МАГІСТЕРСЬКА РОБОТА

МР.ІПм - 13.00.00.000 ПЗ

Група ІПм-24-1

Стасюк Борис

2025

Івано-Франківський національний технічний університет нафти і газу

Інститут післядипломної освіти

Кафедра інженерії програмного забезпечення

Стасюк Борис Борисович

(прізвище, ім'я, по батькові)

УДК 004.9
(індекс)

МАГІСТЕРСЬКА РОБОТА

Моделі та методи визначення аномалій в даних
(назва роботи)

Інженерія програмного забезпечення

(назва освітньої програми)

121 - Інженерія програмного забезпечення

(шифр і назва спеціальності)

Здобувач освітнього ступеня _____Стасюк Борис Борисович_____
(підпис, ініціали та прізвище здобувача)

Науковий керівник ____Шекета Василь Іванович, д.т.н., професор_____
(підпис, прізвище, ім'я, по батькові, науковий ступінь, вчене звання керівника)

Допущено до захисту
Завідувач кафедри
_доц._____Бандура В.В.__
(посада) (підпис) (дата) (ініціали та прізвище)

Нормоконтроль
_асист._____ Ваврик Т.О.__
(посада) (підпис) (дата) (ініціали та прізвище)

Робота містить результати власних досліджень, використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

Івано-Франківськ – 2025

Івано-Франківський національний технічний університет нафти і газу

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення

Освітній рівень магістр

Спеціальність 121 – Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ:
Зав. кафедрою **ІПЗ**
доцент. В.В. Бандура
“ 04 ” вересня 2025 р.

ЗАВДАННЯ
НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ

Стасюк Борис Борисович

(прізвище, ім’я, по-батькові)

1. Тема проекту (роботи) "Моделі та методи визначення аномалій в даних"

керівник проекту (роботи) Шеката Василь Іванович д.т.н., професор

затверджені наказом вищого навчального закладу від “ 17” листопада 2025 р. № 117/7

2. Строк подання студентом проекту (роботи) 15 грудня 2025 р.

3. Вихідні дані до проекту (роботи) Результати і матеріали отримані під час проходження переддипломної практики

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

1. Аналіз моделей і методів виявлення аномалій
2. Дослідження застосування методів виявлення аномалій у комп’ютерних системах
3. Вивчення підходів до класифікації аномалій у даних
4. Розробка алгоритму та програмної реалізації методу виявлення аномалій

5. Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень)

1 Порівняння помилки узагальнення (рис. 1.1, ст. 16)

2 Ефективність kNN у вигляді ROC-кривих (рис. 2.1, ст. 20)

3 Ефективність найближчого сусіда для різних просторово розподілених наборів даних (рис. 2.2, ст. 22)

4 Порівняння алгоритмів з вкладеними циклами, на основі індексу та на основі розділення (рис. 2.3, ст. 24)

6. Консультанти розділів проекту (роботи)

Розділ	Консультант	Підпис, дата

2. Дата видачі завдання **04 вересня 2025 р.**

Керівник _____
(підпис)

Завдання прийняв до виконання_____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів проекту (роботи)	Строк виконання етапів проекту	Примітка
1	Підбір і вивчення наукової та технічної літератури з питань виявлення аномалій у даних	01.10.2025	виконано
2	Аналіз основних концепцій, моделей та алгоритмів визначення аномалій у предметній області	21.10.2025	виконано
3	Вивчення підходів і класифікацій методів виявлення аномалій	30.10.2025	виконано
4	Розробка алгоритмічної моделі з використанням методів визначення аномалій у даних	12.11.2025	виконано
5	Опис розробленого алгоритму виявлення аномалій та принципів його функціонування	25.11.2025	виконано
6	Дослідження функціонування інформаційних інтелектуальних систем на основі методів виявлення аномалій	01.12.2025	виконано
7	Затвердження пояснювальної записки роботи завідувачем кафедри	15.12.2025	виконано

Студент _____
(підпис)

Керівник роботи _____
(підпис)

АНОТАЦІЯ

Магістерська робота: 75 с., 31 рис., 9 табл., 36 джерел.

Метою роботи є теоретичне дослідження сучасних моделей і методів виявлення аномалій у даних, систематизація дискримінативних та генеративних підходів, а також порівняльний аналіз їхніх переваг, обмежень залежно від типу даних і предметної області.

Об'єкт дослідження – процеси виявлення аномалій у даних, що включають ідентифікацію відхилень від нормальної поведінки системи та характеристику природи аномалії (доброякісна чи зловмисна).

Предмет дослідження – теоретичні моделі та методи виявлення аномалій, зокрема дискримінативні та генеративні/ймовірнісні.

Результати дослідження: створено розгорнуту класифікацію методів виявлення аномалій, проведено теоретичний порівняльний аналіз дискримінативних і генеративних підходів, обґрунтовано критерії їх вибору залежно від наявності міток, розмірності даних та вимог до інтерпретуваності.

У вступі обґрунтовано актуальність теми та сформульовано мету й завдання роботи. У першому розділі розглянуто теоретичні основи виявлення аномалій. У другому розділі проаналізовано дискримінативні методи. У третьому розділі досліджено генеративні та ймовірнісні методи.

Висновки: систематизовані сучасні теоретичні підходи до виявлення аномалій дозволяють обґрунтовано обирати метод залежно від характеристик даних і задачі, що сприяє підвищенню точності та зниженню рівня хибних спрацювань у реальних системах моніторингу.

ВИЯВЛЕННЯ АНОМАЛІЙ, OUTLIER DETECTION, ДИСКРИМІНАТИВНІ МЕТОДИ, ГЕНЕРАТИВНІ МЕТОДИ, МАШИНИ ОПОРНИХ ВЕКТОРІВ, ONE-CLASS SVM, НАЙБЛИЖЧІ СУСІДИ, СУМІШ ГАУСІВСЬКИХ РОЗПОДІЛІВ, ПРИХОВАНІ МАРКОВСЬКІ МОДЕЛІ, ДИНАМІЧНІ БАЙЄСІВСЬКІ МЕРЕЖІ.

ANNOTATION

The master's thesis: 75 p., 31 fig., 9 tab., 36 sources.

The aim of the work is a theoretical study of modern models and methods of anomaly detection in data, systematization of discriminative and generative approaches, and a comparative analysis of their advantages and limitations depending on data type and domain.

Research object: processes of anomaly detection in data, including identification of deviations from normal behavior and characterization of anomaly nature (benign or malicious).

Research subject: theoretical models and methods of anomaly detection, particularly discriminative and generative/probabilistic.

Research results: a detailed classification of anomaly detection methods has been created, a theoretical comparative analysis of discriminative and generative approaches has been performed, and criteria for their selection have been substantiated depending on label availability, data dimensionality.

The introduction justifies the relevance of the topic and formulates the goal and objectives. The first chapter examines the theoretical foundations of anomaly detection. The second chapter analyzes discriminative methods. The third chapter investigates generative and probabilistic methods.

Conclusion: the systematized contemporary theoretical approaches to anomaly detection enable a reasoned selection of method based on data characteristics and task requirements, thereby improving accuracy and reducing false positive rates in real-world monitoring systems.

ANOMALY DETECTION, OUTLIER DETECTION, DISCRIMINATIVE METHODS, GENERATIVE METHODS, SUPPORT VECTOR MACHINES, ONE-CLASS SVM, K-NEAREST NEIGHBORS, GAUSSIAN MIXTURE MODELS, HIDDEN MARKOV MODELS, DYNAMIC BAYESIAN NETWORKS.

ЗМІСТ

ВСТУП

У сучасному світі, де обсяги даних зростають експоненційно, а системи стають дедалі складнішими та взаємопов’язанішими, виявлення аномалій у даних є критичною задачею для забезпечення безпеки, надійності та ефективності роботи інформаційних, промислових і кіберфізичних систем. Швидке виявлення відхилень від нормальної поведінки дозволяє вчасно реагувати на збої обладнання, кібератаки, шахрайські операції, деградацію виробничих процесів чи медичні аномалії. Традиційні підходи, що базуються на фіксованих порогах або простих статистичних правилах, вже не справляються з різноманітністю, мінливістю та високою розмірністю сучасних даних, що підкреслює необхідність розробки та дослідження нових моделей і методів виявлення аномалій.

Актуальність теми зумовлена стрімким зростанням складності даних і загроз, які вони приховують. Існуючі комерційні та академічні рішення часто орієнтовані на конкретні предметні області і погано узагальнюються на нові задачі. Більшість методів мають обмежувальні припущення, що робить їх неефективними в реальних умовах. Відсутність універсального підходу, який би надійно працював у різних сценаріях, призводить до високого рівня хибних спрацювань або пропущених аномалій, що може мати катастрофічні наслідки.

Метою роботи є дослідження, систематизація та порівняльний аналіз сучасних моделей і методів виявлення аномалій у даних різних типів (часові ряди, багатовимірні вектори, графові структури, потоки даних), а також розробка рекомендацій щодо вибору та комбінування методів залежно від характеристик предметної області та доступних даних.

Завданнями дослідження є: аналіз предметної області та огляд існуючих підходів до виявлення аномалій; класифікація методів за парадигмами (дискримінативні, генеративні, гібридні, навчання з учителем/без учителя/з частковим учителем; дослідження ключових алгоритмів і моделей; аналіз впливу властивостей даних на ефективність методів; розробка критеріїв та метрик оцінки; експериментальне порівняння методів на відкритих та

синтетичних наборах даних; формулювання рекомендацій щодо вибору та адаптації методів для реальних застосувань.

Об’єктом дослідження є процеси виявлення аномалій у даних різних типів і походження, включаючи виявлення початку несправної або нової поведінки системи, характеристику природи аномалії (доброякісна чи зловмисна, а також визначення можливих причин та кореляційних факторів.

Предметом дослідження є моделі, алгоритми та методичні підходи до виявлення аномалій, зокрема дискримінативні та генеративні методи, методи глибокого навчання, ансамблеві підходи, техніки адаптації до концептуального дрейфу, а також критерії вибору методу залежно від наявності міток, знань предметної області та часових характеристик даних.

Методи дослідження включають: теоретичний аналіз наукових джерел і систематизацію існуючих підходів; порівняльний аналіз алгоритмів за теоретичними властивостями та обмеженнями; експериментальний метод; статистичний аналіз результатів з використанням сучасних метрик; моделювання різних сценаріїв застосування.

Очікуваним результатом роботи є створена систематизована класифікація сучасних методів виявлення аномалій, порівняльна таблиця їхньої ефективності на типових задачах, а також практичні рекомендації та шаблони вибору методу для різних реальних застосувань. Отримані висновки сприятимуть підвищенню надійності систем моніторингу, зниженню кількості хибних спрацювань і швидшому реагуванню на реальні загрози та збої в різних галузях.

Структура та обсяг магістерської роботи

Магістерська робота викладена на 75 сторінках друкованого тексту, який складається із вступу, чотирьох розділів, висновків, списку використаних джерел (36 найменування). Робота містить 9 таблиць, 31 рисунки.

РОЗДІЛ 1
ТЕОРЕТИЧНІ ОСНОВИ ВИЯВЛЕННЯ АНОМАЛІЙ

1.1. Оцінка стану проти виявлення аномалій

У дискримінативних методах основна увага приділяється оптимізації правила прийняття рішень, яке класифікує дані за категоріями, що відповідають нормальним або аномальним режимам поведінки системи. Не докладається жодних зусиль для моделювання причинно-наслідкових зв'язків між даними та базовим системним процесом. З іншого боку, основна увага приділяється генеративним методам.є вивчити модель, яка описує системний процес. За допомогою генеративної моделі можна інтерпретувати систему та зрозуміти причинно-наслідковий зв'язок між прихованим станом системи та її спостережуваною поведінкою. Це відрізняється від дискримінативних методів, які розглядають базову систему як чорну скриньку. Однак, оскільки параметри для генеративних моделей часто вибираються для максимізації правдоподібності даних, ці моделі, як правило, будуть менш оптимізовані для задачі класифікації (виявлення аномалій). Тим не менш, залежно від конкретного застосування, один підхід може краще підходити для певної області, як ми побачимо в наступних розділах.

У цьому розділі ми вводимо позначення, які будуть використовуватися в нашому викладі. Ми використовуємо великі літери для позначення випадкових змінних і малі літери для позначення їхніх інстанцій. Наприклад, якщо бінарна змінна $X \in \{0, 1\}$, X може приймати значення $x = 0$ або $x = 1$.

Ми використовуємо жирний шрифт, коли говоримо про сукупність або набір подібних елементів. Наприклад, якщо маємо d змінних{X1, ..., Xd}, сукупність цих змінних позначається як $\mathbf{X} = \{X1, ..., Xd\}$. Ми також використовуємо жирний шрифт для векторів, оскільки вектори зазвичай є сукупністю більш ніж одного елемента.

Загалом, надрядкові індекси часто використовуються для індексації конкретної точки даних із сукупності точок даних. Наприклад, набір навчальних даних може складатися з N векторів даних, {x (1),.., x (N)}. n-й вектор даних позначається x(n), а його і -й елемент позначається x (n) i. Зверніть увагу, що x (n) і не позначається жирним шрифтом, оскільки це окремий елемент, а не вектор. Крім того, ми використовуємо p(·) для позначення щільності ймовірності та P(·) для позначення функцій ймовірності.

1.2. Дискримінативні та генеративні моделі

Виявлення аномалій тісно пов'язане з класифікацією. Фактично, можна визначити проблему виявлення аномалій як процес класифікації даних за різними категоріями, що відповідають нормальному та аномальному режимам поведінки системи. Як результат, ми розглянемо відмінності між дискримінаційним та генеративним підходами з точки зору їхніх класифікаційних можливостей. Таким чином, ми викладемо математичну теорію цих двох підходів в умовах контрольованої класифікації.

У контрольованій класифікації вхідні ознаки представлені випадковим вектором X, а вихідна мітка представлена випадковою змінною C. Хоча X може мати дійсні або дискретні значення, C вважається дискретним і приймає скінченні значення, що відповідають різним класам. X і C походять від невідомого розподілу ймовірностей p(X, C). Генеративна класифікація використовує підхід апроксимації p(X, C) за допомогою параметричного сімейства моделей, а потім застосовує правило Байєса для обчислення розподілів, обумовлених класом P(C|X). Кожен новий вектор даних x потім присвоюється найбільш ймовірній мітці c відповідно до P(C|X). Доповнюючий підхід дискримінативної класифікації полягає у безпосередньому пошуку правила класифікації з найменшим рівнем помилки. Іншими словами, цей підхід визначає P(C|X) на основі даних без попередньої оцінки спільного розподілу p(X, C). Очевидна відмінність від дискримінативного підходу

полягає в тому, що він не робить жодних припущень щодо вхідного розподілу p(X), тоді як генеративний підхід робить непрямі припущення щодо X під час обчислення спільного розподілу p(X, C) перед обчисленням умовного розподілу P(C|X). Іншими словами, ключова відмінність полягає в наступному: дискримінативні підходи застосовують P(C = k|X = x) для прямого розрізнення значення k для будь-якого екземпляра x, тоді як генеративні підходи оцінюють P(C = k|X = x) з P(C = k) і p(X = x|C = k), останнє з яких може бути використане для генерації випадкових екземплярів x, обумовлених цільовою міткою k.

Тепер ми більш детально розглянемо математичний зв'язок між дискримінаційними та генеративними класифікаторами. Припустимо, що навчальні дані, {x (n), c(n)} N n=1, де x (n) ∈ R d і c (n) ∈ {1, ..., K}, є незалежними і однаково розподіленими відповідно до деякого невідомого розподілу p(X, C). Мета полягає в обчисленні P(C|X), яке буде використовуватися для розробки правила класифікації, що категоризує нові дані з найменшою кількістю помилок. Для цього необхідно обчислити умовну ймовірність класу P(C = k|X) для кожного класу k. Для кожного класу k p(X|C = k) моделюється деяким розподілом fk з параметрами θk, а P(C = k) параметризується попередньою ймовірністю pk.

В цілому, параметри для спільного розподілу є Θ = {p1, ..., pK, θ1, ..., θK}. Припускаючи, що Θ відоме, завдання класифікації зводиться до віднесення нового вектора даних x до класу k, який максимізує

$$P(C = k \mid X = x) = (p_k \cdot f_k(x;\, \theta_k)) \,/\, (\Sigma_{\{c=1\}_c^{\{K\}p}} \cdot f_c(x;\, \theta_c)) \quad (1.1)$$

Як генеративні, так і дискримінативні методології використовують цей самий підхід високого рівня. Їх відмінність полягає в оцінці Θ. За заданих даних {x (n), c(n)} N n=1, параметри генеративного класифікатора вибираються таким чином, щоб максимізувати ймовірність даних, як показано нижче:

$$Theta_{hat_{Gen}} = argmax_{Theta} L_{Gen(Theta)}, where \quad (1.2)$$

$$L_{Gen(Theta)} = sum\ n = 1..N\ of \log\left(p_{c(n)} * f_{c(n)(x^n; Theta)}\right)$$

На відміну від цього, параметри дискримінаційного класифікатора вибираються таким чином, щоб мінімізувати втрати класифікації, які наближено оцінюються за формулою −LDisc, як показано нижче:

$$Theta_{hat_{Disc}} = argmax\ over\ Theta\ of\ L_{Disc(Theta)},$$

where (1.3)

$$L_{Disc(Theta)} = sum\ n = 1..N\ of \log\left(p_{c(n)} * \frac{f_{c(n)(x^n; Theta)}}{\left(sum\ k = 1..K\ of\ p_k * f_{k(x^n; Theta)}\right)}\right)$$

Після розширення LDisc можна легко побачити його зв'язок з LGen:

$$\mathrm{L_{Disc(\Theta)}} = \Sigma(\mathrm{n} = 1 \to \mathrm{N}) \log \mathrm{p_{c(n)}} \cdot \mathrm{f_{c(n)(x^n;\,\theta)}}$$
$$- \Sigma(\mathrm{n} = 1 \to \mathrm{N}) \log \left[\Sigma(\mathrm{k} = 1 \to \mathrm{K}) \mathrm{p_k} \cdot \mathrm{f_{k(x^n;\,\theta)}}\right] \quad (1.4)$$

Таким чином, різниця між LDisc і LGen становить LX, що представляє логарифмічну ймовірність ймовірнісної моделі над вхідним простором X. Це пояснює той факт, що генеративні моделі, як правило, схильні до тих, що максимізують ймовірність навчальних даних, тоді як дискримінативні моделі вільні від помилок упередженості через будь-яке неправильне представлення вхідного розподілу p(X).

Щоб ще більше проілюструвати різні особливості цих двох підходів, ми представляємо добре вивчену дискримінативно-генеративну пару класифікаторів: наївний Байєс і логістична регресія. Тут ми припускаємо, що параметри класифікаторів вже оцінені на основі описаної вище процедури, і наша мета — показати різні правила класифікації, пов'язані з кожним класифікатором. Параметри класифікатора наївного Байєса є оцінками розподілів P(C) та p(X|C), тоді як параметри класифікатора логістичної регресії є вагами {wm} d m=0. Для простоти припустимо, що існує лише два класи,

тобто C ∈ {0, 1}. За заданого нового вхідного вектора x new = {x1, ..., xd}, наївний класифікатор Байєса привласнить x new мітку c new, яка задовольняє

$$c_{new} = \arg\max_k P(C = k) \cdot \prod(i = 1 \to d)\, p(X_i = x_i \mid C = k) \qquad (1.5)$$

тоді як логістична регресія присвоїть x new значення c new = 0, якщо

$$P(C = 1 \mid X = x_{new}) = \frac{1}{1 + exp(w0 + \Sigma(i=1 \to d) w_i \cdot x_i)} < \frac{exp(w0 + \Sigma(i=1 \to d) w_i \cdot x_i)}{1 + exp(w0 + \Sigma(i=1 \to d) w_i \cdot x_i)} \qquad (1.6)$$

що зводиться до

$$1 < exp(\, w0 + sum\{i = 1\ to\ d\} w_i\ x_i\ ) \qquad (1.7)$$

і до c new = 1 в іншому випадку. Порівнюючи ці два випадки, можна побачити, що правила класифікації кардинально відрізняються, що ілюструє різницю в підході між генеративними та дискримінативними класифікаторами.

Загалом, генеративний підхід дає змогу вивчити модель спільного розподілу p(X,C)p(X, C)p(X,C), але оцінка умовного розподілу p(C|X)p(C \mid X)p(C|X) може бути упередженою, якщо модель p(X)p(X)p(X) побудована неточно. Оскільки справжній розподіл p(X)p(X)p(X) майже ніколи не є відомим, генеративна модель зазвичай має певний рівень систематичної похибки. Саме тому дискримінативні класифікатори часто вважаються кращими за свої генеративні аналоги. Проте це поширене переконання є лише частково правильним емпіричному порівнянні наївного байєсівського класифікатора з логістичною регресією.

Дослідження показало, що наївний байєс справді має вищу асимптотичну похибку, ніж логістична регресія. Водночас було виявлено важливу властивість: генеративна модель збігається значно швидше. Якщо ddd —

розмірність вхідного вектора, то наївний байєс досягає стаціонарних параметрів приблизно за O(log d) навчальних прикладів, тоді як логістична регресія потребує O(d) прикладів для збіжності. Це дозволяє сформулювати оптимальну стратегію класифікації: спочатку використовувати генеративну модель, яка швидко досягає прийнятної якості, а після накопичення достатнього обсягу даних — переходити до дискримінативної моделі, що має меншу асимптотичну похибку.

Теоретичні та емпіричні результати підтверджують цю гіпотезу. Експерименти проводилися на 15 наборах даних із репозиторію машинного навчання UCI. На рисунку 1.1 подано емпіричні криві асимптотичної помилки класифікації залежно від кількості навчальних прикладів. Вісім наборів містили неперервні ознаки, а сім — дискретні, що було відповідно позначено. Було встановлено, що у більшості випадків наївний байєс збігався швидше, але до моделі з вищою асимптотичною похибкою порівняно з логістичною регресією. Виняток становили малі набори даних, для яких обсяг тренувальних прикладів був недостатнім, щоб логістична регресія змогла збігтися до своєї оптимальної моделі із нижчою асимптотичною помилкою.

1.3. Основні концепції та класифікації методів аномалій

У багатьох аспектах дискримінативні підходи можна інтерпретувати як задачу апроксимації функції. За заданого XXX дискримінативні методи спрямовані на безпосереднє вивчення відображення від вхідних даних XXX до вихідної мітки класу CCC — або через пряму оцінку умовного розподілу P(C|X)P(C \mid X)P(C|X), або за допомогою інших способів, які мінімізують похибку класифікації.

Перевага дискримінативних класифікаторів полягає в тому, що вони фокусуються на пошуку межі прийняття рішення, яка розділяє класи, наприклад нормальної та аномальної поведінки системи. У результаті вони зазвичай є більш стійкими до викидів у даних порівняно з генеративними

моделями. Проте така зосередженість майже виключно на межі прийняття рішення призводить до того, що структура решти простору ознак ігнорується. Через це дискримінативні підходи забезпечують значно менше знань про внутрішню будову досліджуваної системи та гірше працюють за умов неповних або пропущених даних.

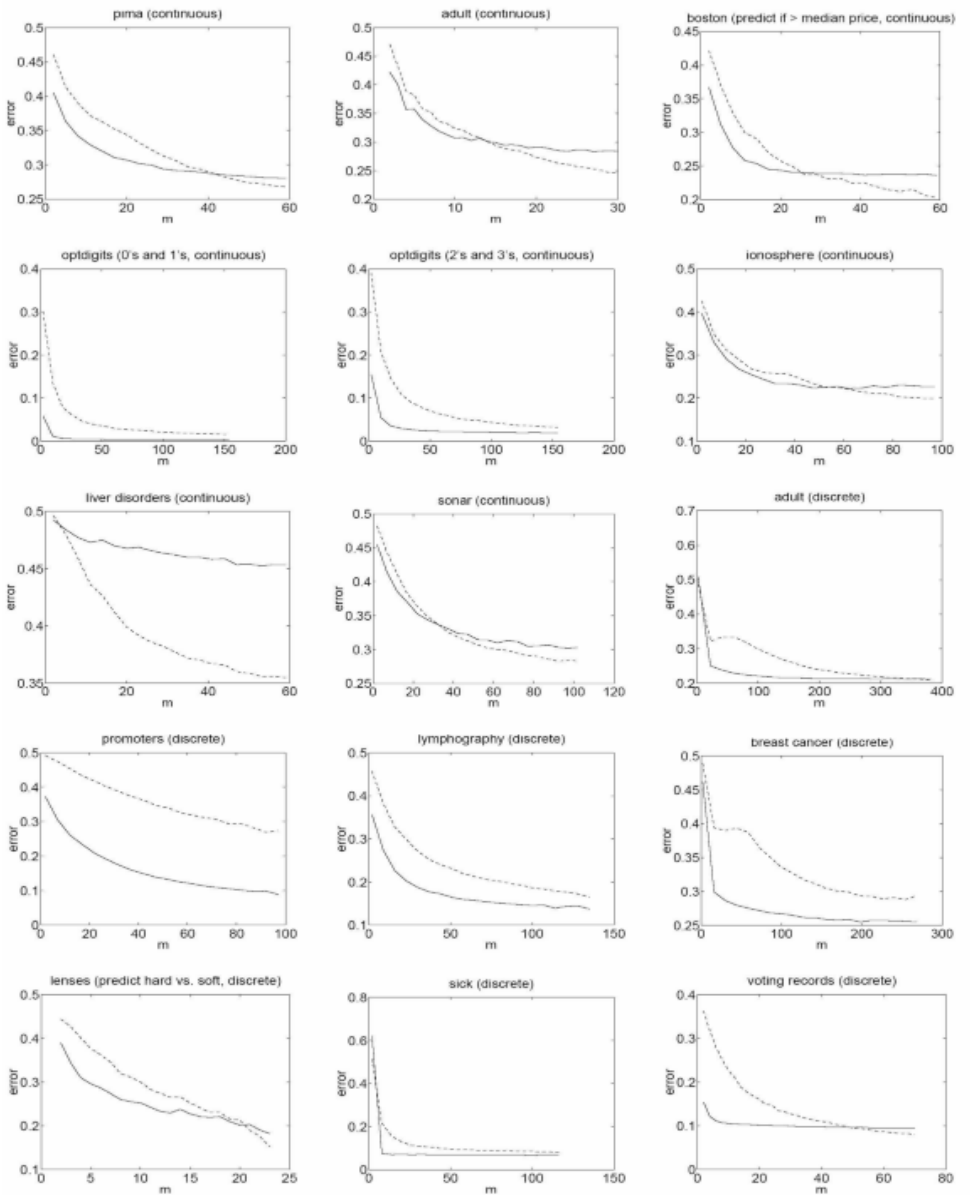

Рис. 1.1 Порівняння помилки узагальнення.

До популярних дискримінативних методів належать логістична регресія, лінійний/квадратичний/регуляризований дискримінантний аналіз, випадкові ліси, методи на основі відстаней, метод опорних векторів та традиційні

нейронні мережі. У цьому розділі основна увага приділяється трьом останнім підходам. Для кожного методу подано коротке пояснення та наведено вибірку сучасних досліджень, що мають особливе значення для задач виявлення аномалій.

Висновки до розділу

У першому розділі розглянуто теоретичні основи виявлення аномалій. Чітко розмежовано задачу оцінки стану системи (health assessment) та задачу виявлення аномалій (anomaly detection), показано, що остання є ширшою і не завжди потребує попереднього знання нормального стану. Проаналізовано фундаментальну відмінність між дискримінативними та генеративними моделями: перші навчаються розділяти «нормальне» і «аномальне», другі моделюють лише нормальну поведінку і вважають аномалією все, що погано пояснюється моделлю. Запропоновано розширену класифікацію методів з урахуванням типу даних, наявності міток, характеру аномалій (точкові, контекстні, колективні) та вимог до інтерпретуваності. Таким чином, створено єдину концептуальну базу, яка дозволяє системно підходити до вибору методу в подальших розділах.

РОЗДІЛ 2
ДИСКРИМІНАТИВНІ МЕТОДИ ВИЯВЛЕННЯ АНОМАЛІЙ

2.1. Методи, що ґрунтуються на відстані

У цьому підрозділі ми розглянемо різноманітні класифікатори та методи виявлення викидів, які використовують поняття просторової відстані для розрізнення нормальних та аномальних векторів ознак. (Нормальні вектори ознак — це ті, що відповідають нормальним класам; аномальні визначаються аналогічно.) Відмінність між класифікаторами та методами виявлення викидів є досить тонкою, але загалом класифікатори навчаються в умовах контрольованого навчання, де тренувальні дані містять мітки класів, тобто тоді як детектори викидів можуть використовувати кластеризацію або методи зниження розмірності, що навчаються в умовах неконтрольованого навчання, де дані не мають міток.

Крім того, багато методів виявлення викидів базуються на двох припущеннях щодо тренувальних даних. Перше — що значна частина навчальних даних є нормальними зразками. Друге — що аномальні вектори ознак можуть бути якісно відрізнені від нормальних. За наявності цих двох припущень — рідкісності та відхилення аномалій від нормальних характеристик — аномальні вектори можна трактувати як викиди, а отже, алгоритми виявлення викидів можуть бути використані для виявлення аномалій.

2.2. Метод найближчого сусіда

Класифікатор найближчого сусіда є одним із найпоширеніших методів виявлення аномалій. Інтуїція методу проста: вектори ознак, що розташовані близько один до одного за певною метрикою відстані, належать до одного

класу. Класифікатор найближчого сусіда передбачає наявність даних із мітками і призначає новий зразок w_{newx} до класу його найближчого сусіда.

Популярним узагальненням цього підходу є метод k найближчих сусідів (kNN), де для визначення класу C_{newC} нового зразка використовуються kkk найближчих сусідів. Один зі способів визначення класу — більшістю голосів, де C_{newC} присвоюється як найбільш поширений клас серед kkk сусідів. Інший спосіб — зважене голосування, коли голос кожного сусіда зважується відповідно до його відстані до X_{newx}: ближчі сусіди мають більший вплив. У такій схемі ваги для кожного класу підсумовуються, і C_{newC} обирається як клас із максимальним сумарним значенням.

Метод kNN успішно застосовувався для виявлення аномалій. У цьому дослідженні kNN використовувався для виявлення вторгнень у мережу на основі трас поведінки програм. Робота спирається на підходи з категоризації текстів та представляє поведінку програм у текстовому форматі, що дозволяє застосовувати kNN для класифікації нормальної та шкідливої поведінки. Зокрема, кожен системний виклик розглядається як «слово», а їхня послідовність під час виконання програми — як «документ».

Спочатку класифікатор kNN навчали на змодельованих даних, що не містили атак, щоб охарактеризувати нормальну поведінку. Новий зразок X_{newx} вважається аномалією (вказівкою на атаку), якщо середня відстань до його kkk найближчих сусідів перевищує заданий поріг. Експерименти проводилися на наборі DARPA, який містив велику кількість моделюваних атак, вбудованих у нормальний трафік.

Для фіксованого kkk якість роботи kNN оцінюється за ROC-кривою, що відображає залежність між точністю виявлення та ймовірністю хибних тривог. На рисунку 2.1 показано продуктивність kNN для різних значень kkk.

Для малих kkk час виконання kNN становить O(N)O(N)O(N), де NNN — кількість процесів у тренувальних даних. Через це kNN може бути неефективним для великих NNN. З метою покращення ефективності kNN може бути поєднаний із перевіркою сигнатур, яка встановлює правила або

властивості, характерні для окремих класів. Поліпшена версія kNN може навчатися новим класам, що відповідають підмножині відомої шкідливої поведінки програм. Таблиця 2.1 демонструє його ефективність для виявлення нових типів шкідливої активності.

Цікава теоретична властивість методу найближчих сусідів полягає в тому, що зі збільшенням числа тренувальних прикладів його похибка ніколи не перевищує подвоєної Байєсівської похибки.

Попри цю властивість, використання методу найближчих сусідів не завжди є доцільним, що теоретично та емпірично показано в роботі. Загалом необхідно переконатися, що дані розподілені у просторі таким чином, що існує чітка різниця між найближчими та найдальшими сусідами для будь-якого типового вектора ознак X_{newx}.

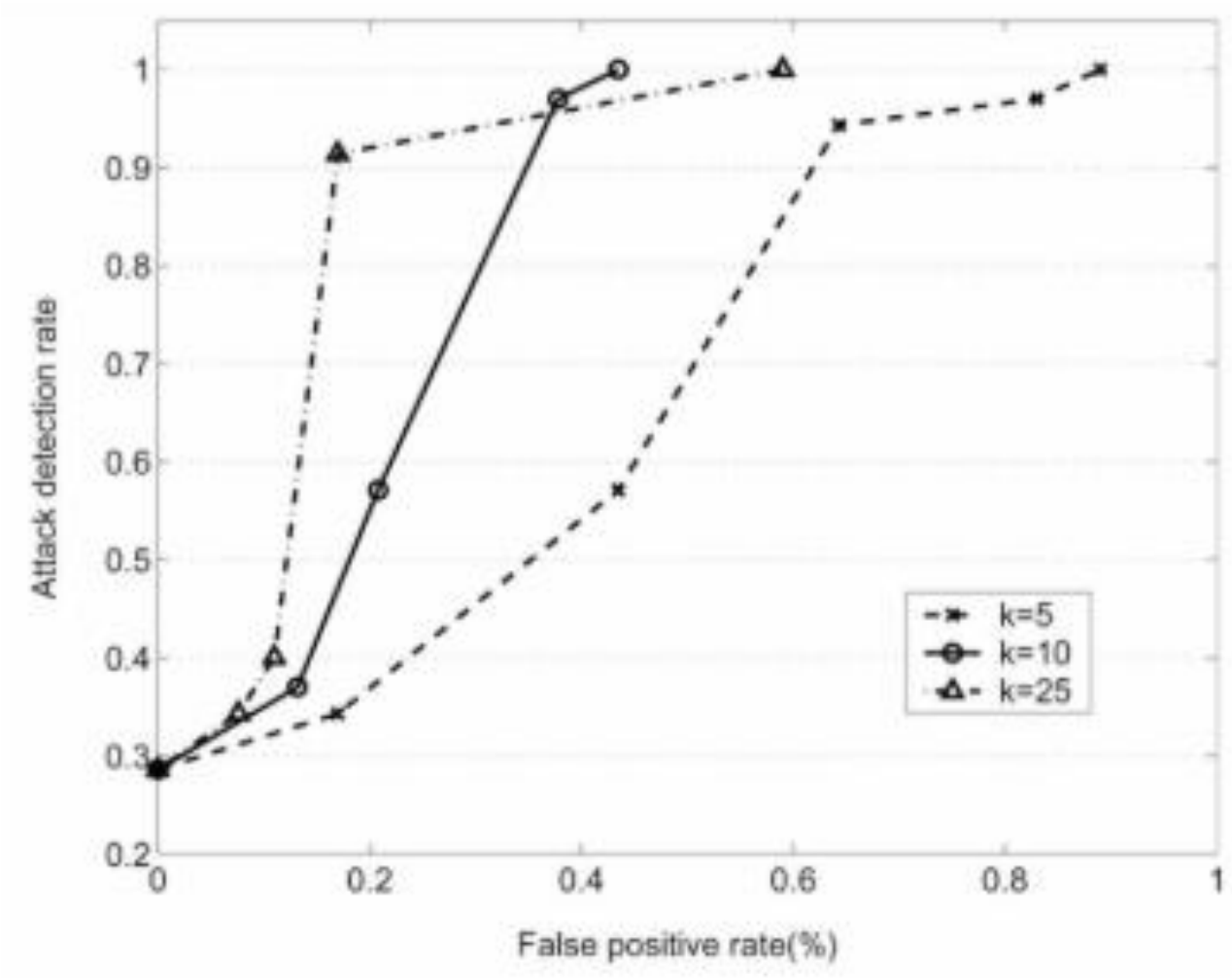

Рис. 2.1 Ефективність kNN у вигляді ROC-кривих

Таблиця 2.1

Рівень виявлення атак для даних DARPA при поєднанні kNN з перевіркою підпису.

Тип атаки	**Випадки**	**Виявлено**	**Рівень виявлення**
Відомі атаки	16	16	100%
Нові атаки	8	6	75%
Всього	24	22	91.7%

(У деякій літературі X_{newx} також називають вектором запиту.) Із зростанням розмірності простору відстань від X_{newx} до його найближчого сусіда зазвичай наближається до відстані до найдальшого сусіда — уже в просторах розмірності 10–15.

Це явище підтверджується емпіричними результатами, показаними на рисунку 2.2, де усереднене відношення відстані до найдальшого сусіда (DMAX) до відстані до найближчого сусіда (DMIN) зображено як функцію розмірності даних (тобто кожен тренувальний вектор має розмірність mmm). Усереднення виконано на 1000 запитах для синтетичних наборів даних, що містять один мільйон об’єктів. Набори даних згенеровані різними розподілами ймовірностей. Лінія «uniform» показує результат для рівномірно розподіленого набору даних. Лінія «recursive» відповідає набору, у якому кожна пара вимірів корельована, а кожний новий вимір має зростаючу дисперсію. Лінія «two degrees of freedom» відображає результат для набору, згенерованого як зважена сума двох рівномірно розподілених випадкових величин.

Для m=1m = 1m=1 відношення $DMAX\_m \,/\, DMIN\_m \approx 10\text{^}7,$ що створює значний контраст між найближчим і найдальшим сусідом. Однак зі збільшенням mmm цей контраст стає незначним, що видно зі зменшення порядків величини $DMAX\_m \,/\, DMIN\_m.$

Щоб підвищити ефективність методу найближчих сусідів у просторах великої розмірності, було запропоновано інтерактивну систему. У цій роботі

описано інтерактивну систему «людина-комп’ютер» для пошуку найближчих сусідів у високих розмірностях, у якій високовимірні тренувальні дані проєктуються на ретельно відібрані простори меншої розмірності. Передбачається, що такі нижчорозмірні проєкції краще відображають суттєві зв’язки між тренувальними даними та вектором запиту X_{newx}.

Проєкції вибираються залежно від того, наскільки добре вони розрізняють (у нижчій розмірності) кластери, що містять X_{newx}, від решти даних. Після того як комп’ютер знаходить такі проєкції, вони відображаються користувачу, який може висловити свої уподобання щодо цих проєкцій, аби сформувати більш значуще уявлення про найближчих сусідів X_{newx} з його власної перспективи. Мотивація цього підходу полягає в тому, що багаторазовий зворотний зв’язок від користувача протягом декількох ітерацій має дозволити системі знайти набір статистично значущих і значущих сусідів

.

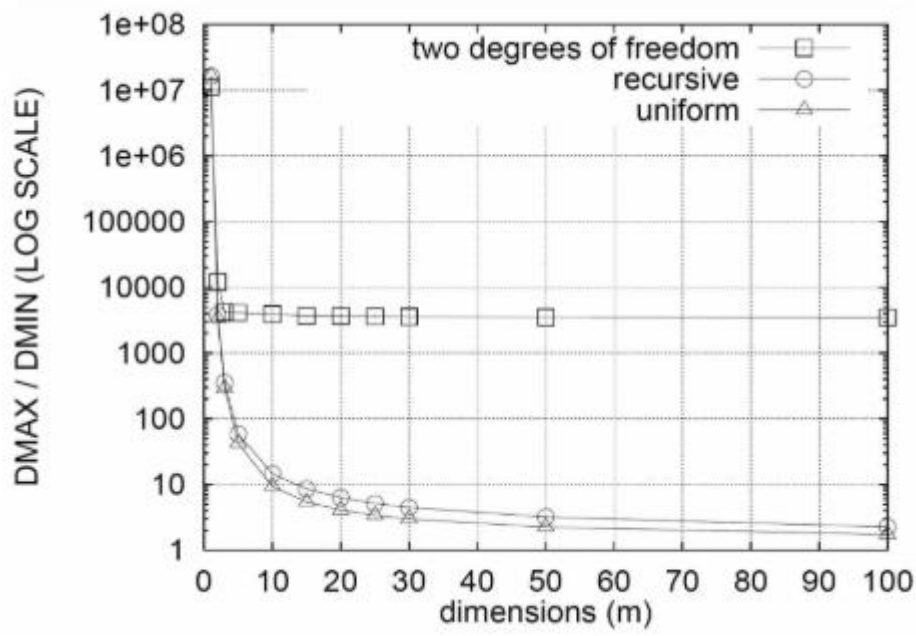

Рис. 2.2 Ефективність найближчого сусіда для різних просторово розподілених наборів даних.

Ця інтерактивна система була протестована на ряді реальних наборів даних з репозиторію машинного навчання UCI. Зокрема, було проведено

порівняння між запропонованим інтерактивним алгоритмом найближчого сусіда та стандартним (повнорозмірним) алгоритмом найближчого сусіда на наборах даних про іоносферу та сегментацію з репозиторію UCI. В ході експерименту вимірювалася точність класифікації найближчих сусідів для 10 векторів запиту. Результати експерименту наведені в таблиці 2.2, де показано, що продуктивність інтерактивної системи є значно вищою.

Таблиця 2.2

Точність класифікації для стандартного алгоритму найближчих сусідів.

Набір даних (розмірність)	**Точність (Стандартна НМ)**	**Точність (Інтерактивна НМ)**
Іоносфера (34)	71%	86%
Сегментація (19)	61%	83%

Альтернативним підходом є уникнення потреби у розмічених навчальних даних шляхом використання методів виявлення викидів. У цих методах аномалії розглядаються як викиди відносно навчальних даних і визначаються виключно за їхнім просторовим розташуванням щодо інших векторів у навчальній вибірці.

Для визначення викидів, що базуються на відстані, використовується відстань від вектора ознак до його k-го найближчого сусіда. У цій концепції кожен вектор у наборі даних ранжується на основі відстані до його k-го найближчого сусіда, і m векторів з найбільшими рангами визначаються як викиди. Ця евристика є інтуїтивною, оскільки вектори з найвищими рангами будуть менш щільно згруповані порівняно з тими, що знаходяться у нижчих рангах, і тому ці вектори є викидами відносно решти даних.

Позначимо через: $D\_k(x)$ — відстань від точки x до її k-го найближчого сусіда. Стандартні алгоритми (такі як вкладені цикли або алгоритми на основі індексів) можуть бути використані для обчислення: $D\_k(x\^(n))$ для кожної

точки x^(n) у наборі даних $\{x\hat{}(n)\}\_(n = 1..N)$, але такі алгоритми є обчислювально дорогими і потребують до: $O(N\hat{}2)$ обчислень.

Щоб усунути цю неефективність, запропоновано алгоритм, що базується на розбитті даних. Цей алгоритм використовує підхід «поділяй і володарюй»: він розбиває набір даних на непересічні підмножини, а потім відсікає ті частини, які гарантовано не містять викидів. Таким чином, необхідно виконати значно менше обчислень для величини: $D\_k(x)$ що призводить до суттєвого прискорення виконання.

Стандартний алгоритм і запропонований алгоритм на основі розбиття були протестовані на синтетичному наборі даних, який містив 100 гіперсферичних кластерів рівномірно розподілених даних, а також 1000 рівномірно розкиданих викидів. На Рисунку 2.3 зображено залежність часу виконання кожного алгоритму від кількості векторів даних N, індекса сусіда k та розмірності вектора даних.

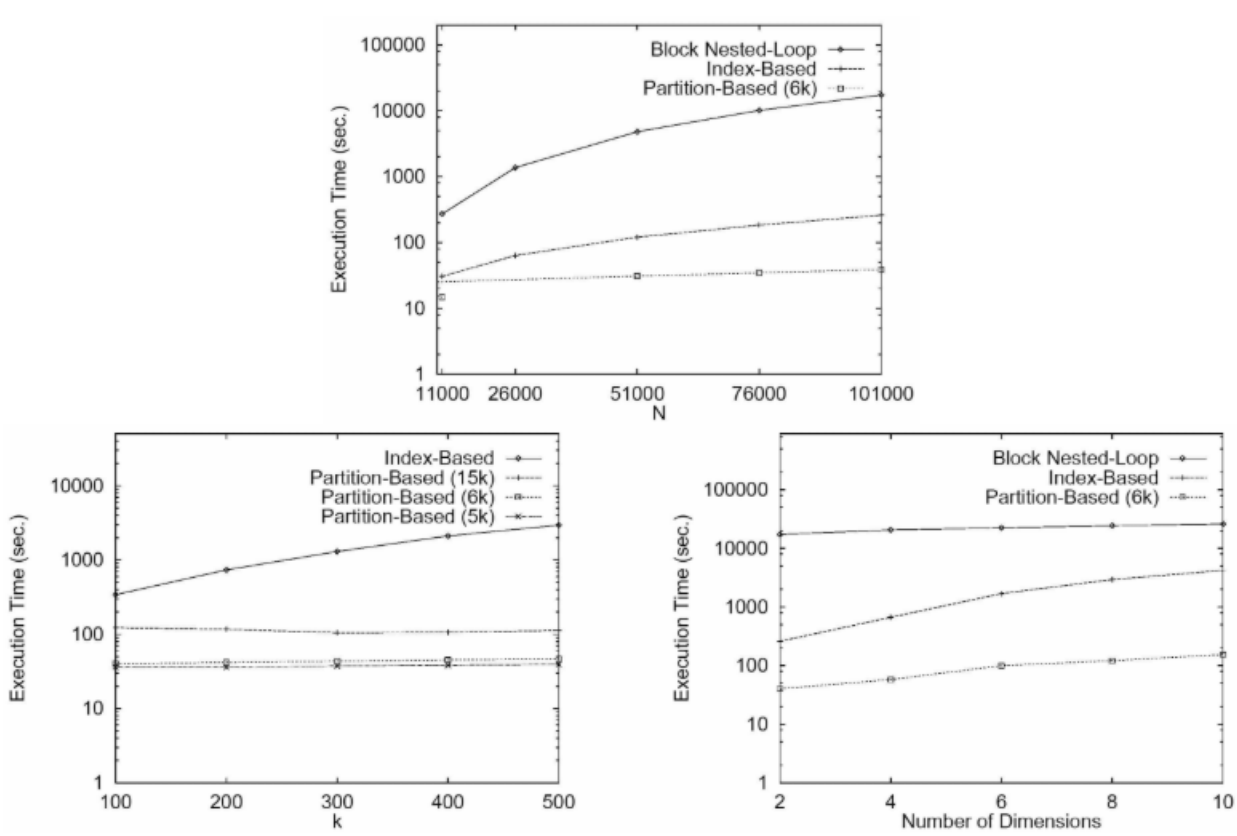

Рис. 2.3 Порівняння алгоритмів з вкладеними циклами, на основі індексу та на основі розділення

З Рисунка 2.3 видно, що алгоритм, заснований на розбитті, є значно швидшим за стандартні алгоритми та добре масштабується як за розміром вибірки, так і за її розмірністю. На додаток до експериментів на синтетичних даних, алгоритм, заснований на розбитті, також був протестований на реальній базі даних NBA (National Basketball Association), де окремих гравців було позначено як викиди через їхнє домінування з великим відривом у певному ігровому аспекті.

У цьому ж дусі підходу «поділяй і володарюй», представлено схожий метод, за яким вектор даних визначається як викид, якщо принаймні частка f елементів набору даних розташована на відстані, більшій за D. Такий викид позначається як DB(f, D) outlier. Знову наведено два прості алгоритми — підхід, заснований на індексах, та підхід з вкладеними циклами — для пошуку DB(f, D) викидів.

Щоб знайти всі DB(f, D) викиди в наборі даних, обидва алгоритми мають найгіршу обчислювальну складність: $O(dN)$ де d — розмірність, а N — розмір набору даних. Також представлено оптимізований алгоритм, заснований на поділі простору на комірки (cell-based), який масштабується лінійно за N, але експоненційно за d. Ідея: вектори даних розбиваються на комірки, і викиди визначаються на основі кожної комірки, а не для кожного вектора окремо. Такий підхід дозволяє швидко відсікти велику кількість векторів, які точно не можуть бути викидами, що суттєво скорочує час виконання.

Експериментальні результати показують, що підхід на основі комірок перевершує методи на основі індексів та вкладених циклів для $d \leq 4$. Цей метод також був застосований до трьох реальних задач, включаючи аналіз статистики NHL (National Hockey League), просторово-часові траєкторії зі спостережних відео та дані про ефективність роботодавців у сфері компенсації працівникам. Рисунок 2.4 демонструє частину результатів із дослідження щодо виявлення викидів на основі відеоспостереження. Викиди визначалися на основі відмінностей у швидкості та/або траєкторії між точками входу й виходу

пішохода. Результати свідчать, що ідею DB-викидів можна успішно застосовувати для виявлення аномалій у просторово-часових даних.

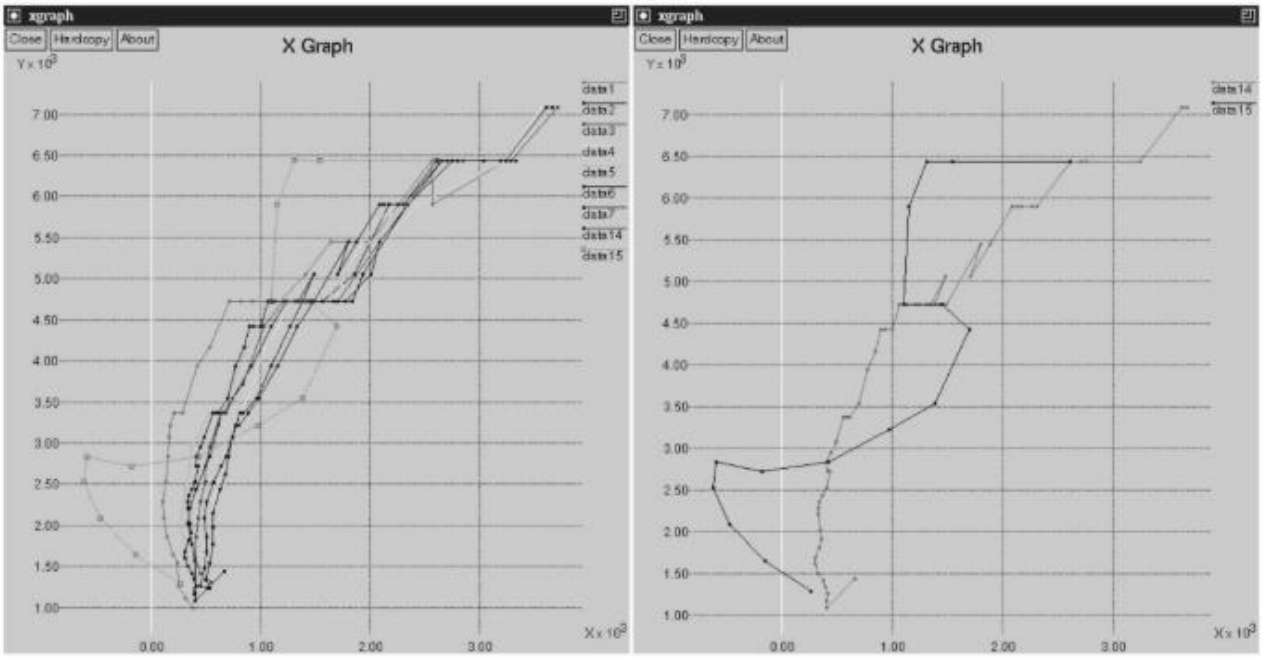

Рис. 2.4 Результати виявлення аномалій у просторово-часових даних з відеоспостереження.

До цього моменту ми розглядали лише алгоритми, які трактують стан «бути викидом» як бінарну властивість: тобто вектор ознак x є викидом або з імовірністю 0%, або з імовірністю 100%. Але в деяких сценаріях може бути більш доцільно приписувати вектору x певний ступінь «викидовості». Саме такий підхід, де запропоновано новий метод виявлення викидів, що базується на понятті локального фактора викиду (LOF — Local Outlier Factor). LOF вимірює ступінь, до якого вектор даних x є викидом, залежно від того, наскільки ізольованим він є відносно свого локального оточення. На відміну від алгоритму k-найближчих сусідів, метод LOF використовує густину точок навколо x, а не лише відстані до k найближчих сусідів.

Метод LOF реалізується у вигляді двоетапного алгоритму. Для кожного вектора x на першому етапі знаходяться всі сусідні вектори, що лежать на відстані не більшій за $D_k(x)$ від x, і їхні фактичні відстані до x зберігаються у

базі даних. Другий етап обчислює значення LOF для всіх точок, використовуючи цю базу. Складність першого етапу залежить від конкретної реалізації й була оцінена як: $O(N\ log\ N)$

Складність другого етапу становить: O(N) де N — кількість векторів у наборі даних.

Алгоритм LOF був протестований на синтетичному двовимірному наборі даних та на двох реальних спортивних наборах — один з хокею та один з футболу. Результати для синтетичного набору наведено на Рисунку 2.5, де представлено наочну графічну картину всіх обчислених значень LOF. Емпіричні результати для реальних наборів також показують, що метод LOF здатний знаходити значущі викиди, які інші підходи не виявляють. Це підтверджується порівняльним дослідженням , де було оцінено популярні методи виявлення викидів, включно з LOF.

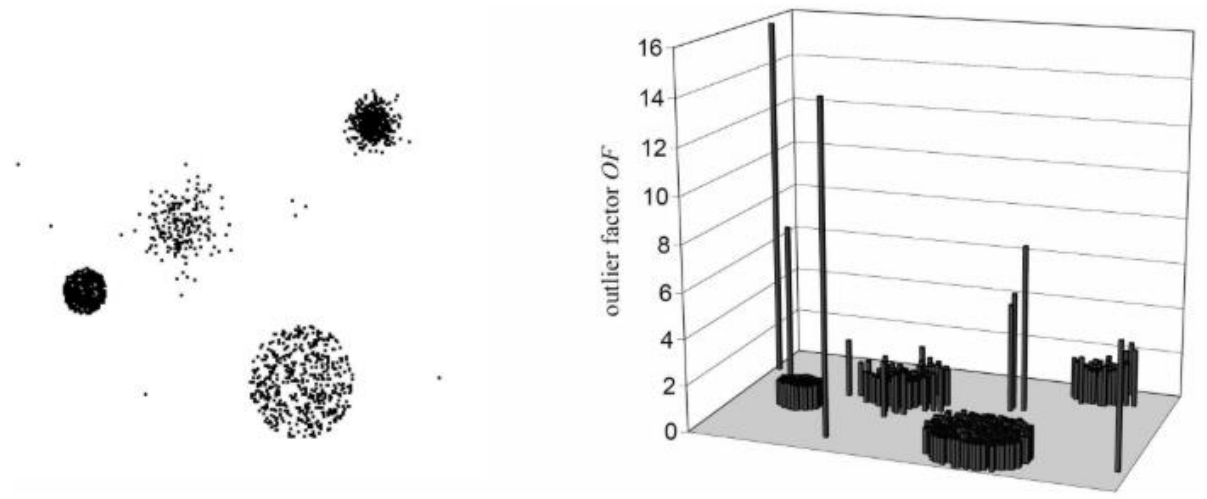

Рис. 2.5 Місцеві фактори відхилення для даних у синтетичному наборі даних.

У цьому дослідженні LOF порівнювали з:

• Найближчим сусідом: вектор ознак є винятковим, якщо відстань до його найближчого сусіда перевищує заданий поріг.

• На основі відстані Махаланобіса: обчислюються середнє значення та стандартне відхилення для навчальних даних. Вектор ознак є винятковим, якщо відстань Махаланобіса до середнього значення навчальних даних перевищує заданий поріг.

• Неконтрольовані машини опорних векторів.

Результати представлено у вигляді кривих ROC на рисунку 2.6, який показує, що LOF перевершує всі інші методи виявлення вторгнень у мережу для набору даних DARPA.

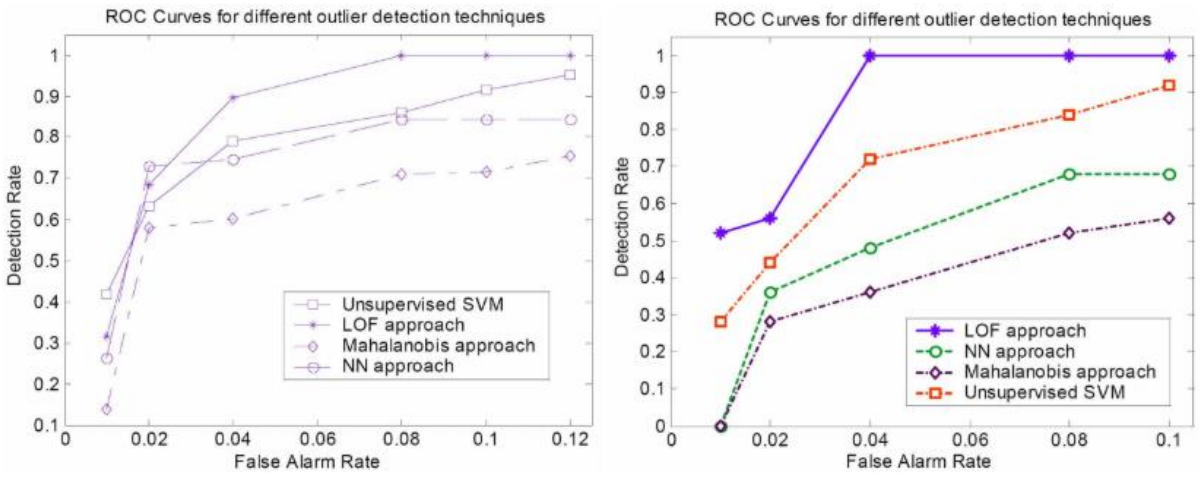

Рис. 2.6 Порівняння різних алгоритмів виявлення аномалій при спалахових атаках (праворуч) і при атаках з одним підключенням (ліворуч)

2.3 Машини опорних векторів

Окрім методів, заснованих на відстані, машини опорних векторів також широко використовуються для виявлення аномалій, особливо в областях виявлення вторгнень і медичної діагностики. У цьому підрозділі ми спочатку пояснимо контрольовану версію машин опорних векторів, а потім коротко обговоримо її неконтрольовану версію на прикладі застосування.

Мета машини опорних векторів (SVM) полягає в тому, щоб визначити гіперплощину рішення, яка розділяє різні класи з найбільшим відступом від найближчих навчальних прикладів. Опорні вектори, як показано та визначено

на Рисунку 2.7, є навчальними прикладами, що визначають оптимальну гіперплощину, яка утворює перпендикулярний бісектрис опорних векторів. По суті, опорні вектори спрямовані на представлення найбільш інформативних шаблонів, які дозволяють найкраще відрізняти різні класи.

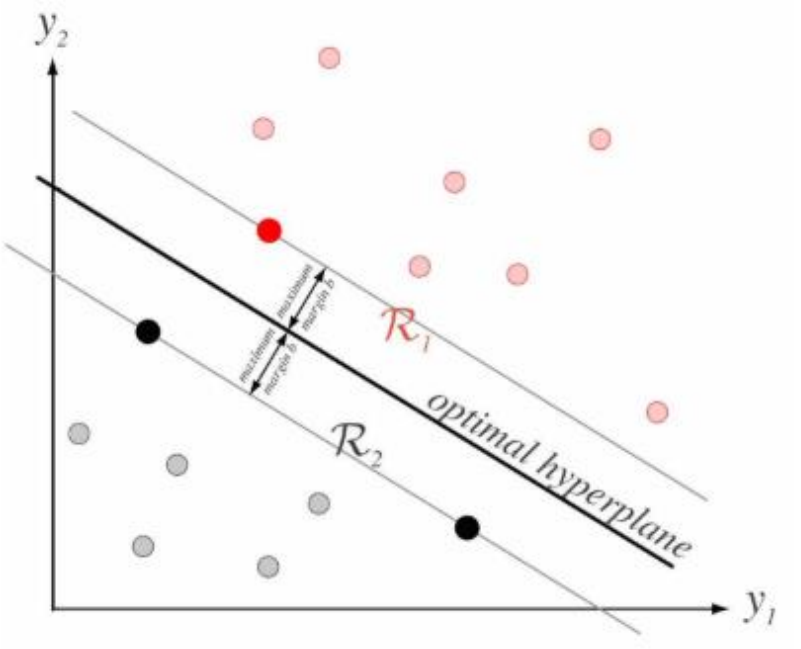

Рис. 2.7 Мета SVM — знайти оптимальну гіперплощину з максимальною відстанню до найближчих зразків.

Щоб визначити ці опорні вектори, SVM застосовують перетворення до даних, щоб шаблони, представлені даними, стали лінійно віддільними (тобто, могли бути розділені гіперплощиною). Це можливо, оскільки нелінійно віддільні шаблони завжди можуть стати лінійно віддільними в достатньо високовимірному представленні. Таким чином, дані відображаються за допомогою відповідної (нелінійної) функції в вищий вимір, і виконується оптимізація для знаходження оптимальної розділяючої гіперплощини.

Варіанти SVM включають SVM з жорстким відступом для віддільних класів, SVM з м'яким відступом для невіддільних класів та робастні SVM, які узагальнюються на шумові дані. Здатність обробляти шумові дані важлива в будь-якому сценарії виявлення чи класифікації, особливо оскільки безшумові

або чисті дані можуть бути важко або дорого отримати для реальних систем, де дані можуть походити з шумових показань сенсорів або бути неправильно поміченими через помилки людини/машини. Крім того, для динамічних систем, де нормальна поведінка може змінюватися з часом, особливо важливо, щоб схема виявлення аномалій могла обробляти шумові дані, оскільки мітки, присвоєні векторам ознак під час навчання, можуть стати ненадійними.

Стандартну та робастну версії SVM порівнюють з класифікатором k-найближчих сусідів (kNN) на чистих і шумових даних. Дослідження використовувало набір даних DARPA Intrusion Detection System Evaluation, де були витягнуті чистий набір даних і шумовий набір даних (див. Таблицю 2.3) для навчання та тестування.

Результати виявлення показані на Рисунку 2.7, де продуктивність робастної SVM, стандартної SVM та класифікатора kNN виражено як ROC-криві на чистих і шумових даних.

Таблиця 2.3

Чисті та зашумлені набори даних, використані в дослідженні.

	Чисті дані	**Шумні дані**
Навчання	300 нормальних процесів	316 нормальних процесів (16 неправильно позначених)
Тестування	28 інтрузивних процесів 5285 нормальних процесів, 22 інтрузивних сесій	12 інтрузивних процесів

На чистих даних швидкість виявлення атак при нульовому рівні хибнопозитивних результатів становила 74,7% для робастної SVM, 50% для стандартної SVM та 13,6% для kNN, тоді як 100% швидкість виявлення атак була досягнута при рівні хибнопозитивних результатів 3% для робастної SVM, 14,2% для стандартної SVM та 8,6% для kNN. Зокрема, здається, що kNN

показав найгірші результати, а робастна SVM — найкращі. На шумових даних швидкість виявлення атак при нульовому рівні хибнопозитивних результатів становила 50% для робастної SVM та 54% для стандартної SVM, тоді як 100% швидкість виявлення атак була досягнута при рівні хибнопозитивних результатів 8% для робастної SVM та 100% для стандартної SVM (що практично марно). Робастна SVM демонструє дуже незначне погіршення продуктивності в присутності шуму, тоді як SVM показує значне погіршення. Стійкість kNN до шуму можна пояснити усередненням, яке воно виконує над k найближчими сусідами тестового вектора, що дозволяє згладити вплив ізольованих шумових навчальних прикладів. Тим не менш, якщо навчальні приклади були неправильно класифіковані, а тестовий вектор виявився одним із цих неправильно класифікованих навчальних прикладів, то класифікатор kNN не зможе виявити вторгнення. Загалом, це дослідження показує, що робастні SVM досить добре підходять для виявлення аномалій у шумових даних, оскільки робастні SVM не так схильні до перенавчання на шумі та також призводять до швидшого часу виконання завдяки зменшеній кількості опорних векторів.

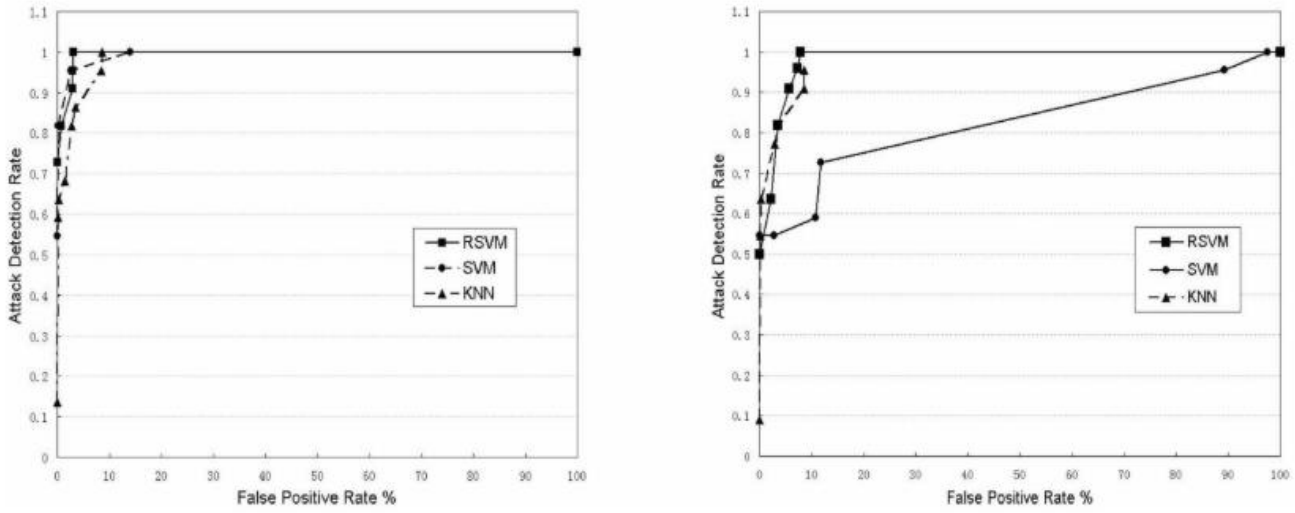

Рис. 2.8 Ефективність надійного SVM, стандартного SVM та класифікатора kNN виражена у вигляді кривих ROC для чистих даних (ліворуч) та даних із шумом (праворуч).

SVM, обговорювані досі, є методами з учителем, де припускається наявність помічених даних для цілей навчання. Однак, за відсутності помічених даних або в присутності високо ненадійних помічених даних, тоді методи без учителя можуть бути бажанішими з практичної точки зору. Високорівнева ідея несупервізійної SVM полягає в тому, що вона знаходить область, де лежить більшість даних, і асоціює ці дані як один клас. Доповнення цих даних тоді вважається належним до окремого класу. Цей алгоритм був оцінений на наборі даних USPS (United States Postal Service) рукописних цифр, який містить 9298 цифрових зображень по 256 пікселів, з яких останні 2007 зображень були використані як тестовий набір для цього емпіричного дослідження. Топ-20 викидів показано на Рисунку 2.8. Під кожним цифровим зображенням курсивне число — це вихід SVM, а жирне число — це мітка класу, присвоєна зображенню. Як видно, ці викиди відповідають нетиповим прикладам, які особливо важко зіставити з їхніми представницькими цифрами.

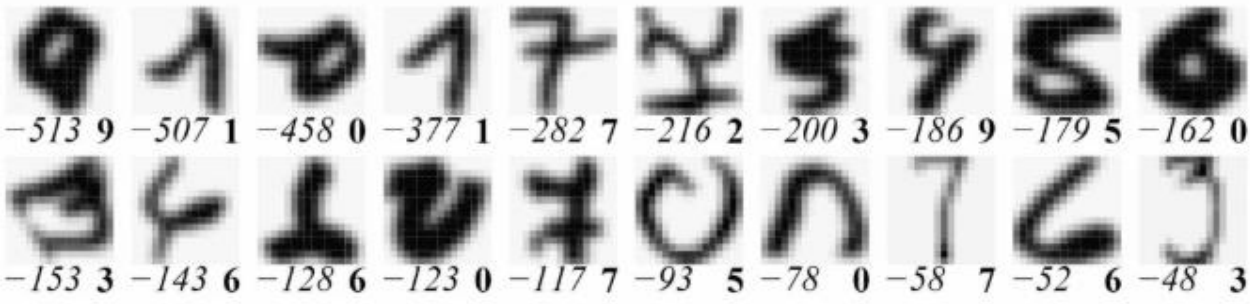

Рис. 2.9 Виявлені неконтрольованою SVM винятки, відсортовані за негативним результатом алгоритму.

Порівняльне дослідження ефективності неконтрольованої SVM та інших алгоритмів виявлення відхилень на основі відстані у виявленні вторгнень. Результати цього дослідження представлено раніше на рисунку 2.9. Окрім систем виявлення вторгнень, SVM також успішно застосовуються для діагностики глаукоми.

2.4 Нейронні мережі

Нейронна мережа є методом обчислення, натхненним біологією, на основі абстрактного представлення мозку. Аналогічно мозку, який складається з великої кількості високо взаємопов'язаних мереж нейронів, нейронна мережа складається з різних одиниць, організованих у шари для імітації процесу навчання мозку.

Як і мозок, нейронна мережа навчається на прикладах, де кожна нейронна мережа тренується для конкретного застосування через процес навчання. Цей процес навчання може бути або з учителем, або без учителя. У навчанні з учителем набір помічених даних обробляється нейронною мережею. Для кожного нейронна мережа порівнює свій вихід класифікації с з істинною міткою і використовує цю помилку для точного налаштування своїх параметрів відповідно. Навпаки, навчання без учителя використовує набір непомічених даних. Процес, за допомогою якого нейронна мережа самоорганізовує дані в різні класи без використання зовнішніх міток, відомий як самоорганізація або адаптація. Загалом, навчання з учителем виконується офлайн, а навчання без учителя — онлайн. Нейронні мережі широко використовуються для виявлення аномалій завдяки їхньому успіху в розпізнаванні шаблонів і класифікації даних. У цьому підрозділі ми зосереджуємося на застосуваннях, де нейронні мережі навчаються в супервізійному режимі.

Базовою одиницею нейронної мережі є нейрон, названий на честь біологічного нейрона, який надихнув на його модель. Кожен нейрон має одну основну функцію: видавати відповідь на зважену суму своїх входів. Характер відповіді залежить від функції активації нейрона. Кожен нейрон може мати різну функцію активації. Але на практиці більшість нейронів мають однакову функцію активації, і часто вибирається логістична функція.

Нейронна мережа складається з вхідного шару, змінної кількості прихованих шарів і вихідного шару нейронів. Кожен шар може мати змінну кількість нейронів, за винятком вхідного шару, який зазвичай обмежений

кількістю нейронів, рівною розмірності вхідного вектора ознак. Вхідний шар приймає на вхід дані, які, в свою чергу, обробляються прихованим шаром (шарами). Результат цієї обробки прихованого шару потім передається на вихідний шар, який видає результат класифікації. Рисунок 2.10 показує базову структуру тришарової нейронної мережі.

Теоретично, тришарова нейронна мережа може реалізувати будь-яку неперервну функцію (або для оцінки щільності, або для класифікації), за умови достатньої кількості прихованих одиниць і правильних параметрів моделі. Однак питання вибору оптимальної структури нейронної мережі для конкретної проблеми все ще залишається певним мистецтвом. (Див. Рисунок 2.10 для різноманітних меж рішень, які можна реалізувати за допомогою нейронних мереж.) Як наслідок, проектування нейронної мережі для будь-якого нетривіального застосування все ще може вимагати людських експертів, які вправні в мистецтві нейронних мереж.

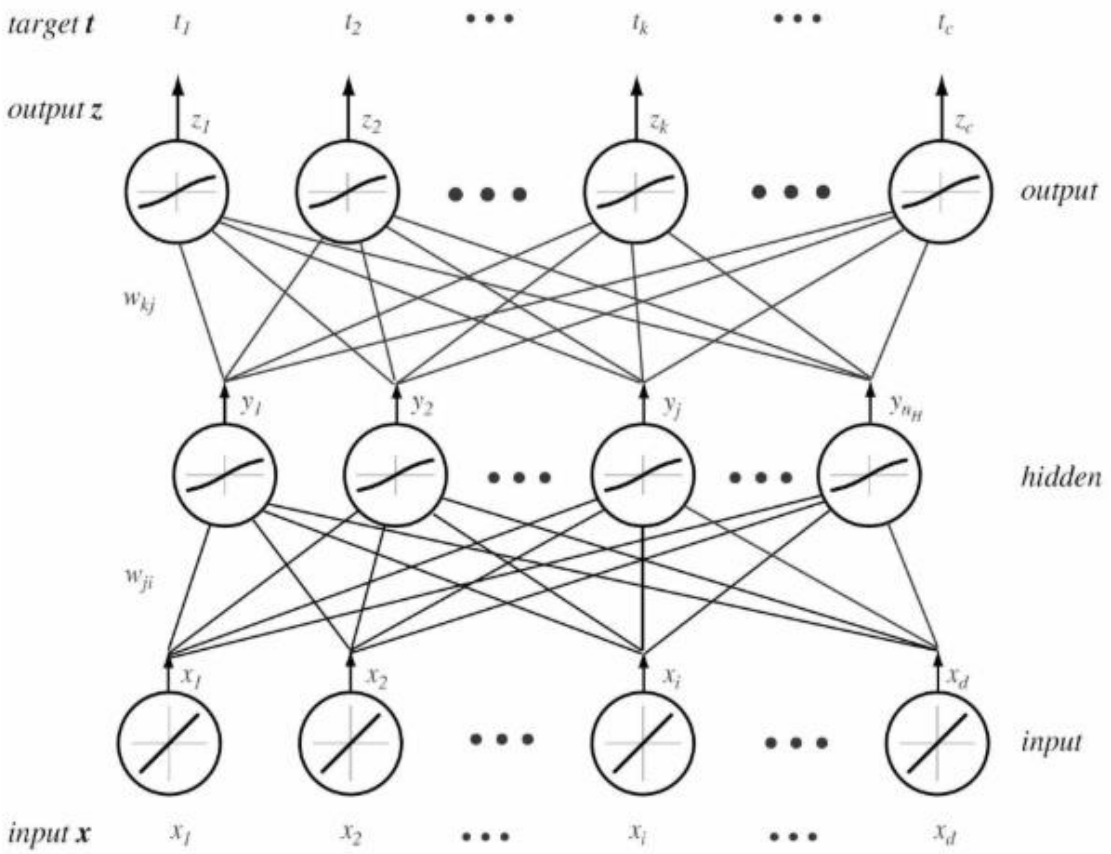

Рис. 2.10 Повністю з'єднана тришарова нейронна мережа.

Нейронну мережу було навчено виявляти вторгнення в мережу на основі аномальної поведінки з боку окремих користувачів. Цей детектор вторгнень на основі нейронної мережі, скорочено NNID, навчається ідентифікувати користувачів комп'ютерів на основі команд, які вони видають протягом дня. Наприкінці кожного дня NNID запускається для виявлення будь-яких аномалій у щоденних сесіях користувачів. Якщо аномалії виявлено, то ініціюється розслідування для діагностики причини аномалій. Система NNID базується на тришаровій нейронній мережі, в якій вхідний шар складався з 100 одиниць, прихований шар — з 30 одиниць, а вихідний шар — з 10 одиниць, по одній для кожного з десяти користувачів, які брали участь у цьому експерименті. Систему NNID було побудовано та протестовано на машині в Університеті Техасу в Остіні, де дані були зібрані з цієї машини протягом 12 днів, що призвело до 89 векторів даних. NNID було навчено на 8 випадково вибраних днях даних (65 векторів даних) і протестовано на решті 4 днів даних (24 вектори даних). В середовищі з 10 користувачами NNID продемонструвала 96% точність виявлення з рівнем помилкових тривог 7%. Ці результати підтверджують перспективність NNID як офлайн-системи моніторингу для виявлення вторгнень.

Було розроблено високо складну нейронну мережу, що складається з шести шарів, для класифікації рукописних цифр з набору даних USPS (United States Postal Service) рукописних цифр. Ця робота використовує ідею, що розпізнавання форм можна покращити шляхом виявлення та комбінування локальних ознак, і переводить цю ідею в архітектуру нейронної мережі, обмежуючи з'єднання в перших кількох шарах як локальні, за допомогою використання карт ознак. Одиниці на карті ознак обмежені виконувати ту саму операцію на різних частинах зображення. Кілька карт ознак витягують різні ознаки з одного і того ж зображення, і тому є необхідним компонентом цієї нейронної мережі. Структура нейронної мережі показана на рисунку 2.11, де кожен прихований шар позначений літерою «H». Поруч із кожною міткою «H»

позначка «m@s × s» означає, що прихований шар складається з m груп одиниць, кожна з яких розташована в площині s-bys.

Нейронна мережа містить 4635 одиниць, 98442 з'єднання та 2578 незалежних параметрів. Після 30 проходів навчання на навчальному наборі з 7291 рукописної цифри та 2549 друкованих цифр нейронна мережа досягла рівня помилок 1,1% та MSE (середнього квадратів помилок) 0,017 на навчальних даних. Під час тестування на тестовому наборі з 2007 рукописних цифр та 700 друкованих символів нейронна мережа досягла рівня помилок 3,4% та MSE 0,024. Помилки класифікації були виключно через неправильне маркування рукописних символів.

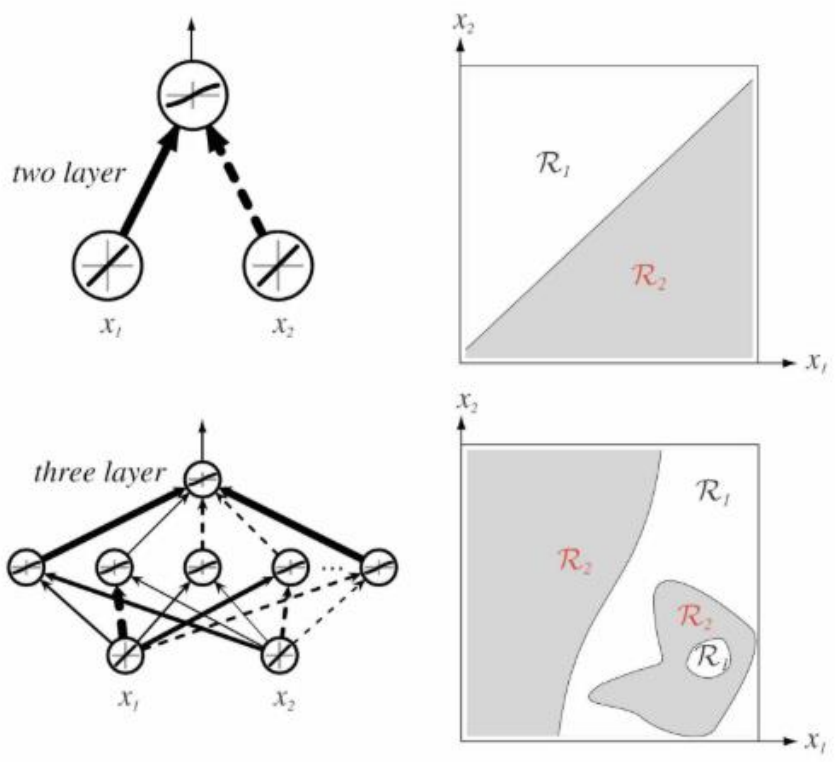

Рис. 2.11 Двошарова нейронна мережа може класифікувати тільки два лінійно роздільні класи.

Нейронні мережі використовувалися в поєднанні з кластеризацією для виявлення нових об'єктів у відеопослідовностях. Під час тестового запуску навчена нейронна мережа обробляє тестові вектори. Будь-які тестові вектори, що призводять до великої розбіжності між фактичними та цільовими виходами

нейронних мереж, асоціюються з одним або кількома новими класами. Ці тестові вектори, що відповідають одному або кільком новим класам, відкладаються в бін. Наприкінці тестового випробування дані в біні кластеризуються, і будь-який кластер, що виявляється статистично відмінним від будь-яких розподілів відомих класів, позначає новий клас.

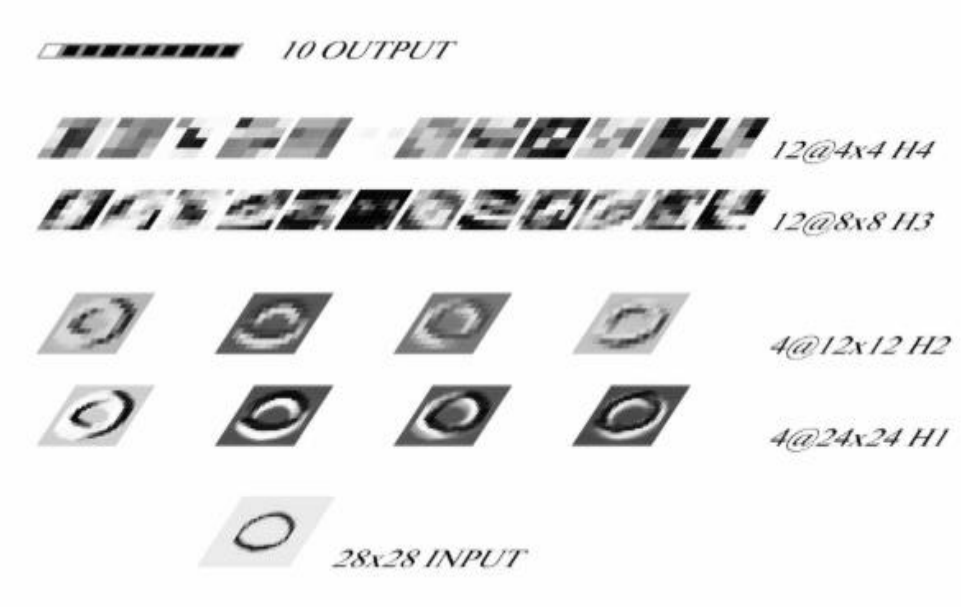

Рис. 2.12 Архітектура шестишарової нейронної мережі, що використовується для класифікації рукописних цифр із набору даних USPS.

Цей алгоритм був реалізований за допомогою тришарової нейронної мережі, яка містить 42 одиниці в першому шарі, 175 одиниць у прихованому шарі та 4 одиниці в останньому шарі. Дані зображень складаються з 3777 зразків, витягнутих з регіонів (таких як дерева, трава, небо та річка, що відображає небо або дерева). Випробування проводилися таким чином, що навчальні дані складалися з усіх класів, окрім одного, а тестові дані складалися з екземплярів виключеного класу (не використовуваного в навчанні), разом з шумовою версією навчальних даних. Результати виявлення класу «Sky» представлено в Таблиці 2.4. У цьому випробуванні дані «Sky» повністю виключено з навчання та використовуються лише для тестування. Таблиця 2.4 показує, що тестові дані класифікуються з точністю 79,6%. З 136 тестових

прикладів з даних «Sky» лише 129 прикладів були правильно призначені в бін для аналізу кластеризації. Склад кластерів (також показано в Таблиці 2.4) потім аналізувався, і «Sky» виявилося статистично достатньо відмінним, щоб бути призначеним новим класом.

Таблиця 2.4

Результати використання нейронної мережі в поєднанні з кластеризацією

	G	**T**	**S**	**Rs**	**Rt**
G	1126	213	0	4	116
T	122	596	0	2	43
S	0	1	0	6	0
Rs	1	0	0	223	0
Rt	24	25	0	0	224

Висновки до розділу

Другий розділ присвячено дискримінативним методам виявлення аномалій. Детально розглянуто методи на основі відстані та щільності (k-NN, LOF, COF), які показали високу ефективність за умови низької або середньої розмірності даних та потреби в інтерпретуації. One-Class SVM виявився універсальнішим завдяки ядровому трюку та здатності працювати з нелінійними границями, хоча потребує ретельного налаштування гіперпараметрів. Нейронні мережі (зокрема, автоенкодери та глибокі SVM) продемонстрували найкращі результати на високовимірних і складних даних, але з втратою інтерпретуваності та підвищенням вимог до обсягу навчальних даних. Загальний висновок розділу: дискримінативні методи найсильніші в задачах unsupervised і semi-supervised навчання, коли потрібна швидкість інференсу та стійкість до шуму, але їхня продуктивність суттєво падає з ростом розмірності без додаткових технік зменшення вимірності.

РОЗДІЛ 3
ГЕНЕРАТИВНІ ТА ЙМОВІРНІСНІ МЕТОДИ ВИЯВЛЕННЯ АНОМАЛІЙ

3.1. Генеративні підходи

На відміну від дискримінативних підходів, генеративні методи оцінюють p(X|C) та P(C) для кожного класу C, а потім застосовують правило Байєса для обчислення умовного за класом розподілу P(C|X):

$$P(C|X) = \frac{p(X|C)P(C)}{p(X)} \tag{3.1}$$

У цій парадигмі завдання класифікації ефективно зводиться до моделювання розподілів p(X|C) та p(C). Загалом, цей підхід вимагає оцінки більшої кількості параметрів. Оскільки ці параметри оцінюються за допомогою максимальної правдоподібності, генеративні моделі зазвичай менш оптимізовані для класифікації порівняно з дискримінативними моделями, які безпосередньо оптимізують помилку класифікації. Тим не менш, генеративні моделі користуються перевагою в спільноті модельно-орієнтованої діагностики, оскільки генеративні моделі надають уявлення про структуру системи та корисні для надання причинних пояснень спостережуваним явищам.

За умови набору спостережуваних даних, генеративна модель пов’язує спостережувані дані з прихованими змінними, які могли спричинити ці спостережувані дані. Спостережувані дані представлені векторами ознак, а приховані змінні становлять невідомі класи C. (У більшості випадків генеративна модель може містити додаткові приховані змінні, що виходять за межі класів, але корисні для покращення прогнозування між вхідними векторами X та їхніми вихідними класами C.) У цій структурі вхідний вектор ознак x інтерпретується як шумове спостереження деякого невідомого процесу

в системі. Цей невідомий процес припускається перемикатися між різними класами або режимами поведінки. Залежно від класу, під яким процес зараз працює, система генеруватиме спостереження, специфічні для цього класу. Таким чином, клас поведінки системи можна вивести через класифікацію спостережень.

Мета генеративного класифікатора — вивести клас c^* , який би згенерував, з найвищою ймовірністю, спостереження, представлене вхідним вектором ознак x. Таким чином, перед тим, як може відбутися класифікація або виявлення аномалій, повинна бути розроблена модель системи шляхом включення апріорної інформації та використання навчання без учителя на непомічених навчальних даних. Після розробки моделі M виконується вивід на M для обчислення P(C = k|X = x; M), ймовірності кожного класу k умовно вхідного вектора ознак x відносно моделі M. Класифікація зводиться до простого вибору класу з найвищою ймовірністю, тобто c* = arg max_k P(C = k|X = x; M). По суті, генеративна модель повинна захоплювати динаміку системи під кожним класом або режимом поведінки. Сама по собі генеративна модель не прогнозуватиме наявність нових класів. Замість цього, для виявлення можливого виникнення нових класів необхідно застосовувати порогову ймовірність, перевірку гіпотез або інші більш складні схеми виявлення до генеративних моделей.

У цьому розділі ми починаємо з представлення двох взаємодоповнюючих методів оцінки щільності, які зазвичай використовуються для створення генеративних моделей. Перший — це непараметричний метод, відомий як вікна Парзена, тоді як інший — це параметричний метод, відомий як моделювання сумішшю гауссівських розподілів. Друга частина цього розділу обговорює більш структуровані представлення, використовувані для генеративного моделювання. Перед тим, як обговорювати генеративні моделі часових процесів, ми пояснимо, як оцінка стану пов’язана з виявленням аномалій, оскільки оцінка стану відіграє ключову роль у виявленні аномалій часових процесів. Нарешті, ми розглянемо популярні моделі часових процесів, такі як

приховані моделі Маркова та динамічні байєсівські мережі, та надамо посилання на недавні роботи, які застосовували ці моделі для класифікації або виявлення аномалій.

3.2 Вікна Парзена

Алгоритм вікон Парзена є методом без учителя непараметричної оцінки щільності та може бути легко адаптований для класифікації. Цей алгоритм використовує функцію ядра для інтерполяції ймовірності вхідного простору, який не підтримується даними. Ця функція ядра може бути досить загальною, за умови, що вона задовольняє властивості дійсної функції щільності ймовірності.

За умови функції ядра, метод вікон Парзена підганяє цю функцію ядра навколо кожного елемента набору даних і використовує лінійну комбінацію цих ядер для апроксимації розподілу ймовірності даних. Для простоти та зручності часто використовується гауссівський розподіл як функція ядра, а ймовірність тестового вектора x апроксимується як суміш радіально симетричних гауссівських розподілів з однаковою дисперсією σ^2. Для набору даних, що складається з N d-вимірних векторів, метод вікон Парзена оцінює істинний розподіл p(x) за допомогою:

$$\widehat{p_{N(x)}} \approx \frac{1}{N} \backslash sum_{\{n=1\}}^{N} \varphi \left(\frac{\left(x - x^{\{(n)\}}\right)}{\sigma} \right) \tag{3.2}$$

З гауссівськими ядрами точки, що знаходяться далеко від тестової точки, практично нерелевантні, оскільки внесок цих точок зменшується експоненційно з квадратом відстані. Ширина ядра визначається дисперсією σ^2. Якщо σ занадто мале, то оцінена розподіл даних буде перенавчатися на даних у формі піків навколо кожної точки даних. Якщо σ занадто велике, то оцінена розподіл страждатиме від низької роздільної здатності, оскільки розподіли різних класів

можуть перекриватися та зливати окремі класи в один єдиний клас. З обмеженою кількістю даних необхідно шукати компроміс між цими двома крайнощами та емпірично фіксувати σ для мінімізації помилки класифікації. (Цей компроміс ілюструється на Рисунку 3.1) Але в разі, коли доступна необмежена кількість прикладів, можна дозволити σ → 0 та досягти асимптотично близької оцінки істинного розподілу даних. Зокрема, для всіх x, коли кількість прикладів N прямує до нескінченності, p_N (x) сходиться до p(x) в сенсі середнього квадрата, де

$$\lim_{N\to\infty} \mathbb{E}\left[p_N(\mathbf{x})\right] = p(\mathbf{x}) \quad (11)$$

$$\lim_{N\to\infty} \mathbf{var}\left[p_N(\mathbf{x})\right] = 0 \quad (12)$$

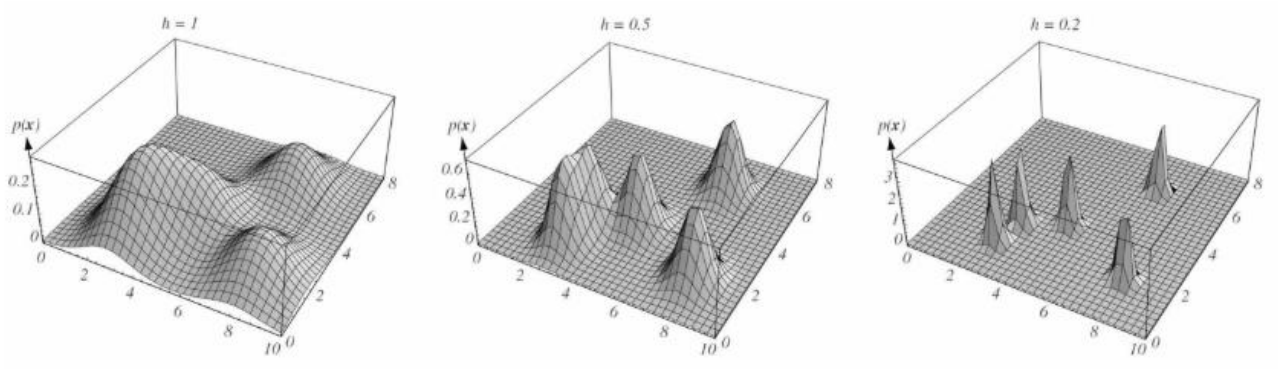

Рис. 3.1 Оцінка даних за допомогою трьох вікон Парзена на основі тих самих п'яти тестових прикладів.

Для класифікації використовується генеративна парадигма: спочатку p(X|C) оцінюється на основі даних за допомогою оцінки щільності за вікнами Парзена, а P(C) оцінюється за допомогою простого розподілу частот (якщо C є скінченним) або суб'єктивного попереднього розподілу, що відображає переконання щодо розподілу класів. (У наших рівняннях ми припускаємо використання частотного підходу для оцінки P(C).) Потім P(C|X) обчислюється за правилом Байєса, як показано в рівнянні. На цьому етапі дані повинні бути розподілені на різні класи на основі форми розподілу ймовірностей (рівняння 3.3), оціненої за методом вікон Парзена. В результаті для кожного класу k можна отримати такі оцінки :

$$p(X = x \mid C = k) \approx (1 / N\_k) \sum\_\{n \in G\_k\} \varphi ((x - x\^\{(n)\}) / \sigma)$$
$$P(C = k) \approx N\_k / N \quad (3.3)$$

де N — кількість векторів даних, з яких Nk векторів належать до класу k. Зверніть увагу, що сума для p(X = x|C = k) береться за індексами в Gk, які відповідають тим векторам даних, що належать до класу k. Іншими словами, |Gk| = Nk. Поєднання цих двох виразів дає локально зважене усереднення даних:

$$p_{N(x)} \cong \left(\frac{1}{N}\right) sum_{\{n=1\ to\ N\}varphi} \left(\frac{(x - x^n)}{sigma}\right) \quad (3.4)$$

$$= \left(\frac{1}{N}\right) su\ m_{\{n=1\ to\ N\}} \left(1 \left((2\ pi)^{\left\{\frac{d}{2}\right\}} sigma^d \right)\right) \exp \left(- \frac{\left|\left|x\ -\ x^n\right|\right|^2}{(2\ sigma a^2)} \right)$$

де я x (n) 7→ k є індикаторною функцією, яка дорівнює 1, якщо x (n) належить до класу k, і дорівнює 0 в іншому випадку. Нарешті, x присвоюється класу k з найвищою ймовірністю P(C = k|X = x).

Насправді, класифікатор Парзена можна інтерпретувати як узагальнення методу k-найближчих сусідів. У класифікаторі k-найближчих сусідів клас тестової точки x визначається більшістю голосів класів від k найближчих сусідів x. Замість того, щоб розглядати лише k найближчих сусідніх векторів, класифікатор Парзена розглядає кожен вектор у наборі даних і зважує їхні голоси за допомогою функції ядра, центрованої на тестовій точці x. Хоча метод розглядає кожен вектор даних, не кожен вектор фактично вносить вклад у більшість голосів, оскільки вектори, розташовані поза функцією ядра, матимуть вагу 0.

У границі нескінченної кількості даних оцінка розподілу даних за допомогою вікна Парзена наближається до істинного розподілу. На практиці може знадобитися багато векторів даних для розумної оцінки розподілу даних. Ця потреба в даних зростає експоненційно з розмірністю даних, обмежуючи

застосовність цього методу через його серйозні обчислювальні та вимоги до пам'яті.

Метод вікон Парзена був успішно застосований для виявлення новизни в виявленні вторгнень. У цій роботі виявлення новизни формулюється як перевірка гіпотези, де логарифмічна правдоподібність тестового вектора L(x) та логарифмічна правдоподібність довільного вектора y, відібраного з нормального класу, L(y), порівнюються. Якщо $P(L(y) \leq L(x)) > \psi$ для деякого рівня помилкових тривог $\psi \in (0, 1)$, то x позначається як належний до нормального класу. В іншому випадку x позначається як аномальний. Дослідження використовувало набір даних KDD Cup, який містить стандартний набір даних для аудиту, включаючи широкий спектр вторгнень, симульованих у військовому мережевому середовищі.

Таблиця 3.1

Порівняння детектора вторгнень.

Метод	**TAR Нормальний**	**Тип вторгнення 1**	**Тип вторгнення 2**	**Тип вторгнення 3**	**Тип вторгнення 4**
На основі Парзена	97.38%	99.17%	96.71%	93.57%	31.17%
Переможець KDD	99.45%	87.73%	97.69%	26.32%	10.27%

Дослідження порівнювало запропоновану систему виявлення вторгнень на основі Парзена з переможцем KDD Cup, використовуючи швидкість істинного прийняття (TAR) та швидкість істинного виявлення (TDR) як метрики продуктивності. TAR вимірює відсоток нормальних екземплярів у тестовому наборі, які були правильно класифіковані як нормальні, тоді як TDR вимірює відсоток вторгнень у тестовому наборі, які були правильно класифіковані як вторгнення. Для оцінки розподілу нормального класу використовувалися 3000 випадково згенерованих прикладів як навчальні дані.

Емпіричні результати показано в Таблиці 3.1, де детектор на основі Парзена перевершив переможця KDD Cup у виявленні вторгнень, з подібними або значно вищими значеннями для TDR. На більш складних типах вторгнень (типи 3 та 4, показані в Таблиці 3.1) переможець KDD Cup показав погані результати, тоді як детектор на основі Парзена зміг домінувати з чітким запасом продуктивності.

3.3 Суміш гаусівських розподілів

Гаусівська суміш щільності — це d-вимірний розподіл ймовірностей, який визначається зваженою сумою M компонентів:

$$p(X|\theta) = \sum_{\{m=1\}}^{M} p(X|m)P(m) \tag{3.5}$$

де кожен компонент p(X|m) є d-вимірним багатовимірним гаусовим розподілом із середнім значенням μm та коваріацією Σm:

$$p(X|m) = 1\left((2\pi)^{\left\{\frac{d}{2}\right\}}|\Sigma_m|^{\left\{\frac{1}{2}\right\}} \right) \exp\left(-\frac{1}{2}(X - \mu_m)^T \Sigma_m^{\{-1\}(X-\mu_m)} \right) \tag{3.6}$$

а P(m) є коефіцієнтом суміші, де P(m) ≥ 0 для m = 1, ..., M і PM m=1 P(m) = 1. Таким чином, щільність гауссової суміші параметризується параметрами всіх її компонентів:

$$\theta = \{\mu_1, \dots, \mu_M, \Sigma_1, \dots, \Sigma_M, P(1), \dots, P(M)\} \tag{3.7}$$

Для класифікації дані з кожного класу k представлені щільністю суміші гауссівських розподілів, де p(X|C = k) △ = p(X|θ_k) (як подано в Рівнянні 3.7). Параметри θ_k зазвичай оптимізуються в сенсі максимальної правдоподібності

за допомогою алгоритму очікування-максимізації, або в байєсівському сенсі за допомогою методів ланцюгів Маркова Монте-Карло для відбору параметрів з апостеріорного розподілу. Ймовірність класу P(C) часто припускається бути дискретним розподілом або багатовимірним гауссівським для обчислювальної ефективності, в якому параметри P(C) зазвичай навчаються за допомогою максимальної правдоподібності або байєсівських технік. Тестовий вектор x призначається до найбільш ймовірного класу c ∈ {1, ..., K} таким чином.

Моделі суміші гауссівських розподілів були успішно застосовані для ідентифікації мовця та активного навчання для виявлення аномалій. Використання щільностей суміші гауссівських розподілів для моделювання ідентичності мовця було мотивоване двома основними причинами: (1) гауссівські компоненти можуть ефективно моделювати акустичні класи мовця та (2) суміші гауссівських розподілів можуть використовуватися для моделювання будь-яких довільних розподілів, що додає їм універсальності. У цьому дослідженні завдання класифікації полягало в правильній класифікації тестового сегмента мовлення та ідентифікації мовця, від якого був згенерований сегмент мовлення. Експерименти були проведені на підмножині бази даних мовлення KING, яка містить зразки розмовного мовлення від 51 чоловічого мовця. Для кожного мовця є 10 незалежних розмов приблизно по 45 секунд. Метрикою продуктивності взято відсоток правильної ідентифікації. Результати дослідження представлено на Рисунку 3.2 та Рисунку 3.3. На Рисунку 3.2 продуктивність ідентифікації мовця нанесено проти кількості компонент суміші в моделі суміші гауссівських розподілів для тестових сегментів та навчальних сегментів різної довжини. Емпіричні результати показують, що понад 16 компонентів спостерігається лише маргінальне покращення продуктивності. Крім того, зі зменшенням кількості навчальних даних стає більш критичним вибір оптимальної кількості компонент для суміші гауссівських розподілів. На Рисунку 3.3 продуктивність ідентифікації мовця нанесено проти довжини тестового сегмента мовлення для різної кількості мовців. Лівий графік представляє результати для чистого мовлення, а правий —

для телефонного мовлення. Для чистого мовлення майже ідеальна ідентифікація досягається, коли тестовий сегмент мовлення становить 15 секунд. Однак для телефонного мовлення, через низьке співвідношення сигнал/шум аудіоданих, спостерігається значне погіршення продуктивності.

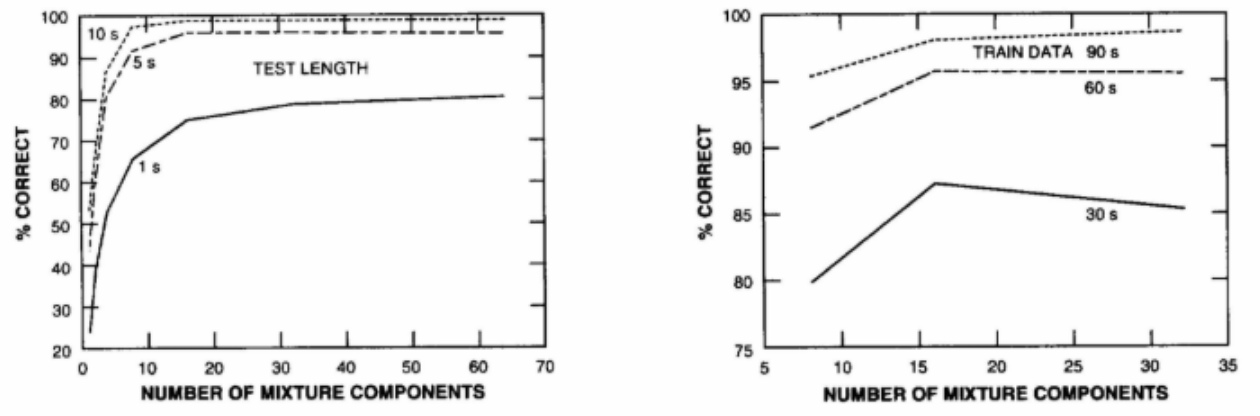

Рис. 3.2 Ефективність ідентифікації мовця як функція кількості компонентів у гаусовій змішаній моделі.

Окрім ідентифікації мовця, гаусові змішані моделі також застосовуються в рамках активного навчання для виявлення рідкісних і корисних аномалій. Підхід активного навчання, запропонований, передбачає, що розподіл даних є надзвичайно асиметричним у бік нормального класу і що для пристосування даних можна використовувати змішану модель. Процес активного навчання проходить у кілька етапів, під час яких комп'ютер намагається вивчити модель даних на основі невеликої кількості позначених навчальних прикладів разом із великою кількістю непозначених навчальних прикладів. Потім він визначає невелику кількість складних прикладів і просить користувача надати позначки для цих складних прикладів. Користувач позначає ці приклади і додає їх до колекції позначених дані, і цикл повторюється. Ця структура є гнучкою в тому, що мітки є необмеженими, і користувач має свободу додавати, видаляти та змінювати класи за бажанням. Як наслідок, ця система дозволяє адаптивне навчання динамічного набору даних.

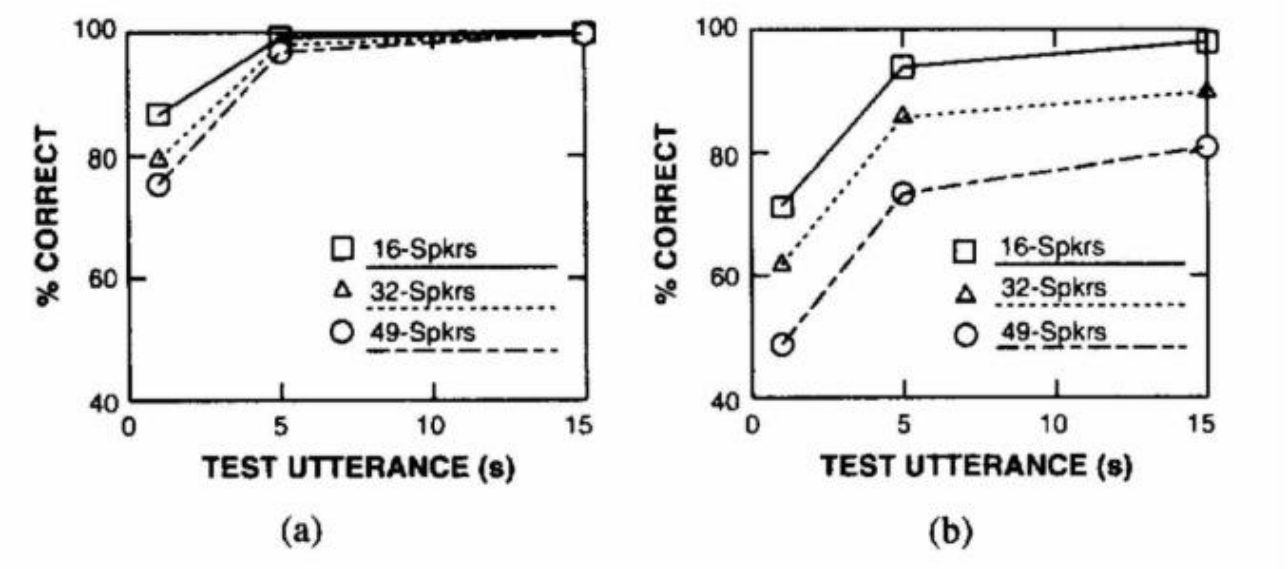

Рис. 3.3 Ефективність ідентифікації мовця як функція довжини тестового сегмента для популяцій розміром 16, 32 та 49 мовців.

Ця робота представляє кілька методів вибору, за допомогою яких вибираються непомічені екземпляри для процесу зворотного зв'язку. Було розглянуто такі критерії вибору підказок:

• LOWLIK: Вибір екземплярів з низькою правдоподібністю, тобто даних, на яких модель працює найгірше. Зокрема, екземпляри ранжуються в зростаючому порядку правдоподібності моделі, і вибираються ті з низькими правдоподібностями.

• AMBIG: Вибір неоднозначних екземплярів, тобто даних, щодо яких комп'ютер має найменшу впевненість. Зокрема, екземпляри ранжуються в спадному порядку ентропії, і вибираються ті з високою ентропією.

• MIX-AMBIG-LOWLIK: Гібридний підхід двох схем вибору, описаних вище.

• INTERLEAVE: Вибір точок на основі ранжування з перспективи лише однієї компоненти, замість суміші компонент.

Експерименти проводилися на синтетичних даних та реальних даних. Останній підхід INTERLEAVE, здається, працює найкраще як на синтетичних, так і на реальних випадках. Виходячи з визначення моделі суміші, цей підхід активного навчання дозволяє кожній компоненті номінувати свої улюблені

запити. Емпіричні результати показують, що цей метод добре працює в присутності шумових даних та надзвичайно рідкісних аномалій.

Рисунки 3.4 та 3.5 показують криві навчання для різних реальних наборів даних, як охарактеризовано в Таблиці 3.2. Криві навчання показують відсоток виявлених класів як функцію кількості підказок, запрошених комп'ютерною системою.

Таблиця 3.2

Властивості реальних наборів даних

Набір даних	**# вимірів**	**# записів**	**# класів**	**найменший клас**	**найбільший клас**
ABALONE	7	4177	20	0.34%	16%
KDD	33	500000	19	0.002%	21.6%
EDSGC	26	1439526	7	0.002%	76%
SDSS	22	517371	3	0.05%	50.6%

Експерименти показують, що цей підхід надзвичайно сильно зменшує кількість прикладів, які людина повинна позначити перед тим, як їх буде передано автоматичному алгоритму навчання. Фактично користувачеві достатньо позначити лише одну-дві сотні прикладів, і вже тоді з величезного набору даних йому починають пред’являтися рідкісні аномалії.

Для забезпечення більшої гнучкості гаусової сумішевої моделі автори запропонували узагальнену гаусову сумішеву модель (generalized Gaussian mixture model), окремі компоненти якої можуть відхилятися від нормального розподілу, щоб представляти платикуртичні та лептокуртичні розподіли.

З метою порівняння узагальнену гаусову сумішеву модель зіставляли зі стандартною гаусовою сумішевою моделлю та алгоритмом кластеризації k-середніх під час ненаглядованого класифікування набору даних, зображеного на рис. 3.5. Кількість класів вважалася відомою, а завдання полягало у навчанні параметрів моделі та класифікації даних. Помилка класифікації становила: – 4,0

% ± 0,5 % — для узагальненої гаусової сумішевої моделі, – 5,5 % ± 0,3 % — для стандартної гаусової сумішевої моделі, – 18,3 % — для алгоритму k-середніх.

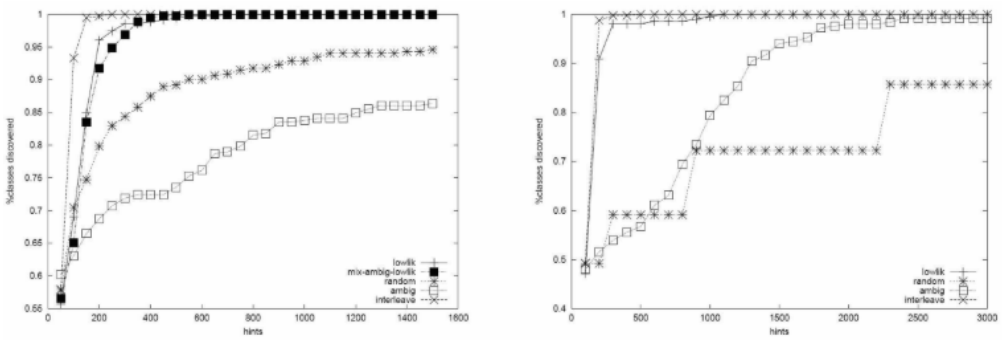

Рисунок 3.4 Криві навчання для наборів даних ABALONE (ліворуч) та KDD.

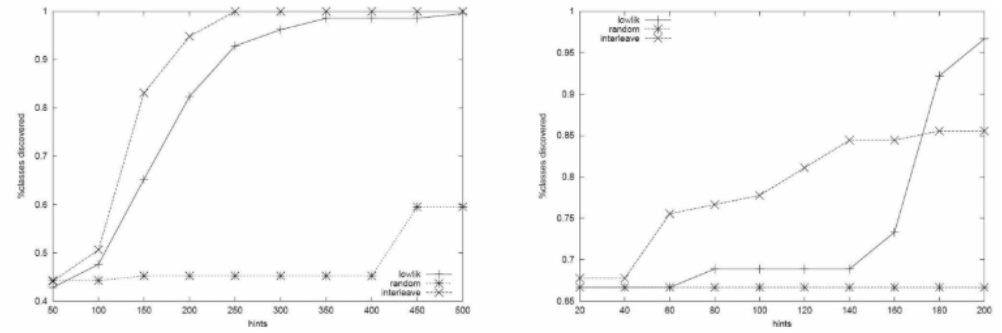

Рис. 3.5 Криві навчання для наборів даних EDSGC (ліворуч) та SDSS (праворуч).

Дослідження показало, що узагальнена гаусова сумішева модель не поступається, а часто й перевершує стандартну гаусову сумішеву модель, особливо на даних, що мають негаусовий характер і містять велику кількість викидів.

3.4 Оцінка стану проти виявлення аномалій

Перш ніж переходити до обговорення генеративних моделей для виявлення аномалій у часовимчасових процесах, корисно чітко пов’язати оцінку стану (state estimation) з виявленням аномалій (anomaly detection).

У генеративній парадигмі ймовірнісні моделі кодують динаміку еволюції системи в часі. У момент часу τ стан системи позначається як Sτ, а вся історія спостережень до цього моменту (включно) — як $y_{1:\tau} = \{y_1, y_2, \ldots, y_\tau\}$.

Апостеріорний розподіл $p(S_\tau \mid y_{1:\tau})$ виражає наші поточні переконання про справжній стан системи в момент τ на основі всієї доступної на цей момент інформації.

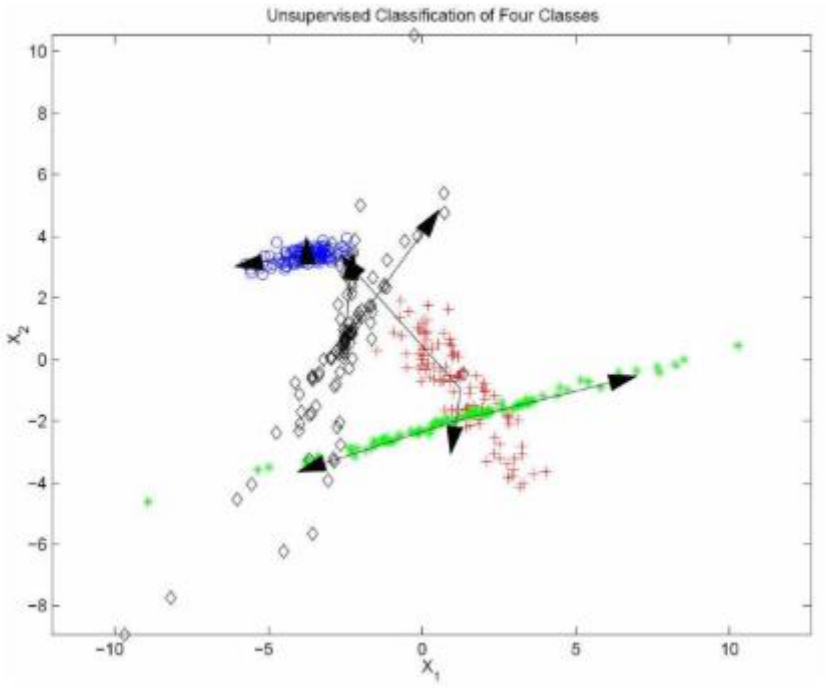

Рис. 3.6 Приклад даних, використаних у завданні неконтрольованої класифікації

Дані в кожному класі були згенеровані за допомогою випадкового розподілу. Оцінка стану є двокроковим процесом, який реалізується шляхом байєсівського оновлення.

Маючи апостеріорний розподіл попереднього кроку $p(S_{t-1} \mid y_{1:t-1})$, крок прогнозування (state prediction) полягає в обчисленні прогнозного розподілу $p(S_t \mid y_{1:t-1})$, який є найкращою оцінкою стану S_t на основі всіх спостережень до моменту $t-1$ включно.

Коли стає доступним нове спостереження y_t у момент t, ця інформація використовується для оновлення прогнозного розподілу, в результаті чого отримуємо апостеріорний розподіл p(S_t | y_{1:t}) на момент t.

Таким чином, два кроки — прогнозування (prediction) та корекція (correction) — утворюють повний цикл оцінки стану, як зображено на рис. 3.6.

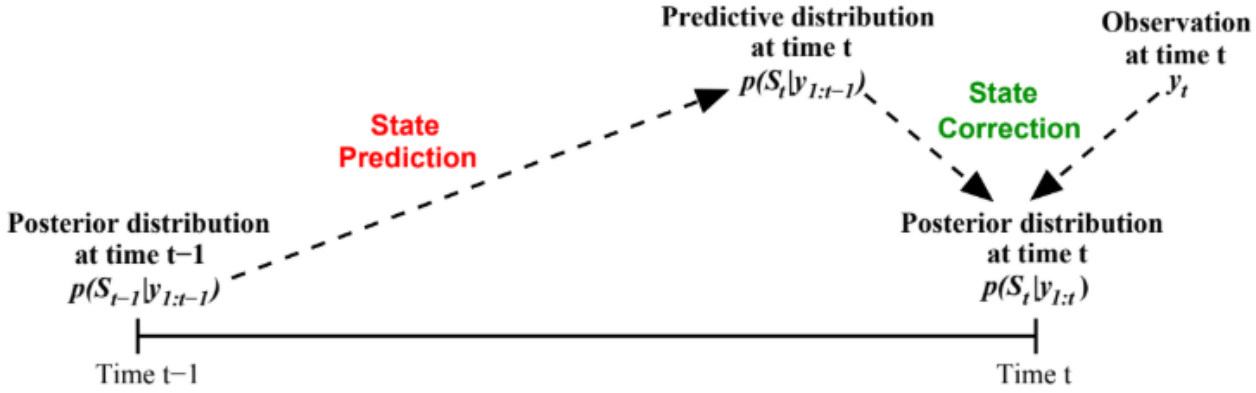

Рис. 3.7 Ітеративний процес оцінки стану.

Крок прогнозування стану (state prediction) полягає в обчисленні прогнозного розподілу $p(S_t \mid y_{1:t-1})$ — найкращої оцінки стану в момент t лише на основі історії спостережень від 1 до t−1. Крок корекції стану (state correction) полягає в оновленні цього прогнозного розподілу $p(S_t \mid y_{1:t-1})$ з урахуванням нового спостереження y_t, в результаті чого отримується апостеріорний розподіл $p(S_t \mid y_{1:t})$, який вже враховує найсвіжішу інформацію.

І класифікація, і виявлення аномалій зазвичай зводяться до певних математичних операцій над апостеріорним розподілом $p(S_t \mid y_{1:t})$.

Класифікація: вибір класу (який зазвичай є частиною стану) з найбільшою імовірністю.

Виявлення аномалій: порогова обробка апостеріорного розподілу з подальшим віднесенням станів з низькою імовірністю до зовсім нового («аномального») класу.

Окрім безпосереднього використання апостеріорного розподілу, додатковим критерієм може слугати значна різниця між прогнозним

розподілом $p(S_t \mid y_{1:t-1})$ та апостеріорним $p(S_t \mid y_{1:t})$. Якщо $p(S_t \mid y_{1:t})$ суттєво відрізняється від $p(S_t \mid y_{1:t-1})$, це зазвичай свідчить про те, що в момент t стався перехід стану, і ефект цього переходу був зафіксований спостереженням y_t.

Використовуючи розбіжність між прогнозним і апостеріорним розподілами як додатковий детектор, можна точніше визначати момент переходу системи з нормального стану в аномальний (і навпаки).

Отже, генеративна класифікація та виявлення аномалій значною мірою спираються на оцінку стану. У свою чергу, ефективність оцінки стану визначається:

класом генеративних моделей, які використовуються для опису системи;

типом алгоритмів виведення (inference), які застосовуються для роботи з цими моделями.

3.5 Приховані марковські моделі

Прихована марковська модель (Hidden Markov Model, HMM) — це графова модель, яка описує марковський процес, стан S_t якого є прихованим і може бути оцінений лише через (зашумлені) спостереження Y_t.

HMM припускає, що система еволюціонує згідно з марковським процесом першого порядку, в якому поточний стан залежить лише від попереднього:

$$p(S_t|S_{\{0:t-1\}}) = p(S_t|S_{\{t-1\}}),\ t = 1,2 \tag{3.8}$$

і, додатково, спостереження залежать лише від поточного стану:

$$p(Y_t|S_{\{0:t\}}) = p(Y_t|S_t),\ t = 1,2,\dots \tag{3.9}$$

Для спрощення більшість HMM вважають ймовірності $p(S_t \mid S_{t-1})$ та $p(Y_t \mid S_t)$ стаціонарними, тобто такими, що не змінюються з часом:

За цих припущень HMM повністю характеризується двома компонентами:

$$p(S_t|S_{\{t-1\}}) = p(S_{\{s+t\}|\, S_{\{s+t-1\}}}),\ s \geq 0 \qquad (3.10)$$

$$p(Y\_t \mid S\_t) = p(Y_{\{s+t\}|\, S_{\{s+t\}}}),\ s \geq 0$$

моделлю переходів $p(S_t \mid S_{t-1})$,

моделлю спостережень $p(Y_t \mid S_t)$.

Таким чином, стан еволюціонує ймовірнісно згідно з моделлю переходів, а спостерігається через зашумлені вимірювання згідно з моделлю спостережень (рис.3.8).

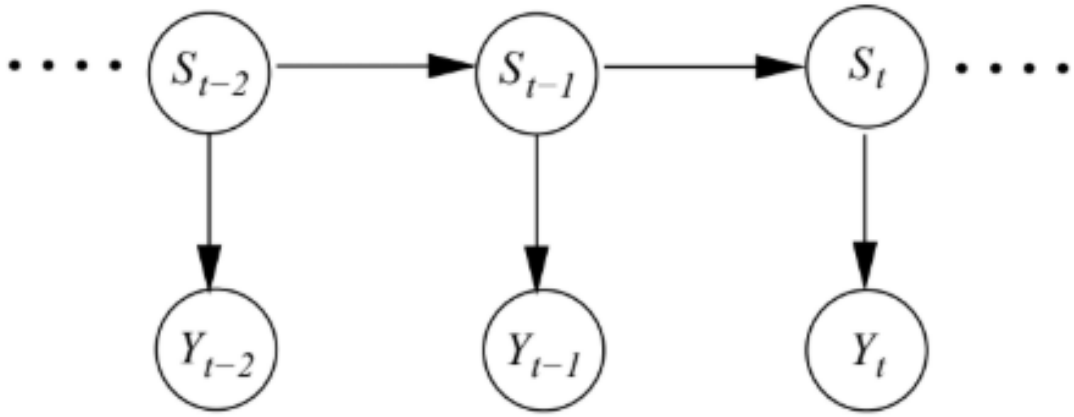

Рис. 3.8 Прихована марковська модель.

Стан S_t еволюціонує ймовірнісно згідно з $p(S_t \mid S_{t-1})$ (стрілка від S_{t-1} до S_t показує причинно-наслідкову залежність). У свою чергу, стан S_t спостерігається через зашумлені вимірювання Y_t згідно з $p(Y_t \mid S_t)$ (стрілка від S_t до Y_t).

HMM добре вивчені, для них існують усталені алгоритми виведення (inference) та навчання параметрів.

Для задач класифікації до скінченностанцевих HMM застосовують алгоритм Вітербі, який знаходить найімовірнішу послідовність прихованих станів, що могла згенерувати дану послідовність спостережень. Оскільки ознаки вхідних даних можна пов'язати зі спостереженнями Y_t, а невідомі класи — з прихованими станами S_t, алгоритм Вітербі дозволяє класифікувати будь-який вхідний вектор до найбільш імовірного класу.

Застосування HMM для класифікації та виявлення аномалій довгий час було популярним у спільноті систем виявлення вторгнень (intrusion detection).

У роботі проведено емпіричне порівняння ефективності динамічних і статичних моделей для виявлення вторгнень. Зокрема, досліджувалась їхня продуктивність на даних програм (системні виклики) та даних користувачів (команди shell).

Динамічна модель — це HMM, натренована на даних нормальної поведінки. Спостереження з низькою правдоподібністю відносно моделі вважаються вторгненнями.

Статична модель — оцінка частотного розподілу подій, який характеризує нормальну поведінку. Критерієм вторгнення є перевищення порогу крос-ентропії між поточним і модельним частотними розподілами.

Ключова відмінність: статична модель ігнорує порядок подій, тоді як динамічна (HMM) його враховує.

Виявлення новизни формулювалось як перевірка статистичних гіпотез, а метриками виступали true acceptance rate (TAR) і true detection rate (TDR).

На першому наборі даних динамічні моделі (HMM) показали кращі результати, ймовірно завдяки сильним часовим залежностям між системними викликами.

На другому наборі статичні моделі випередили динамічні. Отже, автори дійшли висновку, що часові залежності в послідовностях shell-команд можуть бути неінформативними або навіть шумовими ознаками, що погіршують виявлення. Таким чином, HMM доцільно застосовувати лише до задач із вираженими часовими залежностями.

Цю ідею підтримує робота, де HMM застосовувались для виявлення багатостадійних мережевих атак. Такі атаки розтягнуті в часі й складаються з багатьох кроків, у яких дії можуть чергуватись або бути випадковими/обманними. Можливість моделювати довільний порядок дій дозволяє HMM ефективно описувати поведінку зловмисника, який намагається маскувати свої дії.

Завдяки послідовній природі домену використання часової моделі типу HMM є інтуїтивно виправданим: порядок дій часто містить критичну інформацію про характер атаки.

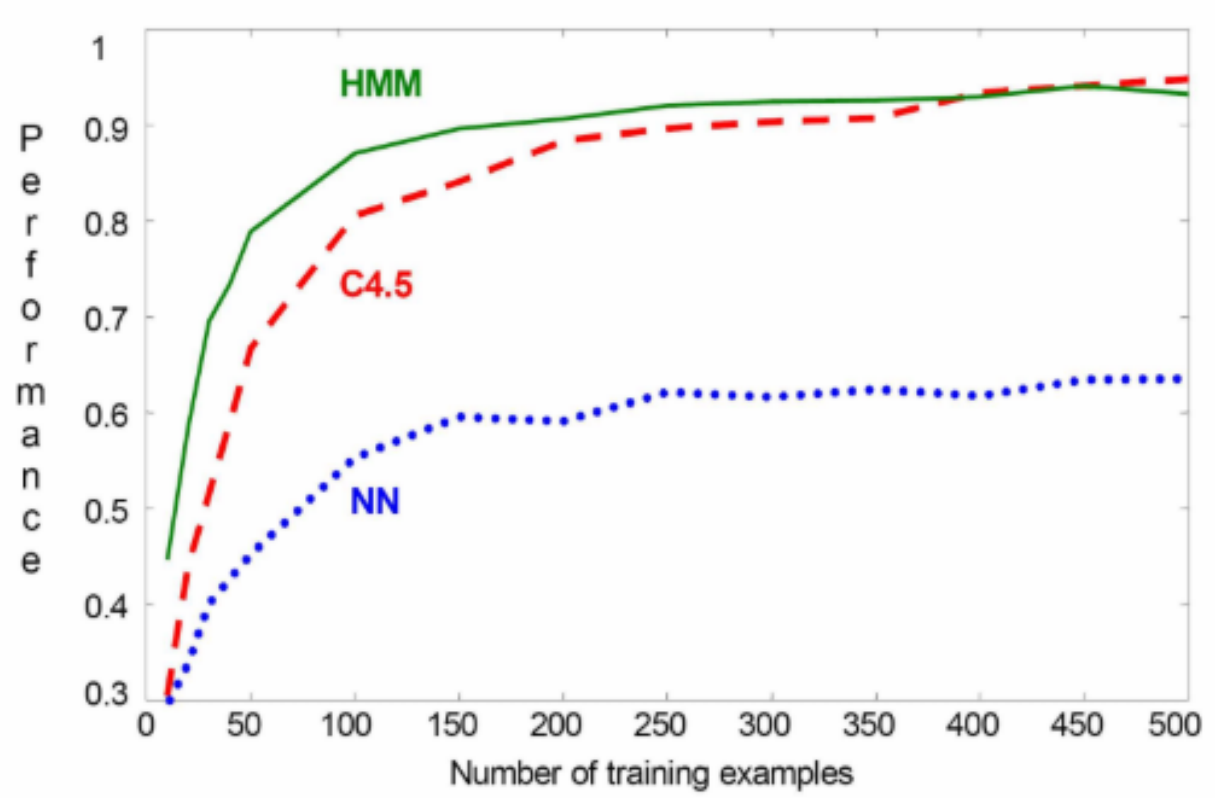

Рис. 3.9 Ефективність виявлення, визначена як частка тестових прикладів

Експерименти проводились на внутрішніх даних мережевих сенсорів; дані напів-автоматично категоризовувались експертами. Кожен навчальний приклад — це послідовність попереджень (alerts), впорядкована за часом, за 24-годинний період. Порівнювались HMM, дерево рішень і нейронна мережа (рис. 3.9). Метрикою була частка правильно класифікованих тестових прикладів.

Матриця плутанини відображає ймовірність того, що тестовий приклад з певного класу буде віднесено до Таблиця 3.3.

Таблиця 3.3

Наведено статистичні показники точності.

Параметр	**Значення**
Навчальні приклади	300
Тестові приклади	100
Продуктивність (Accuracy)	0.9255
Кількість ітерацій вибірки	100

Матриця плутанини відображає ймовірність того, що тестовий приклад з певного класу буде віднесено до Таблиця 3.3.

Діагональ матриці плутанини показує ймовірність того, що тестовий приклад буде правильно віднесений до свого класу. Точність, відтворюваність і F-показник є функціями істинних позитивних результатів (tp), помилкових позитивних результатів (fp) і помилкових негативних результатів (fn) де tp відповідає кількості правильно виявлених вторгнень, fp відповідає кількості помилково виявлених вторгнень, а fn відповідає кількості вторгнень, пропущених системою виявлення.

Параметр α був встановлений на 0,5, щоб забезпечити рівну вагу між точністю та відтворюваністю. Загалом, значення точності, відтворюваності та F-показника демонструють відносно хороші результати, за винятком класів, для яких дані навчання та тестування є недостатніми.

Щоб кількісно оцінити значення додаткових навчальних даних, були проведені експерименти з різною кількістю навчальних прикладів, а ефективність, виражена в ROC-кривих і площі під цими кривими, представлена на рисунках 3.10 і 3.11. Як і очікувалося, можна побачити, що виявлення загалом покращується із збільшенням кількості навчальних даних.

3.6 Динамічні байєсівські мережі

Динамічні байєсівські мережі (Dynamic Bayesian Networks, DBNs) — це компактні представлення марковських процесів. Як і HMM, DBN є графовою моделлю, в якій змінні стану подано вузлами, а причинно-наслідкові впливи між змінними — спрямованими стрілками між вузлами.

На відміну від HMM, де параметри переходу та спостережень визначені над цілим (монолітним) станом S_t, в DBN кожному вузлу S_t (або його компоненті) відповідає власна умовна імовірнісна таблиця $p(S_t \mid Pa(S_t))$, яка описує імовірність цієї змінної за її батьками $Pa(S_t)$ (батьками змінної є ті змінні стану, від яких вона безпосередньо залежить).

Таким чином, DBN узагальнюють HMM, експлуатуючи умовні незалежності між змінними стану, щоб представляти модель переходів і модель спостережень у факторізованій (розкладеній) формі — як добуток нижчевимірних імовірнісних розподілів, кожен з яких описує лише локальну динаміку:

$$p(S_t|S_{\{t-1\}}) = \prod\_i p(S_{\{i,t\}}|_{\setminus\{Pa\}(S_{\{i,t\}})})$$
$$p(Y\_t \mid S\_t) = \prod\_j p(Y_{\{j,t\}}|_{\setminus\{Pa\}(Y_{\{j,t\}})}) \quad (3.11)$$

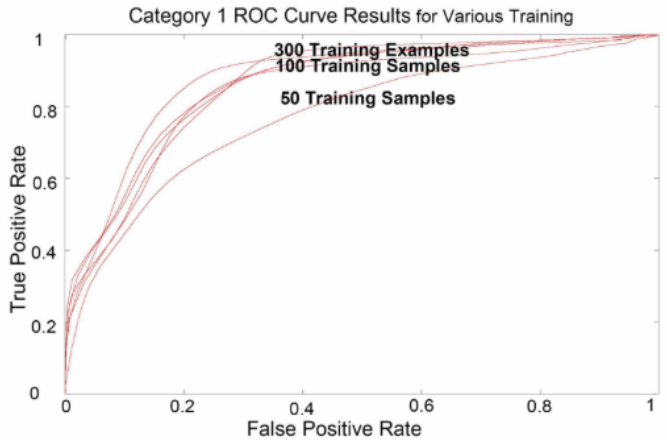

Рис. 3.10 Ефективність виявлення, виражена у вигляді ROC-кривих, для навчальних даних різного розміру.

Представлені результати отримані на основі тестових даних для класу 1.

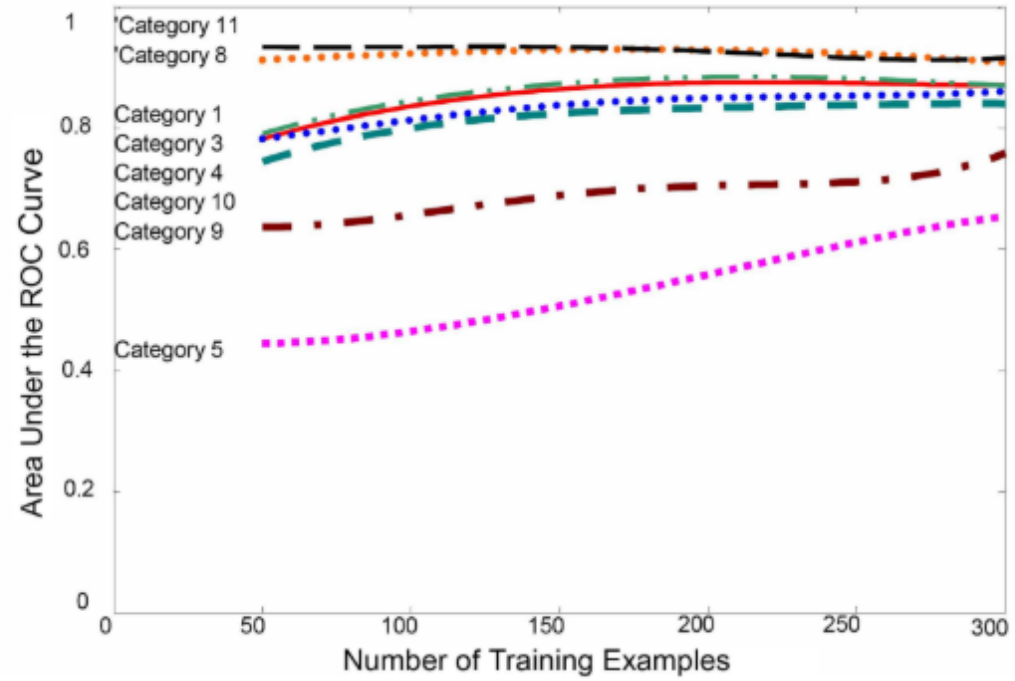

Рис. 3.11 Ефективність виявлення, виражена як площа під кривими ROC, для навчальних даних різних розмірів

де i — індекс по змінних стану, а j — індекс по змінних спостережень.

Використання DBN дуже поширене для моделювання часових процесів у задачах виявлення та діагностики несправностей (fault detection and diagnosis). У цих задачах стан S_t зазвичай розширюють, додаючи змінні несправностей Z_t, які позначають відсутність або наявність тих чи інших дефектів. Оскільки наявність несправності впливає на поведінку системи, цей причинно-наслідковий зв’язок графічно зображається стрілкою від змінних несправностей Z_t до системних змінних V_t .

Зверніть увагу: у цій парадигмі змінні спостережень Y_t відіграють роль вхідних ознак X, а змінні несправностей Z_t тепер представляють вихідні класи. Змінні V_t можна розглядати як допоміжні випадкові змінні, що моделюють інші аспекти системи й допомагають прояснити зв’язок між $Y \leftrightarrow Z$.

У цьому прикладі змінні несправностей Z_t — дискретні (позначені квадратними вузлами), тоді як системні змінні змінні V_t та змінні спостережень Y_t — неперервні (круглі вузли). V_t спостерігається через Y_t, вимірювання якого можуть спотворюватись у присутності несправності, тому є стрілка від Z_t до Y_t.

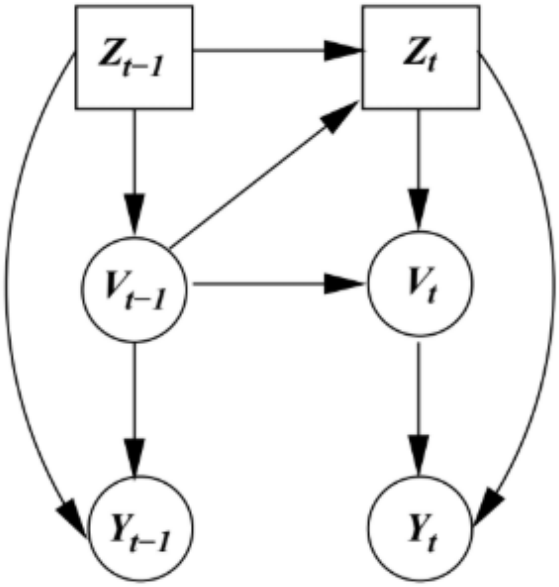

Рис. 3.12 Приклад DBN гібридної системи, що містить змінні несправностей.

У цій парадигмі моделювання стан включає як системні, так і несправні змінні, тобто $S_t = \{V_t, Z_t\}$. За історією значень $y_{1:t}$ (спостережуваних змінних) стандартними методами оцінки стану обчислюють розподіл $P(S_t \mid y_{1:t})$. Далі з $P(Z_t \mid y_{1:t})$ виділяють найбільш імовірні стани несправностей z і передають їх аналітику для остаточної діагностики системи.

У рамках цієї парадигми успішно застосували гібридні DBN (змішані дискретно-неперервні стани) для моделювання промислових установок і діагностики несправностей. У роботі було побудовано DBN-модель системи з кількох послідовно з’єднаних резервуарів на основі специфікацій, заданих темпоральним причинно-наслідковим гра-фом. Модель містила змінні несправностей трьох типів:

• Вимірювальні несправності (measurement faults) — виникають при відмові датчика, через що вимірювання стають надмірно зашумленими. • Розривні несправності (burst faults) — розрив труби, внаслідок чого її опір різко змінюється до невідомого значення.

• Дрейфові несправності (drift faults) — поступова деградація труби через нормальний знос, коли опір труби повільно відхиляється від каліброваного значення.

Система складається з п’яти резервуарів, з’єднаних послідовно трубами. Перетікання рідини між резервуарами відбувається спонтанно, коли рівень в одному резервуарі перевищує висоту з’єднувальної труби (схема — верхня частина рис. 3.13). Фрагмент DBN-моделі системи показаний на рис. 3.12: там видно зв’язки між неперервними системними змінними та дискретними змінними несправностей (D — burst/drift faults, E — measurement faults) для підсистеми з двох резервуарів. Для несправностей, що зберігаються в часі (наприклад, дрейфові), змінна D_t залежить від свого попереднього значення D_{t-1}, що графічно зображено стрілкою від D_{t-1} до D_t.

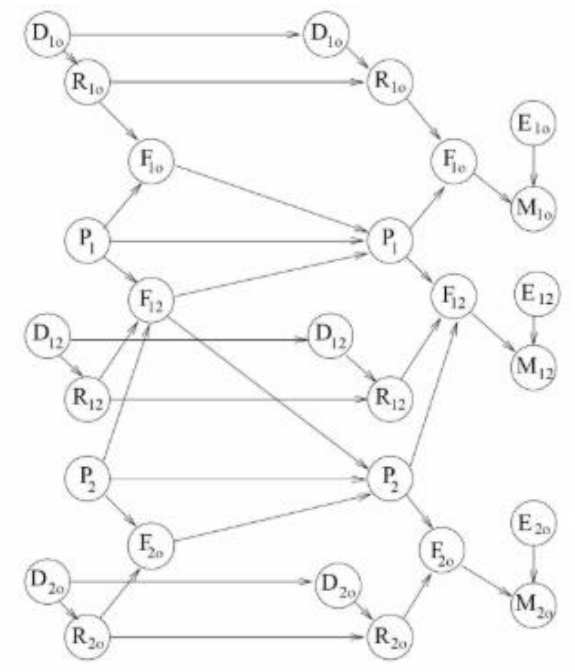

Рис. 3.13 DBN двобалонної системи. Змінні несправності позначаються літерами D і E.

У будь-який момент часу кількість можливих комбінацій несправностей становить 2^{27}. Через це на DBN-моделі застосовувалось апроксимативне виведення (approximate inference): модель розбивалась на 5 підсистем, по одній на кожен резервуар.

Водночас для виявлення миттєвих несправностей, прямі ефекти яких стають видимими не відразу, а лише з малою затримкою, знадобилась процедура згладжування (smoothing). У цій процедурі для виявлення несправностей використовується не $P(Z_t \mid y_{1:t})$, а $P(Z_t \mid y_{1:t+t})$. Ідея полягає в тому, що спостереження, які з'являються через τ кроків після виникнення несправності Z_t, дають додаткові докази її наявності, що суттєво підвищує ймовірність правильного виявлення. Емпіричні результати підтверджують цю інтуїцію: на рис. 3.14 показано вражаючу точність оцінки стану прихованої змінної «Conductance» (провідність). Наводяться лише результати оцінки стану, оскільки якість оцінки стану безпосередньо пов'язана з якістю виявлення несправностей: якщо несправність не була виявлена вчасно, це одразу проявляється в помилках оцінки стану.

Експеримент проводився на повній системі з 5 резервуарів; спостереження генерувались за заздалегідь створеним сценарієм, у якому в проміжку між моментами t=5 і t=25 вводилось багато різних і одночасних несправностей.

Для ілюстрації переваг згладжування розглянемо конкретну. О т=5 до змінної R_{23} (опір труби між резервуарами 2 і 3) була введена дрейфова несправність. У момент виникнення ймовірність дрейфу становила лише 0,012 %. На кроці t=6 ймовірність стрибнула до 71,7 %, а після процедури згладжування зросла до 99,9 %. Алгоритм правильно виявив цю несправність і утримував високу ймовірність до тих пір, поки ефекти дрейфу не зникли.

Отже, дослідження демонструє, що DBN корисні для діагностики несправностей, а поєднання процедури згладжування з апроксимацією підсистемами дозволяє успішно відстежувати складну систему навіть за невеликої кількості вимірювань і за наявності несправностей.

Окрім діагностики несправностей, DBN також застосовувались у новій галузі — виявленні порушень конфіденційності (privacy intrusion detection). Порушення конфіденційності — це незаконне розголошення або будь-яке неналежне використання приватних даних людьми, яким ці дані довірені (наприклад, співробітниками податкової служби або медичної лабораторії). Зі зростанням інформаційних технологій більшість організацій збирають приватну інформацію про своїх клієнтів, і саме організація несе відповідальність за моніторинг і виявлення можливого зловживання цими даними своїми співробітниками. Виявлення порушень конфіденційності реалізується шляхом порівняння дій агента

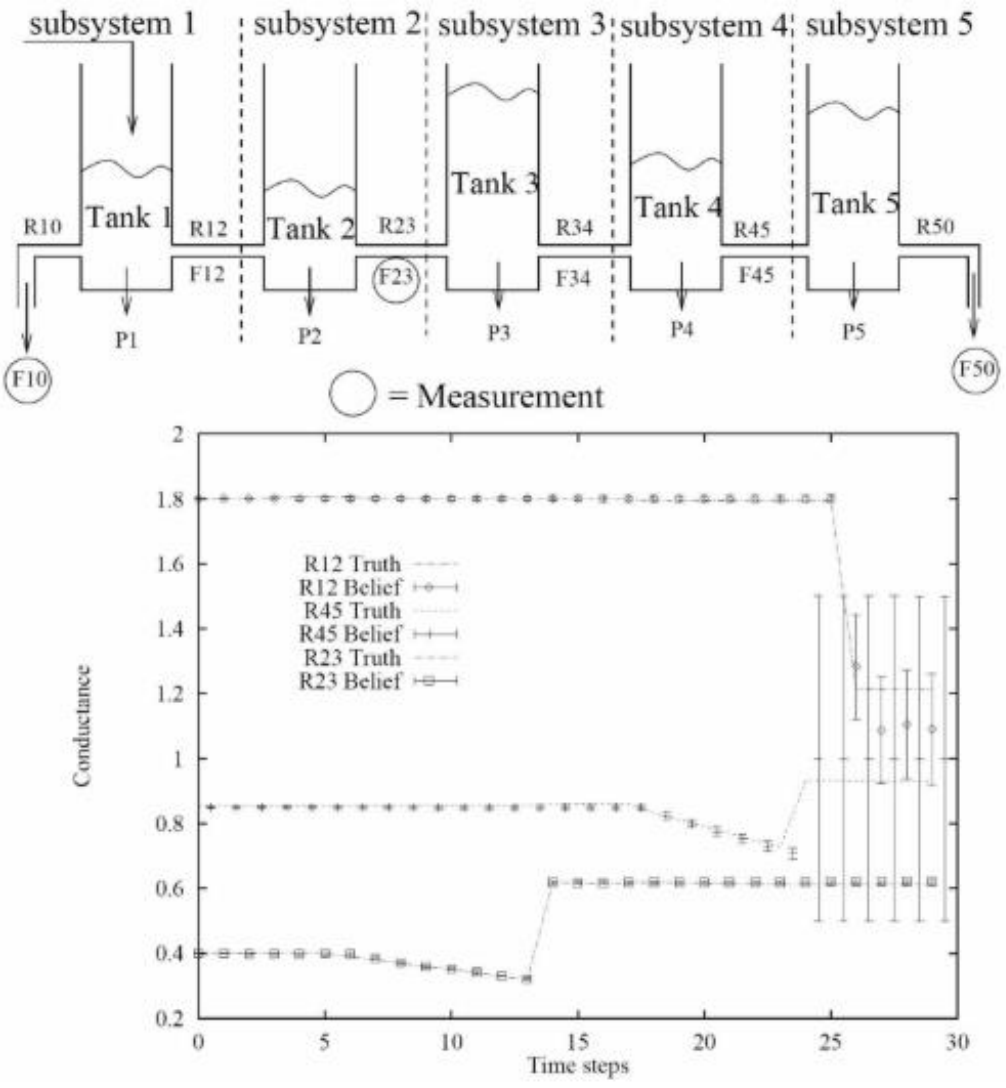

Рис. 3.14 Схема системи з п'ятьма резервуарами (вгорі) та результати діагностики несправностей (внизу).

порівняння поведінки агента з його/її профілем нормальної поведінки виявляється недостатнім для виявлення зловживань (misuse detection). Просто відстеження часу доступу або частоти звернень до певної інформаційної бази може призводити до великої кількості хибних спрацьовувань, оскільки в агента можуть бути цілком легітимні робочі причини для таких дій.

Тому в цьому дослідженні DBN було застосовано для комбінування різноманітних домен-специфічних ознак з метою отримання міри підозрілості (degree of suspiciousness) дій агента.

Загалом цей підхід є розумним, оскільки діяльність агента є стохастичним процесом: агенту може бути призначене завдання, що потребує тривалого часу, а дії, які він виконує для його завершення, ймовірно будуть причинно-наслідково пов’язаними. У роботі представлено DBN (рис. 3.15), спеціально адаптовану для податкової служби, хоча ту саму парадигму моделювання можна застосовувати й до інших галузей.

На рис. 3.15 кожен прямокутник містить випадкові змінні, специфічні для одного часового зрізу (time slice). У кожному зрізі визначені такі змінні:

- F^d — частота використання баз даних
- F^r — частота використання записів
- T^r — кількість часу, витраченого на записи
- M^r — індикатор модифікації записів
- Tk — тип виконуваного завдання (audit або collection/delivery)
- Intr — індикатор порушення конфіденційності
- Hrs — індикатор виконання роботи в робочий час
- A^r — кількість записів
- T^d — кількість часу, витраченого на бази даних

Стрілка від Tk_0 до Tk_1 відображає еволюцію завдань агента, а стрілка від $Intr_0$ до $Intr_1$ — еволюцію факту порушення конфіденційності агентом. У моделі припускається, що агент з більшою ймовірністю вчинить порушення, якщо він уже зараз займається підозрілими діями. Аналогічно, чим довше агент

утримується від таких дій, тим менш імовірним стає порушення в поточному часовому зрізі.

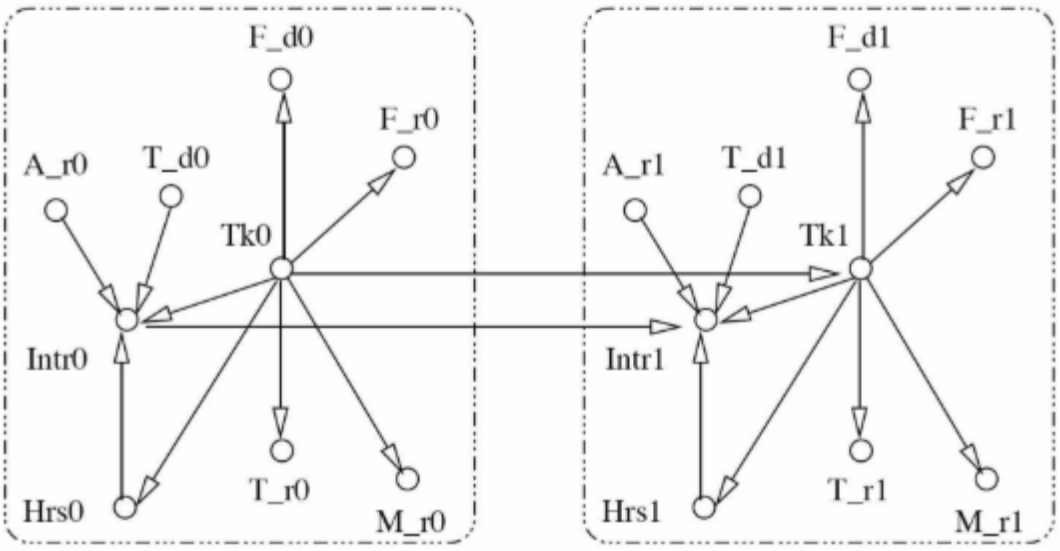

Рис. 3.15 DBN для виявлення вторгнення в приватність

Дослідження валідувало DBN за допомогою «здорового глузду» — прогону правдоподібних сценаріїв. Ймовірно через чутливість реальних даних, конкретних числових результатів виявлення не наводилось. Тим не менш, запропонований підхід цілком придатний для виявлення вторгнень, особливо коли атака полягає у викраденні великого обсягу приватних даних. Більше того, якщо агент отримує доступ до великої кількості даних, що не стосуються його службових обов’язків, DBN працює особливо добре, оскільки нерелевантність даних прямо змодельована в мережі.

Наостанок розглянемо недавнє застосування DBN для виявлення транспортних інцидентів. Виявлення інцидентів на дорогах — важлива практична задача, оскільки оперативне виявлення значно зменшує витрати, пов’язані з аваріями. У дослідженні порівнювалась продуктивність простих уніваріантних детекторів (порогова обробка кожної ознаки окремо) та їх комбінації з методом опорних векторів (SVM). SVM було обрано тому, що він узагальнює лінійні дискримінатори, які реалізують порогові детектори. Алгоритми тестувались на реальних даних руху, зібраних на найбільш

аварійній ділянці автомагістралі в Піттсбурзі. Дані містили статистику дороги (середня швидкість, об'єм — кількість автомобілів, що проїхали, та occupancy — щільність трафіку), зібрану з інтервалом від 30 секунд до 5 хвилин.

Виявилось, що SVM загалом перевершує еталонний алгоритм виявлення — каліфорнійську модель TSC-2. TSC-2 використовує послідовність порогів для оцінки різниць і пропорцій occupancy між сусідніми часовими кроками. Емпірично еталонна модель виявляла в найкращому разі лише третину інцидентів. Порівняно з SVM, TSC-2 зазвичай давала нижчі значення ROC AUC. Проте при дуже низькому рівні хибних спрацьовувань TSC-2 перевершувала SVM завдяки врахуванню доказів з останнього часового кроку. Тому до SVM додали перевірку стійкості (persistence check), що різко підвищило її продуктивність (рис. 3.16).

Це підказує, що часова (динамічна) структура детектора може бути доцільнішою для цієї предметної області. Тому було запропоновано DBN-модель (рис. 3.17), яка містить одну приховану дискретну змінну стану C (синонім невідомого класу) та кілька умовно незалежних унівaріантних гаусових змінних спостережень $O = O_1, \ldots, O_n$. Додатково є бінарна змінна спостереження I — стан інциденту за даними центру керування рухом. Згідно з нашою нотацією, умовний розподіл $p(I_t \mid C_t)$. Ліворуч показано графіки залежності рівня виявлення (вісь Y) від рівня помилкових спрацьовувань (вісь X). Праворуч показано криві ROC, що відображають залежність імовірності справжнього спрацьовування або чутливості (вісь Y) від імовірності помилкового спрацьовування, визначеної як одиниця мінус специфічність (вісь X). Графіки (a) і (b) показують ефективність еталонного детектора California TSC-2. Графіки (c) і (d) показують ефективність SVM. Графіки (e) і (f) показують ефективність вдосконаленого SVM, що включає перевірку стійкості, при якій інцидент повинен бути виявлений у двох послідовних часових точках, перш ніж буде подано сигнал тривоги.

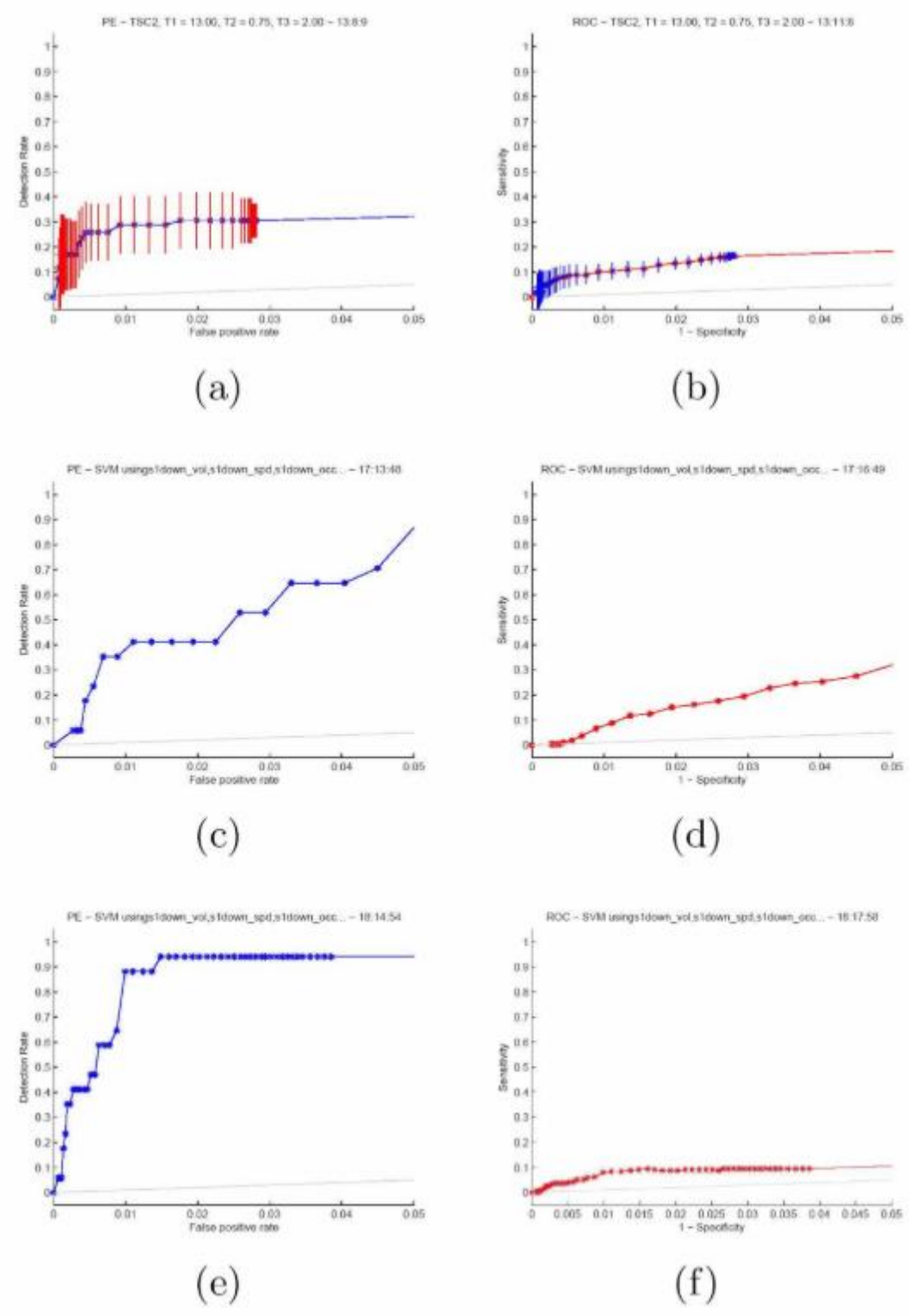

Рис. 3.16 Порівняння SVM та еталонного детектора при низьких рівнях помилкових спрацьовувань.

Результати підходу на основі DBN показані на рисунку 3.17. На жаль, підхід DBN досягнув лише AUC ROC 0,568381, порівняно з 0,810531, досягнутим SVM. Ця низька продуктивність може бути пов'язана з тим, що

структура DBN може не бути найкращою для даних. За допомогою більш складної моделі DBN продуктивність можна покращити. Тим не менш, ця робота проклала шлях для використання DBN у цьому напрямку виявлення аномалій.

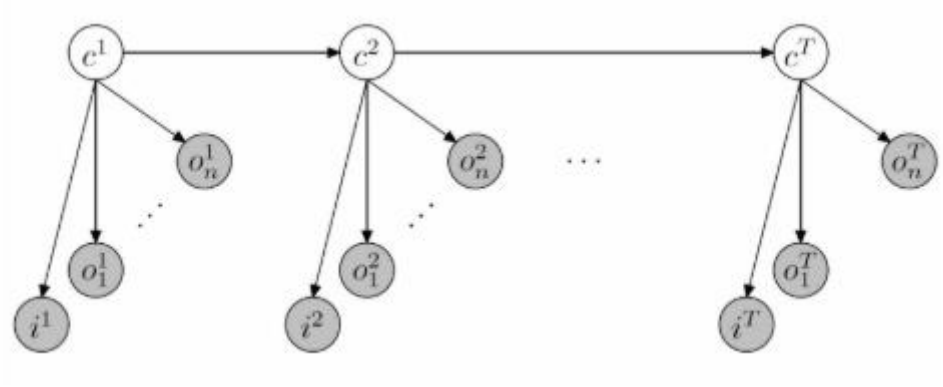

Рис. 3.17 DBN для виявлення дорожньо-транспортних пригод.

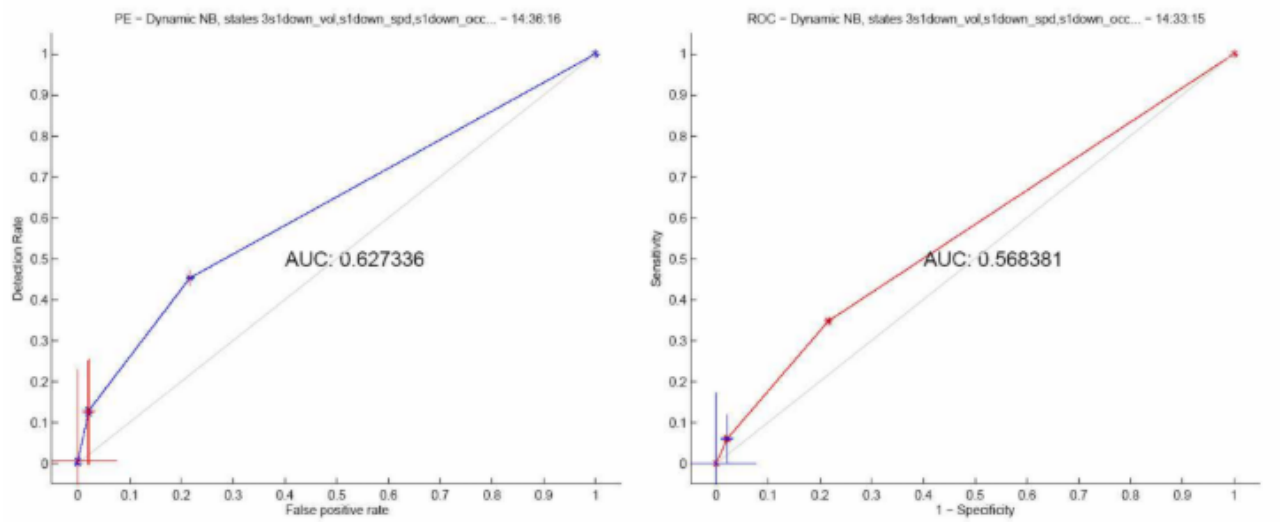

Рис. 3.18 Ефективність детектора на основі DBN.

Висновки до розділу

У третьому розділі досліджено генеративні та ймовірнісні методи. Вікна Парзена та суміші гаусівських розподілів виявилися сильними в задачах з чіткою статистичною структурою даних і невеликої розмірності, забезпечуючи високу точність і ймовірнісну інтерпретацію. Приховані марковські моделі та

динамічні байєсівські мережі показали перевагу при роботі з часовими рядами та послідовними даними, дозволяючи враховувати контекст і виявляти контекстно-залежні аномалії. Повторно підкреслено різницю між оцінкою стану та виявленням аномалій: генеративні моделі природніше підходять для другої задачі, оскільки не потребують прикладів аномалій. Загальний висновок розділу: генеративні методи переважають там, де важлива інтерпретування, можливість генерації синтетичних даних та робота з нестаціонарними процесами, хоча поступаються дискримінативним за швидкістю на великих обсягах даних. Універсального рішення не існує — вибір методу визначається типом і структурою даних.

ВИСНОВКИ

У ході виконання магістерської роботи було проведено комплексне теоретичне дослідження моделей та методів виявлення аномалій у даних, яке охопило аналіз сучасних підходів, систематизацію дискримінативних та генеративних (ймовірнісних) методів, а також порівняльну оцінку їхніх можливостей, переваг і обмежень. Робота показала, що виявлення аномалій залишається однією з найскладніших задач машинного навчання через різноманітність типів даних, мінливість реальних систем і необхідність розрізняти доброякісні та зловмисні відхилення.

Ми розпочали з теоретичних основ , де чітко розмежували задачу оцінки стану системи та задачу виявлення аномалій, розглянули фундаментальні відмінності між дискримінативними та генеративними моделями, а також запропонували розширену класифікацію методів з урахуванням сучасних тенденцій. Це дозволило сформувати єдину концептуальну базу для подальшого аналізу.

У другому розділі детально досліджено дискримінативні методи. Було показано, що методи на основі відстані та найближчих сусідів (k-NN, LOF, COF) добре працюють у випадках, коли доступні лише нормальні дані та важлива інтерпретованість, але чутливі до «прокляття розмірності». One-Class SVM продемонстрував високу ефективність при нелінійних границях, а нейронні мережі (зокрема автоенкодери у режимі one-class) виявилися перспективними для високовимірних даних, хоча потребують значних обчислювальних ресурсів та великої кількості нормальних прикладів для навчання.

Третій розділ присвячено генеративним і ймовірнісним методам. Аналіз вікон Парзена та сумішей гаусівських розподілів підтвердив їхню сильну теоретичну обґрунтованість і високу точність на даних з чіткою статистичною структурою, але водночас виявив обмежену масштабованість при великій розмірності. Приховані марковські моделі та динамічні байєсівські мережі

виявилися особливо ефективними для часових рядів і послідовних даних, дозволяючи моделювати складні часові залежності та виявляти контекстно-залежні аномалії.

Проведене порівняння показало, що універсального методу не існує: вибір залежить від типу даних, наявності міток, вимог до інтерпретуваності, обчислювальної складності та характеру аномалій (точкові, контекстні, колективні). Дискримінативні методи частіше переважають в задачах з високою розмірністю та потребою швидкого інференсу, тоді як генеративні моделі кращі там, де потрібна ймовірнісна інтерпретація та можливість генерації синтетичних даних.

Розроблена в роботі класифікація та критерії вибору методу створюють методичну основу для практичного застосування, допомагаючи спеціалістам обґрунтовано обирати або комбінувати алгоритми залежно від конкретної предметної області (кібербезпека, промисловий моніторинг, фінанси, медицина).

Обмеженням дослідження є переважно теоретичний характер – відсутність широкого експериментального порівняння на реальних великих датасетах.

У майбутньому доцільно розширити роботу в напрямку:

- глибокого навчання (VAE, трансформери, дифузійні моделі);
- ансамблевих та гібридних підходів;
- адаптації до концептуального дрейфу та потокових даних;
- розробки універсальних фреймворків автоматичного вибору та комбінування методів.

Отримані результати можуть бути використані при проєктуванні реальних систем моніторингу та виявлення аномалій, а також як теоретична база для подальших наукових і прикладних розробок у цій галузі.

СПИСОК ПОСИЛАНЬ НА ДЖЕРЕЛА

1. Chandola, V., Banerjee, A., & Kumar, V. Anomaly detection: A survey. *ACM Computing Surveys*, 2009.
2. Aggarwal, C. C. Outlier Analysis. Springer, 2017.
3. Zhang, X., et al. A survey of anomaly detection techniques in IoT networks. *Computer Networks*, 2022.
4. Kwon, D., et al. A survey of deep learning-based network anomaly detection. *Computers & Security*, 2023.
5. Khan, S., & Yairi, T. A review on anomaly detection techniques for time-series data. *Pattern Recognition*, 2018.
6. Pang, G., et al. Deep anomaly detection: A survey. *ACM Computing Surveys*, 2021.
7. Breunig, M. M., et al. LOF: Identifying density-based local outliers. *ACM SIGMOD*, 2000.
8. Liu, F. T., et al. Isolation Forest. *IEEE ICDM*, 2008.
9. Zong, B., et al. Deep autoencoding Gaussian mixture model for unsupervised anomaly detection. *ICLR*, 2018.
10. Malhotra, P., et al. LSTM-based encoder–decoder for anomaly detection in time-series. *ICMLA*, 2016.
11. Hochreiter, S., & Schmidhuber, J. Long short-term memory. *Neural Computation*, 1997.
12. Goodfellow, I., Bengio, Y., & Courville, A. Deep Learning. MIT Press, 2016.
13. Kingma, D. P., & Welling, M. Auto-Encoding Variational Bayes. *ICLR*, 2014.
14. Kingma, D. P., & Ba, J. Adam optimization. *ICLR*, 2015.
15. Ruff, L., et al. Deep One-Class Classification. *ICML*, 2018.
16. Xu, H., et al. Unsupervised anomaly detection for time-series using GANs. *AAAI*, 2018.

17. Schlegl, T., et al. f-AnoGAN: Fast anomaly detection with GANs. *Medical Image Analysis*, 2019.

18. Tuli, S., et al. TranAD: Transformer-based anomaly detection for multivariate time-series. *VLDB*, 2022.

19. Omitaomu, O., & Protopopescu, V. Deep learning for anomaly detection in large-scale systems. *Energy Informatics*, 2021.

20. Lakshminarayanan, B., et al. Deep ensembles: A simple approach to uncertainty estimation. *NeurIPS*, 2017.

21. Zhao, Y., et al. PyOD: A Python toolbox for scalable anomaly detection. *JMLR*, 2020.

22. Feng, J., et al. Anomaly detection in dynamic graphs using GNNs. *CIKM*, 2022.

23. Ding, Y., et al. Graph-based anomaly detection: A survey. *IEEE TKDE*, 2023.

24. Chauhan, S., & Vig, L. Anomaly detection in ECG time-series: LSTM models. *IEEE IJCNN*, 2015.

25. Zhu, L., et al. Anomaly detection in cloud systems using attention networks. *IEEE TNNLS*, 2021.

26. Xu, X., et al. A review of machine learning approaches to anomaly detection in industrial data. *Sensors*, 2021.

27. Doshi-Velez, F., & Kim, B. Interpretable ML for anomaly detection. *arXiv*, 2017.

28. Wang, S., et al. Multivariate time-series anomaly detection with hierarchical VAEs. *NeurIPS*, 2019.

29. Gavai, A., et al. Event anomaly detection in large-scale enterprise data. *KDD*, 2015.

30. Wu, Y., et al. A hybrid deep learning approach for anomaly detection in manufacturing. *IEEE Transactions on Industrial Informatics*, 2022.

31. Hawkins, S. Outliers in Statistical Data. Springer, 1980.

32. Box, G., Jenkins, G. Time Series Analysis: Forecasting and Control. Wiley, 2015.

33. Jiang, L., & Hu, X. Anomaly detection using one-class SVMs: A practical survey. *Knowledge-Based Systems*, 2023.

34. Sosnovik, I., et al. Neural diffusion models for anomaly detection. *CVPR Workshops*, 2023.

35. Li, Z., et al. Diffusion models for unsupervised anomaly localization. *ICCV*, 2023.

36. Kim, M., et al. Real-time anomaly detection using lightweight neural networks on edge devices. *Future Generation Computer Systems*, 2024.