Score
Random Forest	Відсутня	0.624	0.344	0.443
Random Forest	SMOTE	0.895	0.860	0.877
Random Forest	GAN	0.701	0.644	0.671
Isolation Forest	Відсутня	0.566	0.624	0.593
Isolation Forest	SMOTE	0.788	0.819	0.803
Isolation Forest	GAN	0.596	0.733	0.657

ознак міноритарного класу, сформувавши щільні та неперервні кластери, які модель легко відокремила від нормального трафіку.

Використання GAN, хоч і покращило результати порівняно з навчанням на незбалансованих даних, не виправдало очікувань щодо переваги над простішими методами. Для Random Forest конфігурація з GAN досягла F_1 -міри лише 0.671. Хоча це суттєво краще за базовий варіант, метод значно поступився SMOTE як за точністю (0.701 проти 0.895), так і за повнотою (0.644 проти 0.860). Аналогічна тенденція спостерігається і для Isolation Forest: SMOTE забезпечив $F_1 = 0.803$, тоді як GAN — лише 0.657. Головною проблемою GAN стала нездатність досягти високої точності Precision ≈ 0.60 для IF), що вказує на генерацію значної кількості «шумових» прикладів, які розмили границі між класами.

Причини такої поведінки можна пояснити через аналіз геометрії розподілу даних, зокрема за допомогою проекцій t-SNE. У випадку зі SMOTE синтетичні точки заповнюють простір між найближчими сусідами, створюючи компактну область підвищеного ризику. Це спрощує завдання класифікатору: границі рішень стають гладкими та зрозумілими. Натомість для GAN спостерігається ефект фрагментації: синтетичні зразки часто формують окремі локальні «острівці» або ж концентруються навколо моди розподілу (проблема колапсу мод, залишаючи периферійні сценарії атак без покриття. Це призводить до того, що модель «вивчає» лише певні типи загроз, ігноруючи варіативність реальних атак, що й відображається у нижчому показнику Recall (0.644) порівняно зі SMOTE. Аналогічні спостереження щодо геометрії синтетичних даних бачимо також у [41].

Отримані результати свідчать, що в умовах табличних даних певної структури (як у наборі CERT) складність генеративної моделі не гарантує вищої якості навчання. Класичний SMOTE забезпечив кращий компроміс між чутливістю та кількістю хибних спрацювань, створивши більш репрезентативну навчальну вибірку. GAN-аугментація, попри свою технологічну перспективність, у цьому

експерименті виступила лише як проміжний варіант: вона здатна покращити виявлення аномалій відносно неаугментованих даних, але програє конкуренцію методам прямої інтерполяції через складність налаштування генератора та ризик внесення шуму в граничні області простору ознак.

ВИСНОВКИ

Проведено аналіз методів та моделей використання генеративно-змагальних мереж (GAN) для виявлення інсайдерських загроз в інформаційних системах. Встановлено, що традиційні сигнатурні методи захисту втрачають ефективність, оскільки інсайдери діють у межах наданих повноважень, а їхні дії важко формалізувати простими правилами. Перехід до поведінкового аналізу (UEBA) ускладнюється критичним дисбалансом даних у журналах аудиту, де співвідношення інцидентів до норми часто перевищує 1:1000, що призводить до ігнорування загроз класичними алгоритмами.

За результатами виконання поставлених у роботі завдань отримано такі результати:

- обґрунтовано вибір архітектур WGAN-GP та CTGAN як найбільш придатних для моделювання табличних поведінкових даних та генерації змішаних типів ознак;
- розроблено методику попередньої обробки журналів подій CERT r4.2, яка включає агрегацію на рівні «користувач-день», логарифмічну трансформацію та масштабування ознак;
- експериментально досліджено ефективність GAN у режимі самостійного детектора аномалій (навчання без учителя);
- підтверджено ефективність використання GAN для синтетичного збагачення (аугментації) даних при навчанні класифікаторів;
- проведено порівняльний аналіз результатів GAN-аугментації з класичним методом балансування SMOTE.

Для моделювання табличних поведінкових даних найбільш стабільною визначено архітектуру WGAN-GP. Використання відстані Вассерштейна та штрафу градієнта дозволяє уникнути проблеми зникаючих градієнтів, характерної для класичних GAN при роботі з дискретними даними.

Розроблено методику підготовки даних на базі набору CERT r4.2. Ключовим рішенням стало обрання агрегації на рівні «користувач-день», що дозволило сформувати інформативні вектори ознак (розмірністю 146 компонентів), які поєднують динамічні показники активності та статичний контекст. Застосування логарифмічної трансформації та масштабування ознак у діапазон $[-1, 1]$ дозволило нівелювати асиметрію розподілів і забезпечити стабільну збіжність моделей.

Експериментально доведено низьку ефективність використання GAN у

режимі навчання без учителя (як детекторів аномалій). Моделі AnoGAN та BiGAN показали результати, близькі до випадкового вгадування. Найкращий результат продемонструвала модель WGAN-GP ($AUC-ROC \approx 0.94$), проте низькі значення F1-міри та Precision свідчать про непридатність такої моделі як єдиного засобу захисту через значну кількість помилкових спрацювань.

Підтверджено, що використання GAN для генерації синтетичних атак дозволяє покращити якість роботи класифікаторів. Навчання Random Forest на даних, збагачених CTGAN, підвищило F1-міру з 0.443 (на вихідних даних) до 0.671; а Isolation Forest - з 0.624 до 0.733. Це доводить, що генеративні моделі допомагають класифікаторам краще розпізнавати загрози, які ігноруються при навчанні на незбалансованих даних.

Встановлено, що на табличних даних досліджуваного типу класичний метод SMOTE забезпечує вищі показники ефективності ($F1-Score \approx 0.877$), ніж GAN-аугментація. Це пояснюється тим, що геометрична інтерполяція SMOTE створює більш щільні кластери мінорного класу, тоді як GAN стикається з проблемою часткового колапсу мод, покриваючи не всі варіації атак.

Розроблений підхід можливо використовувати як допоміжний інструмент у системах SIEM/UEBA для генерації сценаріїв атак при налаштуванні детекторів, особливо в умовах дефіциту реальних інцидентів. Перспективи подальших досліджень полягають у переході до послідовних архітектур для кращого врахування часових залежностей, які не вловлюються при табличній агрегації.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ian J. Goodfellow та ін. “Generative Adversarial Nets”. АНГЛ. В: Advances in Neural Information Processing Systems 27. Neural Information Processing Systems Foundation, 2014.
2. Ian Goodfellow. “NIPS 2016 Tutorial: Generative Adversarial Networks”. АНГЛ. В: CoRR abs/1701.00160 (2017). arXiv: 1701.00160. URL: <http://arxiv.org/abs/1701.00160>.
3. Antonia Creswell та ін. “Generative Adversarial Networks: An Overview”. АНГЛ. В: IEEE Signal Processing Magazine 35.1 (2018), с. 53—65. DOI: <https://doi.org/10.1109/MSP.2017.2765202>. URL: <https://doi.org/10.1109/MSP.2017.2765202>.
4. Google Developers. Loss Functions. АНГЛ. Last updated 25 Aug 2025. 2025. URL: <https://developers.google.com/machine-learning/gan/loss>.
5. Martin Arjovsky, Soumith Chintala та Léon Bottou. “Wasserstein Generative Adversarial Networks”. АНГЛ. В: Proceedings of the 34th International Conference on Machine Learning. Т. 70. Proceedings of Machine Learning Research. PMLR, 2017, с. 214—223. URL: <https://proceedings.mlr.press/v70/arjovsky17a.html>.
6. Ishaan Gulrajani та ін. “Improved Training of Wasserstein GANs”. АНГЛ. В: Advances in Neural Information Processing Systems 30. 2017.
7. Jeff Donahue, Philipp Krähenbühl та Trevor Darrell. “Adversarial Feature Learning”. АНГЛ. В: Proceedings of the International Conference on Learning Representations (ICLR). 2017. URL: <https://arxiv.org/abs/1605.09782>.
8. Aeryn Dunmore та ін. “A Comprehensive Survey of Generative Adversarial Networks (GANs) in Cybersecurity Intrusion Detection”. АНГЛ. В: IEEE Access

- 11 (2023). DOI: <https://doi.org/10.1109/ACCESS.2023.3296707>. URL: <https://ieeexplore.ieee.org/document/10187144>.
9. Mack Preston. “Insider Threat Detection Data Augmentation Using WCGAN-GP”. АНГЛ. Master’s Thesis. Halifax, Nova Scotia, Canada: Dalhousie University, 2022. URL: <http://hdl.handle.net/10222/81531>.
10. R. G. Gayathri, Atul Sajjanhar та Yong Xiang. “Hybrid Deep Learning Model using SPCAGAN Augmentation for Insider Threat Analysis”. АНГЛ. B: Expert Systems with Applications 249 (2024). DOI: <https://doi.org/10.1016/j.eswa.2024.123533>. URL: <https://doi.org/10.1016/j.eswa.2024.123533>.
11. Ivan Homoliak та ін. “Insight Into Insiders and IT: A Survey of Insider Threat Taxonomies, Analysis, Modeling, and Countermeasures”. АНГЛ. B: ACM Computing Surveys 52.2 (2019). DOI: <https://doi.org/10.1145/3303771>. URL: <https://dl.acm.org/doi/10.1145/3303771>.
12. Ponemon Institute LLC. 2018 Cost of Insider Threats: Global. АНГЛ. Research Report. Traverse City, MI: Ponemon Institute LLC, 2018.
13. Bushra Bin Sarhan та Najwa Altwaijry. “Insider Threat Detection Using Machine Learning Approach”. АНГЛ. B: Applied Sciences 13.1 (2023), с. 259. DOI: <https://doi.org/10.3390/app13010259>. URL: <https://doi.org/10.3390/app13010259>.
14. Serhii Zybin та ін. “Approach of the Attack Analysis to Reduce Omissions in the Risk Management”. АНГЛ. B: Cybersecurity Providing in Information and Telecommunication Systems (CPITS 2021). T. 2923. CEUR Workshop Proceedings. CEUR-WS.org, 2021, с. 318—328. URL: <http://ceur-ws.org/Vol-2923/paper35.pdf>.

15. Ishara Barhoson Galadima, Norafida Bte Ithnin, Nur Haliza Abdulwahab та ін. “Deep Learning Models and Social Engineering Dynamics in Insider Threat Detection: A Systematic Review”. АНГЛ. В: Research Square (2025). Preprint. DOI: <https://doi.org/10.21203/rs.3.rs-7396895/v1>. URL: <https://www.researchsquare.com/article/rs-7396895/v1>.
16. Nitesh V. Chawla та ін. “SMOTE: Synthetic Minority Over-sampling Technique”. АНГЛ. В: Journal of Artificial Intelligence Research 16 (2002), с. 321—357. DOI: <https://doi.org/10.1613/jair.953>. URL: <https://www.jair.org/index.php/jair/article/view/10302>.
17. Mayesh Mohapatra, Anshumaan Phukan та Vijay K. Madiseti. “Generative Adversarial Network Based Approach towards Synthetically Generating Insider Threat Scenarios”. АНГЛ. В: Journal of Software Engineering and Applications 16.11 (2023), с. 586—604. DOI: <https://doi.org/10.4236/jsea.2023.1611030>. URL: <https://www.scirp.org/journal/paperinformation?paperid=129379>.
18. Robin Sommer та Vern Paxson. “Outside the Closed World: On Using Machine Learning for Network Intrusion Detection”. АНГЛ. В: Proceedings of the 2010 IEEE Symposium on Security and Privacy (SP). IEEE, 2010, с. 305—316. DOI: <https://doi.org/10.1109/SP.2010.25>. URL: <https://doi.org/10.1109/SP.2010.25>.
19. Duc C. Le та Nur Zincir-Heywood. “Exploring Anomalous Behaviour Detection and Classification for Insider Threat Identification”. АНГЛ. В: International Journal of Network Management 30.6 (2020). DOI: <https://doi.org/10.1002/nem.2109>. URL: <https://onlinelibrary.wiley.com/doi/10.1002/nem.2109>.

20. Shuhan Yuan та Xintao Wu. “Deep learning for insider threat detection: Review, challenges and opportunities”. АНГЛ. В: Computers & Security 104 (2021). DOI: <https://doi.org/10.1016/j.cose.2021.102221>. URL: <https://doi.org/10.1016/j.cose.2021.102221>.
21. Diana Haidar та Mohamed Medhat Gaber. “Data Stream Clustering for Real-Time Anomaly Detection: An Application to Insider Threats”. АНГЛ. В: Clustering Methods for Big Data Analytics. Unsupervised and Semi-Supervised Learning. Cham: Springer, 2019, с. 115—144. DOI: https://doi.org/10.1007/978-3-319-97864-2_6. URL: https://doi.org/10.1007/978-3-319-97864-2_6.
22. Minhae Jang та ін. “Against Insider Threats with Hybrid Anomaly Detection with Local-Feature Autoencoder and Global Statistics (LAGS)”. АНГЛ. В: IEICE Transactions on Information and Systems E103-D.4 (2020), с. 888—891. DOI: <https://doi.org/10.1587/transinf.2019EDL8180>. URL: <https://doi.org/10.1587/transinf.2019EDL8180>.
23. A. Ludwika, A. S. N. Chakravarthy та C. Priyadarshini. “A Transformer-Based Deep Learning Framework for Insider Threat Detection”. АНГЛ. В: International Journal of Innovative Research in Technology 12.3 (2025), с. 72—78.
24. Nidhi Rastogi та Qicheng Ma. “DANTE: Predicting Insider Threat using LSTM on System Logs”. АНГЛ. В: CoRR abs/2102.05600 (2021). URL: <https://arxiv.org/abs/2102.05600>.
25. Xiangrui Cai та ін. “LAN: Learning Adaptive Neighbors for Real-Time Insider Threat Detection”. АНГЛ. В: IEEE Transactions on Information Forensics and Security 19 (2024). DOI: <https://doi.org/10.1109/TIFS.2024.3488527>.

26. Xiaoyun Ye та ін. “Daily Insider Threat Detection with Hybrid TCN Transformer Architecture”. АНГЛ. В: Scientific Reports 15 (2025). DOI: <https://doi.org/10.1038/s41598-025-12063-x>.
27. Shuang Song та ін. “BRITD: Behavior Rhythm Insider Threat Detection with Time Awareness and User Adaptation”. АНГЛ. В: Cybersecurity 7.2 (2024). DOI: <https://doi.org/10.1186/s42400-023-00190-9>. URL: <https://doi.org/10.1186/s42400-023-00190-9>.
28. Thomas Schlegl та ін. “Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery”. АНГЛ. В: Information Processing in Medical Imaging (IPMI). Springer, 2017, с. 146—157. DOI: https://doi.org/10.1007/978-3-319-59050-9_12. URL: <https://arxiv.org/abs/1703.05921>.
29. Eric Jang, Shixiang Gu та Ben Poole. “Categorical Reparameterization with Gumbel-Softmax”. АНГЛ. В: Proceedings of the International Conference on Learning Representations (ICLR). 2017. URL: <https://arxiv.org/abs/1611.01144>.
30. Jeff Donahue, Philipp Krähenbühl та Trevor Darrell. “Adversarial Feature Learning”. АНГЛ. В: Proceedings of the International Conference on Learning Representations (ICLR). Bidirectional Generative Adversarial Networks (BiGAN). 2017. URL: <https://arxiv.org/abs/1605.09782>.
31. Takaya Saito та Marc Rehmsmeier. “The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets”. АНГЛ. В: PLOS ONE 10.3 (2015). DOI: <https://doi.org/10.1371/journal.pone.0118432>. URL: <https://doi.org/10.1371/journal.pone.0118432>.

32. Alberto Fernández та ін. “SMOTE for Learning from Imbalanced Data: Progress and Challenges, Marking the 15-year Anniversary”. АНГЛ. В: Journal of Artificial Intelligence Research 61 (2018), с. 863—905. DOI: <https://doi.org/10.1613/jair.1.11192>. URL: <https://www.jair.org/index.php/jair/article/view/11192>.
33. Lei Xu та ін. “Modeling Tabular Data using Conditional GAN”. АНГЛ. В: Advances in Neural Information Processing Systems (NeurIPS). Т. 32. 2019. URL: <https://arxiv.org/abs/1907.00503>.
34. Kaiming He та ін. “Deep Residual Learning for Image Recognition”. АНГЛ. В: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016, с. 770—778. DOI: <https://doi.org/10.1109/CVPR.2016.90>. URL: <https://doi.org/10.1109/CVPR.2016.90>.
35. Г. І. Гайдур, С. О. Гахов та Д. Є. Гамза. “Модель виявлення шкідливої активності в інформаційній системі організації на основі гібридної класифікації”. В: Сучасний захист інформації 4(60) (2024), с. 30—38. DOI: <https://doi.org/10.31673/2409-7292.2024.040003>. URL: <https://journals.duikt.edu.ua/index.php/dataprotect/article/download/3056/2945>.
36. Fei Tony Liu, Kai Ming Ting та Zhi-Hua Zhou. “Isolation Forest”. АНГЛ. В: Proceedings of the 2008 Eighth IEEE International Conference on Data Mining (ICDM). Pisa: IEEE, 2008, с. 413—422. DOI: <https://doi.org/10.1109/ICDM.2008.17>.
37. Phavithra Manoharan та ін. “Insider threat detection using supervised machine learning algorithms”. АНГЛ. В: Telecommunication Systems 87 (2024). DOI: <https://doi.org/10.1016/j.tcom.2024.101000>.

//doi.org/10.1007/s11235-023-01085-3. URL: <https://doi.org/10.1007/s11235-023-01085-3>.

38. Taher Al Shehari та ін. “Insider Threat Detection Model Using Anomaly-Based Isolation Forest Algorithm”. АНГЛ. В: IEEE Access 11 (2023), с. 1—3. DOI: <https://doi.org/10.1109/ACCESS.2023.3326750>. URL: <https://doi.org/10.1109/ACCESS.2023.3326750>.
39. Paulo A. A. Resende та Ana Carolina Drummond. “A Survey of Random Forest Based Methods for Intrusion Detection Systems”. АНГЛ. В: ACM Computing Surveys 51.3 (2018). Intrusion detection, Random Forest hyperparameters and evaluation.
40. Fei Tony Liu, Kai Ming Ting та Zhi-Hua Zhou. “Isolation Forest”. АНГЛ. В: Proceedings of the 2008 IEEE International Conference on Data Mining (ICDM). Original description of Isolation Forest anomaly scores and tree-based isolation. 2008.
41. Maayan Frid-Adar та ін. “GAN-based Synthetic Medical Image Augmentation for Increased CNN Performance in Liver Lesion Classification”. АНГЛ. В: Neurocomputing 321 (2018), с. 321—331. DOI: <https://doi.org/10.1016/j.neucom.2018.09.013>. URL: <https://doi.org/10.1016/j.neucom.2018.09.013>.