ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
МІНІСТЕРСТВА ОСВІТИ І НАУКИ УКРАЇНИ

Кваліфікаційна наукова праця
на правах рукопису

ВІЖЕВСЬКИЙ ПЕТРО ВОЛОДИМИРОВИЧ

УДК 004.3:004.5:004.75

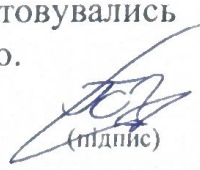
ДИСЕРТАЦІЯ

МЕТОДИ ТА ЗАСОБИ ВИЯВЛЕННЯ І ЗАПОБІГАННЯ ВИТОКУ ДАНИХ В КОРПОРАТИВНИХ МЕРЕЖАХ ЗГІДНО ЕВОЛЮЦІЙНИХ АЛГОРИТМІВ

122 Комп'ютерні науки
(шифр і назва спеціальності)

12 Інформаційні технології
(галузь знань)

Подається на здобуття наукового ступеня доктора філософії
Дисертація містить результати власних досліджень. Подані до захисту наукові положення є власним напрацюванням. Всі використані ідеї, наукові результати, цитати супроводжуються належними посиланнями на їх авторів та джерела опублікування. Всі частини тексту дисертації, під час написання яких використовувались технології штучного інтелекту, перевірені та відредаговані особисто.


(підпис)

П.В. Віжевський

Наукові керівники: Савенко Олег Станіславович, доктор технічних наук, професор
Кравчик Юрій Васильович, кандидат економічних наук, доцент

Хмельницький – 2026

АНОТАЦІЯ

Віжевський П. В. Методи та засоби виявлення і запобігання витоку даних в корпоративних мережах згідно еволюційних алгоритмів. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня доктора філософії за спеціальністю 122 – Комп'ютерні науки. – Хмельницький національний університет, Хмельницький, 2026.

Вирішення задачі підвищення ефективності виявлення і запобігання витокам конфіденційної інформації в корпоративних мережах є однією з важливих наукових задач в галузі інформаційних технологій, орієнтованих на побудову та подальшу експлуатацію систем запобігання витокам даних (ЗВД).

У дисертації здійснено аналіз методів та засобів виявлення витоку даних в корпоративних мережах, наявних ЗВД-рішень, їхніх функціональних обмежень та підходів до класифікації документів, поведінкового аналізу та адаптації політик безпеки. В роботі розроблено моделі, методи виявлення і запобігання витокам конфіденційної інформації в корпоративних мережах на основі еволюційних алгоритмів, а також розроблено відповідні програмні засоби і проведено з ними експериментальні дослідження.

Об'єктом дослідження є процес виявлення і запобігання витокам конфіденційної інформації в корпоративних мережах.

Предметом дослідження є методи та програмні засоби виявлення і запобігання витоку даних в корпоративних мережах на основі еволюційних алгоритмів.

Метою дисертаційного дослідження є підвищення ефективності виявлення і запобігання витокам конфіденційної інформації в корпоративних мережах шляхом розробки еволюційно-адаптивних методів класифікації документів, поведінкового профілювання користувачів та оптимізації політик безпеки на основі генетичних алгоритмів, а також програмних засобів, що реалізують ці методи у складі єдиного програмного комплексу.

Наукова новизна отриманих результатів полягає в такому:

1) удосконалено модель процесу запобігання витокам даних, яка, на відміну від наявних описових підходів з ізольованим розглядом компонентів ЗВД-системи,

об’єднує моделі витоку, корпоративного середовища та даних у єдиній формальній схемі із чітко визначеними точками прикладення адаптивних механізмів, що дає змогу узгоджено проєктувати класифікацію документів, поведінкове профілювання та адаптацію політик як складники одного процесу, а не набір незалежних інструментів;

2) вперше розроблено метод класифікації документів за рівнем конфіденційності на основі генетичного алгоритму, особливістю якого є хромосомне кодування правил у поєднанні з регуляризацією складності моделі та, на відміну від жадібної індукції правил методом послідовного покриття (CN2, RIPPER) і від генетичного відбору правил із наперед сформованої бази знань, глобальний еволюційний пошук цілісних наборів правил безпосередньо в просторі ознак політик безпеки, що дає змогу отримувати компактний, придатний для аудиту та безпосередньої експертної інтерпретації набір правил із конкурентною якістю розпізнавання і за потреби відновлювати виявлення нового типу критичних ідентифікаторів одним дописаним експертом правилом без залучення розмічених прикладів;

3) вперше розроблено метод еволюційної адаптації політик ЗВД за умов дрейфу концепції, особливістю якого є детектор змін на основі підтвердженого багатоознакового статистичного тесту, стійкого до поодиноких хибних спрацювань, який, на відміну від класичних детекторів на потоці помилок класифікатора (ADWIN, DDM, EDDM, KSWIN), працює безпосередньо з розподілом вхідних ознак і не потребує оперативних міток істинності, у поєднанні з еволюційним перенавчанням політик, яке, на відміну від інкрементних і ансамблевих схем адаптації, зберігає інтерпретовану структуру набору правил, що дає змогу суттєво зменшити кількість хибних тривог за повного виявлення справжніх змін і втримати точність класифікації в нестаціонарному потоці;

4) набув подальшого розвитку метод поведінкового профілювання користувачів, який, на відміну від глобальних методів виявлення аномалій з єдиною моделлю нормальної поведінки для всієї сукупності користувачів, ґрунтується на індивідуальному адаптивному профілі кожного користувача з генетичною оптимізацією його конфігурації та ваг ознак, що дає змогу точніше виявляти внутрішніх порушників у межах реалістичного бюджету тривог.

Практичне значення отриманих результатів. На основі запропонованих методів розроблено програмний прототип ЗВД-системи мовою Python із модульною архітектурою. Експериментальне дослідження на відкритих корпусах і реальних потоках даних показало, що генетичний класифікатор документів досягає F1-міри 0,826 на корпусі ai4privacy [117] за компактного набору приблизно з п’яти правил, метод поведінкового профілювання на корпусі CERT r4.2 [119] виявляє інсайдерів на робочому бюджеті тривог приблизно у 2,5 раза точніше за глобальні методи (точність 0,90 у першій двадцятці підозрюваних проти щонайбільше 0,35), правило підтвердження детектора дрейфу зменшує кількість хибних тривог більш як утричі за повного виявлення справжніх змін на потоках з дискретними подіями дрейфу, адаптація з перенавчанням за сигналом детектора підвищує преквенціальну точність на реальних потоках на 6,6-8,0 % порівняно зі статичною політикою, а еволюційне перенавчання правил на потоках зі структурою політики ЗВД дає 0,851 преквенціальної точності проти 0,739 в інкрементної моделі. У сценарії появи нового типу ідентифікаторів одне дописане експертом правило миттєво відновлює повноту виявлення документів нового типу без жодного розміченого прикладу, тоді як непрозорий ансамблевий класифікатор без такого втручання пропускає більшість документів нового типу.

Теоретичні та практичні результати дослідження впроваджені в ТОВ «ІТТ» (м. Хмельницький), ПП «Авіві» (м. Хмельницький), а також в освітньому процесі Хмельницького національного університету при викладанні дисциплін на кафедрі комп’ютерної інженерії та інформаційних систем для здобувачів спеціальності F7 Комп’ютерна інженерія, зокрема в курсах «Безпека та захист комп’ютерних систем», «Методології забезпечення якості, надійності, гарантоздатності та безпеки комп’ютерних систем та мереж».

У вступі представлено обґрунтування актуальності наукової задачі виявлення і запобігання витокам конфіденційної інформації в корпоративних мережах. Здійснено аналіз результатів відомих дослідників за цією тематикою та виокремлено переваги і недоліки відомих рішень, на основі яких сформульовано протиріччя. Визначено об'єкт, предмет, мету та завдання дослідження. Відображено основні наукові результати роботи та їх практичне значення.

У першому розділі здійснено аналіз відомих методів та засобів виявлення витоку даних в корпоративних мережах, систематизацію сучасних комерційних та дослідницьких ЗВД-рішень, механізмів контентної інспекції, контекстного контролю та поведінкового аналізу. Формалізовано функціональні обмеження розглянутих систем. Обґрунтовано доцільність застосування еволюційно-адаптивних методів. Здійснено постановку задачі дослідження.

У другому розділі представлено розробку формальної моделі процесу запобігання витокам даних, що синтезує три взаємопов'язані компоненти: модель витоків даних, яка описує актора, об'єкт, канал передачі та ризик-модель інциденту; модель корпоративного середовища, що формалізує граф мережної інфраструктури, рольовий доступ, потоки даних між зонами безпеки та точки контролю ЗВД; модель даних, де документ подається як кортеж вмісту, метаданих та контексту. Композиція цих моделей утворює результуючу модель ЗВД-системи з чотирма функціональними модулями.

У третьому розділі представлено розроблені методи виявлення витоків даних на основі еволюційних алгоритмів: метод класифікації документів за рівнем конфіденційності з хромосомним кодуванням правил у формі IF-THEN; метод еволюційної адаптації політик ЗВД за умов дрейфу концепції зі статистичним детектором та адаптивним перемиканням між режимами експлуатації та дослідження; метод поведінкового профілювання користувачів з адаптивним середнім на основі експоненціального забування та генетичною оптимізацією ваг ознак.

У четвертому розділі представлено експериментальне дослідження запропонованих методів на відкритих корпусах і реальних наборах даних: синтетично згенерованому та валідованому експертами корпусі документів з персональними даними ai4privacy, корпусі 20 Newsgroups, корпусі інсайдерських загроз CERT Insider Threat r4.2, реальних потоках Electricity та INSECTS і стандартних потоках RandomRBF. Виконано порівняння з відомими методами машинного навчання, оцінено інтерпретованість класифікатора, чутливість до параметрів та обчислювальну вартість, а також наведено наскрізний приклад функціонування системи.

У висновках представлено отримані наукові та практичні результати дисертаційного дослідження.

У додатках представлено наукові публікації, в яких відображено основні наукові результати роботи, акти впровадження результатів роботи, лістинги програмного забезпечення, таблиці результатів експериментів.

Ключові слова: кібербезпека, корпоративна мережа, запобігання витокам даних, генетичний алгоритм, еволюційна оптимізація, класифікація документів, дрейф концепції, поведінкове профілювання, інформаційна безпека, конфіденційна інформація, еволюційні алгоритми, витік даних.

ANNOTATION

Vizhevskyi P. V. Methods and means of data leak detection and prevention in corporate networks using evolutionary algorithms. – Qualification scientific work as a manuscript.

Dissertation for the degree of Doctor of Philosophy in the specialty 122 – Computer Science. – Khmelnytskyi National University, Khmelnytskyi, 2026.

Solving the problem of improving the efficiency of detection and prevention of confidential information leakage in corporate networks is one of the important scientific tasks in the field of information technology, focused on the construction and operation of data loss prevention (DLP) systems.

The dissertation analyzes the methods and means of data leak detection in corporate networks, existing DLP solutions, their functional limitations and approaches to document classification, behavioral analysis and security policy adaptation. The paper develops models, methods for detecting and preventing confidential information leakage in corporate networks based on evolutionary algorithms, as well as develops appropriate software tools and conducts experimental studies.

The object of the study is the process of detecting and preventing confidential information leakage in corporate networks.

The subject of the study is methods and software means of data leak detection and prevention in corporate networks based on evolutionary algorithms.

The purpose of the dissertation research is to improve the efficiency of detecting and preventing confidential information leakage in corporate networks through the development of evolutionary-adaptive methods for document classification, user behavioral profiling and

security policy optimization based on genetic algorithms, as well as of the software means that implement these methods within a single software suite.

The scientific novelty of the results obtained is as follows:

1) the model of the data leak prevention process has been improved; in contrast to the existing descriptive approaches that consider the components of a DLP system in isolation, it unites the leak model, the corporate environment model and the data model within a single formal scheme with clearly defined application points for adaptive mechanisms, which makes it possible to design document classification, behavioral profiling and policy adaptation in a coordinated way, as parts of a single process rather than a set of independent tools;

2) for the first time, a method of document classification by confidentiality level based on a genetic algorithm has been developed, whose distinctive feature is the chromosomal encoding of production rules combined with regularization of model complexity and, unlike greedy rule induction by sequential covering (CN2, RIPPER) and genetic selection of rules from a pre-formed knowledge base, a global evolutionary search over complete rule sets directly in the security-policy feature space, which makes it possible to obtain a compact set of rules suitable for audit and direct expert interpretation with competitive recognition quality and, when necessary, to restore the detection of a new type of critical identifiers with a single expert-added rule without any labeled examples;

3) for the first time, a method of evolutionary adaptation of DLP policies under concept drift has been developed, whose distinctive feature is a change detector based on a confirmed multi-feature statistical test resistant to isolated false triggers and, unlike error-stream detectors (ADWIN, DDM, EDDM, KSWIN), performed directly on the distribution of input features without requiring timely ground-truth labels, combined with evolutionary retraining of policies that, unlike incremental and ensemble adaptation schemes, preserves the interpretable structure of the rule set, which makes it possible to substantially reduce the number of false alarms while fully detecting true changes and to maintain classification accuracy in a non-stationary stream;

4) the method of behavioral profiling of users has been further developed; in contrast to global anomaly detection methods with a single model of normal behavior for the entire population of users, it is based on an individual adaptive profile of each user with genetic

optimization of its configuration and feature weights, which makes it possible to detect insider violators more precisely within a realistic alert budget.

Practical significance of the results obtained. Based on the proposed methods, a prototype DLP system software package was developed in Python with a modular architecture. Experimental evaluation on open benchmark corpora and real-world data streams shows that the genetic rule-based document classifier achieves an F1-score of 0.826 on the ai4privacy PII-Masking-200k corpus [117] with a compact set of about five rules; the behavioral profiling method detects insiders on the CERT Insider Threat r4.2 corpus [119] about 2.5 times more precisely than global anomaly detection methods under a realistic alert budget (precision 0.90 among the top twenty suspects versus at most 0.35); the confirmation rule of the drift detector reduces the number of false alarms more than threefold while fully detecting true changes on streams with discrete drift events; and detector-triggered retraining improves prequential accuracy on real streams by 6.6-8.0% compared with a static policy, while evolutionary rule retraining reaches 0.851 versus 0.739 for the incremental model on streams whose concept has the structure of a DLP policy. In the scenario of a new identifier type, a single rule added by an expert immediately restores detection recall on documents of the new type without a single labeled example, whereas an opaque ensemble model without such an intervention misses most documents of the new type.

The theoretical and practical results of the research are implemented in LLC "ITT" (Khmelnytskyi), PE "Avivi" (Khmelnytskyi), as well as in the educational process of the Khmelnytskyi National University in the teaching of disciplines at the Department of Computer Engineering and Information Systems for applicants of the F7 Computer Engineering specialty, in particular in the courses "Security and Protection of Computer Systems", "Methodology of Quality Assurance, Reliability, guarantee capacity and security of computer systems and networks".

In the introduction, the justification of the relevance of the scientific task of detecting and preventing confidential information leakage in corporate networks is presented. A review of works by domestic and foreign researchers is carried out. The object, subject, goal and tasks of the research are defined. The main scientific results of the work and their practical significance are reflected.

The first section analyzes existing methods and means of data leak detection in corporate networks, systematization of modern commercial and research DLP solutions,

mechanisms of content inspection, contextual control and behavioral analysis. The functional limitations of the reviewed systems are formalized. The expediency of using evolutionary-adaptive methods is substantiated. The research problem is formulated.

The second section presents the development of a formal model of the data leak prevention process that synthesizes three interrelated components: the data leak model, which describes the actor, the object, the transmission channel and the incident risk model; the corporate environment model, which formalizes the network infrastructure graph, role-based access, data flows between security zones and DLP control points; and the data model, in which a document is represented as a tuple of content, metadata and context. The composition of these models forms the resulting DLP system model with four functional modules.

The third section presents the developed methods for data leak detection based on evolutionary algorithms: a method for document classification by confidentiality level with chromosomal encoding of IF-THEN rules; a method for evolutionary adaptation of DLP policies under concept drift with a statistical drift detector and adaptive switching between the exploitation and exploration modes; a method for behavioral profiling of users with adaptive mean based on exponential forgetting and genetic optimization of feature weights.

The fourth section presents an experimental study of the proposed methods on open benchmark and real-world datasets: the synthetically generated, human-validated ai4privacy corpus of documents with personal data, the 20 Newsgroups corpus, the CERT Insider Threat r4.2 corpus, the real Electricity and INSECTS streams, and standard RandomRBF streams. A comparison with known machine learning methods is performed; interpretability, sensitivity to parameters, and computational cost are evaluated; and an end-to-end example of the system operation is provided.

The conclusions present the scientific and practical results of the dissertation research.

The appendices present scientific publications reflecting the main scientific results of the work, acts of implementation of results, software listings, and tables of experiment results.

Keywords: cyber security, corporate network, DLP, genetic algorithm, evolutionary optimization, document classification, concept drift, behavioral profiling, information security, confidential information, evolutionary algorithms, data leak.

Список публікацій здобувача за темою дисертації

Наукові праці, в яких опубліковані основні наукові результати дисертації

1. Віжевський П. В., Савенко О. С. Еволюційна адаптація політик DLP за умов дрейфу концепції у потокових даних. *Центральноукраїнський науковий вісник. Технічні науки*. 2025. № 12(43). С. 9–19. DOI: https://doi.org/10.32515/2664-262X.2025.12(43).2.9-19

2. Віжевський П. В., Савенко О. С. Стратегії виявлення та запобігання витокам даних у корпоративних мережах згідно генетичного алгоритму. *Information Technology: Computer Science, Software Engineering and Cyber Security*. 2025. № 4. С. 42–56. DOI: https://doi.org/10.32782/IT/2025-4-6

3. Віжевський П. В., Кравчик Ю. В. Модель системи запобігання витоку даних з еволюційною адаптацією на основі генетичних алгоритмів. *Вісник Хмельницького національного університету. Серія: Технічні науки*. 2025. № 5. С. 429–436. DOI: https://doi.org/10.31891/2307-5732-2025-357-56

4. Віжевський П. В., Савенко О. С. Модель процесу виявлення витоків даних з еволюційною адаптацією. *Вимірювальна та обчислювальна техніка в технологічних процесах*. 2026. № 1. С. 270–277. DOI: https://doi.org/10.31891/2219-9365-2026-85-34

5. Віжевський П. В., Савенко О. С. Поведінкове профілювання користувачів та виявлення аномалій на основі генетичного алгоритму. *Системні технології*. 2026. № 1 (162). С. 148–159. DOI: https://doi.org/10.34185/1562-9945-5-162-2026-17

Праці, які засвідчують апробацію матеріалів дисертації

6. Rehida P., Savenko O., Sachenko A., Drozd A., Vizhevski P. A trust model that ensures the correctness of computing in grid computing system. *Proceedings of the 5th International Workshop on Intelligent Information Technologies and Systems of Information Security (IntelITSIS-2024)*, Khmelnytskyi, March 2024. Vol. 3675. P. 388–401. URL: https://ceur-ws.org/Vol-3675/paper28.pdf

7. Golovko I., Savenko O., Vizhevskyi P., Sachenko A. Obfuscation process with machine learning module. *2024 14th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, Athens, Greece, 2024. P. 1–7. DOI: https://doi.org/10.1109/DESSERT65323.2024.11122185

8. Golovko I., Savenko O., Vizhevskiy P., Klein O., Salem A.-B. M. Obfuscation technologies of high-level source code using artificial intelligence. Proceedings of the 1st International Workshop on Intelligent & CyberPhysical Systems (ICyberPhyS-2024), Khmelnytskyi, Ukraine, June 28, 2024. Vol. 3736. P. 324–338. URL: https://ceur-ws.org/Vol-3736/paper24.pdf

9. Sachenko A., Vizhevskyi P., Savenko O., Ostroverkhov V., Maslyyak B. Modern strategies for data leak detection and prevention in corporate networks. *Modern Data Science Technologies Doctoral Consortium (MoDaST 2025)*, Lviv, Ukraine, June 15, 2025. 2025. P. 275–292. URL: https://ceur-ws.org/Vol-4005/paper19.pdf

Публікації, які додатково відображають наукові результати дисертації

10. Свідоцтво про реєстрацію авторського права на твір (програму) № 144040. Комп'ютерна програма «Програмний комплекс запобігання витоку даних з еволюційною адаптацією на основі генетичних алгоритмів (ЗВДЕАГА)» / П. В. Віжевський, О. С. Савенко. Дата реєстрації 04.03.2026. URL: https://iprop-ua.com/cr/ny8dq30b/

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ

БД – база даних

ГА – генетичний алгоритм

ЗВД (DLP) – запобігання витокам даних

КМ – корпоративні мережі

ІТ – інформаційні технології

BYOD – використання особистих пристроїв на роботі

IDS – система виявлення вторгнень

IPS – система запобігання вторгненням

UEBA – аналіз поведінки користувачів та сутностей

ВСТУП

Обгрунтування вибору теми дослідження. Сучасні корпоративні мережі (КМ) функціонують в умовах безперервного зростання обсягів оброблюваної інформації та ускладнення ландшафту загроз інформаційній безпеці. Вартість одного інциденту витоку даних у великих компаніях перетнула позначку в 4 мільйони доларів США [4], при цьому значна частка випадків залишається невиявленою протягом тривалого часу [1]. Перенесення даних та процесів у хмарні середовища, поширення мобільних платформ і гібридних моделей роботи суттєво розширили потенційну поверхню атаки [9; 83], що зумовлює підвищену увагу дослідників та індустрії до систем запобігання витокам даних (ЗВД). Такі системи покликані контролювати інформаційні потоки, класифікувати документи за рівнем конфіденційності та блокувати несанкціоновану передачу чутливих даних за межі захищеного периметру [5; 20].

Проблематиці виявлення та запобігання витокам даних у корпоративних мережах присвячено праці низки дослідників. L. Cheng, F. Liu, D. Yao [2] дослідили механізми порушення цілісності корпоративних даних та визначили напрями протидії витокам. I. Herrera Montano [6] здійснив систематичний огляд методів захисту від витоків та підходів до протидії інсайдерським загрозам. S. Syarova [5] систематизувала технології запобігання витокам даних у цифрових конфігураціях. H. Feng [18] запропонував модель та метод виявлення витоків з урахуванням семантики інформаційних потоків. H. Sakr та M. El-Afifi [22] розробили методологію виявлення витоків конфіденційних документів. T. Al-Shehari та R. Alsowail [42] дослідили виявлення інсайдерських витоків із застосуванням методів машинного навчання та надвибірки. J. Domnik та A. Holland [25] адаптували модель зрілості C2M2 для оцінювання ЗВД-процесів організацій. R. Muppalaneni та ін. [23] та D. Esther [24] дослідили застосування штучного інтелекту для підвищення ефективності ЗВД у сучасних корпоративних середовищах.

Окремий напрям досліджень пов’язаний із моделюванням поведінки користувачів та виявленням аномалій, що можуть свідчити про спроби несанкціонованого виведення даних. Song [8] розробив метод виявлення інсайдерських загроз на основі ритмів поведінки з урахуванням часової динаміки.

Mihailescu та ін. [51] запропонував підхід до аналізу поведінки користувачів для підвищення кібербезпеки. Проблему дрейфу концепції у потокових даних, що є критичною для систем моніторингу в реальному часі, досліджували H. Gomes та ін. [57], A. Cano та B. Krawczyk [58; 59], які розробили ансамблеві методи класифікації для нестаціонарних потоків даних. F. Hinder та ін. [60] та A. Arora та ін. [61] систематизували підходи до виявлення та адаптації до дрейфу концепції.

Теоретичні основи еволюційних алгоритмів та їх застосування в інженерних задачах висвітлено у працях S. Katoch, S.S. Chauhan, V. Kumar [64], B. Alhijawi та A. Awajan [65] та A. Slowik, H. Kwasnicka [66], де систематизовано підходи до генетичної оптимізації та визначено перспективні напрями їх розвитку.

Серед українських дослідників вагомий внесок у розвиток методів захисту корпоративних мереж зробили А. Саченко та ін. [12], які запропонували підхід до виявлення ботнетів на основі розподілених систем. Розвинуто також методи виявлення метаморфного зловмисного програмного забезпечення [14]. Роботи О. Корченка та А. Корченко зробили помітний внесок у методи виявлення кібератак: у праці [127] розроблено модель інтелектуального розпізнавання аномалій і кібератак на основі логічних процедур над покриттями матриць ознак, в роботі [16] запропоновано метод виявлення аномалій, спричинених кібератаками в комп’ютерних мережах. В. Харченко та А. Коваленко у співавторстві [128] розвинули методику оцінювання захищеності приватних комунікацій на основі аналізу режимів вторгнень і критичності їхніх наслідків. Р. Ткачук зі співавторами [129] запропонував категорні моделі подання структури та динамічного стану ієрархічних систем для виявлення факторів атак і ризиків. В. Мухін [130] обґрунтував адаптивні механізми керування безпекою комп’ютерних мереж на основі нечіткої логіки. Дотичним до цієї роботи є праця С. Штовби [131]: генетичний алгоритм вибору правил нечіткої бази знань, яка збалансована за критеріями точності та компактності.

Разом із тим аналіз сучасного стану досліджень засвідчує низку невирішених проблем. Наявні ЗВД-системи переважно спираються на статичні правила контентного аналізу, що потребують ручного налаштування та не забезпечують автоматичної адаптації до зміни характеру загроз і робочих процесів організації [90; 91]. Методи машинного навчання, що застосовуються для класифікації документів, часто функціонують як «чорні скриньки» і не забезпечують інтерпретованості рішень,

необхідної для аудиту безпеки [19; 22]. даних КМ залишається недостатньо дослідженою в контексті ЗВД-систем [60; 62]. Крім того, недостатньо формальних моделей, які б інтегрували опис витоків, корпоративного середовища та характеристик даних у єдине підґрунтя для побудови адаптивних захисних механізмів. Проблема дрейфу концепції у потокових даних КМ залишається недостатньо дослідженою в контексті ЗВД-систем [60; 62].

Зазначене обґрунтовує актуальність дисертаційного дослідження, спрямованого на розробку методів та програмних засобів виявлення і запобігання витоку даних у корпоративних мережах на основі еволюційних алгоритмів, що забезпечують адаптивну класифікацію документів, автоматичну оптимізацію політик безпеки та поведінкове профілювання користувачів.

Протиріччя полягає в тому, що динамічне корпоративне середовище з його хмарними сервісами, змінюваними робочими процесами, поступовим дрейфом характеристик даних та інсайдерськими загрозами потребує засобів запобігання витокам, здатних безперервно адаптуватися і водночас пояснювати свої рішення у формі, придатній для аудиту безпеки, тоді як наявні ЗВД-системи спираються на статичні правила з ручним налаштуванням, а методи машинного навчання, що додали б адаптивності, залишаються непрозорими й не дають придатних для аудиту пояснень. Розрив між цією потребою та можливостями наявних засобів і становить суть проблеми.

Зазначена науково-прикладна задача відповідає предметній області Стандарту вищої освіти України зі спеціальності 122 – Комп’ютерні науки для третього (освітньо-наукового) рівня вищої освіти, зокрема, такому об’єкту вивчення та діяльності, як «процеси збору, представлення, обробки, зберігання, передачі та доступу до інформації в комп’ютерних системах».

Зв’язок роботи з науковими програмами, планами, темами. Дисертаційне дослідження виконувалось у рамках науково-дослідної тематики Хмельницького національного університету, зокрема держбюджетної науково-дослідної теми № 1Б-2026 «Система забезпечення стійкості до витоку конфіденційної інформації в корпоративних мережах в умовах впливів комп’ютерних атак» (номер державної реєстрації: 0124U001214), в якій автор дисертації є виконавцем.

Мета і завдання дослідження.

Об'єкт дослідження – процес виявлення та запобігання витокам конфіденційної інформації в корпоративних мережах.

Предмет дослідження – методи та програмні засоби виявлення і запобігання витоку даних в корпоративних мережах на основі еволюційних алгоритмів.

Мета дослідження – підвищення ефективності виявлення та запобігання витокам конфіденційної інформації в корпоративних мережах шляхом розробки еволюційно-адаптивних методів класифікації документів, поведінкового профілювання користувачів та оптимізації політик безпеки на основі генетичних алгоритмів, а також програмних засобів, що реалізують ці методи у складі єдиного програмного комплексу.

Завдання дослідження сформульовано так:

1. Провести системний аналіз відомих методів та платформ ЗВД, формалізувати їхні функціональні обмеження за критеріями масштабованості, адаптивності та точності класифікації.

2. Розробити модель процесу запобігання витокам даних, що інтегрує модель витоку, модель корпоративного середовища та модель даних і враховує характеристики інформаційних ресурсів, профілі користувачів, рівні конфіденційності, показники довіри та оцінки потенційної шкоди від витоку.

3. Розробити метод класифікації документів за рівнем конфіденційності на основі генетичного алгоритму з хромосомним кодуванням продукційних правил, регуляризацією складності моделі та спеціалізованими генетичними операторами.

4. Розробити метод еволюційної адаптації політик безпеки за умов дрейфу концепції зі статистичним детектором змін, що забезпечує безперервне вдосконалення правил контентного та контекстного контролю з мінімізацією інтегральної функції втрат.

5. Розробити метод поведінкового профілювання користувачів з індивідуальними адаптивними профілями та генетичною оптимізацією складу ознак, їхніх ваг і порогів виявлення аномалій.

6. Спроектувати та реалізувати програмний комплекс запобігання витоку з модульною архітектурою і провести експериментальні дослідження запропонованих методів на відкритих корпусах і реальних наборах даних за показниками якості

класифікації, інтерпретованості, обчислювальної вартості та стійкості до дрейфу концепції. За результатами досліджень сформулювати практичні рекомендації щодо інтеграції еволюційно-адаптивних компонентів у наявні корпоративні системи безпеки.

Методи дослідження. Для розв'язання поставлених завдань у дисертаційній роботі використано: теорію множин; теорію графів; теорію ймовірностей та методи математичної статистики; методи машинного навчання; еволюційні алгоритми (генетичні алгоритми); методи оптимізації; елементи теорії інформаційної безпеки.

Наукова новизна отриманих результатів полягає в такому:

1) удосконалено модель процесу запобігання витокам даних, яка, на відміну від наявних описових підходів з ізольованим розглядом компонентів ЗВД-системи, об'єднує моделі витоку, корпоративного середовища та даних у єдиній формальній схемі із чітко визначеними точками прикладення адаптивних механізмів, що дає змогу узгоджено проєктувати класифікацію документів, поведінкове профілювання та адаптацію політик як складники одного процесу, а не набір незалежних інструментів;

2) вперше розроблено метод класифікації документів за рівнем конфіденційності на основі генетичного алгоритму, особливістю якого є хромосомне кодування продукційних правил у поєднанні з регуляризацією складності моделі та, на відміну від жадібної індукції правил методом послідовного покриття (CN2, RIPPER) і від генетичного відбору правил із наперед сформованої бази знань, глобальний еволюційний пошук цілісних наборів правил безпосередньо в просторі ознак політик безпеки, що дає змогу отримувати компактний, придатний для аудиту та безпосередньої експертної інтерпретації набір правил із конкурентною якістю розпізнавання і за потреби відновлювати виявлення нового типу критичних ідентифікаторів одним дописаним експертом правилом без залучення розмічених прикладів;

3) вперше розроблено метод еволюційної адаптації політик ЗВД за умов дрейфу концепції, особливістю якого є детектор змін на основі підтвердженого багатоознакового статистичного тесту, стійкого до поодиноких хибних спрацювань, який, на відміну від класичних детекторів на потоці помилок класифікатора (ADWIN, DDM, EDDM, KSWIN), працює безпосередньо з розподілом вхідних ознак і не

потребує оперативних міток істинності, у поєднанні з еволюційним перенавчанням політик, яке, на відміну від інкрементних і ансамблевих схем адаптації, зберігає інтерпретовану структуру набору правил, що дає змогу суттєво зменшити кількість хибних тривог за повного виявлення справжніх змін і втримати точність класифікації в нестаціонарному потоці;

4) набув подальшого розвитку метод поведінкового профілювання користувачів, який, на відміну від глобальних методів виявлення аномалій з єдиною моделлю нормальної поведінки для всієї сукупності користувачів, ґрунтується на індивідуальному адаптивному профілі кожного користувача з генетичною оптимізацією його конфігурації та ваг ознак, що дає змогу точніше виявляти внутрішніх порушників у межах реалістичного бюджету тривог.

Обґрунтованість і достовірність наукових положень, висновків та рекомендацій дисертаційної роботи забезпечується коректним застосуванням математичного апарату теорії множин, теорії графів, теорії ймовірностей та математичної статистики, використанням апробованих методів еволюційної оптимізації та машинного навчання, проведенням експериментальних досліджень на загальноприйнятих наборах даних (ai4privacy [117], 20 Newsgroups [118], CERT Insider Threat r4.2 [119], INSECTS [120], Electricity [121], стандартні потоки RandomRBF [122], згенеровані корпуси з контрольованими властивостями) із застосуванням стандартних метрик якості класифікації (precision, recall, F1-score, prequential accuracy), статистичною обробкою результатів із побудовою довірчих інтервалів, порівнянням з відомими методами та алгоритмами, апробацією результатів на міжнародних наукових конференціях та публікацією наукових результатів у фахових наукових виданнях.

Практичне значення отриманих результатів. На основі запропонованих методів розроблено програмний прототип ЗВД-системи мовою Python із модульною архітектурою. Експериментальні дослідження на відкритих корпусах і реальних потоках даних підтвердили, що генетичний класифікатор документів досягає F1-міри 0,826 на корпусі ai4privacy [117] за компактного набору приблизно з п’яти правил; метод поведінкового профілювання на корпусі CERT r4.2 [119] виявляє інсайдерів на робочому бюджеті тривог приблизно у 2,5 рази точніше за глобальні методи (точність 0,90 у першій двадцятці підозрюваних проти щонайбільше 0,35); правило

підтвердження детектора дрейфу зменшує кількість хибних тривог більш як утричі за повного виявлення справжніх змін на потоках з дискретними подіями дрейфу; адаптація з перенавчанням за сигналом детектора підвищує преквенціальну точність на реальних потоках на 6,6-8,0 % порівняно зі статичною політикою, а еволюційне перенавчання правил на потоках зі структурою політики ЗВД дає 0,851 преквенціальної точності проти 0,739 в інкрементної моделі. У сценарії появи нового типу ідентифікаторів одне дописане експертом правило миттєво відновлює повноту виявлення документів нового типу без жодного розміченого прикладу, тоді як непрозорий ансамблевий класифікатор без такого втручання пропускає більшість документів нового типу. Програмний комплекс запобігання витоку даних з еволюційною адаптацією на основі генетичних алгоритмів (ЗВДЕАГА), в якому реалізовано запропоновані методи, зареєстровано як об'єкт авторського права [126],

Теоретичні та практичні результати дослідження впроваджені в ТОВ «ІТТ» (акт від 30.06.2026, м. Хмельницький), ПП «Авіві» (довідка від 29.06.2026, м. Хмельницький), а також в освітньому процесі Хмельницького національного університету (акт від 26.06.2026) при викладанні дисциплін на кафедрі комп'ютерної інженерії та інформаційних систем для здобувачів спеціальності F7 Комп'ютерна інженерія, зокрема в курсах «Безпека та захист комп'ютерних систем», «Методології забезпечення якості, надійності, гарантоздатності та безпеки комп'ютерних систем та мереж».

Особистий внесок здобувача. Основні наукові результати, що виносяться на захист, отримано автором самостійно. Дисертація є самостійною науковою працею автора.

У працях, опублікованих у співавторстві, здобувачу належить: у [104] - розробка методу еволюційної адаптації політик ЗВД за умов дрейфу концепції з підтвердженим статистичним детектором змін; у [105] - розробка методу класифікації документів за рівнем конфіденційності на основі генетичного алгоритму з хромосомним кодуванням продукційних правил у складі стратегій виявлення та запобігання витокам даних; у [106] - розробка моделі системи запобігання витоку даних з еволюційною адаптацією на основі генетичних алгоритмів та її формалізація; у [107] - розробка моделі процесу виявлення витоків даних з еволюційною адаптацією, що інтегрує моделі витоку, корпоративного середовища та даних; у [108]

- розробка методу поведінкового профілювання користувачів на основі індивідуальних адаптивних профілів з генетичною оптимізацією їх конфігурації; у [109] - реалізація модуля оцінювання довіри, сформовано список інформації, який можна збирати на обчислювальному елементі шляхом розширення його функціональності; у [110] - розробка модуля машинного навчання для обфускації; у [111] - розробка методів обфускації вихідного коду з використанням штучного інтелекту; у [112] - розробка стратегій виявлення та запобігання витокам даних у сучасних корпоративних мережах.

У роботах, опублікованих у співавторстві, співавторам належать такі результати: [104; 105; 107; 108; 109; 110; 111; 112] – загальна концепція дослідження щодо стратегій виявлення та запобігання витокам даних у сучасних корпоративних мережах, запропоновано визначення ролі обчислювального елемента на основі його поведінки під час функціонування системи (Савенко О.); [106] –формування вимог до моделі (Кравчик Ю.); [109; 112] – постановка задачі надійних розподілених систем, запропоновано використання рейтингу обчислювального елементу та бінарного класифікатора для його коригування (Саченко А.); [109] – розроблення моделі довіри, розроблено модель із ролями обчислювальних елементів розподіленої системи та її реалізація (Регіда П. моделі довіри, розроблено модель із ролями обчислювальних елементів розподіленої системи та її реалізація (Регіда П.Г.); [109] – верифікація моделі, аналіз готових засобів для розгортання бінарного класифікатора (Дрозд А.І.); [110; 111] – дослідження технологій обфускації (Головко І.); [112] – аналіз стратегій виявлення витоків (Островерхов В.); [112] – підготовка експериментальних даних (Маслияк Б.); [111] – дослідження технологій обфускації (Клейн О.); [111] – розроблення підходів штучного інтелекту до застосування в задачі дослідження (Салем А.-Б. М.).

Апробація результатів дисертації. Основні положення та результати дисертаційної роботи доповідались і обговорювались на таких міжнародних наукових конференціях та семінарах: 5th International Workshop on Intelligent Information Technologies and Systems of Information Security (IntelITSIS-2024, м. Хмельницький, березень 2024 р.); 14th IEEE International Conference on Dependable Systems, Services and Technologies (DESSERT 2024, м. Афіни, Греція, 2024 р.); 1st International Workshop on Intelligent & CyberPhysical Systems (ICyberPhyS 2024, м. Хмельницький,

червень 2024 р.); Modern Data Science Technologies Doctoral Consortium (MoDaST 2025, м. Львів, червень 2025 р.).

Публікації. За результатами дисертаційного дослідження опубліковано 10 наукових праць, у тому числі 5 статей у наукових фахових виданнях України [104; 105; 106; 107; 108] і 4 праці у матеріалах міжнародних наукових конференцій [109; 110; 111; 112], з яких 2 опубліковано у виданнях, індексованих у наукометричній базі Scopus [109, 110]; одержано свідоцтво про реєстрацію авторського права на твір (комп'ютерну програму) [126].

Структура та обсяг дисертації. Дисертаційна робота складається з анотації, змісту, переліку умовних скорочень, вступу, чотирьох розділів, висновків, списку використаних джерел та семи додатків. Повний обсяг роботи становить 231 сторінка друкованого тексту, з них анотація – на 11 стор., зміст – на 2 стор., перелік скорочень – на 1 стор., основний текст (вступ, розділи, висновки) – на 150 стор., список зі 137 використаних джерел – на 15 стор., додатки – на 17 стор. Дисертація містить 13 рисунків та 21 таблицю.

РОЗДІЛ 1
АНАЛІЗ МЕТОДІВ ТА ЗАСОБІВ ВИЯВЛЕННЯ ВИТОКУ ДАНИХ В КОРПОРАТИВНИХ МЕРЕЖАХ

У сучасних КМ постійно зростають обсяги даних та ускладняються інформаційні потоки. Зміна бізнес-процесів, перехід на гібридні моделі роботи та інтеграція хмарних технологій видозмінили інформаційний простір підприємств, розширивши поверхню атак та збільшивши ймовірність несанкціонованого витоку конфіденційних даних. За даними звіту [1], майже третина всіх зафіксованих інцидентів витоку даних пов’язана з діяльністю третіх сторін з легітимним доступом: партнерів, постачальників, підрядників. Периметр безпеки організації більше не є чіткою межею, а швидше розмитою зоною взаємодії численних зовнішніх та внутрішніх суб’єктів. Більшість інцидентів є наслідком комбінації технічних вразливостей та організаційних недоліків, а не ізольованих точкових атак [2]. Для запобігання витокам даних необхідні комплексні підходи, що охоплюють як технічні, так і організаційні аспекти захисту інформаційних активів [3].

1.1 Методи виявлення витоку даних

Середня вартість одного інциденту витоку даних досягла 4,88 мільйона доларів США, при цьому майже 40 % випадків залишаються невиявленими протягом понад 200 днів [4]. Чим довше зловмисники зберігають доступ до скомпрометованих систем, тим більше даних вони здатні ексфільтрувати та тим складнішим і дорожчим стає подолання наслідків витоку даних. Репутаційні втрати, відтік клієнтів, регуляторні санкції можуть проявлятися протягом років після інциденту. Раннє виявлення інцидентів залишається ключовою умовою мінімізації їхніх наслідків [5].

Окрему категорію загроз становлять інсайдерські інциденти, які складніше виявити через наявність у порушників повноважень доступу. На інсайдерські загрози припадає значна частка усіх випадків витоку даних, а витрати на ліквідацію наслідків таких інцидентів суттєво перевищують середні показники [6]. Інсайдери можуть діяти з мотивів фінансової вигоди, помсти або ненавмисно, через недостатню обізнаність з політиками безпеки чи помилки. Навмисні та ненавмисні дії потребують різних

підходів до протидії, що ускладнює побудову універсальних механізмів. Агентство з кібербезпеки та захисту інфраструктури США (CISA) класифікує інсайдерські загрози як одну з найбільш недооцінених категорій ризиків корпоративної безпеки [7], а виявлення таких загроз потребує аналізу поведінкових закономірностей [8].

Дедалі ширше використання хмарних середовищ для зберігання корпоративних даних породжує специфічний комплекс вразливостей. Значна частка файлів, розміщених у хмарних сховищах, не охоплена політиками безпеки [9]. Причини цього явища різні: від складності конфігурування багаторівневих систем контролю доступу до відсутності належної інвентаризації хмарних сервісів у використанні. Хмарні сервіси створюють додаткові канали для потенційного витоку через API-інтерфейси, механізми синхронізації хмарних сховищ та функції спільного доступу, контроль яких виходить за межі традиційних засобів захисту. Підходи до маркування та класифікації даних [10] пропонують механізми систематичного відстеження конфіденційної інформації в розподілених середовищах незалежно від їх фізичного розміщення.

Групи операторів програм-вимагачів (ransomware) еволюціонували від простого шифрування даних до моделі подвійного вимагання, де викрадені дані використовуються як додатковий важіль тиску на жертву [11]. Ботнет-мережі забезпечують інфраструктуру для прихованої ексфільтрації інформації через розподілені командні центри [12]. Виявлення такої активності потребує застосування методів аналізу мережного трафіку на основі нечіткої кластеризації [13]. Методи виявлення зловмисного програмного забезпечення [14] та ботнет-мереж [15] продовжують еволюціонувати разом із загрозами. Виявлення аномалій, спричинених кібератаками [16], та рандомізація простору даних [17] пропонують додаткові рівні захисту від прихованої ексфільтрації. Моделі виявлення та запобігання [18] інтегрують множинні індикатори компрометації для підвищення точності ідентифікації загроз.

ЗВД-системи покликані протидіяти описаному комплексу загроз через моніторинг, класифікацію та контроль інформаційних потоків. Сучасні ЗВД-рішення оперують на кількох рівнях, а саме: аналіз безпосереднього вмісту даних та окремо метаінформації про джерело, одержувача та канал передачі [19]. Застосування методів машинного навчання відкриває нові можливості для адаптивної класифікації

документів та виявлення аномальних закономірностей поведінки користувачів [20]. Новітні підходи до виявлення витоків [21] та підходи до захисту конфіденційних документів [22] розширюють можливості традиційних ЗВД-систем. Використання штучного інтелекту демонструє потенціал адаптивних методів для підвищення точності класифікації та зменшення кількості хибних спрацювань [23; 24]. Водночас лише 40% організацій інтегрували свої ЗВД-системи з процедурами автоматизованого реагування на інциденти [26].

Ефективність впровадження захисних технологій суттєво обмежується кадровими чинниками. Недостатня кількість кваліфікованих фахівців з кібербезпеки в операційних центрах безпеки призводить до затримок у реагуванні на інциденти, неповного аналізу сповіщень та накопичення технічного боргу в налаштуванні захисних систем. Оптимізаційні підходи на основі легких генетичних алгоритмів [27] та семантичні комунікаційні моделі [28] пропонують механізми автоматизованої адаптації систем безпеки без постійного втручання адміністраторів.

Різноманіття джерел загроз, каналів ексфільтрації та мотивацій порушників унеможливлює створення єдиної універсальної системи захисту. Необхідні гнучкі методи, здатні пристосовуватися до зміни ландшафту загроз та специфіки інформаційних активів конкретної організації. Саме це обґрунтовує доцільність застосування еволюційних алгоритмів для побудови систем запобігання витоку даних.

Сучасний ландшафт методів виявлення витоків даних формують два такі підходи: контент-орієнтований аналіз, що зосереджується на вмісті інформаційних об’єктів; контекст-орієнтований аналіз, який досліджує обставини та шаблони роботи з даними.

Контент-орієнтовані методи базуються на припущенні, що конфіденційність інформації визначається її змістом. Такі системи аналізують текстові документи, бази даних, мультимедійні файли та комунікаційні потоки, застосовуючи регулярні вирази, словники ключових слів, лінгвістичні моделі та алгоритми машинного навчання для ідентифікації чутливих даних. Контекст-орієнтовані підходи, натомість, зосереджуються на метаданих, поведінкових шаблонах користувачів, часових характеристиках доступу та мережних аномаліях. Ці методи здатні виявляти загрози навіть без прямого аналізу вмісту, що робить їх ефективними проти зашифрованих

каналів витоку та обфускованих даних. На практиці більшість сучасних ЗВД-систем поєднують обидва підходи, прагнучи досягти балансу між точністю виявлення та обчислювальною ефективністю та фокусуються на проактивному запобіганні небажаним інформаційним потокам [30; 32].

Концепція контекстуальної цілісності VACCINE [29] пропонує теоретичний фундамент для оцінки прийнятності інформаційних потоків з урахуванням соціальних норм та очікувань. Одним із перспективних напрямків контент-орієнтованого аналізу є застосування графових структур для моделювання інформаційних потоків. Метод Adaptive Weighted Graph Walk (AGW), запропонований у роботі [31], представляє корпоративну інформаційну систему як зважений граф, вершинами якого є користувачі, документи та системні ресурси, а ребрами – зв'язки доступу та передачі даних. Алгоритм виконує адаптивний обхід графа, динамічно коригуючи ваги ребер залежно від історії взаємодій та контекстних факторів. Перевагою AGW є лінійна обчислювальна складність відносно кількості вершин, що забезпечує масштабованість для великих корпоративних середовищ. Метод демонструє високу ефективність у виявленні аномальних закономірностей доступу, проте його продуктивність суттєво залежить від якості початкової параметризації вагової функції.

Альтернативний підхід до ідентифікації джерел витоків пропонує ймовірнісно-біграфова модель [33], яка зосереджується на встановленні авторства та походження документів. Метод використовує біграфове представлення, де один тип вершин відповідає акторам системи (користувачам, процесам, сервісам), а інший – інформаційним об'єктам. Ймовірнісний механізм виведення дає змогу оцінити апостеріорну ймовірність того, що конкретний актор є джерелом певного документа, враховуючи стилістичні особливості тексту, часові шаблони та історію взаємодій. Такий підхід особливо корисний для розслідування витоків після інциденту, коли необхідно встановити відповідальну особу серед багатьох користувачів з легітимним доступом.

Семантико-орієнтований метод виявлення витоків [34] формує акцент на глибинному розумінні змісту документів через онтологічні моделі. Система будує формальну онтологію предметної області організації, визначаючи концепти, їх властивості та взаємозв'язки, після чого класифікує документи за рівнем

конфіденційності на основі семантичної близькості до захищених концептів. Перевагою цього підходу є стійкість до парафразування та синонімічних замін, тобто типових технік обходу простих контент-фільтрів. Водночас побудова та підтримка актуальної онтології потребує значних експертних зусиль, а семантичний аналіз природної мови залишається обчислювально інтенсивним завданням.

Різновидом контент-орієнтованого аналізу є метод AD-PROM [35], спрямований на виявлення витоків через канал SQL-запитів до баз даних. Система моделює нормальну поведінку користувачів при формуванні запитів за допомогою прихованих марковських моделей, фіксуючи типові послідовності операцій, структуру запитів та обсяги запитуваних даних. Відхилення від встановленого профілю сигналізує про потенційно зловмисну активність, таку як масова вибірка даних або доступ до нетипових таблиць. AD-PROM ефективно доповнює традиційні механізми контролю доступу до баз даних, додаючи поведінковий рівень захисту, проте потребує тривалого періоду навчання для побудови репрезентативних профілів.

На відміну від пасивних методів моніторингу, концепція активного захисту передбачає протидію витокам через контроль над сховищем даних. Підхід DLPFS [36] реалізує файлову систему з вбудованими механізмами запобігання витокам, яка забезпечує прозорий моніторинг операцій без модифікації прикладного програмного забезпечення. Завдяки віртуальному шару між додатками та фізичним сховищем система перехоплює файлові операції. Кожна з них фіксується та аналізується відповідно до політик ЗВД. Такий підхід гарантує повноту моніторингу та забезпечує мінімальний вплив на продуктивність системи при оптимізованій реалізації.

Розширення контролю над інформаційними потоками забезпечується розміщенням модуля ЗВД на рівні гіпервізора. DLP-Visor [37] забезпечує моніторинг всіх операцій гостьових операційних систем, зокрема роботу з пам’яттю, мережний трафік та дискові операції. Гостьова ОС принципово не зможе обійти цей контроль, навіть привілейовані процеси не мають доступу до гіпервізорного шару. Але впровадження DLP-Visor потребує суттєвої перебудови IT-інфраструктури та може спричиняти помітні накладні витрати продуктивності для інтенсивних обчислювальних навантажень.

Простішим, але практичним рішенням є механізм часових міток конфіденційності [38], який асоціює з кожним документом криптографічно захищену

мітку, що фіксує рівень секретності та термін її дії. Система автоматично застосовує політики безпеки залежно від мітки і автоматично знімає обмеження після закінчення терміну конфіденційності. Метод вдало інтегрується з наявними системами документообігу та не потребує значних інфраструктурних змін, хоча його ефективність обмежена сценаріями, де конфіденційність має чітко визначений часовий характер.

Зростання обсягів корпоративних даних та складний характер їх використання стимулювали розвиток ЗВД-методів, адаптованих до парадигми великих даних. Модель Big Data DLP [39] пропонує розподілену архітектуру аналізу інформаційних потоків, здатну обробляти терабайти даних у наближеному до реального часу режимі. Попереднє потокове фільтрування та виділення потенційно чутливих об’єктів застосовується перед поглибленим аналізом на кластерній інфраструктурі. Горизонтальне нарощування обчислювальних ресурсів забезпечує масштабованість, проте впровадження потребує значних інвестицій у інфраструктуру та експертизу роботи з розподіленими системами.

Перспективним напрямком застосування машинного навчання до задач ЗВД є аналіз журналів подій. Метод на основі ансамблю дерев рішень використовує Random Forest для класифікації записів системних журналів за ознакою наявності індикаторів витоку [40]. Алгоритм автоматично виділяє найбільш інформативні ознаки з необроблених журнальних даних, включаючи часові шаблони, ідентифікатори ресурсів, обсяги операцій та послідовності подій. Random Forest демонструє стійкість до шуму в даних та здатність обробляти великі простори ознак, характерні для журнальної інформації. Аналіз журналів безпеки [41] є важливим компонентом сучасних систем моніторингу і дає змогу виявляти аномалії через кореляцію подій з різних джерел. Обмеженням методу є інтерпретованість, тобто коли складні ансамблеві моделі не завжди дають змогу зрозуміти причини конкретного рішення, що ускладнює налагодження та обґрунтування реакцій системи.

Інтерпретованою альтернативою ансамблям традиційно вважають індукцію продукційних правил. Алгоритми CN2 [132] та RIPPER [133] будують набір жадібним послідовним покриттям. Чергове правило нарощується доти, поки покриває прийнятну частку прикладів, після чого покриті об’єкти вилучаються, а пошук триває на залишку. Рішення тут локальні, і сумарна складність набору не виступає

самостійним критерієм оптимізації, тому на зашумлених даних набори схильні розростатися, втрачаючи ту саму прозорість, заради якої їх обирали. Крок до глобальної оптимізації зроблено в дослідженнях з генетичного відбору правил нечіткої бази знань, збалансованої за точністю та компактністю [131], проте там ідеться про відбір із наперед сформованої бази, а нечітка форма правил менш пряма для політик безпеки ЗВД. Для задач цього класу потрібен глобальний пошук компактних наборів чітких правил безпосередньо в термінах політик безпеки.

Значний виклик для методів машинного навчання у сфері ЗВД становить незбалансованість навчальних даних, адже інциденти витоку трапляються набагато рідше за нормальну активність. Підхід SMOTE [42] вирішує цю проблему, поєднуючи аналітику поведінки користувачів із технікою синтетичної генерації прикладів рідкісного класу. Система моделює типові профілі поведінки для кожного користувача та організаційної одиниці, фіксуючи часові ритми роботи, типові ресурси доступу та комунікаційні шаблони. Штучне збагачення навчальної вибірки синтетичними прикладами аномалій покращує здатність класифікатора розпізнавати рідкісні, але критичні події витоку.

Окремою загрозою, що потребує спеціалізованих методів виявлення, є програми-вимагачі, які шифрують корпоративні дані з метою викупу. Система waRLOCK [43] використовує методи навчання з підкріпленням для адаптивного реагування на атаки програм-вимагачів. Агент навчається оптимальній стратегії протидії через взаємодію з симульованим середовищем, що моделює різноманітні сценарії атак. Такий підхід дає системі змогу пристосовуватися до нових варіантів програм-вимагачів без ручного оновлення сигнатур, хоча ефективність у реальних умовах залежить від репрезентативності симуляційного середовища. Систематичні огляди методів протидії програмам-вимагачам [44] наголошують на необхідності багаторівневого захисту. Підходи на основі аналізу шаблонів доступу до файлів [45] та систематичного моніторингу мережної активності [46] доповнюють арсенал методів виявлення. Колаборативні підходи до виявлення вторгнень [47] демонструють ефективність розподіленої архітектури систем аналізу загроз.

Прогностичний підхід до ЗВД [48] переносить фокус з реактивного виявлення на превентивне прогнозування ризиків. Метод застосовує регресійні моделі для оцінки ймовірності витоку на основі агрегованих індикаторів: рівня доступу

користувачів, історії порушень політик, характеру оброблюваних даних та зовнішніх факторів ризику. Система формує рейтинг ризиковості для користувачів та інформаційних ресурсів, що дає службам безпеки змогу зосередити увагу на найбільш вразливих ділянках. Методи аналізу послідовностей дій [49] розширюють можливості прогнозування, виявляючи характерні закономірності поведінки, що передують інцидентам витоку. Поведінковий аналіз загроз [50] уможливлює побудову профілів нормальної активності користувачів та ідентифікацію відхилень, що можуть свідчити про підготовку до ексфільтрації даних. Аналіз поведінкових ритмів користувачів [51] та моніторинг операцій із файловими системами [52] доповнюють цей підхід на рівні конкретних технічних механізмів. Прогностичний підхід особливо цінний для великих організацій, де суцільний моніторинг усіх інформаційних потоків є практично нездійсненним.

В гібридних системах з багаторівневим виявленням витоків даних, виділяють головні компоненти успішних інтеграцій, а саме: рівень збору даних з множинних джерел, рівень нормалізації та корелювання подій, аналітичний рівень з паралельним застосуванням різнорідних методів, рівень прийняття рішень з механізмами консолідації висновків [19]. Такі системи досягають вищої точності виявлення завдяки взаємному доповненню методів, проте їх складність суттєво підвищує вимоги до ресурсів та експертизи адміністрування. Застосування блокчейн-технологій для захисту IoT-середовищ [53] та методи аналізу аномалій для Інтернету речей [54] відкривають нові напрямки розвитку гібридних систем. Виявлення zero-day загроз методами машинного навчання [55] доповнює арсенал контекстних методів захисту.

Мобільний агентний підхід [56] спрямований на захист географічно розподілених та гетерогенних інформаційних систем. Система розгортає автономні програмні агенти, які мігрують між вузлами мережі, збираючи телеметрію та виконуючи локалізований аналіз. Агентна архітектура забезпечує стійкість до часткових відмов мережі та зменшує навантаження на центральну інфраструктуру аналізу. Разом з тим, мобільність агентів породжує додаткові виклики безпеки – необхідність захисту самих агентів від компрометації та гарантування цілісності зібраних ними даних.

Модель C2M2 [25] представляє собою інтегровану модель оцінювання зрілості можливостей захисту інформації організацій, включаючи спроможності ЗВД. На

відміну від технічних методів виявлення, C2M2 зосереджується на організаційних, процесних та управлінських аспектах захисту від витоків. Модель визначає рівні зрілості для різних доменів: управління активами, контролю доступу, моніторингу подій, реагування на інциденти та інших. Застосування C2M2 допомагає організаціям виявити прогалини у системі захисту та формувати обґрунтовані плани розвитку. Обмеженням моделі є її узагальнений характер, коли конкретні технічні рекомендації потребують адаптації до специфіки кожної організації.

Систематизоване зіставлення розглянутих методів дає змогу сформувати висновки щодо їх відносні переваги та обмежень, що важливо для обґрунтованого вибору при проектуванні ЗВД-систем. В таблиці 1.1 узагальнено основні характеристики кожного підходу.

Огляд методів виявлення витоків даних (табл. 1.1) формує кілька системних обмежень таких методів. Передусім, більшість запропонованих підходів оцінювалися на обмежених або синтетичних наборах даних, що ставить під сумнів їх ефективність у реальних корпоративних середовищах з їх гетерогенністю, масштабом та динамічністю. Відсутність стандартизованих систем оцінки для ЗВД-систем ускладнює об’єктивне порівняння методів та відтворення заявлених результатів.

Суттєвою проблемою залишається баланс між повнотою виявлення та рівнем хибних спрацювань. Методи з високою чутливістю генерують значну кількість хибнопозитивних сповіщень, що призводить до втоми аналітиків безпеки та потенційного ігнорування реальних загроз. І навпаки, консервативні налаштування підвищують ризик пропуску справжніх інцидентів. Адаптивне балансування цих характеристик залежно від контексту та профілю ризиків організації залишається відкритим дослідницьким питанням. Методи адаптивних випадкових лісів [57] та ансамблеві підходи для роботи з дрейфом концепції [58; 59] пропонують механізми підвищення стійкості класифікаторів до змін характеристик даних з часом.

Проблема дрейфу концепції досліджується як у контексті моніторингу змін розподілу даних [60], так і з позиції адаптації моделей машинного навчання до нових умов [61]. Огляди методів виявлення деградації моделей [62] та рекурентного дрейфу у потокових даних [63] формують теоретичне підґрунтя для розробки стійких класифікаторів.

Таблиця 1.1 - Порівняння методів виявлення витоків даних

Метод	Вид	Переваги	Недоліки
AGW [31]	Контекстний	Лінійна складність, масштабованість	Чутливість до параметризації
Біграфова модель [33]	Гібридний	Ефективна ідентифікація джерела	Обмежена застосовність для реального часу
Семантичний метод [34]	Контентний	Стійкість до парафразування	Високі витрати на побудову онтології
AD-PROM [35]	Контекстний	Поведінковий захист БД	Тривалий період навчання
DLPFS [36]	Контентний	Прозорий контроль на рівні ФС	Обмеження одним вузлом зберігання
DLP-Visor [37]	Гібридний	Неможливість обходу з гостьової ОС	Значні інфраструктурні зміни
Часова мітка [38]	Контентний	Простота інтеграції	Обмеженість часовими сценаріями
Big Data DLP [39]	Гібридний	Масштабованість до терабайтів	Високі інфраструктурні вимоги
Log-oriented RF [40]	Контекстний	Робастність, автоматичний вибір ознак	Обмежена інтерпретованість
UEBA/SMOTE [42]	Контекстний	Подолання незбалансованості даних	Якість синтетичних прикладів
waRLOCK [43]	Контекстний	Адаптація до нових програм-вимагачів	Залежність від симуляційного середовища
Прогностичний метод [48]	Гібридний	Превентивний характер	Залежність від якості індикаторів
MLDED [19]	Гібридний	Взаємне доповнення методів	Складність адміністрування

Окремої уваги потребує питання, як система взагалі дізнається, що дані змінилися. Класичні детектори дрейфу ADWIN [134], DDM [135], EDDM [136], KSWIN [137] стежать здебільшого за потоком помилок класифікатора або за одновимірними статистиками окремих величин. Така схема передбачає, що правильні

відповіді надходять оперативно, тоді як у ЗВД-контурі мітки істинності з’являються із затримкою або взагалі лише після розслідування інциденту. Реакція за помилками до того ж є реакцією постфактум: зміну фіксують тоді, коли якість уже просіла, тобто частину витоків пропущено. Додає труднощів і чутливість до поодиноких флуктуацій, через яку тривога може виникати без реальної зміни розподілу. Адаптивні ансамблі та методи навчання на нестаціонарних потоках [57; 58; 59] помітно стійкіші до дрейфу, однак досягають цього ціною втрати інтерпретованої структури моделі. Для ЗВД доцільніше стежити безпосередньо за розподілом самих ознак і вимагати підтвердження зміни, а адаптацію проводити так, щоб політика лишалася читабельною для аудиту.

Технологічний прогрес безперервно породжує нові канали потенційних витоків, а саме: хмарні сховища, мобільні пристрої, платформи співпраці, інтернет речей. Більшість розглянутих методів проектувалися для традиційних корпоративних периметрів і потребують суттєвої адаптації до сучасної розподіленої архітектури. Захист даних у гібридних та мультихмарних середовищах, де межі організації розмиваються, становить один з найактуальніших викликів.

Виявлення інсайдерських загроз потребує глибокого розуміння нормальної поведінки кожного користувача та здатності розпізнавати тонкі відхилення, що виходять за межі можливостей переважної більшості сучасних систем. Огляд генетичних алгоритмів [64] засвідчує їх ефективність для задач оптимізації в умовах невизначеності. Дослідження теоретичних основ генетичних операторів [65] та застосування еволюційних алгоритмів до інженерних задач [66] підтверджують широту їх можливостей. Багатокритеріальна оптимізація на основі генетичних алгоритмів [67] та дискретні процедури селекції [68] створюють методологічний фундамент для розробки адаптивних ЗВД-систем. Еволюційні алгоритми для задач реального часу [69] доповнюють цю методологічну базу. Диференціальна еволюція [70; 71] та генетичні алгоритми для задач маскування даних [72] і криптографії [73] демонструють широту застосувань еволюційних методів у суміжних галузях. Дослідження взаємозв’язку еволюційних алгоритмів та нейронних мереж [74] відкривають перспективи гібридних підходів до виявлення загроз. Пошук вразливостей за допомогою генетичних алгоритмів [75] ілюструє здатність еволюційних методів працювати зі складними багатовимірними просторами рішень.

Еволюційні та адаптивні алгоритми, здатні до безперервного навчання та самокоригування, видаються перспективним напрямком подолання цього обмеження, що обґрунтовує доцільність подальшого дослідження їх застосування у контексті ЗВД.

1.2 Особливості забезпечення безпеки корпоративних мереж

Середовищем, у якому розглядають задачу запобігання витокам, слугує корпоративна система, узагальнена структура (рис. 1.1) якої охоплює кілька взаємопов’язаних сегментів. Сегмент автоматизованих робочих місць користувачів взаємодіє із серверним сегментом, до якого входять файлові сервери, системи керування базами даних і сервери застосунків. Окремо виділяють поштову інфраструктуру та шлюз безпеки на периметрі мережі, через який проходить зовнішній трафік. До цього ядра долучаються хмарні сервіси й віддалені або мобільні робочі місця, що розмивають чіткий периметр захисту. Документообіг у такому середовищі зосереджено навколо трьох осередків: файлових сховищ, електронної пошти, хмарної синхронізації. І саме на цих напрямах виникає більшість каналів можливого витоку. Узагальнену структуру такої системи разом з основними точками контролю ЗВД зображено на рисунку 1.1. Організації середнього та великого масштабу з централізованим адмініструванням потребують таких ЗВД-систем.

Об’єктом захисту є конфіденційні дані, подані у формі документів, наприклад, файлів, повідомлень електронної пошти, записів баз даних та вмісту буфера обміну. Саме документ у русі контрольованими каналами обрано за базову одиницю захисту, адже витік стається тоді, коли носій інформації перетинає межу довіреного середовища, а не в момент її зберігання всередині нього. За змістом такі документи поділяють на персональні дані, фінансову інформацію, медичні дані, комерційну таємницю та інтелектуальну власність. Ці категорії неоднорідні за способом виявлення: структуровані ідентифікатори на кшталт номерів платіжних карток чи облікових кодів надійно розпізнаються за формальними шаблонами, тоді як неструктурований вміст вимагає класифікації за сукупністю ознак.

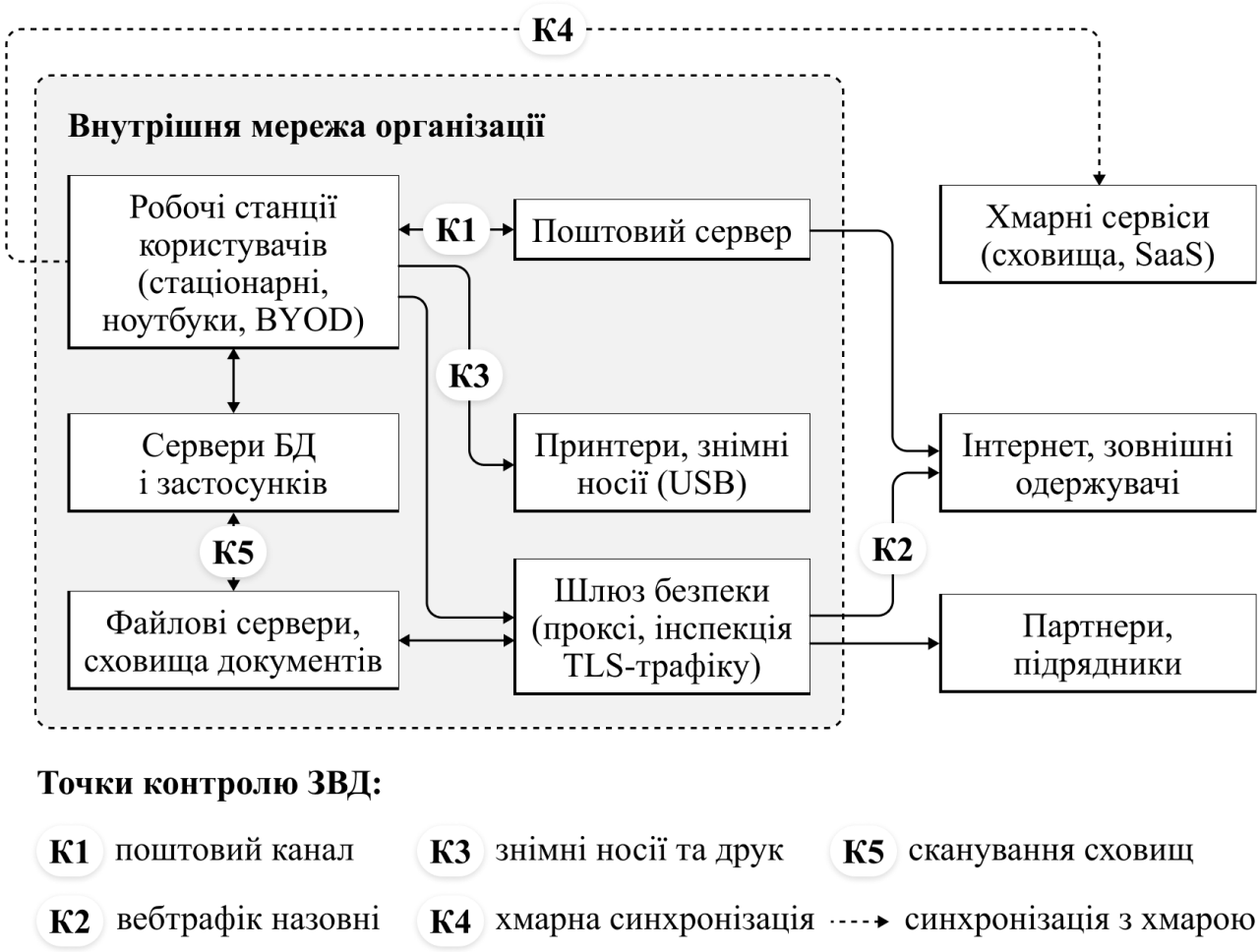


Рисунок 1.1 – Типова структура корпоративних мереж з точками контролю запобігання витокам даних

Розглянуто три групи сценаріїв витоку, кожну з яких співвіднесено з певними об’єктами захисту. Перша група охоплює передавання документа з конфіденційними даними контрольованим каналом, наприклад, електронною поштою, хмарною синхронізацією, завантаженням на знімний носій або друк, причому як навмисне, так і внаслідок помилки користувача.. Друга група пов’язана з поступовою зміною характеристик документопотоку та робочих процесів, тобто з дрейфом концепції, через який колись налаштовані правила з часом втрачають точність. Третя група стосується аномальної активності легітимного користувача, інсайдера, який готує або здійснює виведення даних у межах наданих йому повноважень.

Перелічені сценарії окреслюють межі області дослідження. Поза його розглядом залишаються атаки на інфраструктуру як таку (зловмисне програмне забезпечення, відмова в обслуговуванні), фізичні канали витоку на кшталт

фотографування екрана, стеганографічне приховування даних, а також компрометація облікових записів адміністративного рівня.

Окремої уваги потребує співвідношення розроблюваних засобів із криптографічними механізмами. Шифрування каналів зв'язку та сховищ надійно захищає дані від перехоплення сторонньою особою, проте не запобігає витоку через легітимного користувача, який має права доступу до відкритого вмісту. Тому аналіз ЗВД виконують у точках, де вміст доступний незашифрованим, тобто агентом на кінцевій точці до моменту шифрування або шлюзом з інспекцією TLS-трафіку. Криптографічні засоби, зокрема шифрування сховищ, криптографічно захищені часові мітки конфіденційності [38] та інтеграція шифрування із ЗВД-системами [81], розглянуто як суміжний рівень захисту, що доповнює контентно-поведінковий аналіз; власне криптографічні методи винесено за межі предмета дослідження.

Перенесення даних і обчислень в хмару та поширення периферійних обчислень створюють принципово нові виклики для систем захисту даних. Динамічний характер хмарних конфігурацій означає, що мережна топологія організації може змінюватися щохвилини, бо віртуальні машини створюються та знищуються, контейнери масштабуються залежно від навантаження, а мікросервіси взаємодіють через численні API-інтерфейси. Це визначає потребу в динамічному захисті хмарних середовищ, який здатен адаптуватися до постійних змін інфраструктури [76]. Статичні правила фільтрації та фіксовані політики контролю доступу виявляються неспроможними відстежувати ці трансформації в реальному часі, що потребує багаторівневого адаптивного підходу до захисту даних та управління правами [77; 78]. Аналіз журналів хмарної безпеки, в свою чергу, забезпечує механізми ретроспективного розслідування інцидентів, результати якого можуть бути використані для навчання ЗВД-систем [79].

Організації все активніше впроваджують інтелектуальні системи, що оперують не лише структурованими даними, а й складними семантичними зв'язками між інформаційними об'єктами. Ці зв'язки породжують нові вектори вразливостей, адже зловмисник може отримати конфіденційну інформацію не через пряме викрадення, а шляхом аналізу метаданих та непрямих кореляцій. Моделювання мережної залежності вимагає врахування складних топологічних структур та мультимережних графів, що значно ускладнює задачу виявлення потенційних каналів витоку [80].

Конфіденційно-орієнтовані генетичні алгоритми для хмарних середовищ [73] та інтеграція шифрування з ЗВД-системами [81] пропонують механізми захисту даних, що враховують специфіку семантичних зв'язків та метаінформації.

Залежність від сторонніх постачальників перетворилася на фактор ризику. Периферійні (edge) пристрої, VPN-шлюзи, CI/CD системи, як компоненти, часто надаються зовнішніми провайдерами, безпекова практика яких залишається поза межами контролю організації. Ця тенденція особливо небезпечна для галузей з підвищеними вимогами до конфіденційності, як-от медичні установи, що масово впроваджують хмарні сервіси без належної оцінки ризиків [9].

Людський фактор залишається найменш передбачуваним елементом системи безпеки. Технічні засоби захисту виявляються марними, коли керівництво не усвідомлює важливості безпекової культури або свідомо нехтує встановленими процедурами. Неетична поведінка співробітників, включаючи навмисне порушення політик безпеки, становить значну частку інцидентів з витоку інформації [82]. Ситуацію погіршує поширення парадигми BYOD, коли працівники використовують особисті пристрої для доступу до корпоративних ресурсів, значна кількість компаній дозволяє таку практику, проте багато з них не мають жодних формалізованих політик щодо захисту даних на особистих пристроях [83].

Масштаб організації створює специфічні безпекові виклики. Малий та середній бізнес стикається з ресурсною асиметрією: значна частка таких компаній не має спеціалістів з інформаційної безпеки [84]. Великі корпорації, попри значні інвестиції в безпеку, страждають від структурних ризиків. Дослідження 30 найбільших публічних компаній Німеччини виявило, що лише трохи більше половини інцидентів вдається виявити внутрішніми засобами моніторингу [85]. Безпека мультихмарних середовищ [86] та предиктивна аналітика для виявлення загроз [87] стають важливими компонентами корпоративної стратегії захисту. Аналіз поведінки користувачів [51] забезпечує емпіричну основу для розробки поведінкових моделей виявлення загроз. Динамічний захист персональних даних [88] інтегрує принципи privacy-by-design у архітектуру ЗВД-систем. Комплексний аналіз безпеки корпоративних мереж засвідчує необхідність переходу від реактивних до проактивних моделей захисту [89]. Класичні ЗВД-системи, побудовані на статичних

сигнатурах та незмінних політиках, не відповідають реаліям сучасного корпоративного середовища.

Поєднання технологічної складності, динамічної інфраструктури, людського фактору та різноманіття організаційних контекстів формує запит на принципово нові підходи до запобігання витоку даних. Системи захисту мають не просто реагувати на відомі загрози, а еволюціонувати разом з інфраструктурою, адаптуючи свої політики до змінних умов функціонування. Саме ця потреба в адаптивності обґрунтовує доцільність застосування еволюційних методів, у вигляді генетичних алгоритмів, для автоматизації процесів класифікації даних та оптимізації політик ЗВД-систем.

1.3 Засоби виявлення витоку даних в корпоративних мережах

Аналіз ринку ЗВД-систем від Gartner фіксує стійку тенденцію переходу від контент-центрованої архітектури до ризик-адаптивних моделей захисту [90]. Традиційний підхід, що спирався переважно на регулярні вирази, словники та цифрові відбитки документів, поступово віддає першість гібридним системам, які враховують контекст дій користувача, історію його поведінки та рівень довіри до конкретного співробітника. Така еволюція відображає усвідомлення індустрією того факту, що витік даних є не лише технічною подією передачі забороненого контенту, а є складним поведінковим процесом із множинними передумовами. Моделі впровадження ЗВД-систем [91] та поведінкові моделі на основі COM-B [92] пропонують структуровані підходи до організаційного прийняття технологій захисту даних.

Корпоративні ЗВД-платформи пропонують централізоване керування політиками через єдину консоль адміністрування, що допомагає формувати узгоджену стратегію захисту для всієї організації. Серед переваг таких рішень виділено такі: деталізовані дії реагування, тобто від попередження користувача до повного блокування; інтеграція з модулями аналітики поведінки користувачів і сутностей; можливість кореляції подій із різних каналів. Проте ці переваги супроводжуються суттєвими недоліками. Вартість розгортання та підтримки корпоративних платформ залишається високою [84], а складність налаштування вимагає залучення кваліфікованих фахівців.

Альтернативою виступають інтегровані комплекси, де функціональність запобігання витокам вбудована в інші засоби безпеки такі як мережні екрани, шлюзи електронної пошти, хмарні брокери доступу. Такий підхід зменшує початкові інвестиції, однак призводить до фрагментації консолей керування та необхідності ручної кореляції інцидентів між різними системами, що ускладнює оперативне реагування та комплексний аналіз загроз.

Аналіз провідних комерційних рішень демонструє різноманітність архітектурних підходів. Forcepoint DLP реалізує ризик-адаптивний захист, динамічно змінюючи рівень контролю залежно від поточної оцінки ризику користувача [93]. FortiDLP функціонує як SaaS-рішення з вбудованою поведінковою аналітикою та тісною інтеграцією з екосистемою продуктів Fortinet [94]. GTB Technologies спеціалізується на багатоканальному захисті з розширеними можливостями оптичного розпізнавання символів та ідентифікації документів за цифровими відбитками [95].

Окремий напрям представляють рішення, що акцентують увагу на поведінковій телеметрії. DTEX зосереджується на аналізі дій користувачів без глибокого дослідження контенту, що зменшує вплив на продуктивність і ризики для приватності [96]. CrowdStrike Falcon Data Protection розширює можливості платформи виявлення та реагування на кінцевих точках (EDR) функціями захисту даних, забезпечуючи єдиний агент для множинних завдань безпеки [97]. Cyberhaven DDR пропонує інноваційний підхід до відстеження повного ланцюжка життєвого циклу даних – від створення через трансформації до виходу за межі організації [98].

Екосистемні рішення великих вендорів також займають помітну частку ринку. Microsoft Purview забезпечує глибоку інтеграцію з Microsoft 365 та розвинені механізми автоматичної класифікації чутливих даних [99]. Symantec DLP включає модуль Information Centric Analytics для виявлення інсайдерських загроз на основі поведінкових аномалій [100]. Серед спеціалізованих рішень Nightfall позиціонується як хмарно-орієнтована (cloud-native) платформа з API-орієнтованою архітектурою [101], Proofpoint зосереджується на захисті каналу електронної пошти [102], а SkyGuard EDLP оптимізований для високонавантажених корпоративних середовищ [103].

Головні тенденції розвитку ринку визначаються кількома факторами. Ризик-орієнтований підхід стає домінуючою парадигмою, витісняючи статичні політики на основі виключно контентних правил. Машинне навчання дедалі активніше застосовується для класифікації даних, виявлення аномалій та зменшення кількості хибних спрацювань. Прогнози Gartner свідчать, що найближчим часом більшість корпоративних рішень інтегрують елементи адаптивного захисту як базову функціональність [90].

Попри технологічний прогрес, комерційні ЗВД-системи мають низку універсальних обмежень. Ефективність захисту залежить від якості початкової класифікації даних, серед яких неточна або неповна таксономія чутливої інформації призводить до пропуску реальних інцидентів або надмірного блокування легітимних операцій. Проблема хибних спрацювань залишається актуальною навіть для рішень із машинним навчанням, оскільки контекст бізнес-операцій часто неможливо повністю формалізувати. Інтеграція ЗВД у сучасні DevOps-процеси та CI/CD-конвеєри становить окремий виклик через специфіку автоматизованих потоків даних.

Узагальнене зіставлення розглянутих комерційних платформ і дослідницьких прототипів за моделлю розгортання, охопленням каналів, методами аналізу та механізмами адаптивності подано в таблиці 1.2. Таке зіставлення унаочнює розрив між ризик-орієнтованими деклараціями ринку та переважно статичними механізмами реалізації політик, який і мотивує цю роботу.

Таблиця 1.2 - Порівняння комерційних і дослідницьких ЗВД-платформ

Система (джерело)	Модель розгортання	Канали контролю	Методи аналізу	Механізми адаптивності
1	2	3	4	5
Forcepoint DLP [93]	локальна або гібридна	мережа, кінцеві точки, хмара, пошта	контентний аналіз, цифрові відбитки, OCR	ризик-адаптивні політики за оцінкою ризику користувача
FortiDLP (Fortinet) [94]	SaaS	кінцеві точки, хмарні застосунки	контентний аналіз, поведінкова аналітика	вбудована поведінкова аналітика

Продовження таблиці 1.2

1	2	3	4	5
GTB Technologies [95]	локальна або хмарна	мережа, кінцеві точки, пошта	цифрові відбитки, OCR, регулярні вирази	статичні політики з ручним налаштуванням
DTEX [96]	SaaS	кінцеві точки	поведінкова телеметрія без глибокої інспекції вмісту	поведінкові індикатори ризику
CrowdStrike Falcon Data Protection [97]	SaaS	кінцеві точки	контекстний аналіз потоків даних, спільний агент з EDR	кореляція з телеметрією EDR
Cyberhaven DDR [98]	SaaS	кінцеві точки, хмара	відстеження походження та життєвого циклу даних	динамічне простеження потоків даних
Microsoft Purview [99]	хмарна (Microsoft 365)	хмарні сервіси, пошта, кінцеві точки	автоматична класифікація, мітки чутливості	адаптивний захист за ризиком користувача
Symantec DLP (Broadcom) [100]	локальна або гібридна	мережа, кінцеві точки, хмара, пошта	контентний аналіз, цифрові відбитки	поведінковий модуль Information Centric Analytics
Nightfall [101]	SaaS (API)	хмарні застосунки	класифікація на основі машинного навчання	навчання детекторів
Proofpoint [102]	SaaS	пошта, хмара, кінцеві точки	контентний аналіз, аналітика поведінки користувачів	людино-центрична модель ризику

Кінець таблиці 1.2

1	2	3	4	5
SkyGuard EDLP [103]	локальна	мережа, кінцеві точки	контентний аналіз	статичні політики, оптимізація під високі навантаження
DLPFS (дослідницький прототип) [36]	локальна, рівень файлової системи	файлові операції	перехоплення файлових операцій згідно з політиками	статичні політики
DLP-Visor (дослідницький прототип) [37]	рівень гіпервізора	операції гостьових ОС	моніторинг пам’яті, трафіку та дискових операцій	статичні політики

Вибір конкретного рішення має враховувати масштаб організації, наявну інфраструктуру безпеки, регуляторні вимоги галузі та готовність до інвестицій у навчання персоналу. Для організацій із гетерогенним середовищем перевагу матимуть платформи з широким покриттям каналів, тоді як компанії з домінуванням хмарних сервісів можуть обрати хмарно-орієнтовані рішення з API-інтеграцією. Значним критерієм залишається здатність системи адаптуватися до еволюції загроз без постійного ручного переналаштування політик. Саме на подолання цього обмеження спрямовано еволюційно-адаптивний підхід, за якого класифікацію документів, політики безпеки та поведінкові профілі користувачів оптимізують генетичні алгоритми, а самі рішення безперервно пристосовуються до змін середовища, зберігаючи інтерпретовану форму правил.

1.4 Постановка задачі дослідження

Окреслене протиріччя між потребою в безперервно адаптивному захисті та переважно статичною природою наявних засобів визначає науково-прикладну задачу дослідження, завданнями якої є розроблення методів та програмних засобів виявлення і запобігання витоку даних у корпоративних мережах на основі еволюційних алгоритмів. Ці програмні засоби повинні забезпечити адаптивну

класифікацію документів за рівнем конфіденційності, автоматичну адаптацію політик безпеки за умов дрейфу концепції та поведінкове профілювання користувачів, не втрачаючи при цьому інтерпретованості ухвалюваних рішень. Вимога інтерпретованості є принциповою, бо результат роботи системи повинен залишатися придатним для перевірки й аудиту, а не зводитися до непрозорого рішення. Саме поєднання безперервної адаптивності з прозорістю ухвалюваних рішень визначає специфіку поставленої задачі.

Проведений аналіз сучасного стану досліджень у галузі запобігання витокам даних засвідчив наявність низки невирішених науково-технічних проблем. Наявні ЗВД-системи переважно спираються на статичні правила та заздалегідь визначені шаблони, що обмежує їхню ефективність в умовах динамічного корпоративного середовища. Недостатньо дослідженими залишаються механізми автоматичної адаптації політик безпеки до змін у характері даних та поведінці користувачів, а також методи оптимізації класифікаторів документів за умов дрейфу концепції. Це зумовлює актуальність розробки еволюційно-адаптивних підходів до побудови ЗВД-систем нового покоління.

Зіставлення запропонованого підходу з найближчими аналогами за розрізняльними ознаками наведено в таблиці 1.3.

Таблиця 1.3 - Розрізняльні ознаки запропонованого підходу та найближчих аналогів

Найближчий аналог	Основний принцип	Обмеження в задачі ЗВД	Відмінність запропонованого підходу
1	2	3	4
CN2 і RIPPER [132; 133]	Жадібне послідовне покриття, правила нарощуються по одному	Локальні рішення, сумарна складність набору не контролюється	Глобальний популяційний пошук цілісних наборів із регуляризацією складності
Генетичний відбір правил нечіткої бази [131]	Еволюційний відбір із наперед сформованої бази знань	Обмеження простором готових правил, нечітка форма умов	Еволюція чітких правил у просторі ознак політик безпеки

Кінець таблиці 1.3

1	2	3	4
Еволюційні системи правил GABIL, XCS [123; 124]	Еволюція наборів класифікаційних правил загального призначення	Не враховують специфіки ознак документів і аудиту	Ознаковий простір ЗВД, шаблонні детектори, спеціалізовані оператори
Ансамблеві класифікатори для журналів безпеки [40]	Агрегація багатьох дерев рішень	Висока точність без пояснення причин рішення	Прозорий набір правил за конкурентної якості розпізнавання
Детектори дрейфу ADWIN, DDM, EDDM, KSWIN [134; 135; 136; 137]	Контроль потоку помилок або одновимірних статистик	Потребують оперативних міток, реагують після деградації якості	Багатоознаковий тест на розподілі вхідних ознак із підтвердженням
Адаптивні ансамблі для потоків [57; 58; 59]	Безперервне оновлення ансамблю моделей	Втрата інтерпретованої структури політики безпеки	Еволюційне перенавчання правил із валідацією перед розгортанням

Метою дисертаційного дослідження є підвищення ефективності виявлення та запобігання витокам конфіденційної інформації в корпоративних мережах шляхом розробки еволюційно-адаптивних методів класифікації документів, поведінкового профілювання користувачів та оптимізації політик безпеки на основі генетичних алгоритмів, а також програмних засобів, що реалізують ці методи у складі єдиного програмного комплексу.

Для досягнення поставленої мети необхідно вирішити такі завдання:

1. Провести системний аналіз відомих методів та платформ ЗВД, формалізувати їхні функціональні обмеження за критеріями масштабованості, адаптивності та точності класифікації.

2. Розробити модель процесу запобігання витокам даних, що інтегрує модель витоку, модель корпоративного середовища та модель даних і враховує характеристики інформаційних ресурсів, профілі користувачів, рівні конфіденційності, показники довіри та оцінки потенційної шкоди від витоку.

3. Розробити метод класифікації документів за рівнем конфіденційності на основі генетичного алгоритму з хромосомним кодуванням продукційних правил, регуляризацією складності моделі та спеціалізованими генетичними операторами.

4. Розробити метод еволюційної адаптації політик безпеки за умов дрейфу концепції зі статистичним детектором змін, що забезпечує безперервне вдосконалення правил контентного та контекстного контролю з мінімізацією інтегральної функції втрат.

5. Розробити метод поведінкового профілювання користувачів з індивідуальними адаптивними профілями та генетичною оптимізацією складу ознак, їхніх ваг і порогів виявлення аномалій.

6. Спроектувати та реалізувати програмний прототип ЗВД-системи з модульною архітектурою і провести експериментальні дослідження запропонованих методів на відкритих корпусах і реальних наборах даних за показниками якості класифікації, інтерпретованості, обчислювальної вартості та стійкості до дрейфу концепції. За результатами досліджень сформулювати практичні рекомендації щодо інтеграції еволюційно-адаптивних компонентів у наявні корпоративні системи безпеки.

Послідовне виконання окреслених завдань забезпечить науково обґрунтований перехід від традиційної парадигми статичних ЗВД-систем до еволюційно-адаптивної моделі захисту, здатної автономно вдосконалювати власні механізми виявлення загроз відповідно до змін корпоративного інформаційного середовища.

1.5 Висновки до першого розділу

1. Аналіз сучасного ландшафту кіберзагроз виявив стійке зростання і частоти, і складності інцидентів, пов’язаних із витоком корпоративних даних. Переміщення інформаційних активів у хмарні та периферійні середовища, поширення мобільних платформ і гібридних моделей роботи істотно розширили потенційну поверхню атаки, через що традиційні підходи до захисту інформації потребують перегляду.

2. Систематизація наявних ЗВД-рішень показала, що жодна окрема технологія не здатна самотужки протидіяти всьому спектру загроз витоку даних. Перспективні системи захисту потребують поєднання поведінкової аналітики, семантичної

інспекції вмісту та контекстного моделювання дій користувачів, без чого прийнятного рівня виявлення досягти не вдається.

3. Порівняння архітектурних підходів засвідчило, що гібридні рішення, які поєднують мережний та агентський моніторинг, дають найкращий баланс між точністю виявлення і швидкодією. Ефективність такої архітектури прямо залежить від якості та повноти телеметрії, що надходить із різнорідних джерел корпоративної інфраструктури.

4. Дослідження ролі людського фактору підтвердило вагу інсайдерської активності як джерела витоків, яку посилюють практика використання особистих пристроїв та недостатня культура інформаційної безпеки. Ці обставини обґрунтовують перехід від статичних політик доступу до механізмів динамічної оцінки довіри до суб’єктів корпоративного середовища.

5. Аналіз практики впровадження комерційних ЗВД-платформ виявив суттєві бар’єри для їх повноцінного використання: високу вартість володіння, складність налаштування правил класифікації, фрагментованість ринку рішень. Найгостріше ці чинники даються взнаки малим і середнім підприємствам, яким зазвичай бракує ресурсів на підтримку комплексних систем захисту.

6. Обґрунтовано перспективність еволюційно-адаптивних методів для автоматичної оптимізації політик запобігання витокам даних у потоковому режимі. Безперервно припасовуючи параметри системи до змінюваних умов операційного середовища, такий підхід дає змогу мінімізувати сукупні втрати від хибних спрацювань і пропущених інцидентів.

7. Зіставлення виявлених потреб із можливостями наявних засобів увиразнює основне протиріччя: динамічне корпоративне середовище вимагає захисту, який безперервно адаптується і водночас пояснює свої рішення для аудиту, тоді як наявні системи покладаються або на статичні правила з ручним налаштуванням, або на непрозорі моделі машинного навчання. Розв’язання цього протиріччя визначає науково-прикладну задачу дослідження.

Основні результати розділу опубліковані у працях [104; 105; 106; 107; 108; 109; 110; 111; 112].

РОЗДІЛ 2
МОДЕЛЬ ПРОЦЕСУ ЗАПОБІГАННЯ ВИТОКАМ ДАНИХ У КОРПОРАТИВНИХ МЕРЕЖАХ

2.1 Модель даних та їх представлення для аналізу витоків

Ефективність системи запобігання витокам даних безпосередньо залежить від її здатності коректно ідентифікувати та класифікувати конфіденційну інформацію. Побудова моделі даних для ЗВД-системи вимагає врахування кількох суперечливих вимог. З одного боку, модель має бути достатньо виразною для захоплення семантики різноманітних типів конфіденційної інформації. З іншого боку, обчислювальна складність аналізу має залишатися прийнятною для застосування в реальному часі, коли система обробляє потоки документів на каналах передачі. Крім того, модель повинна підтримувати як структуровані дані з чіткими шаблонами, так і неструктуровані текстові документи, де чутливість визначається семантикою вмісту. Класифікацію основних типів конфіденційних даних у корпоративному середовищі наведено в таблиці 2.1.

Наведена класифікація демонструє, що різні категорії конфіденційних даних вимагають різних підходів до виявлення. Структуровані дані, такі як номери платіжних карток або ідентифікаційні коди, мають характерні формати, що ефективно описуються регулярними виразами з додатковою валідацією контрольних сум. Неструктуровані дані, наприклад, комерційні таємниці чи бізнес-стратегії, потребують семантичного аналізу вмісту та застосування методів машинного навчання. Ця дихотомія визначає архітектуру ознакового простору, де поєднуються ознаки на основі шаблонів та ознаки на основі статистичного представлення тексту.

Для автоматизації класифікації документів системою ЗВД необхідно формалізувати поняття категорії конфіденційних даних. Функція категоризації встановлює однозначну відповідність між документом та множиною категорій, до яких він належить.

Таблиця 2.1 - Класифікація типів конфіденційних даних у корпоративному середовищі

Категорія	Підкатегорія	Приклади	Методи виявлення	Регуляторні вимоги
1	2	3	4	5
Персональні дані (PII)	Прямі ідентифікатори	ПІБ, дата народження, ІПН, номери документів	Регулярні вирази, NER	GDPR, Закон України про захист персональних даних
	Контактні дані	Email, телефон, адреса	Регулярні вирази	GDPR
	Біометричні дані	Відбитки пальців, зображення обличчя	Аналіз формату файлів	GDPR (особлива категорія)
Фінансова інформація	Платіжні дані	Номери карток, CVV, терміни дії	Алгоритм Луна, регулярні вирази	PCI DSS
	Банківські реквізити	IBAN, номери рахунків	Регулярні вирази з валідацією	PCI DSS
	Корпоративна звітність	Баланси, P&L, прогнози	Семантичний аналіз, ключові слова	SOX
Медичні дані (PHI)	Діагнози та лікування	Історії хвороб, призначення	NER, медичні онтології	HIPAA
	Результати досліджень	Аналізи, знімки	Аналіз форматів DICOM, HL7	HIPAA
Комерційна таємниця	Технології та процеси	Рецептури, методики	Семантичний аналіз	Комерційне право
	Бізнес-стратегії	Плани розвитку, M&A	Ключові слова, класифікація	Комерційне право

Кінець таблиці 2.1

1	2	3	4	5
	Клієнтські бази	Контакти, історія взаємодії	Структурний аналіз БД	Комерційне право
Інтелектуальна власність	Вихідний код	Репозиторії, скрипти	Розпізнавання мов програмування	Авторське право, патенти
	Технічна документація	Специфікації, схеми	Семантичний аналіз	Патентне право
	Патентні матеріали	Заявки, дослідження	Ключові слова, класифікація	Патентне право

Класифікацію типів даних задамо функцією приналежності до категорій. Нехай K позначає множину основних категорій конфіденційних даних, а саме $K = \{PII, FIN, MED, TRADE, IP\}$. Тоді функцію категоризації визначимо так:

$$\kappa_{cat}: D \to 2^K, \tag{2.1}$$

де 2^K – множина всіх підмножин множини K.

Один документ може одночасно належати до кількох категорій: медична картка пацієнта містить як персональні дані, так і медичну інформацію, фінансовий звіт може охоплювати відомості про клієнтів, що становлять комерційну таємницю.

Чутливість у вигляді ваг категорій визначимо спираючись на регуляторні вимоги [115; 116] та внутрішню політику організації так:

$$S_{cat}: K \to [0,1], \quad S_{cat}(k) = w_k, \tag{2.2}$$

де S_{cat} – функція, яка кожній категорії конфіденційних даних зіставляє її вагу чутливості зі значеннями у відрізку [0,1];

K – множина основних категорій конфіденційних даних;

k – окрема категорія з цієї множини;

$S_{cat}(k)$ – вага, призначена категорії k;

w_k – вага категорії k, що відображає її відносну чутливість у політиці організації.

Початкові значення ваг задаються регуляторними вимогами й експертно, а далі автокоригуються за історією інцидентів.

Загальну чутливість документа, що належить до кількох категорій, визначимо як адитивно-імовірнісну агрегацію так:

$$S(d) = 1 - \prod_{k \in \kappa_{cat}(d)} \left(1 - w_k \cdot \nu_k(d)\right), \quad (2.3)$$

де $\nu_k(d) \in (0,1]$ – нормована насиченість категорії k у документі (через кількість входжень n_k: $\nu_k = min(1, n_k/n_{ref})$);

n_{ref} – поріг, при якому категорія вважається повністю насиченою, тобто її внесок у чутливість досягає максимуму.

Значення n_{ref} задається регуляторно ($n_{ref} = 1$ для критичних категорій) або статистично за історичними даними з подальшим автокоригуванням.

Для побудови системи аналізу та класифікації необхідно формалізувати поняття документа як базової одиниці обробки. У контексті ЗВД-системи документом вважається будь-який інформаційний об’єкт, що може бути переданий через контрольовані канали: файл, повідомлення електронної пошти, запис у базі даних, вміст буфера обміну тощо. Модель документа подаємо як кортеж, в якому d_{id} позначає унікальний ідентифікатор документа у системі (d_{id} належить I_{id}), d_{cont} представляє вміст документа як послідовність символів над алфавітом, d_{feat} визначає вектор ознак документа у m-вимірному просторі (d_{feat} належить R^m), d_{meta} містить структуру метаданих документа (d_{meta} належить M), а d_{sens} фіксує рівень чутливості як значення з інтервалу $[0, 1]$:

$$d = \langle d_{id}, d_{cont}, d_{feat}, d_{meta}, d_{sens} \rangle, \quad (2.4)$$

де d – документ як базова одиниця обробки у системі ЗВД;

d_{id} – унікальний ідентифікатор документа у системі, $d_{id} \in I_{id}$;

d_{cont} – вміст документа як послідовність символів над алфавітом;

d_{feat} – вектор ознак документа у m-вимірному просторі, $d_{feat} \in R^m$;

m – розмірність вектора ознак документа;

d_{meta} – структура метаданих документа, $d_{meta} \in M$;

d_{sens} – рівень чутливості документа зі значеннями з інтервалу [0, 1].

Вміст документа d_{cont} містить необроблені дані, що підлягають аналізу. Для текстових документів це послідовність символів Unicode, для бінарних файлів – послідовність байтів, яка може бути частково конвертована у текст залежно від формату. Необроблений вміст не використовується безпосередньо для класифікації: він слугує джерелом для видобування ознак, що відбувається на етапі попередньої обробки.

Вектор ознак $d_{feat} = (x_1, x_2, \dots, x_m)$ є результатом попередньої обробки та видобування характеристик документа. Саме цей вектор слугує вхідними даними для алгоритмів машинного навчання та класифікаторів. Розмірність m визначається обраним методом представлення та може варіюватися від сотень для методів на основі ключових слів до десятків тисяч для повних моделей «мішка слів» на великих корпусах.

Метадані охоплюють атрибути документа, які не є частиною його вмісту, але несуть важливу інформацію для класифікації:

$$d_{meta} = \langle m_{auth}, m_{crt}, m_{mod}, m_{fmt}, m_{size}, m_{src}, m_{perm} \rangle, \tag{2.5}$$

де d_{meta} – структура метаданих документа, введена в (2.4);

m_{auth} – ідентифікатор автора або останнього редактора документа;

m_{crt} – часова мітка створення документа;

m_{mod} – часова мітка останньої модифікації документа;

m_{fmt} – тип файлу через MIME-тип або розширення;

m_{size} – розмір документа у байтах;

m_{src} – джерело документа (локальний диск, мережний ресурс, email-вкладення);

m_{perm} – поточні права доступу до документу.

Модель документа підтримує ієрархічні та складні структури. Архівний файл, повідомлення електронної пошти з вкладеннями, контейнерний документ можуть містити вкладені документи. Функцією ν_{ch} відображаємо документ на множину його вкладених компонентів:

$$\nu_{ch}: D \to 2^D, \tag{2.6}$$

де ν_{ch} – функція вкладеності, яка кожному документу зіставляє множину його вкладених компонентів;

D – множина документів системи.

Для таких складених документів чутливість визначимо рекурсивно, як максимум між власною чутливістю та чутливістю всіх вкладених компонентів:

$$S(d) = \max\left(S_{self}(d), \max_{c \in \nu_{ch}(d)} S(c)\right), \quad (2.7)$$

де $S(d)$ – чутливість складеного документа d;

$S_{self}(d)$ – власна чутливість документа d без урахування вкладених компонентів;

$\nu_{ch}(d)$ – множина вкладених компонентів документа d;

c – окремий вкладений компонент із множини ν_{ch}(d);

$S(c)$ – чутливість вкладеного компонента c, а результатом слугує максимум серед власної чутливості документа та чутливостей усіх його вкладених компонентів.

Цей принцип гарантує, що конфіденційний вкладений файл у відкритому архіві чи листі буде коректно виявлений системою. Якщо працівник намагається вивести секретний документ, вклавши його в архів із незловмисними файлами, система розпізнає загрозу завдяки рекурсивному аналізу. Ієрархічну структуру складеного документа з рекурсивним обчисленням чутливості зображено на рисунку 2.1.

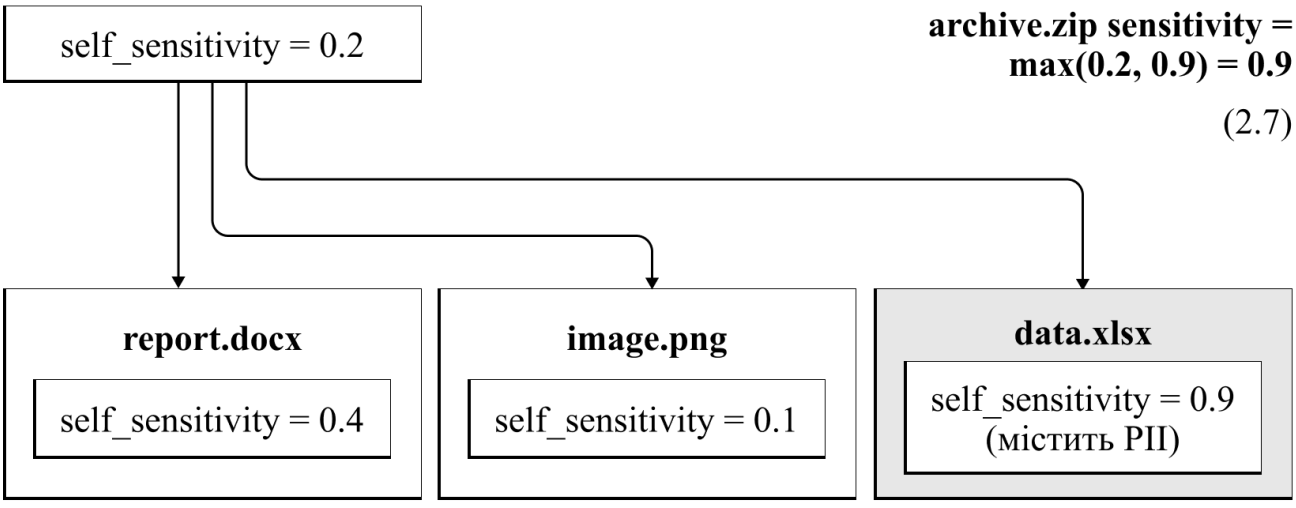


Рисунок 2.1 – Ієрархічна структура складеного документа з рекурсивним обчисленням чутливості

Перетворення документа з необробленого вмісту у вектор ознак, придатний для машинного навчання, є важливим етапом аналізу. Ознаковий простір системи ЗВД охоплює три основні категорії характеристик: текстові ознаки, що відображають

семантику вмісту; метадані, що описують атрибути документа; контекстні ознаки, що характеризують обставини передачі.

Текстові ознаки утворюють основу контентного аналізу та охоплюють різноманітні способи представлення лінгвістичного вмісту документа. Найпоширенішим підходом є видобування окремих термінів, де кожен унікальний терм t зі словника V стає потенційною ознакою. Значенням ознаки слугує частота терму у документі або його зважене представлення за схемою TF-IDF, яка надає вищі значення термам, що часто зустрічаються у конкретному документі, але рідко в корпусі загалом. Для документа d та терму t зважене значення визначимо як добуток нормалізованої частоти терму та логарифму оберненої частоти документів, де N позначає загальну кількість документів у корпусі, $n(t,d)$ – кількість входжень терму t у документ d, а $n_{df}(t)$ – кількість документів, що містять терм t:

$$\psi(t,d) = \frac{n(t,d)}{\sum_{t' \in d} n(t',d)} \cdot log \frac{N}{n_{df}(t)}, \quad (2.8)$$

де $\psi(t,d)$ – зважене значення ознаки терму t у документі d за схемою TF-IDF;

V – словник термів корпусу;

d – документ, для якого обчислюється зважене значення;

$n_{t,d}$ – кількість входжень терму t у документ d;

t' – терм, що пробігає всі терми документа d, а сума в знаменнику першого множника дає загальну кількість входжень усіх термів документа;

N – загальна кількість документів у корпусі;

$n_{df}(t)$ – кількість документів, що містять терм t.

N-грами розширюють підхід на основі окремих термів, розглядаючи послідовності з n суміжних елементів. Біграми та триграми здатні захоплювати локальний контекст та сталі словосполучення. Наприклад, біграма “номер картки” є значно інформативнішою для виявлення фінансових даних, ніж окремі слова “номер” та “картки”. Множину n-грам документа утворюють послідовності суміжних токенів, де $(w_1, w_2, \dots, w_{|d|})$ позначає послідовність токенів документа. Задамо її так:

$$\Gamma_n(d) = \{\langle w_i, w_{i+1}, \dots, w_{i+n-1}\rangle : i = 1, \dots, |d| - n + 1\}, \quad (2.9)$$

де $\Gamma_n(d)$ – множина n-грам документа d;

d – документ, для якого будується множина n-грам;

w_i – і-й токен послідовності токенів документа d;

$|d|$ – кількість токенів у документі d.

Ознаки на основі регулярних виразів відіграють особливу роль у виявленні структурованих конфіденційних даних. Номери кредитних карток, телефонів, адреси електронної пошти, ідентифікаційні коди мають характерні формати, які ефективно описуються шаблонами. Ознаки на основі шаблонів можуть бути бінарними індикаторами наявності або лічильниками входжень. Для шаблону p та документа d кількість входжень визначимо як потужність множини всіх збігів шаблону у вмісті документа так:

$$x_{pattern}(d,p) = |\{m: m \in входженням\, p\, у\, d_{cont}\}|, \quad (2.10)$$

де $x_{pattern}(d,p)$ – ознака, що дорівнює кількості входжень шаблону p у документ d;

d – документ, який аналізується;

p – шаблон (регулярний вираз) для виявлення структурованих даних;

m – окреме входження шаблону;

d_{cont} – текстовий вміст документа d, а значення ознаки дорівнює потужності множини всіх входжень p у цей вміст.

Типові шаблони для виявлення персональних даних охоплюють вирази для електронної пошти, номерів телефону , номерів кредитних карток з опціональними роздільниками, ідентифікаційних номерів платників податків ([0-9]{10}). Для платіжних карток додатково застосовується валідація за алгоритмом Луна, що відфільтровує випадкові числові послідовності.

Ознаки на основі ключових слів формуються зі списків термів, характерних для певних категорій конфіденційних даних. Словник ключових слів W_k для категорії k будується експертами або автоматично на основі аналізу корпусу відомих конфіденційних документів. Ознака наявності ключових слів категорії вимірює частку знайдених ключових слів від загального розміру словника:

$$x^k_{keyword}(d) = \frac{|v(d) \cap W_k|}{|W_k|}, \quad (2.11)$$

де $x^k_{keyword}(d)$ – ознака наявності ключових слів категорії k у документі d;

k – категорія конфіденційних даних;

d – аналізований документ;

$v(d)$ – множина термів, наявних у документі d;

W_k – словник ключових слів категорії k, значення ознаки становить частку ключових слів словника, знайдених у документі, тобто відношення потужності перетину $v(d) \cap W_k$ до потужності словника W_k.

Метадані документа надають цінну інформацію для класифікації, яка не залежить від аналізу вмісту. Атрибути авторства охоплюють інформацію про створювача та редакторів документа. Документи, створені керівництвом організації або працівниками певних підрозділів (юридичний, фінансовий, HR), апріорно мають вищу ймовірність містити конфіденційну інформацію. Ознаку авторства визначимо через функцію w_{sens}, що відображає авторів на рівень чутливості їхніх типових документів:

$$x_{author}(d) = w_{sens}\big(m_{auth}(d)\big), \tag{2.12}$$

де $x_{author}(d)$ – ознака авторства документа d;

d – сам документ;

$m_{auth}(d)$ – метадані авторства, тобто автор чи редактори документа;

w_{sens} – функція, що зіставляє авторові рівень чутливості його типових документів.

Часові атрибути можуть бути індикаторами актуальності та важливості. Нещодавно створені або модифіковані документи частіше містять поточну конфіденційну інформацію, тоді як старі документи можуть бути вже деактуалізованими. Тип файлу є важливим індикатором потенційного вмісту: електронні таблиці часто містять фінансові дані, презентації – стратегічну інформацію, бази даних – структуровані персональні дані. Розмір документа може корелювати з обсягом потенційно конфіденційної інформації, де логарифмування застосовується для зменшення впливу екстремальних значень.

Контекстні ознаки характеризують обставини, за яких документ передається або обробляється. Ці ознаки не є властивостями самого документа, але суттєво впливають на оцінку ризику витоку. Канал передачі визначає технічний механізм переміщення даних, де різні канали характеризуються різними профілями ризику. Електронна пошта на зовнішні домени, завантаження у публічні хмарні сховища, копіювання на USB-носії мають підвищений ризик порівняно з передачею всередині

корпоративної мережі. Роль відправника та отримувача впливає на оцінку легітимності передачі: передача фінансових документів від фінансового директора до аудитора є очікуваною, тоді як аналогічна передача від рядового працівника на особисту пошту виглядає підозрілою.

Загальний вектор ознак документа формується конкатенацією всіх груп ознак, де m_{text}, m_{meta} та $m_{context}$ позначають розмірності відповідних компонентів:

$$\boldsymbol{x}(d,c,a,t) = \left(x_{text}(d), x_{meta}(d), x_{context}(d,c,a,t)\right) \in \mathbb{R}^m, \quad m = m_{text} + m_{meta} + m_{context}, \tag{2.13}$$

де $x(d,c,a,t)$ – загальний вектор ознак документа d, утворений конкатенацією текстових, метаданих та контекстних ознак;

$x_{text}(d)$ – компонент текстових ознак;

$x_{meta}(d)$ – ознаки метаданих;

$x_{context}(d,c,a,t)$ – контекстний компонент, що залежить від документа d, каналу передачі c, актора a та моменту часу t;

R^m – простір ознак розмірності m;

m_{text} – розмірність текстового компонента;

m_{meta} – розмірність компонента метаданих;

$m_{context}$ – розмірність контекстного компонента, а загальна розмірність m дорівнює їхній сумі.

Перед видобуванням текстових ознак вміст документа проходить етап попередньої обробки, метою якого є нормалізація тексту та видалення шуму, що не несе семантичної інформації. Якість попередньої обробки безпосередньо впливає на ефективність подальшої класифікації, оскільки зашумлені або ненормалізовані дані призводять до розрідженого ознакового простору та погіршення узагальнювальної здатності моделей.

Токенізація є першим кроком обробки, що полягає у розбитті суцільного тексту на окремі токени: слова, числа, розділові знаки. Для європейських мов, включаючи українську, базова токенізація спирається на пробільні символи та розділові знаки як розділювачі. Проте проста токенізація за пробілами є недостатньою для мов з багатою морфологією та для текстів зі спеціальними форматами. URL-адреси, email, числа з роздільниками потребують спеціальної обробки. Для задач ЗВД особливу увагу слід приділяти збереженню структурованих даних під час токенізації: номер кредитної

картки “4111-1111-1111-1111” не повинен розбиватися на окремі числові фрагменти, оскільки саме повна послідовність є ідентифікатором конфіденційних даних. Тому токенізація для ЗВД часто виконується у два етапи: спочатку застосовуються регулярні вирази для виділення структурованих шаблонів, а потім залишковий текст токенізується стандартним чином.

Нормалізація регістру приводить усі літери до нижнього регістру, що усуває штучне розрізнення токенів за регістром написання. Ця трансформація доречна для загального аналізу тексту, проте для деяких шаблонів регістр є значущим, тому нормалізація застосовується вибірково.

Видалення стоп-слів усуває з тексту загальновживані службові слова, що не несуть семантичного навантаження: прийменники, сполучники, частки, допоміжні дієслова. Для української мови типовий список стоп-слів налічує 200-300 одиниць: “і”, “та”, “в”, “на”, “з”, “що”, “як”, “це”, “але”, “для” тощо. Видалення стоп-слів зменшує розмірність представлення та знижує шум, проте в окремих випадках стоп-слова можуть бути частиною значущих шаблонів.

Стемінг та лематизація спрямовані на приведення словоформ до канонічної форми, що дає змогу ототожнювати морфологічні варіанти одного слова. Стемінг відсікає закінчення та суфікси за евристичними правилами, отримуючи основу слова: “документ”, “документи”, “документа”, “документом” приводяться до стему “докумен”. Для української мови стемінг є складнішим, ніж для англійської, через багатшу морфологію. Лематизація є точнішим підходом, що спирається на морфологічний аналіз та словник для визначення леми – словникової форми слова. Лема дієслова є його інфінітивом, іменника – називним відмінком однини, прикметника – називним відмінком однини чоловічого роду. Лематизація потребує більших обчислювальних ресурсів та мовних ресурсів у вигляді морфологічних словників, але забезпечує вищу якість нормалізації.

Конвеєр попередньої обробки визначимо як композицію функцій трансформації, де кожен етап отримує результат попереднього:

$$П_p = f_{lem} \circ f_{stop} \circ f_{norm} \circ f_{tok} \circ f_{pat}, \quad (2.14)$$

де $П_p$ – конвеєр попередньої обробки документа, заданий композицією послідовних етапів;

f_{pat} – функція виділення структурованих шаблонів;

f_{tok} – токенізація тексту;

f_{norm} – нормалізація токенів;

f_{stop} – вилучення стоп-слів;

f_{lem} – лематизація;

$\circ$ – композиція функцій, за якої кожен етап застосовується до результату попереднього.

Результатом є нормалізована послідовність токенів та окремо набір виділених структурованих шаблонів, які формують відповідні групи ознак. Загальна схема роботи конвеєра обробки документів зображена на рис. 2.2.

Серед методів векторного представлення текстів базовим є підхід "мішка слів", що ігнорує порядок слів, зосереджуючись на частотних характеристиках. Незважаючи на простоту, цей метод забезпечує достатню точність для багатьох задач класифікації документів.

Модель «мішка слів» представляє документ як невпорядковану мультимножину слів, ігноруючи порядок та граматичну структуру. Словник $V = \{w_1, w_2, \dots, w_{|V|}\}$ формується з усіх унікальних токенів корпусу документів. Кожен документ представимо вектором розмірності $|V|$, де і-та компонента відповідає терму w_i:

$$\boldsymbol{x}^{BoW}(d) = \left(x_1, x_2, \dots, x_{|V|}\right), \quad x_i \in \{0,1, count(w_i, d), tf(w_i, d)\}, \tag{2.15}$$

де $x^{BoW}(d)$ – вектор подання документа d за моделлю «мішка слів»;

V – словник, утворений з усіх унікальних токенів корпусу;

x_i – i-та компонента вектора, що відповідає термові w_i;

w_i – i-те слово словника;

$|V|$ – потужність словника і тим самим розмірність вектора;

$count(w_i, d)$ – абсолютна кількість входжень терма w_i у документ d;

$tf(w_i, d)$ – нормалізована частота цього терма, кожна компонента набуває одного зі значень: 0 або 1 (індикатор наявності), абсолютної або нормалізованої частоти.

Компонента x_i може бути бінарним індикатором наявності терму, абсолютною частотою входжень або нормалізованою частотою. Модель «мішка слів» є простою

та ефективною, але має суттєві обмеження: втрата порядку слів унеможливлює розрізнення конструкцій з різним значенням, а великий словник призводить до розріджених векторів високої розмірності.

TF-IDF представлення вдосконалює «мішок слів» шляхом зважування термів за їхньою інформативністю. Терміни, що зустрічаються у багатьох документах, отримують низькі ваги, тоді як специфічні терміни – високі. Для задач ЗВД це означає, що характерні терміни конфіденційних документів (назви проектів, технологій, специфічна термінологія) матимуть вищу вагу порівняно із загальновживаною лексикою.

Нормалізація векторів ознак є важливим етапом підготовки даних для алгоритмів машинного навчання. Різні ознаки можуть мати різні масштаби: частота терму має значення від 0 до 1, розмір файлу варіюється від кілобайтів до гігабайтів. Без нормалізації ознаки з більшими абсолютними значеннями домінуватимуть у моделях, що базуються на відстанях або градієнтах. Min-max нормалізація приводить значення ознаки до діапазону [0, 1], де min_j та max_j позначають мінімальне та максимальне значення j-ї ознаки на навчальній вибірці:

$$x_j' = \frac{x_j - min_j}{max_j - min_j}, \tag{2.16}$$

де $x_{j'}$ – нормалізоване значення j-ї ознаки після мін-макс нормалізації зі значеннями у діапазоні [0, 1];

x_j – вихідне значення j-ї ознаки;

min_j – мінімальне значення j-ї ознаки на навчальній вибірці;

max_j – максимальне значення j-ї ознаки на навчальній вибірці.

Використаємо Z-score нормалізацію, щоб центрувати ознаки відносно нуля та масштабувати за стандартним відхиленням:

$$x_j' = \frac{x_j - \mu_j}{\sigma_j}, \tag{2.17}$$

де μ_j та σ_j – середнє та стандартне відхилення j-ї ознаки.

Після такої нормалізації ознаки мають нульове середнє та одиничну дисперсію, що покращує збіжність градієнтних методів оптимізації. Для виродженого випадку сталої ознаки (нульового стандартного відхилення) нормоване значення

встановлюється рівним нулю. Так само за рівності максимуму й мінімуму в мін-макс нормалізації.

L2-нормалізацію застосовуємо до всього вектора документа, приводячи його до одиничної евклідової норми:

$$\boldsymbol{x'} = \frac{\boldsymbol{x}}{\|\boldsymbol{x}\|_2} = \frac{\boldsymbol{x}}{\sqrt{\sum_j x_j^2}}, \tag{2.18}$$

де x' – нормалізований вектор ознак документа;

x – вихідний вектор ознак;

$\| x \|_2$ – евклідова (L2) норма вектора x;

x_j – j-та компонента вектора x.

Нормалізація ділить вектор на його норму, що дорівнює кореню із суми квадратів усіх компонент, приводячи результат до одиничної довжини. Ця нормалізація особливо корисна для обчислення косинусної подібності, яка є стандартною мірою схожості документів у векторному просторі. Для двох векторів x_1 та x_2 косинусна подібність визначається як їхній скалярний добуток, нормалізований добутком норм:

$$\cos(\boldsymbol{x_1}, \boldsymbol{x_2}) = \frac{\sum_j x_{1j} \cdot x_{2j}}{\sqrt{\sum_j x_{1j}^2} \cdot \sqrt{\sum_j x_{2j}^2}}, \tag{2.19}$$

де $cos(x_1, x_2)$ – косинусна подібність векторів ознак x_1 та x_2;

x_1 – вектор ознак першого документа;

x_2 – вектор ознак другого документа;

x_{1j} – j-та компонента вектора x_1;

x_{2j} – j-та компонента вектора x_2.

Значення косинусної подібності належить діапазону [0, 1] для невід'ємних векторів TF-IDF. Значення, близьке до 1, свідчить про схожість документів, близьке до 0 – про їхню несхожість. У контексті ЗВД косинусна подібність застосовується для порівняння нового документа з еталонними зразками конфіденційної інформації. Якщо схожість перевищує порогове значення, документ класифікується як потенційно конфіденційний. Якщо один із векторів нульовий, то подібність встановлюється рівною нулю.

Текстові дані є найбільш придатними для аналізу і охоплюють документи, вміст яких безпосередньо представлений як послідовність символів читабельного тексту. Структуровані дані зберігаються у форматах з чіткою схемою: бази даних, електронні таблиці, XML/JSON з визначеною структурою. Аналіз таких даних потребує врахування семантики полів та їхніх взаємозв'язків. Поле з назвою "SSN" або "Credit Card" у заголовку стовпця бази даних є сильним індикатором конфіденційності, навіть без аналізу вмісту. Повний конвеєр обробки документа, що перетворює необроблений текст на вектор ознак, зображено на рисунку 2.2.

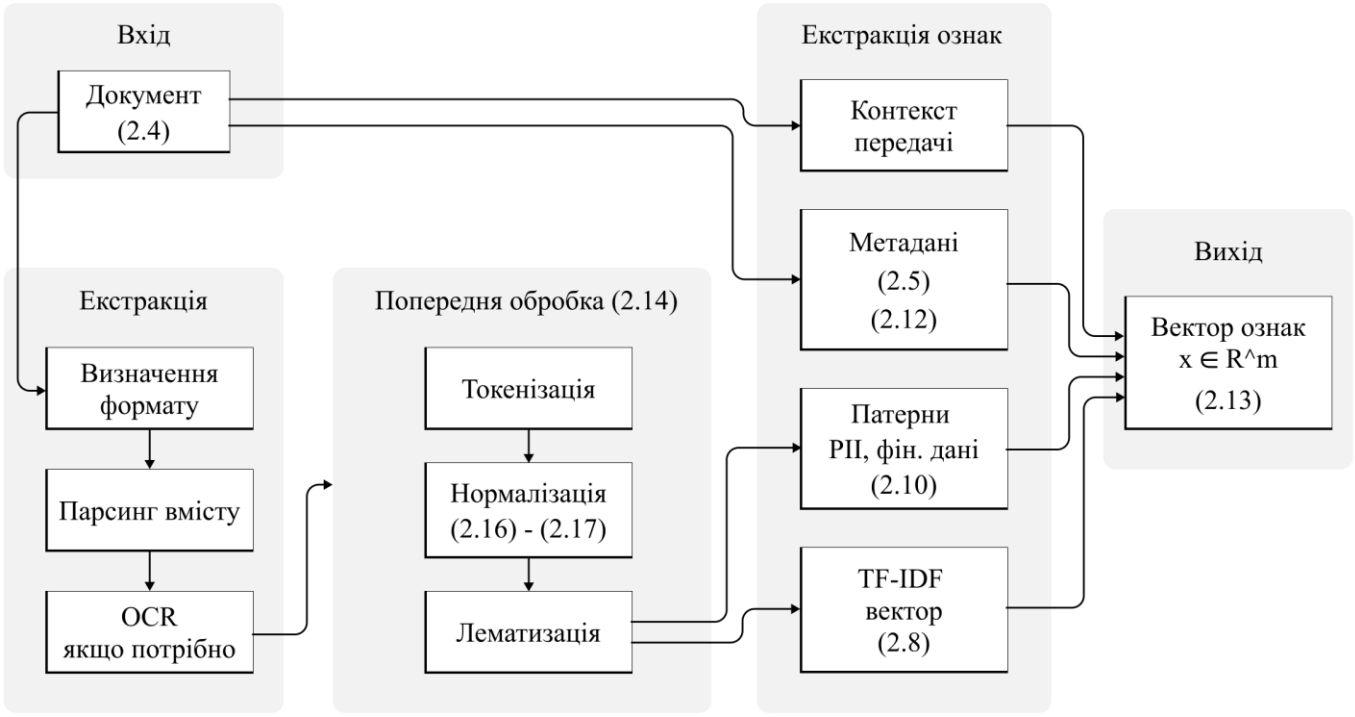


Рисунок 2.2 – Повний конвеєр обробки документа від необробленого тексту до вектору ознак

Для структурованих даних застосовуємо спеціалізовані методи, де f агрегує результати аналізу схеми, вмісту полів та виявлених шаблонів:

$$S_{struct}(d) = f\left(\varphi_{schema}(d), \varphi_{field}(d), \varphi_{pattern}(d)\right), \quad (2.20)$$

де $S_{struct}(d)$ – оцінка чутливості структурованих даних документа d;

d – документ зі структурованими даними;

f – функція, що агрегує окремі оцінки;

$\varphi_{schema}(d)$ – результат аналізу схеми даних;

$\varphi_{field}(d)$ – результат аналізу вмісту полів;

$\varphi_{pattern}(d)$ – результат аналізу виявлених шаблонів.

Бінарні дані охоплюють формати, що не мають безпосереднього текстового представлення: виконувані файли, мультимедійні дані, архіви, зашифровані дані. Можливості контентного аналізу для бінарних даних є суттєво обмеженими. Для виконуваних файлів можливий аналіз рядкових констант та метаданих компіляції, проте обфускація та пакування ускладнюють такий аналіз. Для зображень, окрім метаданих EXIF, контентний аналіз потребує застосування технологій комп'ютерного зору та OCR для розпізнавання тексту, що є ресурсомістким і може пропустити текст на складних фонах або у нестандартних шрифтах. Для архівів система має здійснювати рекурсивну декомпресію та аналіз вкладених файлів, а захищені паролем архіви не піддаються такому аналізу без знання пароля. Для зашифрованих даних контентний аналіз принципово неможливий, і система може лише фіксувати факт передачі зашифрованого вмісту та застосовувати альтернативні методи контролю.

Обмеження на аналіз задаємо функцією придатності, значення якої відображає частку вмісту, що піддається контентному аналізу (придатність різних типів даних до такого аналізу зіставлено в таблиці 2.2):

$$\eta_a: D \to [0,1], \tag{2.21}$$

де η_a – функція придатності документа до контентного аналізу зі значеннями у відрізку [0,1];

D – множина документів, значення функції дорівнює частці вмісту документа, доступній для контентного аналізу.

Для документів з низькою придатністю до аналізу система має покладатися на альтернативні індикатори: метадані, контекст передачі, поведінковий аналіз відправника. Агреговану оцінку ризику з урахуванням придатності обчислимо як зважену комбінацію контентного та контекстного ризиків:

$$R_{adj}(d) = \eta_a(d) \cdot R_{cont}(d) + \big(1 - \eta_a(d)\big) \cdot R_{ctx}(d), \tag{2.22}$$

де $R_{adj}(d)$ – скоригована агрегована оцінка ризику для документа d;

$\eta_a(d)$ – придатність документа до контентного аналізу, яка слугує ваговим коефіцієнтом;

$R_{cont}(d)$ – ризик, оцінений за вмістом;

$R_{ctx}(d)$ – ризик, оцінений за контекстом передачі;

$\left(1-\eta_a(d)\right)$ – множник, який задає вагу контекстної складової, тоді зі зниженням придатності зростає роль контексту.

Таблиця 2.2 - Придатність різних типів даних до контентного аналізу

Тип даних	Екстракція тексту	Патерни	Семантика	Аналізованість	Рекомендований підхід
Чистий текст	Повна	Повні	Повна	1,0	Контентний аналіз
Офісні документи	Висока	Повні	Повна	0,85-0,95	Екстракція + контент
PDF текстовий	Висока	Повні	Повна	0,80-0,95	Екстракція + контент
PDF сканований	OCR (помірна)	Часткові	Часткова	0,40-0,70	OCR + метадані
Бази даних	За полями	Повні	Схема + вміст	0,50-0,80	Структурний аналіз
Зображення	OCR (низька)	Низькі	Мінімальна	0,10-0,40	Метадані + контекст
Архіви	Рекурсивна	Залежить	Залежить	0,60-0,90	Декомпресія + аналіз
Зашифровані	Неможлива	Неможливі	Неможлива	0,0	Контекст + політика

Для зашифрованих даних, де $\eta_a(d)=0$, рішення повністю ґрунтується на контексті: хто передає, куди передає, у який час, з якого пристрою.

Оцінка чутливості документа є центральною задачею контентного аналізу в системі ЗВД. На відміну від бінарної класифікації “конфіденційно/неконфіденційно”, багаторівнева модель чутливості дає змогу диференціювати реакцію системи залежно від ступеня важливості даних. Функція чутливості $Sensitivity: D \rightarrow [0,1]$ відображає кожен документ на неперервну шкалу від 0 (публічна інформація без обмежень) до 1 (інформація найвищого ступеня конфіденційності). Цю функцію будуємо як композицію кількох оцінок.

Оцінка на основі категорії даних використовує результати класифікації за типами конфіденційних даних згідно з формулою (2.1). Оцінка на основі виявлених шаблонів враховує кількість та тип структурованих конфіденційних даних, знайдених

у документі, де P позначає множину шаблонів, w_p – вагу шаблону, а $n_p(d)$ – кількість його входжень:

$$S_{pattern}(d) = min\left(1, \sum_{p \in P} w_p \cdot n_p(d)\right), \quad (2.23)$$

де $S_{pattern}(d)$ – оцінка чутливості документа d за виявленими шаблонами;

P – множина шаблонів структурованих конфіденційних даних;

p – окремий шаблон з цієї множини;

w_p – вага шаблону p;

$n_p(d)$ – кількість входжень шаблону p у документ d, оцінка дорівнює зваженій сумі входжень усіх шаблонів, обмеженій зверху одиницею функцією min.

Функція мінімуму обмежує значення одиницею, запобігаючи виходу за межі інтервалу при великій кількості входжень.

Агреговану чутливість обчислюємо як зважену комбінацію компонентів, де ваги $\alpha_i \geq 0$ та їхня сума дорівнює одиниці:

$$S(d) = \sum_i \alpha_i \cdot S_i(d), \quad \sum_i \alpha_i = 1, \quad (2.24)$$

де $S(d)$ – агрегована чутливість документа d;

$S_i(d)$ – i-та компонента чутливості;

α_i – невід'ємна вага цієї компоненти, $\alpha_i \geq 0$;

i – індекс компонента, сумарна чутливість є зваженою сумою компонент, причому ваги α_i у сумі дають одиницю.

Формули (2.3), (2.7) і (2.24) описують той самий показник чутливості на різних рівнях деталізації. Категорійна агрегація (2.3) виступає однією з компонент S_i, а рекурсивне правило (2.7) поширює оцінку на складені документи.

Альтернативно може застосовуватися максимум серед компонентів, що реалізує принцип обережності: якщо хоча б одна компонента вказує на високу чутливість, документ вважається конфіденційним.

Для практичного застосування неперервну шкалу чутливості дискретизуємо у рівні класифікації $L = \{l_{pub}, l_{int}, l_{conf}, l_{sec}, l_{top}\}$ за допомогою порогових значень $0 < \tau_1 < \ldots < \tau_i < \ldots < \tau_n < 1$.

Рівні класифікації $L = \{l_{pub}, l_{int}, l_{conf}, l_{sec}, l_{top}\}$ не вводяться довільно, а спираються на чинну нормативну базу класифікації інформації. Закон України «Про

інформацію» за порядком доступу поділяє інформацію на відкриту та інформацію з обмеженим доступом, а останню розрізняє як конфіденційну, таємну та службову. Захист персональних даних регулює Закон України «Про захист персональних даних». Класифікацію інформаційних активів організації описують стандарти ISO/IEC 27001 та ISO/IEC 27002. У корпоративному контексті найвищі рівні чутливості відповідають комерційній таємниці, поняття якої визначає Цивільний кодекс України (стаття 505). Відповідність рівнів моделі та нормативних категорій подано в таблиці 2.3.

Таблиця 2.3 - Відповідність рівнів класифікації нормативним категоріям

Рівень моделі	Нормативна категорія	Джерело
l_{pub} (публічний)	Відкрита інформація	ЗУ «Про інформацію»
l_{int} (внутрішній)	Службова інформація	ЗУ «Про інформацію»
l_{conf} (конфіденційний)	Конфіденційна інформація; персональні дані	ЗУ «Про інформацію»; ЗУ «Про захист персональних даних»
l_sec, l_top (таємний, особливо таємний)	Таємна інформація; комерційна таємниця	ЗУ «Про інформацію»; ЦК України, ст. 505

Така прив'язка дає кожному рівневі чутливості зовнішнє нормативне обґрунтування, бо поріг класифікації документа спирається не лише на емпіричну оцінку, а й на правовий статус інформації, що підлягає захисту. Стандарти ISO/IEC 27001 та 27002 при цьому слугують методологічною основою для процедури класифікації активів у межах системи управління інформаційною безпекою.

Порогові значення налаштовуються відповідно до політики організації та бажаного балансу між безпекою та зручністю використання. Консервативніші пороги (нижчі значення) призводять до класифікації більшої кількості документів як конфіденційних, що підвищує безпеку ціною збільшення хибних спрацювань. Ліберальніші пороги (вищі значення) зменшують кількість хибних спрацювань, але підвищують ризик пропуску справжніх витоків.

Кожен рівень асоціюється з набором дозволених операцій та каналів передачі. Функцією Ψ_{ch} відображаємо рівень класифікації на множину каналів, через які дозволена передача:

$$\Psi_{ch}: L \to 2^C, \tag{2.25}$$

де Ψ_{ch} – функція, що кожному рівню класифікації зіставляє множину дозволених каналів передачі;

L – рівні класифікації;

C – множина каналів передачі.

Наприклад, документи публічного рівня можуть передаватися будь-якими каналами, внутрішнього – лише корпоративними каналами.

Запропонована модель даних та ознаковий простір для аналізу витоків мають кілька принципових відмінностей від підходів, описаних у літературі. Більшість відомих ЗВД-систем обробляють контентні, метаданні та контекстні ознаки окремими модулями з незалежними порогами прийняття рішень, тоді як розроблений ознаковий простір (формула (2.13)) об'єднує всі три категорії характеристик у єдиний вектор, що подається на вхід класифікатору. Механізм рекурсивного обчислення чутливості для складених документів, де чутливість контейнера визначається як максимум серед чутливостей його компонентів, є оригінальним рішенням проблеми обходу ЗВД через вкладення конфіденційних файлів у архіви. Функція придатності до аналізу забезпечує коректну деградацію оцінки для бінарних та зашифрованих даних: замість повного ігнорування таких об'єктів система переключається на контекстні та поведінкові індикатори. Дискретизація неперервної шкали чутливості у рівні класифікації із відповідними множинами дозволених каналів передачі забезпечує практичний інтерфейс між аналітичним модулем та модулем прийняття рішень.

2.2 Модель витоку даних

Більшість відомих моделей витоків даних розглядають інцидент як бінарну подію витоку або його відсутності, залишаючи поза увагою градації ризику, зумовлені характеристиками суб'єкта, об'єкта та каналу передачі. Також контекстне маркування даних розглядають ізольовано від оцінки поведінки актора. Відсутність єдиного формального апарату, який би пов'язував ці компоненти, призводить до розриву між теоретичними моделями та реальними потребами ЗВД-систем, що мають враховувати кумулятивний характер ризику. Формалізація витоку як багатовимірного

процесу, а не точкової події, дає змогу побудувати кількісну ризик-модель, придатну для автоматизованого прийняття рішень.

У контексті даного дослідження витік даних визначається як подія передачі конфіденційних даних за межі авторизованого периметру їх використання. Нехай D позначає множину об'єктів даних у корпоративній системі, A – множину акторів (суб'єктів), C – множину каналів передачі, а P – множину політик безпеки. Подію $e = (a, d, c, t)$ називаємо витоком даних, якщо виконується умова:

$$L_e \Leftrightarrow \neg Au(a, d, c) \wedge Tr(a, d, c, t), \tag{2.26}$$

де $Au(a, d, c)$ – предикат, що визначає наявність у актора a авторизації на передачу даних d через канал c згідно з чинними політиками Pol;

$Tr(a, d, c, t)$ – факт передачі даних у момент часу t, компоненти a, d, c, t тут є складниками самої події e, тому окремого квантування вони не потребують.

Предикат авторизації формально пов'язує витік із порушенням політик безпеки організації: $Au(a, d, c) \equiv \exists p \in Pol: Pr(p, a, d, c)$, де $Pr(p, a, d, c)$ означає, що політика p дозволяє актору a передавати дані d через канал c.

Функція конфіденційності даних визначає ступінь чутливості інформаційного об'єкта та є важливим елементом для прийняття рішень системою ЗВД. Задаємо її відображенням $S: D \to [0,1]$, де значення $S(d) = 0$ означає публічну інформацію без обмежень на розповсюдження, $S(d) = 1$ відповідає інформації найвищого рівня конфіденційності, а проміжні значення відображають градації чутливості. На практиці функція конфіденційності часто реалізується через класифікацію даних за дискретними рівнями: публічна інформація ($S = 0$), внутрішня ($S \approx 0{,}25$), конфіденційна ($S \approx 0{,}5$), таємна ($S \approx 0{,}75$) та особливо таємна ($S = 1$).

Різноманітність сценаріїв витоку даних у корпоративному середовищі потребує систематичної класифікації, яка б охоплювала основні характеристики інцидентів та слугувала основою для проектування захисних механізмів. Класифікація структурується за трьома ортогональними вимірами: джерело витоку, намір актора та канал реалізації. Узагальнену класифікацію типів витоків наведено у таблиці 2.4.

Таблиця 2.4 - Класифікація типів витоків даних у корпоративних мережах

Вимір	Тип	Підтипи	Характеристика
За джерелом	Внутрішній	Працівники, підрядники, партнери	Легітимний доступ, знання інфраструктури
	Зовнішній	Хакери, конкуренти	Подолання периметру, експлуатація вразливостей
За наміром	Умисний	Шпигунство, саботаж, фінансова вигода	Цілеспрямований вибір даних, маскування
	Ненавмисний	Помилки, недбалість	Хибний адресат, неправильні налаштування
За каналом	Мережний	Email, Web, Cloud, Messaging	Високий обсяг, можливість інспекції
	Локальний	USB, Print, Bluetooth	Фізичний доступ, агентський контроль
	Фізичний	Крадіжка пристроїв, винос документів	Поза цифровим контролем

Внутрішні витоки реалізуються особами, які мають легітимний доступ до інформаційних ресурсів організації. Інсайдерські загрози становлять особливу небезпеку з кількох причин: інсайдери володіють знанням про місцезнаходження цінних даних та способи обходу систем захисту, їхні дії часто виглядають як легітимна робоча активність, що ускладнює виявлення, і вони можуть мати прямий доступ до даних без необхідності долати технічні бар’єри. Зовнішні витоки здійснюються особами, які не мають авторизованого доступу до корпоративних ресурсів; такі інциденти зазвичай поєднуються з порушенням систем захисту і охоплюють хакерські атаки, експлуатацію вразливостей та соціальну інженерію для отримання облікових даних.

Умисні витоки здійснюються свідомо з метою заподіяння шкоди організації або отримання особистої вигоди. Мотивацією може бути фінансова зацікавленість (продаж даних конкурентам), помста (незадоволений працівник), ідеологічні міркування або шпигунство на користь третіх сторін. Такі витоки характеризуються

цілеспрямованим вибором цінних даних, застосуванням методів маскування та спробами приховати сліди. Ненавмисні витоки відбуваються внаслідок помилок, недбалості або недостатньої обізнаності персоналу: відправлення електронного листа з конфіденційним вкладенням хибному адресату, публікація внутрішнього документа у загальнодоступному хмарному сховищі, втрата незашифрованого носія інформації.

Мережні витоки реалізуються через канали передачі даних корпоративної мережі та мережі Інтернет: електронну пошту, хмарні сховища та файлообмінні сервіси, месенджери та засоби командної комунікації. Ці канали є найпоширенішим вектором витоку в сучасному корпоративному середовищі, що зумовлює пріоритетну увагу до їхнього моніторингу з боку ЗВД-систем. Локальні витоки здійснюються через периферійні пристрої та локальні інтерфейси робочих станцій: копіювання на знімні USB-носії, запис на оптичні носії, виведення на друк. Фізичні витоки пов’язані з матеріальними носіями інформації: крадіжка або втрата ноутбуків, мобільних телефонів з корпоративними даними, вилучення жорстких дисків або серверного обладнання.

Витік даних не є миттєвою подією, а становить процес, що розгортається у часі та проходить через послідовність етапів. Розуміння життєвого циклу витоку важливе для проектування ефективних захисних механізмів, оскільки різні етапи вимагають різних стратегій виявлення та протидії. Життєвий цикл витоку даних охоплює шість основних етапів: мотивація та планування, розвідка, отримання доступу, збір та агрегація даних, ексфільтрація та приховування слідів.

На етапі мотивації та планування формуються передумови для витоку. Для умисних витоків це може бути незадоволення працівника, отримання пропозиції від конкурента, фінансові труднощі або ідеологічні мотиви. Виявлення загрози на цьому етапі можливе через аналіз поведінкових індикаторів: зміни в робочих шаблонах, ознаки незадоволення, контакти з конкурентами. Етап розвідки передбачає ідентифікацію цінних даних та дослідження доступних шляхів їх отримання та виведення. Інсайдер визначає місцезнаходження цільової інформації, аналізує системи захисту та моніторингу, шукає канали з мінімальним контролем. На етапі отримання доступу актор забезпечує собі можливість читання або копіювання цільових даних. Етап збору та агрегації полягає в накопиченні цільової інформації перед її виведенням. Ексфільтрація є центральним етапом витоку, на якому

відбувається фактична передача даних за межі організації; саме цей етап є основною ціллю для ЗВД-систем. Приховування слідів є завершальним етапом, на якому актор намагається усунути докази своєї діяльності.

Актор є важливим елементом моделі витоку даних, оскільки саме його дії призводять до несанкціонованої передачі інформації. Модель актора охоплює характеристики, релевантні для оцінки ризику та прийняття рішень системою ЗВД. Актора $a \in A$ визначимо як кортеж:

$$a = \langle a_{id}, a_{role}, a_{access}, a_{context} \rangle, \tag{2.27}$$

де $a_{id} \in I_{id}$ – унікальний ідентифікатор актора у системі;

$a_{role} \in R_a$ – організаційна роль, що визначає службові повноваження;

$a_{access} \subseteq D$ – множина об'єктів даних, до яких актор має авторизований доступ;

$a_{context} \in X$ – контекстуальні атрибути поточної сесії.

Кількісне оцінювання надійності актора є необхідним для автоматизованого прийняття рішень системою ЗВД. Без формалізованої метрики довіри система не зможе диференціювати реакцію на дії різних користувачів залежно від їхньої історії та контексту.

Рівень довіри є центральним елементом моделі, що агрегує різноманітні фактори для оцінки надійності актора. Функцію довіри будуємо на основі статичних та динамічних характеристик:

$$T_a(a,t) = w_T \cdot T_{static}(a) + (1 - w_T) \cdot T_{dynamic}(a,t), \tag{2.28}$$

де $w_T \in [0,1]$ – ваговий коефіцієнт, що визначає баланс між компонентами;

$T_{static}(a) = f\left(s_{tenure}(a), \mathrm{s}_{clearance}(a), \mathrm{s}_{history}(a)\right)$ - статична компонента довіри зі значеннями у відрізку [0,1], яку визначаємо на основі постійних характеристик актора: стажу роботи в організації $s_{tenure}(a)$ (тривала робота зазвичай асоціюється з вищою лояльністю), рівня допуску до конфіденційної інформації $s_{clearance}(a)$ (наявність допуску свідчить про проходження перевірок) та історії інцидентів безпеки $s_{history}(a)$.

Динамічна компонента відображає поточний поведінковий профіль актора та обчислюємо її як $T_{dynamic}(a,t) = 1 - An(a,t)$, де $An(a,t) \in [0,1]$ – показник аномальності поведінки актора у момент часу t.

$a_{context} = \langle a_{location}, a_{time}, a_{device}, a_{auth}, a_{session} \rangle$ – це контекстуальні атрибути, які охоплюють характеристики поточної сесії актора: географічне розташування або мережну адресу, час сесії (робочі години, вихідні, нічний час), пристрій (корпоративний, особистий, невідомий), метод автентифікації (пароль, двофакторна, біометрична) та тривалість поточної сесії. Контекст суттєво впливає на оцінку ризику: доступ до конфіденційних даних з корпоративного офісу в робочий час є нормальною активністю, тоді як аналогічний доступ вночі з невідомої геолокації через особистий пристрій є підозрілим і вимагає посиленого контролю.

Множина ролей R_a визначає організаційну структуру доступу. Кожна роль $r \in R_a$ асоціюється з набором дозволів через відображення $R_a \to 2^{D \times O}$, де $O = \{read, write, copy, send, print, \dots\}$ – множина можливих операцій над даними. Через Perm(r) позначимо множину дозволів ролі r, тобто підмножину D × O, приписану цій ролі. Актор може виконувати операцію op над об'єктом d, якщо $(d, op) \in Perm(r(a))$.

Об'єкт витоку (конфіденційні дані) є центральним елементом, захист якого є метою ЗВД-системи. Модель об'єкта даних представляє характеристики, потрібні для класифікації, оцінки чутливості та прийняття рішень щодо дозволу або блокування передачі. Об'єкт даних $d \in D$ визначимо як кортеж:

$$d = \langle d_{id}, d_{sens}, d_{cont}, d_{feat}, d_{meta}, d_{class}, d_{lin} \rangle, \quad (2.29)$$

де $d_{id} \in I_{id}$ – унікальний ідентифікатор об'єкта даних;

$d_{sens} \in [0,1]$ – рівень чутливості;

$d_{cont} \in \Sigma^*$ – вміст об'єкта (множина всіх скінченних послідовностей символів алфавіту Σ);

d_{feat} – вектор ознак документа, успадкований від моделі (2.4);

$d_{meta} \in M$ – метадані об'єкта;

$d_{class} \in L$ – класифікаційна мітка;

$d_{lin} \in G$ – походження та історія трансформацій.

Отже, кортеж, який задано за формулою (2.29), є явним розширенням моделі документа, який задано за формулою (2.4). Перші компоненти успадковано зі збереженням змісту, а класифікаційну мітку та походження додано для потреб аналізу витоків.

Функцію чутливості визначаємо як комплексну оцінку, що враховує як явну класифікацію, так і результати аналізу вмісту:

$$d_{sens} = \max\big(S_{label}(d), S_{cont}(d)\big), \tag{2.30}$$

де $S_{label}(d)$ – чутливість, визначена на основі класифікаційної мітки (якщо присутня);

$S_{cont}(d) = f_{classifier}\big(d_{feat}\big)$ – чутливість, обчислена на основі аналізу вмісту за допомогою методів машинного навчання або правил, d_{feat} отримуємо з видобування ознак d_{cont}.

Метадані $d_{meta} = \langle d_{author}, d_{created}, d_{modified}, d_{type}, d_{size}, d_{location}, d_{tags}\rangle$ охоплюють описові атрибути об’єкта і мають самостійну цінність для аналізу: документ, створений керівництвом, модифікований нещодавно та розташований у теці “Стратегія”, ймовірно, є більш чутливим порівняно з документом аналогічного вмісту, створеним рядовим працівником.

Класифікаційною міткою $d_{class} \in L$ представляємо явне маркування документа відповідно до корпоративної політики класифікації. Множина $L = \{l_{pub}, l_{int}, l_{conf}, l_{sec}, l_{top}\}$ є лінійно впорядкованою за рівнем чутливості з відношенням порядку $l_{pub} < l_{int} < l_{conf} < l_{sec} < l_{top}$. Походження даних $d_{lin} \in \mathrm{G}$ відстежує історію об’єкта і моделюється як орієнтований ациклічний граф $d_{lin}(d) = (V, E)$, де вершини V представляють об’єкти даних, а ребра E – трансформації або операції копіювання. Відстеження походження важливе для виявлення витоків похідних даних: якщо конфіденційний документ був скопійований, модифікований та перейменований, система має розпізнавати похідний об’єкт як конфіденційний на основі його походження.

Канал передачі є третім основним компонентом моделі витоку, що визначає технічний механізм виведення даних за межі організаційного периметру. Канал передачі $c \in \mathrm{C}$ визначається як кортеж:

$$c = \langle c_{type}, c_{bandwidth}, c_{encryption}, c_{monitoring}, c_{destination}\rangle, \tag{2.31}$$

де $c_{type} \in T_C$ – тип каналу (email, web, USB, print, cloud тощо);

$c_{bandwidth} \in \mathbb{R}^+$ – пропускна здатність каналу;

$c_{encryption} \in \{none, transport, e2e\}$ – рівень шифрування;

$c_{monitoring} \in [0,1]$ – ступінь моніторингу каналу системою ЗВД;

$c_{destination} \in E_{dest}$ – кінцева точка призначення.

Тип каналу визначає принципову категорію механізму передачі. Множина типів структурується ієрархічно: $T_C = \{network, local, physical\}$ з подальшою деталізацією, де $network = \{email, web, cloud, messaging, ftp, \dots\}$, $local = \{usb, \text{print}, \text{bluetooth}, \text{screenshot}, \dots\}$, $physical = \{device, document, \dots\}$. Кожен тип каналу характеризується специфічним профілем ризику та вимагає відповідних методів контролю.

Пропускна здатність $c_{bandwidth}$ визначає максимальний обсяг даних, що може бути переданий через канал за одиницю часу. Канали з високою пропускною здатністю становлять більшу загрозу, оскільки дозволяють швидко вивести великі обсяги інформації. Ступінь моніторингу $c_{monitoring} \in [0,1]$ відображає, наскільки повно канал контролюється ЗВД-системою: значення 1,0 відповідає повній інспекції вмісту, проміжні значення – частковій інспекції (метадані, вибіркова перевірка) або логуванню, нуль – відсутності контролю. Канали з низьким моніторингом становлять підвищений ризик і є привабливими для зловмисників.

Кінцева точка призначення $c_{destination} \in E_{dest}$ визначає, куди спрямовуються дані, і характеризується кортежем $c_{destination} = \langle ds_{category}, ds_{reputation}, ds_{geography} \rangle$, де $ds_{category} \in \{internal, partner, customer, public, competitor, unknown\}$ – категорія отримувача, $ds_{reputation} \in [0,1]$ – репутаційний рейтинг (для зовнішніх сервісів), $ds_{geography}$ – географічне розташування. Передача до внутрішніх систем або довірених партнерів має низький ризик, тоді як передача на особисті поштові скриньки, файлообмінники з низькою репутацією або до юрисдикцій з неадекватним захистом даних – високий.

Функцію ризику каналу визначимо як агрегат цих характеристик для оцінки загального рівня загрози:

$$R_{channel}(c) = f\left(c_{type}, c_{bandwidth}, 1 - c_{monitoring}, R_{dest}(c_{destination})\right), \qquad (2.32)$$

де R_{dest} – функція ризику кінцевої точки. Канали з високою пропускною здатністю, низьким моніторингом та ризикованими кінцевими точками отримують найвищі оцінки ризику.

Об’єднання моделей актора, об’єкта та каналу дає змогу побудувати комплексну ризик-модель витоку даних, що є основою для прийняття рішень системою ЗВД. Ризик витоку визначимо як добуток скалярів ймовірності реалізації витоку та потенційного впливу (збитків) від інциденту:

$$R(e) = P(L_e) \cdot I(d), \tag{2.33}$$

де $R(e)$ – ризик витоку для події e;

e – подія передачі даних, що оцінюється;

$P_L(e)$ – ймовірність того, що подія e є витоком (несанкціонованою передачею);

d – об’єкт (документ), задіяний у події e;

$I(d)$ – потенційний вплив від витоку об’єкта d.

Ймовірність витоку обчислюємо на основі характеристик актора, об’єкта та каналу. Компонента актора $P_{actor}(a,t) = \left(1 - \mathrm{T}_a(a,t)\right) \times An(a,t)$ оцінює ймовірність того, що актор здійснює несанкціоновану передачу: низька довіра та висока аномальність поведінки підвищують ймовірність зловмисних дій. Компонента об’єкта $\mathrm{d}_{sens}(d)$ враховує чутливість даних. Компонента каналу $R_{channel}(c)$ оцінює ризикованість каналу передачі. Агреговану ймовірність визначимо як функцію від цих компонент: Тут $P(L_e)$ визначаємо як прогнозовану системою ймовірність $p_e = P(Y_e = 1|x_e)$, де Y_e - істинна бінарна мітка витоку за формулою (2.26), а x_e - спостережувані ознаки події. Істинна мітка та її прогноз є різними об’єктами, і в ризик-моделі використовується саме прогноз. Формулу задамо так:

$$P(L_e) = f\left(P_{actor}(a,t), d_{sens}(d), R_{channel}(c)\right), \tag{2.34}$$

де $P(L_e)$ – агрегована ймовірність події витоку L_e;

f – функція, що поєднує компоненти актора, об'єкта та каналу;

$P_{actor}(a,t)$ – компонента актора, тобто ймовірність несанкціонованої передачі актором a у момент t;

$d_{sens}(d)$ – чутливість об'єкта даних d;

$R_{channel}(c)$ – ризикованість каналу передачі c.

Функція впливу $I: D \to R^+$ оцінює потенційні збитки від витоку конкретного об’єкта даних, для практичного застосування спростимо її до лінійної комбінації:

$$I(d) = w_1 \cdot d_{sens}(d) + w_2 \cdot d_{volume}(d) + w_3 \cdot d_{regulatory}(d) + w_4 \cdot d_{business}(d), \quad (2.35)$$

де d_{volume} – обсяг даних;

$d_{regulatory}$ – регуляторні наслідки (штрафи, санкції);

$d_{business}$ – бізнес-наслідки (конкурентні втрати, репутаційні збитки);

w_i – ваги, які визначаються політикою організації.

На основі обчисленого ризику система ЗВД приймає рішення згідно з пороговими значеннями:

$$Dc(e) = \begin{cases} Dc_{allow}, & якщо\ R(e) < \tau_q \\ Dc_{quarantine}, & якщо\ \tau_q \le R(e) < \tau_b, \\ Dc_{block}, & якщо\ R(e) \ge \tau_b \end{cases} \quad (2.36)$$

де τ_q – поріг карантину (передача відкладається для перевірки аналітиком);

τ_b – поріг блокування (передача забороняється);

Dc_{allow}, $Dc_{quarantine}$ та Dc_{block} – відповідно рішення «дозволити», «відправити на карантин» і «заблокувати» передачу для події e;

$R(e)$ – ризик цієї події, який обчислюємо за формулою (2.33).

Таке трирівневе рішення забезпечує гнучкість реагування: низькоризикові операції виконуються безперешкодно, середньоризикові потребують додаткової верифікації, високоризикові блокуються автоматично.

Практичне налаштування ЗВД-системи вимагає об'єктивного критерію для порівняння альтернативних конфігурацій. Функція вартості помилок зводить різнотипні наслідки (пропущені витоки та хибні блокування) до єдиної числової шкали, придатної для оптимізації.

Для кількісної оцінки ефективності системи застосуємо функцію вартості помилок:

$$C_{total} = C_{FP} \cdot FP + C_{FN} \cdot FN, \quad (2.37)$$

де C_{FP} – вартість хибнопозитивного спрацювання (блокування легітимної передачі);

C_{FN} – вартість хибнонегативного результату (пропущений витік);

FP та FN – кількості відповідних помилок.

Типово $C_{FN} \gg C_{FP}$, оскільки пропущений витік може призвести до значних збитків, тоді як хибне блокування лише створює незручності та затримки. Проте надмірна кількість хибних спрацювань призводить до “втоми від тривог” та ігнорування попереджень системи, тому необхідний баланс.

Детерміністичні правила прийняття рішень не завжди адекватно відображають невизначеність реальних ситуацій. Баєсівський підхід уможливлює коректне комбінування апріорних знань про частоту витоків з поточними спостереженнями, забезпечуючи більш обґрунтовані ймовірнісні оцінки.

Ймовірнісна модель витоку може бути деталізована з використанням баєсівського підходу. Апостеріорну ймовірність витоку за умови спостережуваних ознак **x** обчислимо за формулою Баєса:

$$P(l|x) = \frac{P(x|l) \cdot P(l)}{P(x)}, \tag{2.38}$$

де $P(l)$ – апріорна ймовірність витоку (базовий рівень інцидентів в організації);

$P(\mathbf{x}|l)$ – правдоподібність ознак за умови витоку;

$P(\mathrm{x})$ – повна ймовірність спостереження ознак.

Вектор ознак $\mathrm{x} = \left(x_{actor}, x_{object}, x_{channel}, x_{context}\right)$ включає характеристики всіх компонентів. За умови незалежності компонентів: $P(\mathrm{x}|l) = P(x_{actor}|l) \cdot P\left(x_{object}\middle|l\right) \cdot P(x_{channel}|l) \cdot P(x_{context}|l)$. Параметри моделі оцінюються на основі історичних даних про інциденти та легітимну активність.

Синтез розглянутих компонентів дозволяє побудувати узагальнену модель витоку даних у корпоративних мережах. Модель витоку даних визначимо як систему:

$$M_{leak} = (A, D, C, Pol, E, R, Dc), \tag{2.39}$$

де A – множина акторів;

D – множина об’єктів даних, як у (2.6);

C – множина каналів передачі з (2.25);

Pol – множина політик безпеки;

$E \subseteq A \times D \times C \times T$ – простір подій передачі;

$R: E \to \mathbb{R}^+$ – функція оцінки ризику;

$Dc: E \to \left\{Dc_{allow}, \mathrm{D}c_{quarantine}, \mathrm{D}c_{block}\right\}$ – функція прийняття рішень.

Взаємозв’язки між компонентами моделі відображають потік обробки події передачі даних у системі ЗВД. Для кожної події $e = (a, d, c, t)$ система виконує послідовність кроків: ідентифікація актора a та отримання його профілю; аналіз об’єкта даних d з оцінкою чутливості; характеризація каналу c з визначенням рівня моніторингу та ризику кінцевої точки; обчислення агрегованого ризику $Risk(e)$; порівняння з пороговими значеннями та прийняття рішення; виконання відповідної дії (дозвіл, карантин або блокування); логування події та оновлення профілів. Узагальнену модель витоку даних та потік прийняття рішень у ЗВД-системі зображено на рисунку 2.3.

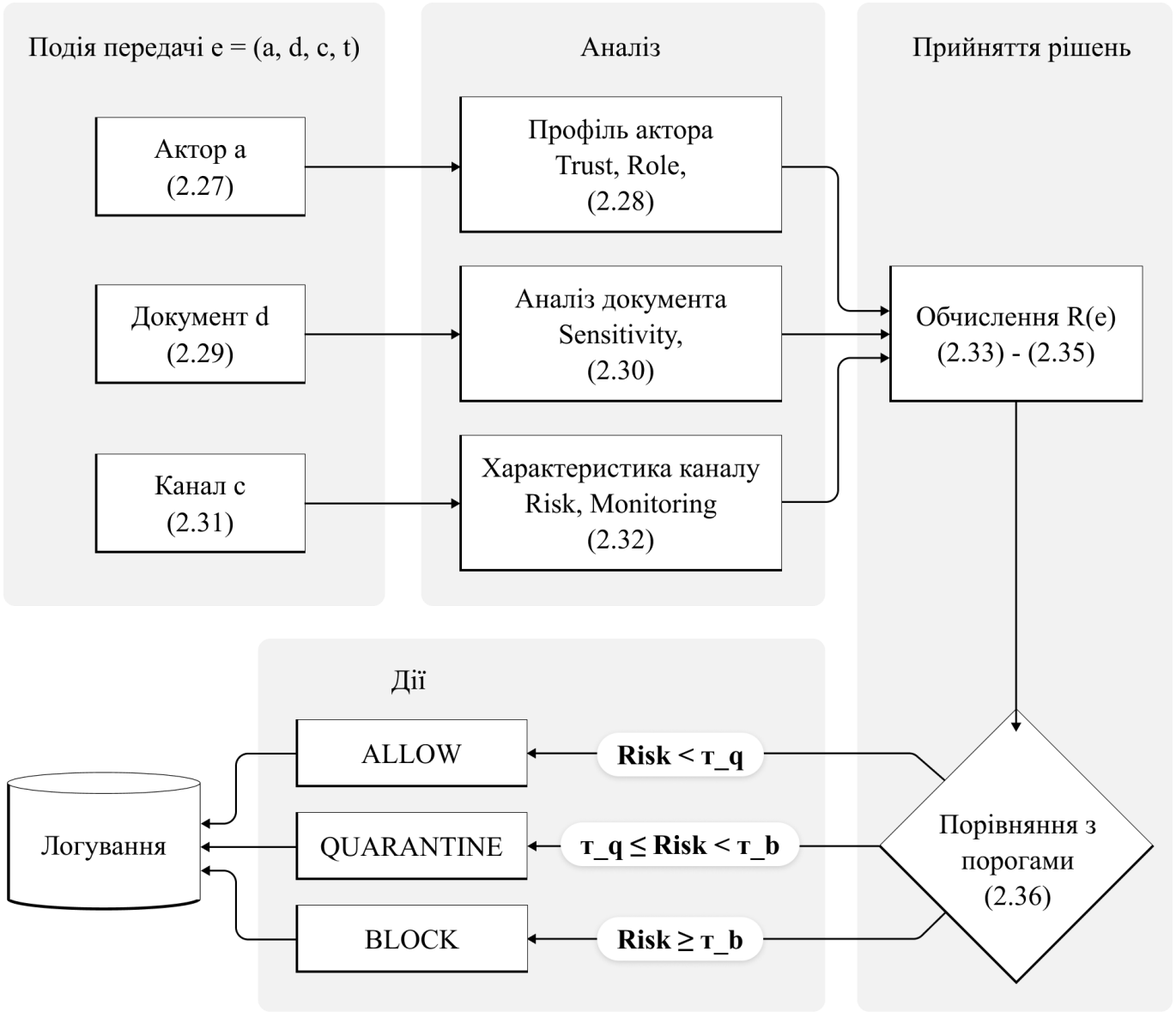


Рисунок 2.3 – Узагальнена модель витоку даних та потік прийняття рішень у ЗВД-системі

Головною властивістю моделі є її адаптивність. Функція довіри до актора, параметри класифікатора та порогові значення рішень можуть налаштовуватися з часом на основі зворотного зв'язку. Це дозволяє системі адаптуватися до зміни поведінки користувачів, появи нових типів конфіденційних даних та еволюції загроз. Модель також враховує можливість каскадних витоків, коли дані передаються через послідовність проміжних точок: $e_{cascade} = (a_1, d, c_1, t_1) \to (a_2, d', c_2, t_2) \to \cdots \to (a_n, d^{(n)}, c_n, t_n)$, де $d^{(i)}$ – можливо трансформовані версії початкових даних. Система має відстежувати походження даних для виявлення таких ланцюжків.

Порівняння запропонованої ризик-моделі з наявними підходами до формалізації витоків даних виявляє кілька принципових відмінностей. На відміну від роботи [18], де аналіз обмежується семантичними характеристиками контенту без урахування суб'єктних та канальних факторів, побудована модель інтегрує рівень довіри до актора, чутливість об'єкта та ризик каналу в єдину кількісну оцінку ризику витоку (2.33). Підхід [10] стосується контекстного маркування документів, однак не пропонує механізму агрегації різнорідних факторів для автоматизованого прийняття рішень; розроблена ж модель формалізує трирівневе рішення (дозвіл, карантин, блокування) через порогові значення інтегрованого ризику (2.36). Баєсівське розширення моделі забезпечує можливість безперервного навчання на основі історичних даних: апостеріорна ймовірність витоку уточнюється з кожним новим спостереженням, що відрізняє запропонований підхід від статичних моделей з фіксованими параметрами. Функція вартості помилок дає змогу об'єктивно порівнювати альтернативні конфігурації системи з урахуванням асиметрії наслідків хибнопозитивних та хибнонегативних рішень.

Запропонована узагальнена модель витоку даних забезпечує теоретичний фундамент для розробки системи запобігання витокам. Модель охоплює всі головні компоненти: суб'єктів (акторів), об'єкти (конфіденційні дані), механізми реалізації (канали) та критерії прийняття рішень на основі ризик-орієнтованого підходу.

В якості прикладу розглянемо співробітника фінансового відділу, який намагається надіслати зовнішньому адресатові електронною поштою документ-рахунок, що містить три номери платіжних карток та два прізвища клієнтів.

Контентний аналіз виявляє у документі дві категорії конфіденційних даних: фінансові реквізити, персональні дані, тобто $\kappa_{cat}(d) = \{FIN, PII\}$. Ваги категорій

візьмемо такі: $w_{FIN} = 0{,}9$, $w_{PII} = 0{,}6$. Кількість входжень: $n_{FIN} = 3$ (три номери карток), $n_{PII} = 2$ (два прізвища). Референсні значення: $n_{ref,FIN} = 1, n_{ref,PII} = 3$. Тоді нормовані насиченості категорій становлять $\nu_{FIN} = \min\left(1, \frac{3}{1}\right) = 1{,}0$ та $\nu_{PII} = \min\left(1, \frac{2}{3}\right) \approx 0{,}67$.

Агреговану чутливість за формулою (2.3) обчислюємо як адитивно-імовірнісну агрегацію так:

$$S(d) = 1 - (1 - w_{FIN} \cdot \nu_{FIN})(1 - w_{PII} \cdot \nu_{PII}) = 1 - (1 - 0{,}9 \cdot 1{,}0)(1 - 0{,}6 \cdot 0{,}67) \approx 1 - 0{,}1 \cdot 0{,}60 \approx 0{,}94.$$

Документ отримує високу чутливість, оскільки містить кілька платіжних реквізитів. Для порівняння, один номер картки без прізвищ дав би менше значення, тобто модель розрізняє поодинокий реквізит та їх сукупність.

Функцію впливу за формулою (2.35) обчислюємо як зважену суму чутливості, обсягу, регуляторних та бізнес-наслідків. Беручи $d_{sens} = 0{,}94$, нормований обсяг $d_{volume} = 0{,}3$, регуляторну складову $d_{regulatory} = 0{,}9$ і бізнес-складову $d_{business} = 0{,}5$ з вагами (0,4; 0,2; 0,3; 0,1), маємо $I(d) = 0{,}4 \cdot 0{,}94 + 0{,}2 \cdot 0{,}3 + 0{,}3 \cdot 0{,}9 + 0{,}1 \cdot 0{,}5 \approx 0{,}76$.

Актор є фінансовим працівником із доволі високим рівнем довіри: $T_a(a, t) = 0{,}7$. Канал передачі (зовнішня електронна пошта) і знижена довіра, властива спробі вивести платіжні дані за периметр, дають імовірність витоку $P(L_e) \approx 0{,}40$. Ризик події за формулою (2.33) дорівнює $R(e) = P(L_e) \cdot I(d) = 0{,}40 \cdot 0{,}76 \approx 0{,}30$.

Порівнюємо ризик із пороговими значеннями $\tau_q = 0{,}2$ та $\tau_b = 0{,}6$ за правилом (2.36): оскільки $\tau_q \leq R(e) < \tau_b$, система ухвалює рішення «карантин». Документ не блокується остаточно, проте його передачу відкладено для перевірки аналітиком.

Таким чином, побудовано узагальнену модель витоку даних, яка пов’язує в єдину формальну схему актора, об’єкт передачі, канал і ризик-орієнтоване правило прийняття рішень. Чутливість документа агрегується за категоріями виявлених даних, вплив інциденту поєднує регуляторні та бізнес-наслідки, а трирівневе порогове правило перетворює оцінку ризику на конкретну дію системи. Наскрізний числовий приклад показав, що модель коректно розрізняє поодинокі та множинні реквізити і приводить до обґрунтованого рішення «карантин». Водночас модель описує подію витоку абстраговано від середовища, в якому вона відбувається. Для

практичного розгортання точок контролю потрібна формалізація самої корпоративної інфраструктури, її вузлів, зон безпеки та потоків даних.

2.3 Модель середовища виникнення витоків в корпоративних системах

Витоки даних виникають не у вакуумі, а є наслідком взаємодії акторів, інформаційних ресурсів та комунікаційних каналів у конкретному технологічному та організаційному контексті. Для практичного застосування цієї моделі необхідне глибоке розуміння середовища, в якому такі події відбуваються. Корпоративна інформаційна система є складною екосистемою, де технічна інфраструктура переплітається з організаційними процесами, політиками безпеки та людським фактором. Адекватна формалізація цього середовища є передумовою для проєктування ефективних механізмів запобігання витокам.

Сучасна корпоративна система суттєво відрізняється від класичних уявлень про замкнений периметр із чітко визначеними межами. Поширення хмарних технологій, мобільних пристроїв та моделі віддаленої роботи розмило традиційні кордони організації. Дані постійно переміщуються між локальними серверами та хмарними платформами, між корпоративними робочими станціями та особистими пристроями працівників, між внутрішніми системами та сервісами партнерів. Ця гетерогенність створює множинні точки потенційного витоку та ускладнює контроль над інформаційними потоками. Мета цього підрозділу: побудувати формальну модель корпоративної системи, яка б охоплювала компоненти інфраструктури, зв’язки між ними, потоки даних та механізми контролю, забезпечуючи теоретичний фундамент для розміщення точок контролю ЗВД-системи.

Топологія корпоративної мережі природно моделюється за допомогою теорії графів, де вузли представляють обчислювальні ресурси, а ребра – комунікаційні канали між ними. Таке представлення відкриває можливість застосувати розвинений математичний апарат для аналізу структури мережі, виявлення критичних точок та оптимізації розміщення захисних механізмів. Корпоративну мережу моделюватимемо як орієнтований мультиграф з атрибутами:

$$G = \langle V, E_G, \tau_V, \tau_E, \omega \rangle, \tag{2.40}$$

де G – орієнтований мультиграф з атрибутами, що моделює топологію корпоративної мережі;

V – скінченна множина вершин (вузлів мережі);

$E_G \subseteq V \times V \times T_E$ – типізовані ребра (канали зв'язку);

$\tau_V: V \to T_V$ – типізація вершин;

$\tau_E: E_G \to T_E$ – типізація ребер;

$\omega: E_G \to R^+$ – ваги ребер (пропускна здатність, латентність);

T_V – множина типів вершин мережі;

T_E – множина типів ребер, тобто типів каналів зв'язку;

R_+ – множина додатних дійсних чисел, до якої належать ваги ребер.

Множина типів вершин $T_V = \{server, workstation, mobile, network_{device}, security_{appliance}, cloud_{service}, external\}$ охоплює різноманітні категорії мережних вузлів, кожен з яких характеризується специфічним набором властивостей та вимог до захисту. Сервери зберігають та обробляють дані, виступаючи основними сховищами конфіденційної інформації. Робочі станції є точками взаємодії користувачів із корпоративними ресурсами, де дані візуалізуються, редагуються та можуть бути скопійовані на локальні або знімні носії. Мобільні пристрої розширюють периметр мережі за межі фізичного офісу, часто працюючи через недовірені мережі. Мережне обладнання забезпечує маршрутизацію та комутацію трафіку. Засоби безпеки реалізують функції захисту. Хмарні сервіси представляють зовнішні ресурси, інтегровані в корпоративну інфраструктуру. Зовнішні вузли моделюють точки виходу в Інтернет та взаємодії з некорпоративними системами.

Множина типів ребер $T_E = \{lan, wan, vpn, wifi, cloud_{connect}, internet, usb\}$ класифікує канали зв'язку за їхніми технічними характеристиками та профілем ризику. Локальні мережі забезпечують високошвидкісний зв'язок у межах офісу або дата-центру, трафік у них зазвичай не шифрується на мережному рівні, що спрощує його інспекцію засобами ЗВД. Глобальні мережі з'єднують географічно розподілені локації та часто проходять через недовірену інфраструктуру. Віртуальні приватні мережі створюють захищені тунелі через публічні мережі, забезпечуючи конфіденційність трафіку, проте термінація тунелю на концентраторі VPN створює точку, де можлива інспекція вмісту. Хмарні з'єднання пов'язують локальну

інфраструктуру з сервісами публічних та приватних хмар, при цьому частина інфраструктури перебуває поза межами прямого контролю організації.

Для детальнішого опису вузлів вводиться функція атрибутів $h: V \to \Xi$, де простір атрибутів включає тип, зону безпеки, критичність, операційну систему, перелік сервісів, власника та прапорець наявності агента ЗВД. Атрибут критичності $\xi_{crit} \in [0,1]$ відображає важливість вузла для бізнес-процесів, а прапорець $\xi_{ЗВД} \in \{0,1\}$ фіксує наявність агентського програмного забезпечення, здатного моніторити операції з файлами, буфером обміну, друком та периферійними пристроями. Приклад графа корпоративної мережі з виділенням зон безпеки та точок контролю подано на рисунку 2.4.

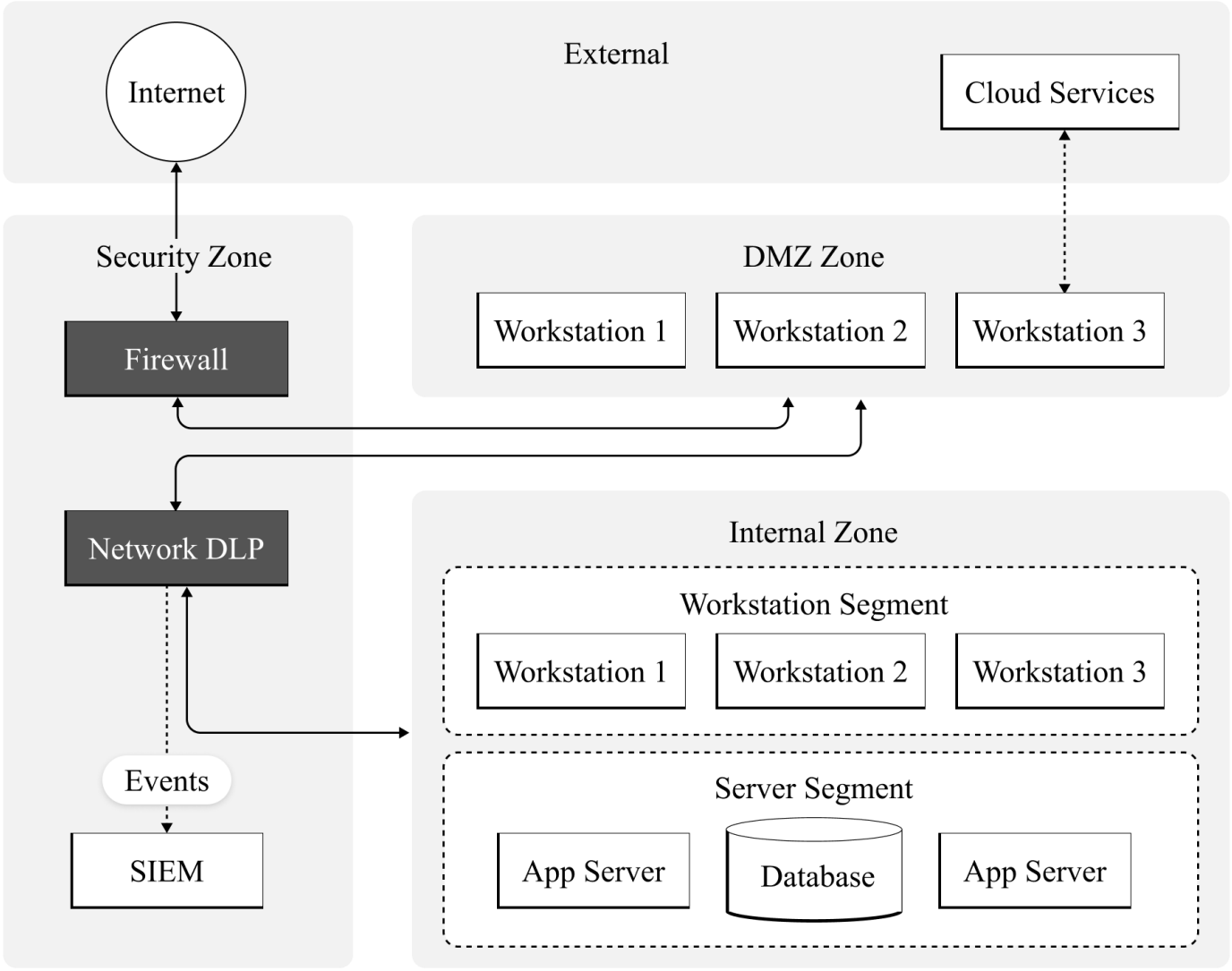


Рисунок 2.4 – Граф корпоративної мережі з виділенням зон безпеки та точок контролю ЗВД

На рівні мережних вузлів функціонують сервіси та додатки, які безпосередньо обробляють користувацькі дані та забезпечують їх передачу. Кожен тип сервісу створює специфічний канал потенційного витоку з власними характеристиками. Множина сервісів $Serv$ формально визначається як множина кортежів, де кожен сервіс характеризується типом, вузлом розміщення, протоколом взаємодії, типами даних, що обробляються, та методом автентифікації. Електронна пошта залишається одним із найпоширеніших каналів витоку даних: листи можуть містити конфіденційну інформацію як у тілі повідомлення, так і у вкладеннях, що робить контроль вихідної пошти класичною функцією ЗВД-систем. Файлові сервери та системи управління документами зберігають структуровані колекції корпоративних документів, інтеграція з системами класифікації даних забезпечує автоматичне маркування файлів. Хмарні сховища стали невід'ємною частиною робочих процесів, синхронізація файлів із хмарою створює ризик виведення даних за межі контрольованого периметру. Корпоративні месенджери поєднують текстові повідомлення з обміном файлами, швидкість та неформальність спілкування у месенджерах підвищують ризик ненавмисного розголошення інформації.

Контроль доступу до інформаційних ресурсів є механізмом безпеки, що визначає, які актори можуть виконувати які операції над якими даними.

Модель рольового управління доступом (Role-Based Access Control, RBAC) є домінуючою парадигмою в корпоративних системах завдяки балансу між гнучкістю та керованістю. Модель RBAC визначимо як кортеж:

$$B = \langle U, R, P, S, \Phi_{ua}, \Phi_{pa}, f_{user}, f_{role} \rangle, \tag{2.41}$$

де U – множина користувачів;

R – множина ролей;

P – множина дозволів;

S – множина сесій;

$\Phi_{ua} \subseteq U \times R$ – відношення призначення користувачів ролям;

$\Phi_{pa} \subseteq P \times R$ – відношення призначення дозволів ролям;

$f_{user}: S \to U$ – функція відображення сесії на користувача;

$f_{role}: S \to 2^R$ - функція активних ролей у сесії. Множина дозволів структурується як декартів добуток операцій та ресурсів $P = O \times D$, де множина

операцій O та сама, що введена в підрозділі 2.2, і $O = \{read, write, delete, copy, send, print, download, upload\}$ охоплює типові дії над даними, а D - множина об'єктів даних. Користувач u має дозвіл p тоді і лише тоді, коли існує роль, призначена користувачу та асоційована з цим дозволом:

$$H_p(u,p) \Leftrightarrow \exists r \in R: (u,r) \in \Phi_{ua} \wedge (p,r) \in \Phi_{pa}. \quad (2.42)$$

де $H_p(u,p)$ – предикат наявності у користувача u дозволу p;

r – окрема роль.

Введемо відношення часткового порядку ієрархією ролей $\Phi_{rh} \subseteq R \times R$, де $(r_1, r_2) \in \Phi_{rh}$ означає, що роль r_1 успадковує всі дозволи ролі r_2. Така структура дозволяє ефективно керувати правами доступу у великих організаціях: замість індивідуального призначення дозволів кожному користувачу, адміністратори визначають ролі та їхню ієрархію, а потім призначають користувачів відповідним ролям. Для інтеграції з ЗВД-системою модель RBAC розширюється контекстуальними обмеженнями, де дозвіл активується лише при виконанні контекстуальних умов, таких як час доби, локація або тип пристрою. Інтеграція RBAC з ЗВД реалізується через перевірку авторизації при кожній події передачі даних, система ЗВД звертається до сервісу авторизації для визначення, чи має актор право на виконання операції з урахуванням додаткових обмежень на дозволені канали передачі для певної ролі.

Дані у корпоративній системі перебувають у постійному русі: створюються, модифікуються, копіюються, передаються між вузлами та зберігаються. Моделювання потоків даних дозволяє ідентифікувати шляхи потенційного витоку та визначити оптимальні точки контролю. Потік даних між вузлами мережі визначимо функцією:

$$\Phi: V \times V \times T \to 2^D, \quad (2.43)$$

де $\Phi(v_1, v_2, t)$ – множина об'єктів даних, що передаються від вузла v_1 до вузла v_2 у момент часу t.

Для аналізу ризиків вводиться зважений потік з урахуванням чутливості даних:

$$\Phi_{sens}(v_1, v_2, t) = \sum_{d \in \Phi(v_1, v_2, t)} S(d) \cdot s(d), \quad (2.44)$$

де Φ_{sens} – зважений за чутливістю потік даних від вузла v_1 до вузла v_2 у момент t;

$\Phi(v_1, v_2, t)$ – множина переданих цим потоком об'єктів, за якою ведеться підсумовування;

d – окремий об'єкт даних потоку;

$S(d)$ – чутливість цього об'єкта за формулою (2.3);

$s(d)$ – його обсяг, який взято з поля метаданих m_{size}.

Зважений потік у такий спосіб ураховує як чутливість, так і кількість переданих даних.

Потоки класифікуються за напрямком відносно організаційного периметру. Внутрішні потоки циркулюють між вузлами однієї зони безпеки. Вихідні потоки спрямовані за межі контрольованого периметру і є основною ціллю моніторингу ЗВД, оскільки вони представляють можливість виведення даних за межі організації. Вхідні потоки надходять з-за меж периметру і є менш релевантними для запобігання витокам, хоча важливі для виявлення командних каналів зловмисного програмного забезпечення. Транзитні потоки проходять через проміжні вузли, їхній аналіз дозволяє виявляти обхідні шляхи передачі даних.

Матриця потоків F агрегує інформацію про трафік між вузлами за певний період часу. Елементи матриці з високими значеннями вказують на інтенсивний обмін чутливими даними та потребують посиленого контролю. На основі матриці потоків будуємо граф потоків даних $G_F = (V, E_F)$, де $E_F = \{(v_i, v_j) : F[v_i, v_j] > \theta_F\}$, а θ_F – порогове значення для фільтрації несуттєвих потоків. Аналіз цього графу дозволяє виявити вузли з високою централізованістю, потенційні точки витоку та аномальні закономірності потоків. Вузли з високим ступенем центральності за посередництвом є особливо важливими для моніторингу, оскільки через них проходить значна частина потоків між іншими вузлами мережі. Виявлення нових ребер у графі потоків, особливо спрямованих до зовнішніх вузлів, може свідчити про появу нових каналів витоку або зміну шаблонів роботи користувачів, що потребує уваги аналітиків безпеки.

Ефективне запобігання витокам даних вимагає стратегічного розміщення точок контролю на шляхах потоків даних. Кожна точка контролю здатна інспектувати,

фільтрувати або блокувати передачу даних відповідно до політик безпеки. Точку контролю ЗВД визначимо як кортеж:

$$q = \langle q_{loc}, q_{type}, q_{cap}, q_{cov}, q_{mode} \rangle, \quad q \in Q, \tag{2.45}$$

де $q_{loc} \in V \cup E_G$ – розташування точки контролю в мережі (вузол або ребро);

$q_{type} \in \{network, endpoint, cloud, discovery\}$ – її тип;

$q_{cap} \subseteq \{inspect, log, alert, block, quarantine, encrypt\}$ – набір можливостей;

$q_{cov} \subseteq C$ – контрольовані канали передачі;

$q_{mode} \in \{inline, tap, api\}$ – режим роботи.

Мережні точки контролю розміщуються на периметрі мережі та критичних внутрішніх сегментах, вони інспектують трафік, аналізуючи вміст пакетів та застосовуючи політики; режим inline дозволяє блокування, але вносить латентність, режим tap забезпечує пасивний моніторинг без впливу на швидкість. Точки контролю на кінцевих пристроях реалізуються через агентське програмне забезпечення, яке перехоплює операції на рівні операційної системи, контролюючи доступ до файлів, периферійних пристроїв та буфера обміну; перевага агентів – можливість контролю локальних каналів, недолік – необхідність розгортання на всіх робочих станціях. Хмарні точки контролю забезпечують моніторинг взаємодії з хмарними сервісами через інтеграцію з API платформ. Точки виявлення сканують сховища даних для ідентифікації конфіденційної інформації, працюючи офлайн і періодично класифікуючи документи. Порівняльні характеристики типів точок контролю зведено в таблиці 2.5.

Таблиця 2.5 - Характеристики типів точок контролю ЗВД

Тип точки контролю	Розташування	Режим роботи	Охоплені канали	Можливість блокування
Мережний ЗВД	Периметр, шлюзи	Inline/Tap	Email, Web, FTP	Так (inline)
Endpoint ЗВД	Робочі станції	Inline	USB, Print, Clipboard	Так
Cloud ЗВД (CASB)	Хмарні API	API	Cloud upload, Share	Так
Discovery ЗВД	Файлові сервери	Offline	Storage	Ні (лише alert)

Повноту покриття визначимо через частку контрольованих каналів так:

$$\kappa_{cov} = \frac{|\{c \in C : \exists q \in Q,\ c \in q_{cov}\}|}{|C|}, \tag{2.46}$$

де κ_{cov} – повнота покриття каналів контролем;

c – окремий канал передачі;

C – множина всіх каналів з (2.25), потужність якої стоїть у знаменнику;

q – точка контролю;

Q – множина розгорнутих точок контролю ЗВД;

q_{cov} – канали, охоплені точкою q. Чисельник в формулі (2.46) дорівнює кількості каналів, які контролює хоча б одна точка.

Ідеальне покриття рідко досяжне на практиці через технічні обмеження та вартість. Канали з наскрізним шифруванням, де ключі контролюються кінцевими користувачами, унеможливлюють контентну інспекцію без компрометації криптографічного захисту. Мобільні пристрої за межами корпоративної мережі можуть передавати дані через стільникові канали, що обминають точки контролю. Тіньові IT-сервіси, які працівники використовують без санкції організації, також залишаються поза моніторингом. Стратегія розміщення точок контролю має балансувати повноту покриття, продуктивність системи та бюджетні обмеження, враховуючи ці обмеження та компенсуючи їх альтернативними методами контролю, такими як поведінковий аналіз або політики обмеження використання неконтрольованих каналів. Типове розміщення точок контролю ЗВД у корпоративній інфраструктурі показано на рисунку 2.5.

Сегментація корпоративної мережі на зони безпеки є принципом захисту, що реалізує концепцію глибокого захисту. Кожна зона характеризується рівнем довіри та набором застосовуваних політик. Розподіл вузлів мережі за зонами безпеки визначимо функцією:

$$\zeta : V \to Z, \quad Z = \{z_{ext}, z_{dmz}, z_{int}, z_{res}, z_{cloud}\}, \tag{2.47}$$

де ζ – функція розподілу вузлів мережі за зонами безпеки;

V – множина вузлів мережі;

Z – зони безпеки;

z_{ext} – зовнішня зона;

z_{dmz} – демілітаризована зона;

z_{int} – внутрішня;

z_{res} – обмежена;

z_{cloud} – хмарна.

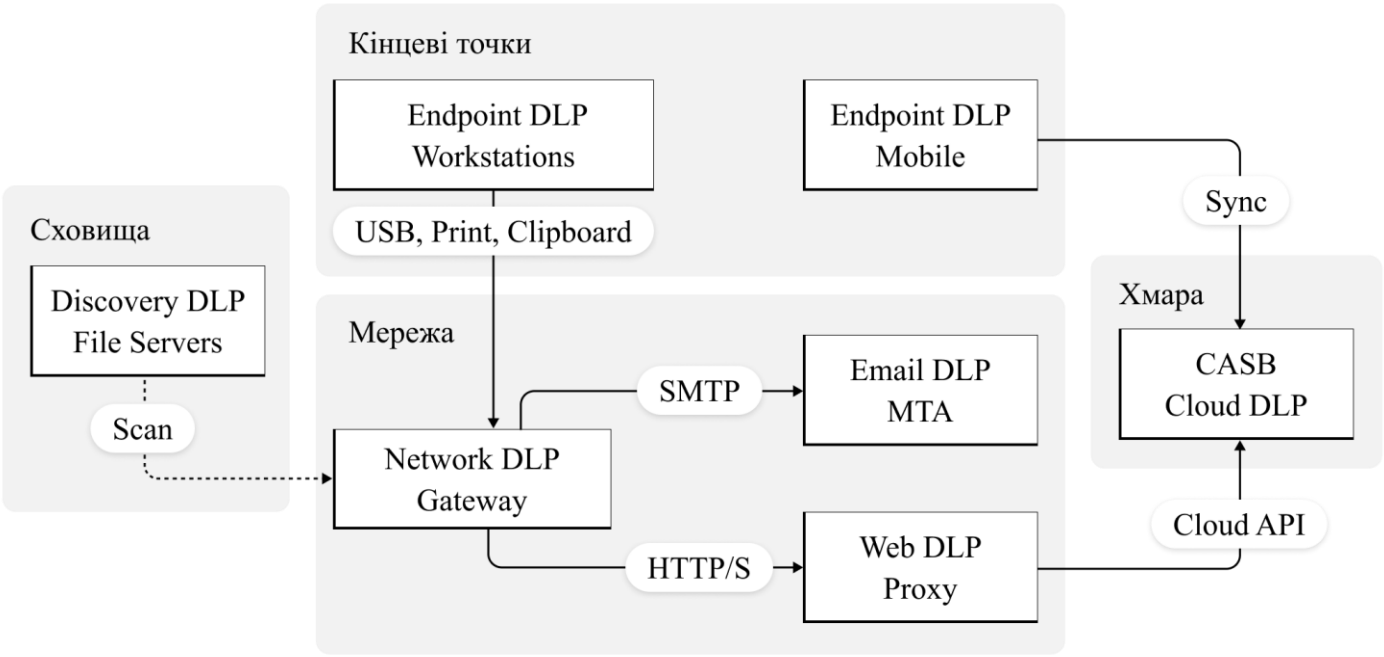


Рисунок 2.5 – Розміщення точок контролю ЗВД у корпоративній інфраструктурі

Зони характеризуються рівнем довіри $T_z: Z \to [0,1]$ з типовими значеннями від нуля для зовнішньої зони до значень близько одиниці для обмеженої зони. Зовнішня зона охоплює мережу Інтернет та будь-які ресурси за межами контролю організації, весь трафік з цієї зони вважається потенційно ворожим. Демілітаризована зона містить сервіси, доступні ззовні, які не мають прямого доступу до внутрішніх ресурсів. Внутрішня зона охоплює основну корпоративну інфраструктуру, саме тут реалізується більшість витоків через дії інсайдерів. Обмежена зона містить ресурси з максимальним рівнем захисту. Хмарна зона представляє ресурси у хмарних середовищах з обмеженим контролем над інфраструктурою.

Межі довіри визначимо як переходи між зонами з різними рівнями довіри:

$$B_{trust} = \{(v_1, v_2) \in E_G : \zeta(v_1) \neq \zeta(v_2)\}, \quad (2.48)$$

де B_{trust} – множина меж довіри;

E_G – множина ребер (каналів зв'язку) графа мережі;

v_1 – початковий вузол ребра;

v_2 – кінцевий вузол ребра;

ζ – зонування вузла, тобто віднесення вузла до зони безпеки;

$\zeta(v_1) \neq \zeta(v_2)$ – умова відбору ребер, що з'єднують вузли різних зон.

На межах довіри застосовуються посилені контролі: передача даних з внутрішньої зони до зовнішньої вимагає посиленої інспекції вмісту, доступ із зовнішньої зони до внутрішньої вимагає строгої автентифікації та авторизації. Матриця переходів між зонами $\Psi_z: Z \times Z \to \{\psi_a, \psi_i, \psi_b\}$ визначає дозволені напрямки потоків та необхідний рівень контролю: ψ_a - дозволити,ψ_i – інспектувати, ψ_b – заборонити. Матрицю дозволених переходів між зонами безпеки наведено в таблиці 2.6. Пари вершин у цьому правилі пробігають ребра графа мережі. Формально $(v_1, v_2, t_e) \in E_G$, причому тип ребра t_e у правилі переходу не використовується.

Таблиця 2.6 - Матриця дозволених переходів між зонами безпеки

Джерело / Призначення	Зовнішня	Демілітаризована	Внутрішня	Обмежена	Хмара
Зовнішня	-	ψ_i	ψ_b	ψ_b	ψ_i
Демілітаризована	ψ_a	-	ψ_i	ψ_b	ψ_i
Внутрішня	ψ_i	ψ_a	-	ψ_i	ψ_i
Обмежена	ψ_i	ψ_i	ψ_i	-	ψ_i
Хмара	ψ_i	ψ_i	ψ_i	ψ_b	-

Концепція Zero Trust розширює модель зон, відмовляючись від імпліцитної довіри до внутрішніх ресурсів. У цій моделі замість довіри на основі мережного розташування застосовується безперервна верифікація кожного запиту з урахуванням ідентичності актора, стану пристрою та контексту. Інтеграція ЗВД з Zero Trust передбачає контроль на рівні кожної транзакції, а не лише на межах зон. Практична реалізація Zero Trust у контексті ЗВД означає, що навіть передача даних між вузлами однієї внутрішньої зони підлягає перевірці авторизації та аналізу вмісту, якщо чутливість даних перевищує певний поріг. Це суттєво підвищує вимоги до

продуктивності системи ЗВД, оскільки обсяг трафіку, що підлягає інспекції, зростає на порядки порівняно з периметральним підходом.

Система ЗВД функціонує не ізольовано, а як компонент комплексної архітектури безпеки організації. Ефективна інтеграція з суміжними системами підвищує точність виявлення, прискорює реагування та забезпечує цілісне бачення стану безпеки. Екосистема безпеки включає системи управління інформацією та подіями безпеки (SIEM), системи виявлення та запобігання вторгненням (IDS/IPS), антивірусне програмне забезпечення, системи управління ідентифікацією та доступом (IAM), брокери безпеки хмарного доступу (CASB) та системи захисту кінцевих точок (EDR). Інтеграція ЗВД з SIEM дозволяє корелювати події передачі даних з подіями з інших джерел для виявлення складних атак: комбінація події автентифікації з незвичної локації, аномальної мережної активності та спроби завантаження великого обсягу файлів може вказувати на компрометацію облікового запису. Інтеграція з IAM дозволяє ЗВД звертатися до сервісу авторизації для перевірки прав на операції з даними. Антивірусні системи та EDR надають контекст про стан безпеки кінцевих пристроїв, виявлення зловмисного програмного забезпечення на робочій станції є сигналом для ЗВД про підвищення ризику витоку з цього вузла. Автоматизація реагування на інциденти через платформи SOAR оркеструє дії між системами: ЗВД виявляє спробу витоку, генерує тривогу, SOAR автоматично блокує обліковий запис користувача, ізолює його робочу станцію та створює інцидент для розслідування. Обмін розвідданими про загрози збагачує контекст рішень ЗВД інформацією про відомі зловмисні домени, хеші зловмисних файлів та шаблони ексфільтрації, що дозволяє виявляти передачі на відомі зловмисні ресурси.

Узагальнивши розглянуті компоненти, сформулюємо комплексну формальну модель корпоративної системи як середовища виникнення витоків даних:

$$E_{corp} = \langle G, B, Serv, \Phi, Q, T_a, \zeta, \Lambda \rangle, \tag{2.49}$$

де G – граф мережної інфраструктури;

B – модель рольового доступу;

$Serv$ – множина сервісів та додатків;

Φ – функція потоків даних;

Q – множина точок контролю ЗВД;

T_a – функція довіри до акторів (2.28);

ζ – функція зонування;

Λ – множина інтегрованих систем безпеки.

Модель витоку, яку задано за формулою (2.39) розширюємо контекстом середовища через розширену функцію авторизації:

$$Au_{env}(a,d,c,t) = H_p(a,d) \wedge H_c(a,c) \wedge \Psi_z(v_{src}, v_{dst}) \neq \psi_b \wedge T_a(a,t) \geq \tau_{min}(d), \quad (2.50)$$

де Ψ_z перевіряє допустимість передачі між зонами, а умова на довіру вимагає мінімального рівня довіри залежно від чутливості даних $\tau_{min}(d) = \gamma \cdot S(d)$;

$\gamma \in [0,1]$ – жорсткість вимог;

$H_p(a,d)$ – предикат наявності в актора a права на операції з об’єктом d;

$H_c(a,c)$ – предикат допустимості каналу c для актора a;

аргументами Ψ_z для стислості записано вершини, фактично умова застосовується до їхніх зон $\zeta(v_{src})$ і $\zeta(v_{dst})$ та вимагає, щоб перехід між зонами не був заборонений (значення, відмінне від ψ_b).

Розширюючи ризик-модель (2.33), функцією ризику витоку у контексті середовища враховуємо характеристики всіх компонентів:

$$R_{env}(a,d,c,t) = P_{env}(a,d,c,t) \times I(d), \quad (2.51)$$

де R_{env} – ризик витоку з урахуванням контексту середовища;

a – актор;

d – об'єкт даних;

c – канал передачі;

t – момент часу;

P_{env} – ймовірність витоку в цьому середовищі;

$I(d)$ – функція впливу, що оцінює потенційні збитки від витоку об'єкта d, ймовірність витоку залежить від рівня довіри до актора, чутливості даних, ступеня моніторингу каналу та ризику переходу між зонами.

$R_z(c)$ оцінює ризик переходу як невід’ємну різницю між рівнями довіри зон джерела та призначення: передача до зони з нижчою довірою має підвищений ризик.

Ефективність покриття середовища точками контролю оцінюємо через частку контрольованих чутливих потоків:

$$\eta_{eff} = \frac{\sum_{\varphi \in \Phi^{+}_{sens}} \mu_{ch}(c(\varphi)) \cdot S(d(\varphi))}{\sum_{\varphi \in \Phi^{+}_{sens}} S(d(\varphi))}, \quad (2.52)$$

де $\mu_{ch}(c(\varphi))$ – ступінь моніторингу каналу $c(\varphi)$, яким передається потік φ, і відповідає компоненту $c_{monitoring}$ моделі каналу з формули (2.31);

$S(d(\varphi))$ – чутливість об’єкта $d(\varphi)$, що передається потоком φ;

Φ^{+}_{sens} – множина чутливих потоків, тобто потоків із додатною зваженою чутливістю згідно з формулою (2.44), за якою ведеться підсумовування. За порожньої множини чутливих потоків показник не визначений і покладається рівним одиниці, бо немає що контролювати.

Модель середовища слугує основою для оптимізаційної задачі розміщення точок контролю: максимізувати ефективність покриття за обмежень на вартість розгортання та допустимий вплив на продуктивність системи.

Побудована модель корпоративної системи має низку переваг порівняно з наявними підходами до моделювання середовища функціонування ЗВД. Традиційні моделі мережної безпеки оперують пласкою структурою вузлів без типізації, тоді як запропоноване графове представлення, яке задане за формулою (2.49), з типізованими вершинами та зваженими ребрами дає змогу диференціювати робочі станції, сервери, мережне обладнання та хмарні ресурси, ідентифікувати шляхи потоків даних та важливі вузли для аналізу централізованості та виявлення вузьких місць. Інтеграція концепції Zero Trust у модель ЗВД-системи переносить принцип верифікації кожного запиту на задачу запобігання витокам даних, забезпечуючи динамічну оцінку надійності акторів на основі їхніх характеристик та поточної поведінки. Зонна модель безпеки з матрицею дозволених переходів та порогами довіри формалізує інтуїтивне уявлення про глибокий захист у вигляді, придатному для алгоритмічної перевірки політик доступу та оптимального розміщення точок контролю. Метрика ефективності покриття середовища надає кількісний критерій для оптимізаційної задачі розміщення точок контролю, яка у відомих роботах розглядається переважно лише евристично. Запропонована модель є достатньо абстрактною для застосування до організацій різного масштабу, водночас достатньо деталізованою для практичного використання завдяки параметризації атрибутів вузлів, типізації ребер та порогових значень.

2.4 Модель процесу запобігання витокам даних

Модель витоків, яку задано за формулою (2.39), формалізує природу витоків як подій несанкціонованої передачі даних, визначає ризик-модель, яку задану за формулою (2.33), для оцінки загроз. Модель корпоративного середовища (2.49) описує мережну топологію, рольову модель доступу, потоки даних та точки контролю. Модель даних визначає ознаковий простір в формулі (2.13) для класифікації та методи контентного аналізу. Тепер постає завдання синтезувати ці окремі компоненти у цілісну архітектуру системи запобігання витокам даних, здатну ефективно функціонувати у реальному корпоративному середовищі.

Синтез моделей передбачає не просте механічне об’єднання компонентів, а побудову системи з емерджентними властивостями, де взаємодія модулів породжує якості, відсутні в окремих складових. Модуль класифікації контенту, що спирається на ознаковий простір в формулі (2.13), має взаємодіяти з модулем поведінкового аналізу, який оцінює дії актора згідно з моделлю актора (формула (2.27)) та функцією довіри (формула (2.28)). Результати обох модулів агрегуються у модулі прийняття рішень, що враховує контекст середовища (формула (2.49)) та поточні політики безпеки. Ця взаємодія відбувається у реальному часі для кожної події передачі даних, що вимагає чіткої формалізації архітектури та протоколів обміну інформацією між компонентами.

Синтез моделей попередніх підрозділів у єдину формальну структуру забезпечує цілісне уявлення про ЗВД-систему як об'єкт проектування. Композиційний підхід дозволяє чітко визначити інтерфейси між компонентами та їхні функціональні залежності.

Результуючу модель системи ЗВД будуємо як композицію моделей попередніх підрозділів. Нехай M_{leak} позначає модель витоків (2.39), E_{corp} – модель корпоративного середовища (2.49), D_{model} – модель даних, визначену кортежем документа та ознаковим вектором (2.4), (2.13). Тоді результуючу модель системи ЗВД визначимо як кортеж, що об’єднує ці компоненти з додатковими елементами, специфічними для інтегрованої системи.

$$S_{ЗВД} = \langle M_{leak}, E_{corp}, D_{model}, Pol, F, \Omega, \Delta \rangle, \tag{2.53}$$

де $S_{\text{ЗВД}}$ – результуюча модель системи запобігання витокам даних;

M_{leak} – модель витоків;

E_{corp} – корпоративне середовище;

D_{model} – модель даних;

Pol – множина політик безпеки, узгоджена з (2.39);

F – функції обробки та трансформації даних;

Ω – цільова функція оптимізації;

Δ – функція прийняття рішень.

Архітектура системи ЗВД структурується за модульним принципом, де кожен модуль реалізує специфічну функціональність та взаємодіє з іншими через чітко визначені інтерфейси. Такий підхід забезпечує гнучкість налаштування, можливість незалежного оновлення компонентів та масштабованість системи. Виділяються чотири основні функціональні модулі: модуль класифікації контенту, модуль поведінкового аналізу, модуль адаптації політик та модуль прийняття рішень. Додатково система включає інфраструктурні компоненти: шину подій для обміну інформацією, сховище даних для зберігання профілів та політик, інтерфейс адміністрування для налаштування та моніторингу.

Сигнатура модуля класифікації задається відображенням:

$$M_{content}: D \times X \to [0,1] \times 2^K, \tag{2.54}$$

де $M_{content}$ – модуль класифікації контенту;

D – множина документів;

X – контексти передачі, область значень поєднує оцінку чутливості в [0,1] з підмножиною категорій;

K – множина категорій конфіденційних даних.

На вхід надходить документ $d \in D$, разом з контекстом передачі $x \in X$. Результатом є числове значення чутливості у діапазоні від нуля до одиниці та множина категорій $k \in K$, до яких належить документ.

Внутрішня структура модуля класифікації контенту охоплює підсистему видобування тексту з документів різних форматів, підсистему попередньої обробки для нормалізації та токенізації, підсистему видобування ознак для побудови векторного представлення, підсистему класифікації для застосування навчених

моделей. Ці підсистеми утворюють конвеєр обробки, де результат кожного етапу є вхідними даними для наступного.

Внутрішня оцінка чутливості обчислюється за формулою:

$$S(d) = max\left(\alpha_1 \cdot S_K(d) + \alpha_2 \cdot S_{pattern}(d) + \alpha_3 \cdot S_{cl}(d),\ S_{label}(d)\right), \quad (2.55)$$

де S_K – категорійна оцінка, обчислювана за (2.3);

$S_{pattern}$ – оцінка на основі структурованих шаблонів;

S_{cl} – оцінка класифікатора;

S_{label} – явна мітка класифікації, якщо вона присутня.

Коефіцієнти α_i визначають відносну вагу компонентів при сумі, рівній одиниці. Операція максимуму гарантує, що явна мітка має пріоритет над автоматичною класифікацією.

Контентний аналіз виявляє конфіденційні дані, але не враховує контекст дій користувача. Поведінковий модуль доповнює контентну класифікацію, ідентифікуючи аномальні закономірності активності, що можуть свідчити про підготовку до витоку ще до фактичної передачі даних.

Модуль поведінкового аналізу оцінює дії акторів на предмет відповідності їхнім типовим закономірностям активності та виявляє аномалії, що можуть свідчити про спробу витоку. Цей модуль спирається на модель актора (формула (2.27)) та функцію довіри (формула (2.28)), розширюючи їх механізмами статистичного профілювання та виявлення відхилень. Формально модуль поведінкового аналізу задамо відображенням:

$$M_{behavior}: A \times E \times T \to [0,1] \times \mathbb{R}^+, \quad (2.56)$$

де $M_{behavior}$ – модуль поведінкового аналізу;

A – множина акторів;

E – події передачі;

T – множина моментів часу, вихід поєднує показник аномальності в $[0,1]$ з R^+, тобто з невід'ємною оцінкою ризику поведінки.

Входом слугують ідентифікатор актора $a \in A$, подія передачі $e \in E$ та момент часу t; виходом є показник аномальності в діапазоні $[0,1]$ і похідна від нього оцінка

ризику поведінки. Нормальну поведінку актора a за вікно спостереження W задамо профілем так:

$$\Pi_a = (\mu_a, \Sigma_a), \tag{2.57}$$

де μ_a – вектор математичних сподівань ознак активності актора, а Σ_a – коваріаційна матриця цих ознак, оцінені за подіями вікна W.

Ознаки активності походять із моделі актора (2.27), а сам профіль доповнює функцію довіри (2.28) статистичним виміром норми, описуючи не лише рівень довіри до суб'єкта, а й характерну для нього картину дій. Відхилення поточного спостереження x_t від профілю Π_a вимірюється відстанню Махаланобіса (2.58):

$$D_M\big(\boldsymbol{x}_t, \Pi(a)\big) = \sqrt{(\boldsymbol{x}_t - \mu_a)^T \Sigma_a^{-1} (\boldsymbol{x}_t - \mu_a)}, \tag{2.58}$$

де D_M – відстань Махаланобіса між спостереженням і профілем;

x_t – вектор ознак поточного спостереження в момент t;

$\Pi(a)$ – профіль нормальної поведінки актора a;

μ_a – вектор середніх значень ознак профілю;

Σ_a – коваріаційна матриця цих ознак;

Σ_a^{-1} – обернена до неї матриця;

верхній індекс T - транспонування вектора відхилення.

Показник аномальності отримується шляхом нормалізації (2.58) через функцію розподілу хі-квадрат з n ступенями свободи, що відповідає кількості ознак у профілі

$$An(a,t) = F_{\chi_n^2}\left(D_M^2\big(\boldsymbol{x}_t, \Pi(a)\big)\right), \tag{2.59}$$

де $An(a,t)$ – показник аномальності поведінки актора a в момент t;

$F_{\chi_n^2}$ – функція розподілу хі-квадрат із n ступенями свободи;

n – кількість ознак профілю (число ступенів свободи);

D_M^2 – квадрат відстані Махаланобіса;

x_t – вектор ознак поточного спостереження;

$\Pi(a)$ – профіль нормальної поведінки актора.

Таке нормування спирається на наближену багатовимірну нормальність ознак вікна та невироджену коваріаційну матрицю. Для коротких вікон оцінку коваріації слід регуляризувати (стисканням до діагоналі) або використовувати псевдообернену матрицю, а відповідність квантилів перевіряти емпірично.

Значення показника аномальності інтерпретується як ймовірність того, що випадкова активність з нормального профілю матиме меншу відстань від центра розподілу. Високі значення, наприклад більші за 0,95, свідчать про активність, що є статистично нетиповою для цього актора.

Модуль адаптації політик забезпечує динамічне коригування параметрів системи ЗВД відповідно до змін у середовищі та накопиченого досвіду. Статичні політики, визначені на етапі впровадження, неминуче застарівають: з'являються нові типи конфіденційних даних, змінюються бізнес-процеси, еволюціонують шаблони легітимної та зловмисної активності. Адаптивний механізм дозволяє системі підтримувати ефективність без постійного ручного втручання адміністраторів.

$$M_{adapt}: Pol \times H \times \Psi_{fb} \to Pol', \tag{2.60}$$

де M_{adapt} – модуль адаптації політик;

Pol – поточна конфігурація політик;

H – історія подій та рішень;

Ψ_{fb} – зворотний зв'язок від аналітиків;

Pol' – оновлена конфігурація політик.

Модуль адаптації приймає поточну конфігурацію політик Pol, історію подій та рішень H, зворотний зв'язок від аналітиків Ψ_{fb}, і генерує оновлену конфігурацію політик Pol'. Процес адаптації може відбуватися періодично за розкладом або ініціюватися накопиченням достатнього обсягу зворотного зв'язку.

Політика системи ЗВД формалізується як набір правил, що визначають реакцію на події передачі даних. Кожне правило p складається з чотирьох компонентів: умови, дії, пріоритету та статусу активності. Умова є предикатом над простором подій, що може включати обмеження на чутливість даних, характеристики актора, властивості каналу, контекстуальні фактори. Дія належить множині можливих реакцій: дозвіл, блокування, карантин, сповіщення, запит підтвердження. Пріоритет визначає порядок застосування правил при конфлікті. Прапорець дозволяє тимчасово деактивувати правило без його видалення.

Адаптація політик охоплює кілька механізмів. Коригування порогових значень змінює параметри τ_q та τ_b, що визначають межі між рішеннями дозволу, карантину та блокування. Оновлення ваг класифікатора модифікує параметри моделі

машинного навчання на основі нових прикладів. Розширення шаблонів додає нові регулярні вирази для виявлення структурованих конфіденційних даних. Налаштування пріоритетів правил змінює порядок їх застосування на основі статистики ефективності.

$$\tau_q(t+1) = \tau_q(t) - \eta \cdot \frac{\partial L}{\partial \tau_q}, \quad \tau_b(t+1) = \tau_b(t) - \eta \cdot \frac{\partial L}{\partial \tau_b}, \tag{2.61}$$

де τ_q – поріг карантину;

τ_b – поріг блокування;

t – номер кроку адаптації;

η – швидкість навчання;

L – функція втрат, що враховує хибні спрацювання та пропущені витоки;

$\partial L/\partial \tau_q,\ \partial L/\partial \tau_b$ – частинні похідні функції втрат за відповідним порогом.

Оскільки кількості хибних спрацювань і пропусків є кусково сталими функціями порогів, то похідні тут розглядаються як похідні згладженої апроксимації функції втрат. На практиці оновлення порогів виконується безградієнтним еволюційним пошуком, а в формулі (2.61) відіграє роль концептуальної схеми напряму корекції.

Формула (2.61) описує градієнтне оновлення порогів, де L – функція втрат, що враховує хибні спрацювання та пропущені витоки, η – швидкість навчання.

Модуль прийняття рішень є центральним компонентом, що агрегує результати інших модулів та генерує фінальне рішення щодо кожної події передачі даних. Цей модуль реалізує логіку ризик-моделі (2.33) з пороговими значеннями рішень (2.36), з урахуванням контексту середовища (2.49):

$$M_{decision}: [0,1] \times [0,1] \times C \times Pol \to \{Dc_{allow}, Dc_{quarantine}, Dc_{block}\} \times R^{+}, \tag{2.62}$$

де $M_{decision}$ – модуль прийняття рішень, а перші два множники [0,1] відповідають оцінці чутливості та показнику аномальності;

C – множина каналів передачі;

Pol – конфігурація політик;

Dc_{allow} – рішення «дозволити»;

$Dc_{quarantine}$ – рішення «карантин»;

Dc_{block} – рішення «блокувати» (усі три відповідають рішенням правила з формули (2.36));

R^+ – оцінка впевненості (невід'ємне дійсне число).

Модуль приймає оцінку чутливості від модуля класифікації контенту, показник аномальності від модуля поведінкового аналізу, характеристики каналу передачі та поточну конфігурацію політик. Результатом є рішення з множини допустимих дій та числова оцінка впевненості у цьому рішенні.

Окремі модулі системи генерують різнорідні оцінки ризику: чутливість контенту, аномальність поведінки, характеристики каналу. Для прийняття фінального рішення необхідний механізм агрегації, що комбінує ці сигнали у єдину метрику з урахуванням їхньої відносної важливості.

Агрегація оцінок ризику від різних компонентів здійснюється через зважене комбінування. Нехай $r_{content}$ – ризик на основі чутливості контенту, $r_{behavior}$ – ризик на основі поведінкового аналізу, $r_{channel}$ – ризик каналу передачі, $r_{context}$ – контекстуальний ризик. Розширюючи ризик-модель (2.33), інтегрований ризик події обчислюємо за формулою:

$$R_{agg}(e) = w_1 \cdot r_{content} + w_2 \cdot r_{behavior} + w_3 \cdot r_{channel} + w_4 \cdot r_{context}, \qquad (2.63)$$

де $R_{agg}(e)$ – інтегрований ризик події e;

w_1, w_2, w_3, w_4 – вагові коефіцієнти компонент;

$r_{content}$ – ризик на основі чутливості контенту;

$r_{behavior}$ – ризик за результатами поведінкового аналізу;

$r_{channel}$ – ризик каналу передачі;

$r_{context}$ – контекстуальний ризик.

Ваги w_i є параметрами системи, що визначають відносну важливість різних факторів ризику. Їхня сума дорівнює одиниці. Показник з формули (2.63) є агрегованою оцінкою (score), яку порогове правило з формули (2.36) використовує замість прогнозованої ймовірності. На відміну від формули (2.33), він не є добутком імовірності на вплив, і обидві величини не тотожні.

Альтернативно може застосовуватися мультиплікативна модель, де ризики перемножуються, або модель максимуму, де враховується лише найвищий з ризиків. Вибір моделі агрегації залежить від політики безпеки організації та характеру загроз.

Рішення приймається на основі порівняння інтегрованого ризику з пороговими значеннями. Ця логіка узагальнює (2.36), додаючи можливість контекстуального модифікування порогів.

$$Dc(e) = \begin{cases} Dc_{allow}, & R_{agg}(e) < \tau_q \cdot \kappa(ctx) \\ Dc_{quarantine}, & \tau_q \cdot \kappa(ctx) \le R_{agg}(e) < \tau_b \cdot \kappa(ctx), \\ Dc_{block}, & R_{agg}(e) \ge \tau_b \cdot \kappa(ctx) \end{cases} \quad (2.64)$$

де $Dc(e)$ – рішення для події e;

$Dc_{allow}, Dc_{quarantine}, Dc_{block}$ – рішення з (2.62);

$R_{agg}(e)$ – інтегрований ризик події;

τ_q, τ_b – пороги з (2.36);

$\kappa(ctx)$ – контекстний множник, що модифікує пороги;

ctx – контекст події.

$\kappa(ctx)$ модифікує пороги залежно від контексту: для особливих періодів, таких як кінець фінансового року або період аудиту, пороги можуть знижуватися, посилюючи контроль, тоді як у звичайні періоди застосовуються стандартні значення.

Взаємодія між модулями організована через шину подій, що забезпечує асинхронний обмін повідомленнями. Кожна подія передачі даних ініціює послідовність обробки, де результати одних модулів використовуються іншими. Потік обробки події може бути описаний алгоритмічно.

При надходженні події передачі e система паралельно запускає аналіз контенту та поведінковий аналіз. Модуль класифікації контенту обчислює чутливість документа $S(d)$ та визначає категорії конфіденційних даних $\kappa catd(\ )$. Модуль поведінкового аналізу обчислює показник аномальності $An(a,t)$ та оновлює профіль актора. Результати передаються до модуля прийняття рішень, який обчислює інтегрований ризик $R_{agg}(e)$ та генерує рішення $Dc(e)$. Рішення застосовується до події: дозвіл, карантин або блокування. Подія логується разом з усіма проміжними оцінками для подальшого аналізу та навчання. Модуль адаптації політик періодично аналізує накопичені журнали та зворотний зв’язок, оновлюючи параметри системи. Наскрізний потік обробки події передачі даних зображено на рисунку 2.6.

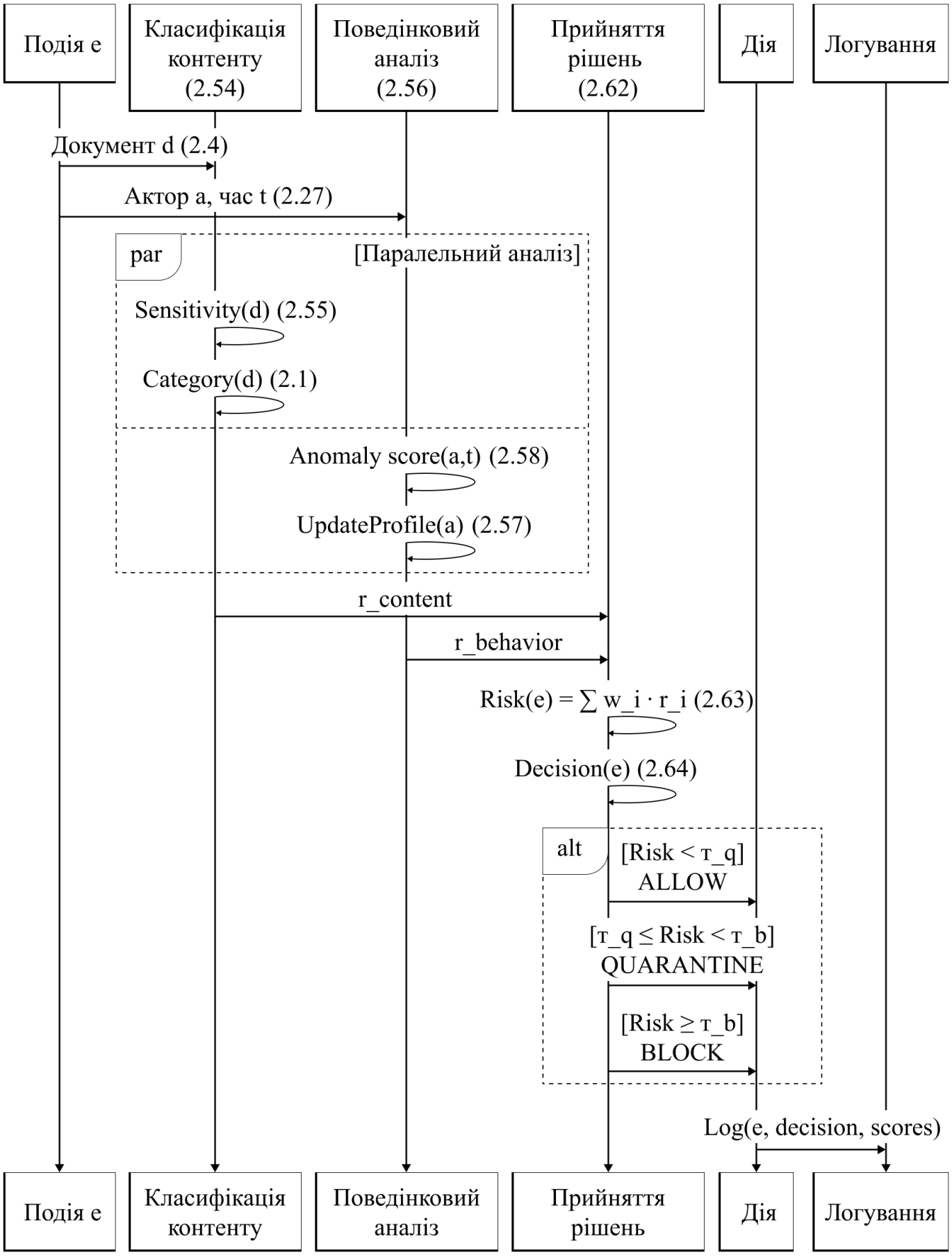


Рисунок 2.6 – Потік обробки події передачі даних у системі ЗВД

Формалізація взаємодії компонентів вимагає визначення інтерфейсів та протоколів обміну. Кожен модуль надає набір операцій, які можуть викликатися іншими модулями або зовнішніми системами. Модуль класифікації контенту надає операцію $Op_{classify}(d)$ для визначення чутливості та категорій. Модуль поведінкового аналізу надає операції $Op_{evaluate}(a, e)$ для оцінки аномальності та $Op_{updateprofile}(a, e)$ для оновлення профілю. Модуль адаптації політик надає операцію $Op_{adapt}(f)$ для ініціювання циклу адаптації. Модуль прийняття рішень надає операцію $Op_{decide}(e)$ для генерації рішення.

Застосування паралельної обробки для класифікації контенту та поведінкового аналізу, які є незалежними, суттєво скорочує загальну латентність. Типові вимоги до латентності становлять сотні мілісекунд для мережного трафіку та секунди для операцій з файлами.

Для формалізації потоку обробки події визначається скінченний автомат станів. Множина станів S включає: очікування події, аналіз контенту, поведінковий аналіз, прийняття рішення, застосування дії, логування. Функція переходів δ_{st} визначає зміну стану при надходженні відповідних сигналів від модулів. Початковим станом є очікування, кінцевим – логування після успішної обробки.

Цільова функція оптимізації системи ЗВД відображає баланс між безпекою та операційною ефективністю. Ідеальна система блокувала б усі витоки, не створюючи жодних хибних спрацювань. На практиці ці цілі конфліктують: підвищення чутливості системи зменшує пропущені витоки, але збільшує хибні блокування легітимної активності. Розширюючи формулу (2.37), формалізуємо цей компроміс цільовою функцією так:

$$\Omega = \min_{\theta} \left(C_{FN} \cdot N_{FN}(\theta) + C_{FP} \cdot N_{FP}(\theta) + C_{op} \cdot l(\theta) \right), \quad (2.65)$$

де θ – вектор параметрів системи, що включає порогові значення, ваги класифікаторів, параметри профілювання;

N_{FN} – кількість пропущених витоків;

N_{FP} – кількість хибних блокувань;

$l(\theta)$ – середня латентність обробки;

тут символ Ω - мінімальне досяжне значення цільової функції, тобто оптимум виразу в дужках за параметрами θ.

Коефіцієнти C_{FN}, C_{FP}, C_{op} визначають відносну вартість кожного типу помилок та операційних витрат.

Вектор ваг інтегрованого ризику $w = (w_1, w_2, w_3, w_4)$ з формули (2.63) не задається раз і назавжди експертно, а входить до вектора параметрів системи θ цільової функції оптимізації (2.65). Початкові експертні значення відіграють роль лише першого наближення: вони задають точку старту, від якої система уточнює ваги так, щоб мінімізувати інтегральну функцію втрат на реальному потоці подій. Це знімає проблему суб'єктивності фіксованих експертних оцінок, оскільки остаточні значення ваг визначаються спостереженими наслідками рішень, а не апріорним судженням.

Те саме стосується решти вагових коефіцієнтів моделі: експертне значення є початковим, а робоче формується автоналаштуванням. Позначення α_i у формулах (2.24) та (2.55) стосуються різних, незалежних наборів ваг: у (2.24) це ваги складових агрегованої чутливості в моделі даних, у (2.55) це ваги складових оцінки чутливості в модулі класифікації контенту; обидва набори нормовані до одиниці й налаштовуються окремо. Зведення всіх вагових коефіцієнтів подано в таблиці 2.7.

Таблиця 2.7 - Вагові коефіцієнти моделі

Символ	Формула	Призначення	Діапазон	Спосіб визначення (початкове → робоче)
1	2	3	4	5
w_k	(2.2), (2.3)	Чутливість категорії конфіденційних даних k	[0, 1]	Регуляторно та експертно (напр., $w_{FIN} = 0{,}9$; $w_{PII} = 0{,}6$) → автокоригування за історією інцидентів
$n_{ref,k}$	(2.3)	Референсна кількість входжень категорії k (масштаб насичення за обсягом)	ціле ≥ 1	Регуляторно ($n_{ref} = 1$ для критичних категорій) або статистично → автокоригування

Кінець таблиці 2.7

1	2	3	4	5
α_i (модель даних)	(2.24)	Ваги складових агрегованої чутливості (категорійна, шаблонна, класифікаторна)	[0, 1], $\Sigma = 1$	Експертно (0,5; 0,3; 0,2) → автоналаштування
α_i (модуль контенту)	(2.55)	Ваги складових оцінки чутливості в модулі класифікації контенту	[0, 1], $\Sigma = 1$	Експертно (0,4; 0,4; 0,2) → автоналаштування
$w_1 \dots w_4$ (вплив)	(2.35)	Внески чутливості, обсягу, регуляторних та бізнес-наслідків у I(d)	[0, 1], $\Sigma = 1$	Експертно (0,4; 0,2; 0,3; 0,1) → автоналаштування
$w_1 \dots w_4$ (ризик)	(2.63)	Баланс контентного, поведінкового, канального та контекстного ризику в R(e)	[0, 1], $\Sigma = 1$	Експертно (0,4; 0,3; 0,2; 0,1) → оптимізація у складі θ (2.65)
w_T	(2.28)	Баланс статичної та динамічної складових функції довіри до актора	[0, 1]	Експертно (0,6…0,8) → автоналаштування
$\psi_{sev}(d)$	(2.66)	Серйозність потенційного інциденту для документа d	R+	Експертно, за категорією та рівнем конфіденційності
$\psi_{biz}(e)$	(2.66)	Бізнес-критичність операції, заблокованої подією e	R+	Експертно, за типом бізнес-процесу
$\psi_{pri}(e)$	(2.67)	Пріоритетність бізнес-процесу у вартості хибного блокування	R+	Експертно (політика безпеки)

Функція вартості помилок деталізується з урахуванням типу та серйозності інцидентів. Не всі пропущені витоки однаково критичні, бо наприклад, витік публічної інформації, помилково класифікованої як конфіденційна, відрізняється від витоку реальних персональних даних чи комерційних таємниць. Аналогічно,

блокування важливого бізнес-процесу має вищу вартість за блокування рутинної операції. Визначимо її так:

$$C = \sum_{i \in FN} c_{fn}(i) \cdot \psi_{sev}(d_i) + \sum_{j \in FP} c_{fp}(j) \cdot \psi_{biz}(e_j), \quad (2.66)$$

де FN та FP – множини хибнонегативних і хибнопозитивних рішень (індекси i та j пробігають відповідні документи d_i та події e_j);

c_{fn} та c_{fp} – вартості пропущеного витоку та хибного блокування, деталізовані у формулі (2.67);

$\psi_{sev}(d_i)$ – ваговий коефіцієнт серйозності потенційного інциденту для документа d_i;

$\psi_{biz}(e_j)$ – коефіцієнт бізнес-критичності операції, заблокованої подією e_j, обидва коефіцієнти задаються експертно відповідно до політики безпеки організації (табл. 2.7).

Функція вартості помилок уточнюється для різних сценаріїв. Хибнопозитивне спрацювання при низькій чутливості має максимальну питому вартість: блокування явно неконфіденційної передачі є найменш виправданим втручанням у робочий процес, і саме це відображає множник $(1 - S(d))$ у формулі (2.67). Хибнонегативний результат при високій чутливості має максимальну вартість через серйозність потенційного витоку. Проміжні сценарії зважуються пропорційно.

$$\begin{aligned} c_{fn}(d) &= c_{base} \cdot e^{\gamma \cdot S(d)}, \\ c_{fp}(e) &= c_{base} \cdot \psi_{pri}(e) \cdot \big(1 - S(d)\big), \end{aligned} \quad (2.67)$$

де cfn(d) - вартість пропущеного витоку для документа d, що зростає експоненційно з його чутливістю;

$c_{fp}(e)$ – вартість хибнопозитивного спрацювання для події e;

c_{base} – базова вартість помилки;

$\psi_{pri}(e)$ – ваговий коефіцієнт пріоритетності бізнес-процесу події;

$S(d)$ – чутливість об'єкта даних d;

$\big(1 - S(d)\big)$ – множник, що знижує вартість для менш чутливих даних;

γ – коефіцієнт зростання вартості пропуску зі збільшенням чутливості.

Експоненційна залежність вартості пропущеного витоку від чутливості відображає непропорційно високі наслідки витоку цінних даних порівняно з менш

чутливими. Вартість хибного блокування зменшується зі зростанням чутливості, оскільки блокування потенційно конфіденційних даних є більш виправданим.

Компроміс між хибними спрацюваннями та пропущеними витоками візуалізується кривою ROC, що відображає залежність True Positive Rate від False Positive Rate при варіюванні порогу рішення. Оптимальну робочу точку визначимо мінімізацією загальної вартості помилок:

$$\theta^* = argmin_\theta\big(C_{FN} \cdot \big(1 - TPR(\theta)\big) \cdot N_{pos} + C_{FP} \cdot FPR(\theta) \cdot N_{neg}\big), \qquad (2.68)$$

де θ^* – оптимальний поріг рішення;

$argmin_\theta$ – оператор, що повертає те значення θ, що мінімізує вираз;

θ – сам поріг рішення;

C_{FN} – вартість пропущеного витоку;

C_{FP} – вартість хибного спрацювання;

$TPR(\theta)$ – частка істинно позитивних при порозі θ;

$FPR(\theta)$ – частка хибнопозитивних;

N_{pos} – кількість конфіденційних прикладів;

N_{neg} – кількість неконфіденційних.

На відміну від значення в формулі (2.65), де θ позначає повний вектор параметрів системи, тут θ є лише скалярним порогом рішення. При співвідношенні вартостей $\frac{C_{FN}}{C_{FP}}$, що значно перевищує одиницю, оптимальна точка зміщується у бік вищої чутливості за рахунок більшої кількості хибних спрацювань.

Взаємодія чотирьох модулів організована через шину подій. Таке архітектурне рішення зумовлене трьома перевагами.

По-перше, модулі працюють з різною швидкістю. Класифікація контенту та прийняття рішень виконуються для кожної події майже миттєво, поведінкове профілювання накопичує статистику за вікнами спостереження, а адаптація політик запускається періодично, за накопиченим досвідом. Асинхронний обмін повідомленнями через шину дозволяє кожному модулю споживати та публікувати дані у власному темпі, не блокуючи решту. Обмін реалізовано за моделлю «публікація-підписка»: модуль публікує свій результат, а зацікавлені модулі підписуються на потрібні типи подій.

По-друге, шина розв'язує модулі за контрактами повідомлень, а не за прямими викликами. Завдяки цьому реалізацію будь-якого модуля (наприклад, класифікатор контенту) можна замінити чи оновити незалежно від інших, поки зберігається формат повідомлень. Ця властивість безпосередньо забезпечує вимоги до гнучкості налаштування та масштабованості системи, сформульовані на початку підрозділу.

По-третє, кожна подія передачі та всі проміжні оцінки проходять через шину і фіксуються в журналі. Журнал шини утворює повний відтворюваний слід прийняття кожного рішення, потрібний для розслідування інцидентів та для аудиту, тобто реалізує вимогу аудиту, закладену в моделі.

Прямі синхронні виклики відкинуто, оскільки вони зв'язали б життєві цикли та темпи модулів. Повільна адаптація політик блокувала б швидкий тракт прийняття рішень, а заміна одного компонента вимагала б змін у решті. Архітектуру взаємодії компонентів через шину подій зображено на рисунку 2.7.

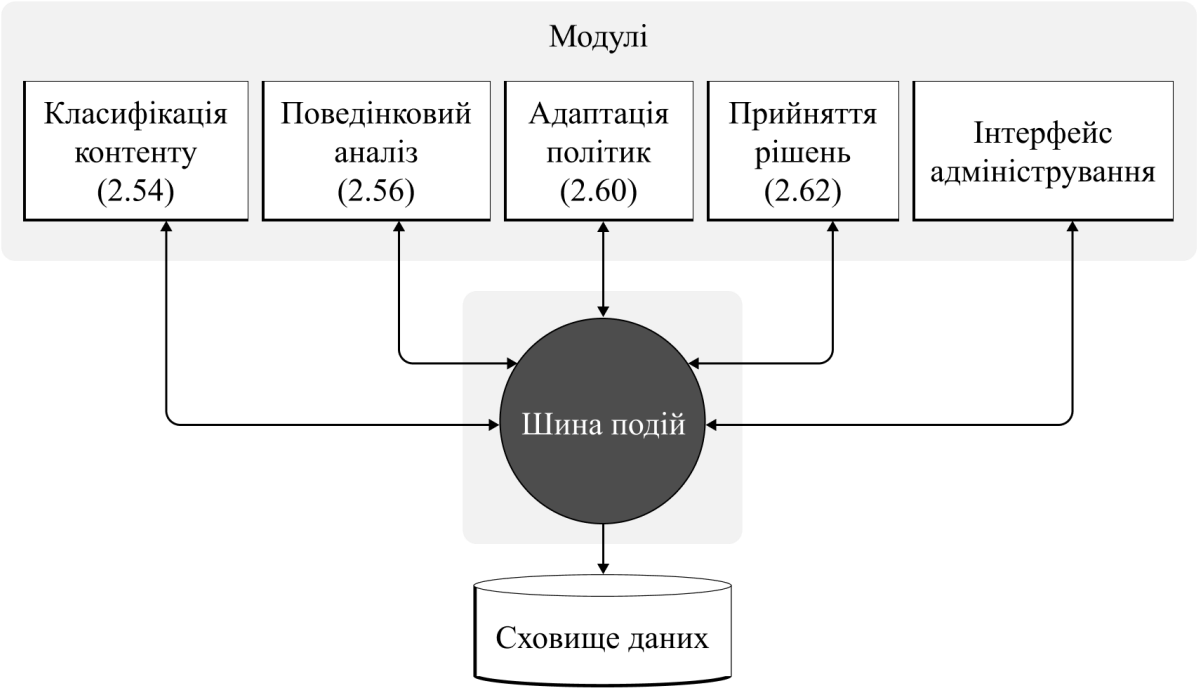


Рисунок 2.7 – Архітектура інтеграції компонентів через шину подій

Сховище даних системи ЗВД містить кілька категорій інформації: профілі користувачів з історією активності та статистичними характеристиками, конфігурацію політик з правилами та пороговими значеннями, моделі класифікаторів з навченими параметрами, журнали подій та рішень для аудиту та навчання. Вимоги до сховища включають високу доступність, швидкий доступ для операцій реального часу та можливість аналітичних запитів для звітності.

Інтерфейс адміністрування надає можливості налаштування системи, моніторингу її роботи та реагування на інциденти. Адміністратор може переглядати та модифікувати політики, аналізувати статистику спрацювань, переглядати деталі окремих інцидентів, надавати зворотний зв’язок для навчання системи. Інтерфейс аналітика безпеки зосереджений на обробці карантинних подій та розслідуванні інцидентів.

Специфікацію системи ЗВД завершуємо переліком інваріантів, тобто властивостей, що мають виконуватися за будь-яких умов. Інваріант повноти вимагає, щоб кожна подія передачі отримала рішення: не існує події, що залишилася без обробки. Інваріант несуперечності вимагає, щоб однакові події отримували однакові рішення при незмінній конфігурації. Інваріант аудиту вимагає збереження повної інформації про кожне рішення для можливості пізнішого аналізу.

Верифікація цих інваріантів забезпечується архітектурою системи. Повноту гарантує надійна черга подій з повторною доставкою та ідемпотентною обробкою (у прототипі реалізовано синхронне спрощення з послідовною обробкою кожної події до ухвалення рішення), детермінована логіка рішень – несуперечність, транзакційне логування – аудит.

Масштабованість системи ЗВД є важливим фактором для корпоративного застосування. Обсяг подій передачі даних у великій організації може сягати мільйонів на день, що вимагає розподіленої архітектури. Горизонтальне масштабування досягається реплікацією модулів аналізу з розподілом навантаження між екземплярами. Модуль прийняття рішень може масштабуватися за умови узгодженого доступу до політик. Сховище даних застосовує шардування та реплікацію для забезпечення продуктивності та доступності.

Відмовостійкість забезпечується резервуванням компонентів та механізмами відновлення. При відмові екземпляра модуля його навантаження перерозподіляється на інші екземпляри. При недоступності сховища політик система переходить у режим за замовчуванням з підвищеною обережністю. При повній недоступності системи ЗВД трафік може або блокуватися, або пропускатися залежно від політики безпеки організації.

Інтеграція з екосистемою безпеки організації реалізується через стандартні протоколи та інтерфейси. Система ЗВД обмінюється подіями з SIEM для кореляції з

іншими джерелами безпеки. Інтеграція з IAM забезпечує актуальну інформацію про ролі та дозволи користувачів. Взаємодія з EDR дозволяє отримувати інформацію про стан безпеки кінцевих пристроїв. Інтеграція з CASB розширює покриття на хмарні сервіси.

Модульне подання моделі ЗВД-системи об'єднує описані функціональні модулі у цілісну структуру:

$$S_{\text{мод}} = \langle M_{content}, M_{behavior}, M_{adapt}, M_{decision}, Pol, \Omega, I_{ext} \rangle, \quad (2.69)$$

де $S_{\text{мод}}$ – модульне подання моделі ЗВД-системи;

$M_{content}$ – модуль класифікації контенту;

$M_{behavior}$ – модуль поведінкового аналізу, побудований на моделі актора з M_{leak};

M_{adapt} – модуль адаптації політик;

$M_{decision}$ – модуль прийняття рішень;

Pol – множина політик безпеки;

Ω – цільова функція оптимізації;

I_{ext} – інтерфейси інтеграції із зовнішніми системами.

Кортеж (формула (2.69)) є модульним поданням моделі (2.53): модулі реалізують множину функцій обробки F та функцію прийняття рішень Δ, а I_{ext} охоплює інтерфейси інтеграції із зовнішніми системами.

Функціонування системи описується як ітеративний процес обробки потоку подій з періодичною адаптацією параметрів. На кожній ітерації система отримує чергову подію передачі, застосовує модулі аналізу та прийняття рішень, логує результат та оновлює статистику. Після накопичення достатнього обсягу даних модуль адаптації коригує параметри для покращення майбутніх рішень.

Окремим аспектом архітектури є підтримка пояснюваності рішень. Система ЗВД має надавати зрозумілі пояснення, чому конкретна передача була заблокована або направлена на карантин. Це важливо з кількох причин: для забезпечення довіри користувачів до системи, для можливості оскарження хибних рішень, для відповідності регуляторним вимогам щодо автоматизованого прийняття рішень.

Пояснення рішення структурується як перелік факторів, що вплинули на результат, з вказівкою їхнього внеску. Для рішення блокування пояснення може

включати: виявлені конфіденційні дані з прикладами, аномальні характеристики поведінки відправника, ризикові властивості каналу передачі, правила політики, що спрацювали.

Принцип найменших привілеїв застосовується до компонентів системи: кожен модуль має доступ лише до тих ресурсів, які необхідні для виконання його функцій. Модуль класифікації контенту читає вміст документів, але не має доступу до профілів користувачів. Модуль поведінкового аналізу працює з агрегованою статистикою, не маючи доступу до необробленого вмісту документів. Модуль адміністрування має широкі повноваження, але потребує посиленої автентифікації та аудиту.

Результуюча модель ЗВД-системи, що складається з моделі витоків (2.39), корпоративного середовища (2.49) та даних (2.13), має низку характеристик, що відрізняють її від наявних рішень. На відміну від монолітних ЗВД-проєктів, запропонована архітектура побудована за модульним принципом із взаємодією компонентів через шину подій, що забезпечує незалежну еволюцію та масштабування кожного модуля. Багатокритеріальна цільова функція оптимізації формалізує компроміс між безпекою, зручністю та продуктивністю – цей компроміс у більшості комерційних систем встановлюється інтуїтивно без кількісного обґрунтування. Модель прийняття рішень із трьома рівнями реагування (дозвіл, карантин, блокування) є гнучкішою за бінарну схему, оскільки проміжний рівень карантину дозволяє залучати аналітика лише для неоднозначних випадків. Механізм агрегації оцінок ризику від різних модулів через зважене комбінування уможливлює адаптацію відносної ваги контентного, поведінкового та канального факторів залежно від політики організації. Формальні інваріанти системи забезпечують верифікованість архітектури, що є вимогою регуляторних стандартів.

2.5 Висновки до другого розділу

Формалізовано модель витоку даних як подію несанкціонованого переміщення інформації, що описується кортежем із моделей актора, об'єкта даних та каналу передачі. Модель актора враховує композитну функцію довіри, яка поєднує статичну та динамічну складові, а чутливість даних визначається як максимум між мітковою та контентною оцінками. Для каналів передачі введено функцію ризику, що залежить

від типу каналу, пропускної здатності, рівня моніторингу та призначення. На основі байєсівського розширення ризикової моделі побудовано трирівневу модель прийняття рішень із оптимізовуваними порогами, що мінімізує вартість помилок класифікації обох типів.

Побудовано модель корпоративного середовища як типізований зважений граф із рольовою моделлю доступу, функцією потоків даних із ваговим урахуванням чутливості та моделлю контрольних точок ЗВД-системи з метрикою покриття. Запровадження зон безпеки з межами довіри та інтеграція середовищних характеристик у функції авторизації й оцінки ризику дають змогу враховувати мережну топологію та організаційну структуру при прийнятті рішень про блокування чи пропуск інформаційних потоків. Розроблено модель даних, що охоплює категоризацію за типами конфіденційної інформації, обробку складених і вкладених документів із рекурсивним обчисленням чутливості, а також конвеєр попередньої обробки тексту й побудову ознакового простору на основі частотних, шаблонних, словникових та метаданих-характеристик. Функція аналізованості контенту забезпечує коригування ризику з урахуванням балансу між результатами контентного аналізу та контекстними ознаками.

На основі запропонованих моделей синтезовано узагальнену модель ЗВД-системи, що об'єднує модель витоку, модель корпоративного середовища та модель даних із чотирма функціональними модулями: класифікації контенту, поведінкового аналізу на основі відстані Махаланобіса, адаптації політик та прийняття рішень із зваженою агрегацією ризиків. Сформульовано задачу багатокритеріальної оптимізації, що балансує повноту виявлення витоків, продуктивність інформаційної системи та навантаження на персонал безпеки, з вибором оптимальних порогів за ROC-аналізом. Для синтезованої системи сформульовано та обґрунтовано інваріанти повноти, несуперечності та аудитованості, а також передбачено інтеграцію з SIEM/IAM-системами, горизонтальну масштабованість та відмовостійкість. Удосконалена модель процесу запобігання витокам даних становить перший науковий результат дисертації.

Основні результати розділу опубліковані у працях [104; 105; 106; 107; 108; 112].

РОЗДІЛ 3
МЕТОДИ ВИЯВЛЕННЯ І ЗАПОБІГАННЯ ВИТОКУ ДАНИХ В КОРПОРАТИВНИХ МЕРЕЖАХ НА ОСНОВІ ЕВОЛЮЦІЙНИХ АЛГОРИТМІВ

Розроблена архітектура системи виявлення витоків даних в корпоративних мережах та математичні моделі її компонент та середовища корпоративної мережі є основою створення нових систем виявлення витоків даних в корпоративних мережах, які зможуть адаптуватися до постійних змін і еволюції методів та засобів ексфільтрації даних.

Організація функціонування системи виявлення витоків даних в корпоративних комп'ютерних мережах відповідно до розроблених архітектури та математичних моделей потребує розроблення та реалізації в них методів, які б забезпечували вчасне виявлення та реагування на витоки цими системами, а також давали б змогу уникати хибнопозитивних спрацювань, водночас зберігаючи функціонування корпоративної мережі на належному рівні та не використовуючи надмірну кількість ресурсів.

3.1 Метод класифікації документів на основі генетичного алгоритму

Ефективна система запобігання витокам даних потребує надійного механізму ідентифікації конфіденційної інформації серед масиву корпоративних документів. Класифікація документів за рівнем чутливості є задачею, від якості розв'язання якої безпосередньо залежить здатність ЗВД-системи виконувати свою основну функцію – запобігати несанкціонованій передачі конфіденційних даних за межі організаційного периметру. У попередньому розділі було формалізовано модель даних та ознаковий простір для класифікації; цей підрозділ присвячено розробці методу автоматичної побудови правил класифікації із застосуванням генетичного алгоритму.

У формальних термінах задача класифікації документів формулюється як задача бінарної класифікації. Нехай D позначає множину всіх документів корпоративної інформаційної системи, а $Y = \{0,1\}$ – множину класів, де $y = 1$ відповідає конфіденційним документам, $y = 0$ – неконфіденційним. Необхідно

побудувати класифікатор $f: D \to \mathrm{Y}$, який для довільного документа $d \in D$ визначає його належність до одного з класів. Практична реалізація класифікатора передбачає попередню трансформацію документів у числове представлення: кожен документ d відображається на вектор ознак $\mathrm{x} = (x_1, x_2, \ldots, x_m) \in \mathrm{R}^m$, де m – розмірність ознакового простору, визначеного у формулі (2.13). Таким чином, класифікатор фактично визначається як функція $f: \mathrm{R}^m \to Y$, що приймає рішення на основі ознакового представлення документа.

Специфіка задачі класифікації документів у контексті ЗВД-систем доповнює суттєві обмеження на прийнятні розв'язки. Насамперед система працює в умовах значного дисбалансу класів. Конфіденційні документи зазвичай становлять невелику частку загального документообігу організації. Частка конфіденційних документів у типовій корпоративній мережі рідко перевищує 5-15 % від загального обсягу. Цей дисбаланс суттєво ускладнює навчання класифікатора та вимагає спеціальних підходів до оцінювання якості. Наступною вимогою є інтерпретованість рішень класифікатора. У реальних умовах експлуатації ЗВД-системи аналітики безпеки мають розуміти причини, з яких документ було класифіковано як конфіденційний. Можливість пояснити рішення необхідна для верифікації коректності спрацювань, налаштування політик та реагування на інциденти. Третя вимога стосується обчислювальної ефективності. ЗВД-система має аналізувати документи в реальному або близькому до реального часі, щоб не створювати затримок у робочих процесах. Нарешті, класифікатор має бути адаптивним до змін у характері конфіденційної інформації: з'являються нові типи чутливих даних, змінюються регуляторні вимоги, оновлюється структура документів.

Критерій якості класифікації задаємо через міру F_1, яка балансує повноту та точність. У контексті ЗВД-систем вартість помилок різних типів є асиметричною: пропущений конфіденційний документ потенційно може призвести до витоку даних з усіма наслідками – фінансовими збитками, регуляторними санкціями, репутаційними втратами. Хибне спрацювання створює незручності для користувачів та додаткове навантаження на аналітиків безпеки, проте не несе прямих ризиків безпеці. Водночас надмірна кількість хибних спрацювань призводить до ігнорування сигналів системи та зниження загальної ефективності захисту.

Розв’язання задачі класифікації документів потребує методу, здатного одночасно задовольнити кілька суперечливих вимог: забезпечити високу точність класифікації, генерувати інтерпретовані моделі та підтримувати ефективну адаптацію до нових даних. У цьому підрозділі пропонується метод класифікації документів на основі генетичного алгоритму для еволюційного виведення правил класифікації.

Генетичний алгоритм оперує популяцією потенційних розв’язків, застосовуючи еволюційні оператори селекції, схрещування та мутації для пошуку оптимальної або близької до оптимальної конфігурації. Кожна особина популяції кодує набір IF-THEN правил, а функція пристосованості оцінює якість класифікації на навчальній вибірці. Еволюційний процес поступово вдосконалює популяцію правил, відбираючи найбільш ефективні та комбінуючи їхні елементи через операцію схрещування.

Результатом роботи алгоритму є явний набір правил класифікації у формі, зрозумілій людині. Правило виду “ЯКЩО документ містить ключові слова з фінансової термінології ТА створений фінансовим відділом ТА розмір понад 100 КБ, ТО документ є конфіденційним” може бути безпосередньо проаналізоване та верифіковане експертом з безпеки. Генетичний алгоритм здійснює неявний відбір ознак: у процесі еволюції правила, що використовують нерелевантні ознаки, відсіюються через нижчу пристосованість, що дозволяє автоматично ідентифікувати найбільш інформативні характеристики документів. Еволюційний процес природно підтримує багатокритеріальну оптимізацію: функція пристосованості може містити не лише точність класифікації, а й штраф за складність правил, забезпечуючи компроміс між якістю та простотою моделі.

Ідея еволюційного виведення класифікаційних правил має тривалу історію: їй присвячено класичні системи навчання понять на основі генетичних алгоритмів на кшталт GABIL [123], сімейство навчальних класифікаційних систем, найвідомішим представником якого є XCS [124], та їхні численні нащадки, детально оглянуті в [125]. Запропонований метод продовжує цю лінію, але не зводиться до перенесення відомих схем. Специфіка задачі ЗВД вимагає ознакового простору, що поєднує частотні характеристики тексту із шаблонними детекторами критичних ідентифікаторів, регуляризації складності, яка утримує набір правил у межах, прийнятних для регуляторного аудиту, та операторів, узгоджених зі змішаною структурою умов над

числовими й категоріальними ознаками. Саме ця комбінація, а не сам факт еволюційного пошуку правил, становить внесок методу. Правила формулюються безпосередньо в термінах політик безпеки, а їхня кількість і складність контролюються як самостійний критерій якості.

Жадібні алгоритми індукції правил, CN2 і RIPPER, використано лише як джерело частини початкової популяції. Вони швидко дають розумні стартові набори, але самі по собі нарощують правила по одному й спираються на локальні рішення. Пошук натомість ведеться популяційно і глобально, оцінюється якість цілісного набору правил разом із його сумарною складністю, що й дозволяє утримувати модель компактною. На відміну від генетичного відбору правил із наперед сформованої нечіткої бази знань [131], тут еволюціонують чіткі продукційні правила безпосередньо у просторі ознак політик безпеки, тож експерт може втрутитися напряму: дописати правило й отримати робочу політику без перенавчання та без розмітки.

Ефективність генетичного алгоритму суттєво залежить від способу представлення потенційних розв'язків у формі хромосом. Для задачі класифікації документів пропонується спеціалізоване кодування, що представляє набір IF-THEN правил класифікації у структурованому вигляді, придатному для застосування генетичних операторів. Правило класифікації документів визначимо формою продукційного правила:

$$r\colon \bigwedge_{i=1}^{c_r} C_i \Longrightarrow (y = 1), \tag{3.1}$$

де r – правило класифікації документів у формі продукційного правила;

i – номер умови у правилі;

C_i – і-та умова, що перевіряється для документа;

c_r – кількість умов правила в межах $1 \le c_r \le C_{max}$;

C_{max} – максимально допустима кількість умов в одному правилі;

y – мітка класу документа, значення y = 1 відповідає конфіденційному класу.

Кожне правило голосує лише за конфіденційний клас: якщо всі його умови виконано, документ визнається конфіденційним, тому окремі поля класу та ваги правила не потрібні.

Кожна умова C_i визначає обмеження на значення певної ознаки документа та подається кортежем $C_i = (f_i, op_i, v_i)$, де $f_i \in \{1,2, \dots, m\}$ – індекс ознаки, $op_i \in \{<, >\}$

– оператор порівняння, v_i – порогове значення. Для числових ознак використовуються оператори порівняння, для категоріальних – оператор належності до множини значень. Саме такий клас односторонніх порогових умов реалізовано у програмному прототипі.

Правило класифікації подамо як кортеж:

$$r = \Psi_r = \{C_1, C_2, \dots, C_{c_r}\}, \tag{3.2}$$

де Ψ_r – множина умов правила, які об'єднуються кон'юнктивно. Правило ототожнюється зі своєю множиною умов, оскільки всі правила мають той самий позитивний результуючий клас.

Правило активується для документа d з вектором ознак $\mathbf{x} = (x_1, \dots, x_m)$, якщо виконуються всі його умови:

$$M(r, \boldsymbol{x}) = \bigwedge_{i=1}^{c_r} \varepsilon_r(C_i, \boldsymbol{x}), \tag{3.3}$$

де $\varepsilon_r(C_i, \mathbf{x})$ – функція обчислення істинності умови C_i для вектора $\mathbf{x}$, що визначається як $\varepsilon_r\big((f, op, v), \mathbf{x}\big) = x_f \; op \; v$.

Для застосування генетичного алгоритму необхідно закодувати повний класифікатор (набір правил) у формі хромосоми – лінійної структури, до якої можна застосовувати генетичні оператори схрещування та мутації.

Хромосому подамо як набір правил змінної довжини $g = r_1, r_2, \dots, r_k$, де кількість правил k змінюється в процесі еволюції, а стала K обмежує її зверху. Окремий бітовий вектор активності не потрібен: вилучення чи додавання правила виконується операторами мутації безпосередньо над набором. Задамо її так:

$$g = r_1, r_2, \dots, r_k, 1 \le k \le K, \tag{3.4}$$

$r_1, r_2, \dots, r_k$ – окремі правила класифікації, що входять до хромосоми;

k – кількість правил у хромосомі, змінна в процесі еволюції за рахунок операторів додавання та вилучення правил;

K – стала, що обмежує кількість правил зверху.

Кількість правил хромосоми дорівнює k і змінюється операторами додавання та вилучення правил. Структура хромосоми для класифікації документів візуалізується на рисунку 3.1.

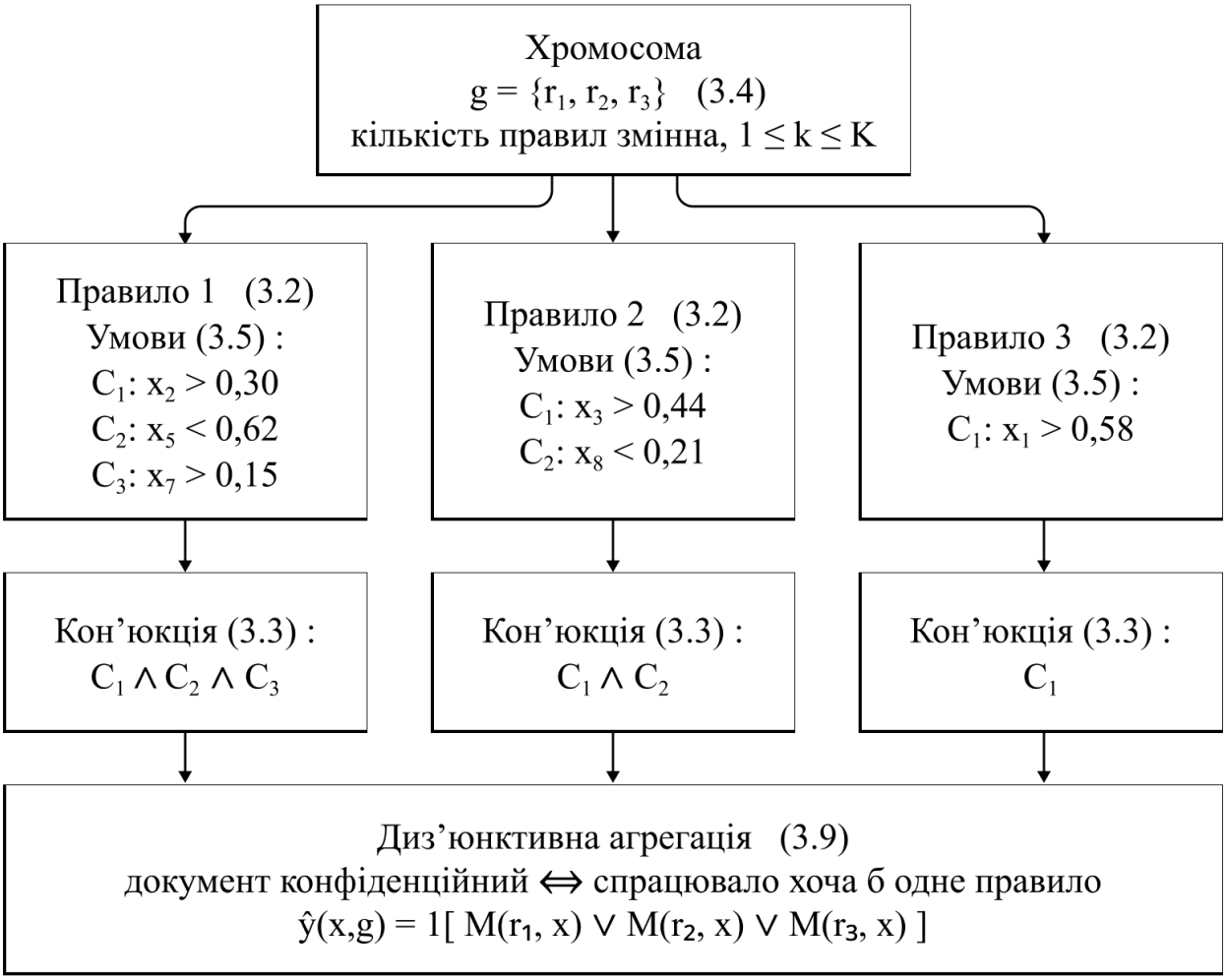


Рисунок 3.1 – Структура хромосоми для кодування правил класифікації документів

Кодування умови для числової ознаки складається з індексу ознаки, знака порівняння та порога: умова інтерпретується як $x_f > \theta$ або $x_f < \theta$. Поріг обирається в межах діапазону значень ознаки в навчальних даних так:

$$C_{numeric} = (f, op, \theta), op \in \{<, >\}, \theta \in [min_f, max_f], \quad (3.5)$$

де min_f та max_f – мінімальне та максимальне значення ознаки f у навчальних даних.

Категоріальні характеристики документа (тип файлу, підрозділ автора тощо) кодуються числовими ознаками ще на етапі побудови ознакового простору, тому окремого категоріального кодування умов не потребують.

Ознаковий простір для класифікації документів формуємо з двох категорій характеристик:

1) контентних ознак, що описують вміст документа;

2) метаданих, що характеризують його властивості.

Контентні ознаки охоплюють частоту появи ключових термінів з предметних словників конфіденційної інформації, наявність регулярних виразів, що відповідають чутливим шаблонам, статистичні характеристики тексту та тематичні показники на основі латентного семантичного аналізу. Метадані охоплюють тип файлу, розмір файлу, автора та підрозділ, дату створення та модифікації, розташування у файловій системі та історію доступу. Вектор ознак документа d визначимо так:

$$\boldsymbol{x}(d) = \left(x_1^{content}, \dots, x_p^{content}, x_1^{meta}, \dots, x_q^{meta}\right), \tag{3.6}$$

де p – кількість контентних ознак;

q – кількість метаданих;

$m = p + q$ – повна розмірність ознакового простору.

Особливістю запропонованого підходу є комбінування текстових ознак із метаданими в єдиному ознаковому просторі. Це дає правилам класифікації змогу враховувати як зміст документа, так і його контекстуальні характеристики. Така комбінація забезпечує вищу точність класифікації порівняно з підходами, що розглядають вміст або метадані окремо.

Лексичні ознаки базуються на аналізі словникового складу документа. Для виявлення конфіденційної інформації використовуються спеціалізовані словники термінів, характерних для різних категорій чутливих даних. Насиченість документа термінами кожного словника вимірюємо словниковим покриттям, нормалізованим на кількість унікальних термінів документа:

$$x_{dict}^{k} = \frac{|v(d) \cap W_k|}{|v(d)|}, \tag{3.7}$$

де W_k – словник категорії k;

$v(d)$ – множина термінів документа d, показник є словниковим покриттям, тобто часткою унікальних термінів документа, що належать словнику, він не є частотою входжень, а нормування на $|v(d)|$ робить його незалежним від обсягу документа, від показника з формули (2.11) він відрізняється знаменником, бо там береться частка знайдених слів словника, а тут частка словникових термінів у документі.

Патернові ознаки виявляють структуровані фрагменти тексту, що відповідають чутливим типам даних. Кожен шаблон генерує числову ознаку відповідно до формули (2.10).

Перетворення вихідного документа на точку простору ознак спирається на конвеєр попередньої обробки $П$, визначений як композиція трансформацій $П = f_{lem} \circ f_{stop} \circ f_{norm} \circ f_{tok} \circ f_{pat}$ (формула (2.14)), і явно фіксує параметри, від яких залежить результат: розмір словника V, порядок n для n-грам та список шаблонів P. Лексичні ознаки обчислюють за словниками категорій W_k згідно з формулою (3.7), а шаблонні за співвідношеннями з формул (2.10) і (2.23).

Алгоритм видобування вектору ознак документа отримує на вхід документ d; параметри конвеєра $П$ розмір словника V, порядок n-грам n, список шаблонів $P = (P_1, \dots, P_r)$; словники категорій $W_1, \dots, W_t$. Вихід: вектор ознак $x(d) \in R^m, m = p + q$. Послідовність кроків:

1. Нормалізувати текст (f_{norm}): звести до нижнього регістру, прибрати службову розмітку, уніфікувати пробіли й пунктуацію; виокремити текстову частину $t(d)$ та блок метаданих $\mu(d)$.

2. Токенізувати $t(d)(f_{tok})$ і побудувати множину термінів $v(d)$, враховуючи n-грами порядку до n; службові слова відсікти (f_{stop}), за потреби звести словоформи (f_{lem}).

3. Для кожної категорії $k = 1, \dots, t$ обчислити лексичну ознаку за (3.7): $x_{dict}^k = \frac{|v(d) \cap W_k|}{|v(d)|}$; за потреби зважити терміни за схемою tf-idf відносно словника обсягу V.

4. Застосувати шаблони P (f_{pat}): для кожного P_j дістати шаблонну ознаку згідно з (2.10), а агреговану шаблонну чутливість за (2.23): $S_{pattern}(d) = \min(1, \sum_p w_p \cdot n_p(d))$.

5. З блоку μ(d) видобути метадані: ознаки автора й підрозділу, розмір, тип, мітки часу; закодувати їх у числові компоненти x_{meta}.

6. Скласти повний вектор: $x(d) = (x_1^{content}, \dots, x_p^{content}, x_1^{meta}, \dots, x_q^{meta})$.

7. Нормалізувати вектор мін-макс покомпонентно задля сумірності ознак різної природи, повернути x(d).

Кроки 2-4 повністю визначаються параметрами V, n і P, тому за однакових параметрів процедура детермінована: той самий документ завжди дає той самий вектор. Це й робить ознаки відтворюваними, а отримані згодом правила – перевірюваними на будь-якому документі.

Метадані забезпечують контекстуальну інформацію про документ. Атрибути авторства охоплюють ідентифікатор автора, його підрозділ, посаду та рівень доступу. Документи, створені керівництвом організації або відділами з підвищеними вимогами до конфіденційності, мають вищу апріорну ймовірність бути конфіденційними. Часові атрибути охоплюють дату створення, дату останньої модифікації та час з моменту створення. Атрибути розташування охоплюють шлях у файловій системі, теку та сервер зберігання. Атрибути формату описують тип файлу, розмір файлу та кодування.

Функція пристосованості визначає критерій оптимізації для генетичного алгоритму та спрямовує еволюційний пошук у напрямку високоякісних класифікаторів. Для задачі класифікації документів функція пристосованості враховує як точність класифікації, так і складність результуючого набору правил.

Основним компонентом функції пристосованості є міра F_1, що забезпечує баланс між точністю та повнотою класифікації. Для хромосоми g, що кодує класифікатор, міра F_1 обчислюється на навчальній або валідаційній вибірці:

$$F_1(g) = \frac{2 \cdot Pr(g) \cdot Re(g)}{Pr(g) + Re(g)}, \tag{3.8}$$

де $F_1(g)$ – міра F_1 класифікатора, закодованого хромосомою g, обчислена як гармонічне середнє точності й повноти;

g – хромосома, що кодує класифікатор;

$Pr(g)$ – точність класифікації на навчальній чи валідаційній вибірці;

$Re(g)$ – повнота класифікації на тій самій вибірці.

Класифікація документа d з вектором ознак x здійснюється шляхом застосування активних правил хромосоми. Агрегація правил диз'юнктивна: документ визнається конфіденційним, щойно для нього спрацювало хоча б одне таке правило набору:

$$y(x, g) = 1[\bigvee_{i=1}^{k} M(r_i, x)], \tag{3.9}$$

де $y(x, g)$ – клас, приписаний документу з вектором ознак x (одиниця відповідає конфіденційному класу, нуль неконфіденційному);

$M(r_i, x)$ – предикат спрацювання правила r_i за (3.3);

$1[\cdot]$ – індикаторна функція, що дорівнює одиниці за виконання умови в дужках і нулю інакше;

k – кількість правил хромосоми g.

За диз'юнктивної агрегації порядок правил не впливає на рішення. Пояснення позитивного рішення складається з переліку правил, що спрацювали; негативне рішення пояснюється тим, що жодна з умов активації не виконалася.

Надмірно складні класифікатори з великою кількістю правил та умов є небажаними з кількох причин: вони схильні до перенавчання на тренувальних даних, їх важче інтерпретувати та верифікувати, а обчислювальна вартість застосування зростає зі складністю. Для контролю складності до функції пристосованості додається штрафний член:

$$\Gamma(g) = \frac{1}{2} \cdot \frac{k}{K} + \frac{1}{2} \cdot \frac{\sum_{i=1}^{k} |\Psi_i|}{K \cdot C_{max}}, \tag{3.10}$$

де k – кількість правил хромосоми;

K – максимальна кількість правил;

$|\Psi_i|$ – кількість умов у правилі i;

C_{max} – максимальна кількість умов у правилі.

Перший доданок штрафує за кількість правил, другий - за сумарну кількість умов; обидва мають вагу 1/2, тому $\Gamma(g) \in [0,1]$ і внесок штрафу в (3.11) обмежений величиною λ.

Остаточну функцію пристосованості формулюємо як різницю між мірою якості та штрафом за складність:

$$F(g) = F_1(g) - \lambda \cdot \Gamma(g), \tag{3.11}$$

де $F(g)$ – функція пристосованості хромосоми g;

$F_1(g)$ – міра якості класифікації за (3.8);

$\lambda \in [0,1]$ – коефіцієнт регуляризації, що задає компроміс між точністю та простотою моделі;

$\Gamma(g)$ – штраф за складність моделі.

Функція пристосованості потребує обробки граничних випадків. Якщо класифікатор не приймає жодного позитивного рішення, але позитивні приклади існують, еквівалентна форма $F_1 = 2TP/(2TP + FP + FN)$ дає нуль, і ділення на нуль

не виникає; невизначеність можлива лише тоді, коли нульовим є весь знаменник, і в цьому разі F_1 покладається рівною нулю. Крім того, штраф за складність не дає алгоритму виродити класифікатор у порожній набір правил: хромосома щонайменше з одним правилом і однією умовою завжди присутня.

Стандартні генетичні оператори потребують адаптації до специфічної структури хромосом, що кодують правила класифікації. Розроблені оператори враховують семантику представлення: односторонні порогові умови над числовими ознаками, до яких на етапі побудови ознакового простору зведено й категоріальні характеристики.

Для задачі класифікації документів розроблено спеціалізовані оператори, адаптовані до структури кодування правил. Початкова популяція формується комбінованим підходом: частина хромосом генерується випадково, частина – за допомогою евристик на основі аналізу навчальних даних. Випадкова ініціалізація створює хромосому шляхом незалежної генерації кожного правила. Для правила i генерується кількість умов k_i з рівномірного розподілу на інтервалі від 1 до C_{max}, для кожної умови j генерується індекс ознаки f_{ij} з рівномірного розподілу на множині $\{1, \dots, m\}$, а знак порівняння та поріг умови генеруються відповідно рівноймовірно з множини $\{<, >\}$ та рівномірно з діапазону значень обраної ознаки.

Евристична ініціалізація базується на аналізі розподілу ознак у конфіденційних та неконфіденційних документах навчальної вибірки. Для кожної ознаки f обчислюємо статистичні характеристики:

$$\mu_f^{(c)} = E[x_f | y = c], (\sigma_f^{(c)})^2 = Var[x_f | y = c], c \in \{0,1\}, \quad (3.12)$$

де $\mu_f^{(c)}$ – середнє значення ознаки f у документах класу c;

E – оператор математичного сподівання за умови $y = c$;

x_f – значення ознаки f;

y – мітка класу документа;

$c \in \{0,1\}$ – клас документа (нуль для неконфіденційного, одиниця для конфіденційного);

$\sigma_f^{(c)}$ – стандартне відхилення (корінь з відповідної дисперсії) ознаки f у класі c;

Var – дисперсія ознаки за умови $y = c$.

Ознаки з найбільшою розрізнювальною здатністю, тобто з найбільшою різницею $\left|\mu_f^{(1)} - \mu_f^{(0)}\right|$ відносно стандартних відхилень, використовуємо для побудови початкових правил так:

$$C_f = \left(f, op_f, \theta_f\right), \theta_f = \frac{\mu_f^{(1)} + \mu_f^{(0)}}{2}, \tag{3.13}$$

де op_f – знак порівняння, що обирається за розташуванням класових середніх: >, якщо $\mu_f^{(1)} > \mu_f^{(0)}$, і < в іншому разі;

θ_f – середина між середніми значеннями ознаки у двох класах, додатковими кандидатами порога слугують квантилі розподілу ознаки в позитивному класі.

Початкова популяція комбінує три групи хромосом: близько десятої частини походить від жадібного послідовного покриття у контексті алгоритмів індукції правил CN2 та RIPPER (правила нарощуються умовами, що максимізують якість щодо ще не покритих позитивних прикладів) з мутаційними варіаціями; ще близько третини складають евристичні правила за (3.13); решту генеровано випадково. Така структура поєднує сильні стартові розв’язки з різноманітністю.

Селекція визначає механізм відбору батьківських хромосом для створення нового покоління. Турнірна селекція обрана як компроміс між селекційним тиском та збереженням різноманітності. Алгоритм турнірної селекції полягає у випадковому виборі k кандидатів з популяції та поверненні кандидата з найвищою пристосованістю. Параметр k контролює селекційний тиск: більші значення k забезпечують вищий тиск, менші – більшу ймовірність вибору середніх особин, що сприяє збереженню різноманітності. Для даної задачі використовується значення $k = 3$, яке забезпечує помірний селекційний тиск.

Оператор схрещування створює нащадків шляхом комбінування правил двох батьківських хромосом на рівні цілих правил. Нехай $g_1 = r_1^{(1)}, \dots, r_{k_1}^{(1)}$ та $g_2 = r_1^{(2)}, \dots, r_{k_2}^{(2)}$ - батьківські хромосоми з k_1 та k_2 правилами. Для кожної з них незалежно обирається точка розрізу, і нащадок успадковує префікс правил першого батька та суфікс другого:

$$g' = (r_1^{(1)}, \dots, r_{p_1}^{(1)}, r_{p_2+1}^{(2)}, \dots, r_{k_2}^{(2)}), \tag{3.14}$$

де p_1 та p_2 – незалежно обрані точки розрізу, що задовольняють умови $0 \le p_1 \le k_1$, $0 \le p_2 \le k_2$.

Довжина нащадка обмежується величиною K відкиданням надлишкових правил. Другий нащадок утворюється переставленням ролей батьків. Цілісність кожного правила при цьому зберігається.

Схрещування застосовується до кожної пари відібраних турніром батьків; варіативність нащадків забезпечують незалежні точки розрізу, а мутація додатково урізноманітнює результат.

Мутація вносить випадкові зміни до хромосоми, забезпечуючи дослідження нових областей простору пошуку. Для кожного нащадка застосовується один із типів мутації. Мутація порога зсуває порогове значення обраної умови:

$$(f, op, \theta) \to (f, op, \theta + \Delta), \Delta \sim N\left(0;\ 0{,}2 \cdot \sigma_f\right), \tag{3.15}$$

де σ_f – стандартне відхилення ознаки f у навчальних даних, причому після зсуву поріг лишається в діапазоні значень ознаки.

Мутація ознаки замінює індекс ознаки в умові на випадковий, встановлюючи поріг у медіану нової ознаки. Структурні мутації додають або вилучають умову правила чи ціле правило в межах обмежень C_{max} і K. Окремим типом є спрямована спеціалізація. До правила з найбільшою кількістю хибних спрацювань додається умова з найбільшим приростом пристосованості, що безпосередньо зменшує кількість хибних тривог.

Повний алгоритм методу класифікації документів на основі генетичного алгоритму інтегрує описані компоненти у цілісний ітеративний процес.

1. Алгоритм приймає на вхід навчальну вибірку документів з мітками класів $D_{train} = \{(d_i, y_i)\}_{i=1}^{N}$ та параметри конфігурації: розмір популяції N_{pop}, максимальна кількість поколінь N_{gen}, максимальна кількість правил K, коефіцієнт регуляризації λ, ймовірності схрещування та мутації p_c, p_m. Вихідними даними є оптимальна хромосома g^* – набір правил класифікації.

2. Виконуємо попередню обробку: перетворення документів у вектори ознак $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{N}$.

3. Ініціалізуємо популяцію $P_0 = \left\{g_1, g_2, \dots, g_{N_{pop}}\right\}$ комбінованим методом. Для кожної хромосоми обчислюємо пристосованість за формулою (3.11).

4. Виконуємо еволюційний цикл для $t = 1$ до N_{gen}. На кожній ітерації формується нова популяція P_t:

а) копіюємо n_{elite} найкращих хромосом з P_{t-1} до P_t (елітизм);

б) поки $|P_t| < N_{pop}$ обираємо турнірним методом двох батьків, схрещуванням формуємо нащадка, з імовірністю p_m мутуємо його та додаємо до P_t;

в) обчислюємо пристосованість P_t і замінюємо близько десятої частини найгірших особин новими евристичними або випадковими хромосомами (ін'єкція різноманітності);

г) перевіряємо критерій зупинки: досягнення максимальної кількості поколінь, відсутність покращення пристосованості протягом n_{stag} поколінь або досягнення цільового значення F_1.

5. Після завершення еволюційного циклу обираємо найкращу хромосому:

$$g^* = arg \max_{g \in P_{N_{gen}}} F(g), \tag{3.16}$$

де $g*$ – найкраща хромосома, відібрана після завершення еволюційного циклу;

g – хромосома популяції, за якою виконується вибір;

$P(Ngen)$ – популяція на останньому, N_{gen}-му поколінні;

N_{gen} – кількість поколінь еволюційного циклу;

$F(g)$ – пристосованість хромосоми g.

6. Кроки 3-5 повторюються кілька разів з незалежними ініціалізаціями (мультистарт), і серед найкращих хромосом усіх запусків обирається підсумкова модель.

Обчислювальна складність однієї ітерації алгоритму визначається кількістю операцій оцінювання пристосованості. Для популяції розміром N_{pop} з хромосомами, що містять до K правил із до C_{max} умов кожне, на навчальній вибірці з N документів, загальну складність для N_{gen} поколінь визначимо як:

$$T_{total} = O\big(N_{gen} \cdot N_{pop} \cdot K \cdot C_{max} \cdot N\big), \tag{3.17}$$

де T_{total} – загальна обчислювальна складність еволюційного циклу;

N_{gen} – кількість поколінь;

N_{pop} – розмір популяції;

K – максимальна кількість правил у хромосомі;

C_{max} – максимальна кількість умов в одному правилі;

N – кількість документів навчальної вибірки.

На практиці обчислення можуть бути прискорені шляхом паралелізації оцінювання хромосом та використання інкрементного оновлення метрик при мутаціях.

Параметри алгоритму адаптують залежно від характеристик конкретного набору даних. Для великих наборів даних доцільно збільшувати розмір популяції та кількість поколінь; для наборів з великою кількістю ознак – збільшувати максимальну кількість умов у правилі.

Наведемо типові приклади правил, що виводяться алгоритмом для класифікації документів у корпоративному середовищі. Правило для фінансової документації може мати вигляд: «ЯКЩО словникове покриття фінансової термінології > 0,15 ТА кількість шаблонів платіжних реквізитів > 0, ТО документ конфіденційний». Правило для персональних даних: «ЯКЩО кількість PII-шаблонів > 0,05 ТА частка цифр у тексті > 0,06, ТО документ конфіденційний»; воно виявляє документи з ідентифікаційними даними на основі регулярних виразів для номерів документів, адрес і телефонів. Правило для технічної документації: «ЯКЩО словникове покриття технічних термінів > 0,20 ТА розмір файлу > 0,42 (у нормованих одиницях), ТО документ конфіденційний».

При класифікації конкретного документа система може генерувати пояснення, перелічуючи правила, що спрацювали. Така деталізація дає аналітику безпеки змогу верифікувати коректність класифікації та, за потреби, ініціювати перегляд правил.

Правила класифікації можуть бути візуалізовані у формі таблиці покриття. Для набору з k активних правил таблиця покриття показує, які документи навчальної вибірки покриваються кожним правилом. Покриття визначається як частка документів, для яких правило активується, точність – як частка коректних класифікацій серед покритих документів, а внесок у F_1 – як приріст загальної міри F_1 завдяки даному правилу.

Інтерпретована форма правил дає експертам з безпеки змогу вручну коригувати класифікатор. Типові операції охоплюють додавання нового правила на основі

експертних знань про нові типи конфіденційних даних, модифікацію порогових значень для підвищення точності або повноти, видалення правил, що генерують хибні спрацювання, та об'єднання схожих правил для спрощення моделі. Можливість ручного втручання є необхідною для практичного застосування, оскільки автоматичні методи не завжди здатні врахувати специфічні вимоги організації або регуляторні обмеження.

Практична реалізація методу класифікації документів вимагає інтеграції з іншими компонентами ЗВД-системи. Архітектура модуля класифікації охоплює компоненти для видобування ознак, навчання класифікатора та застосування правил у режимі реального часу. Структура модуля класифікації представлена на рисунку 3.2.

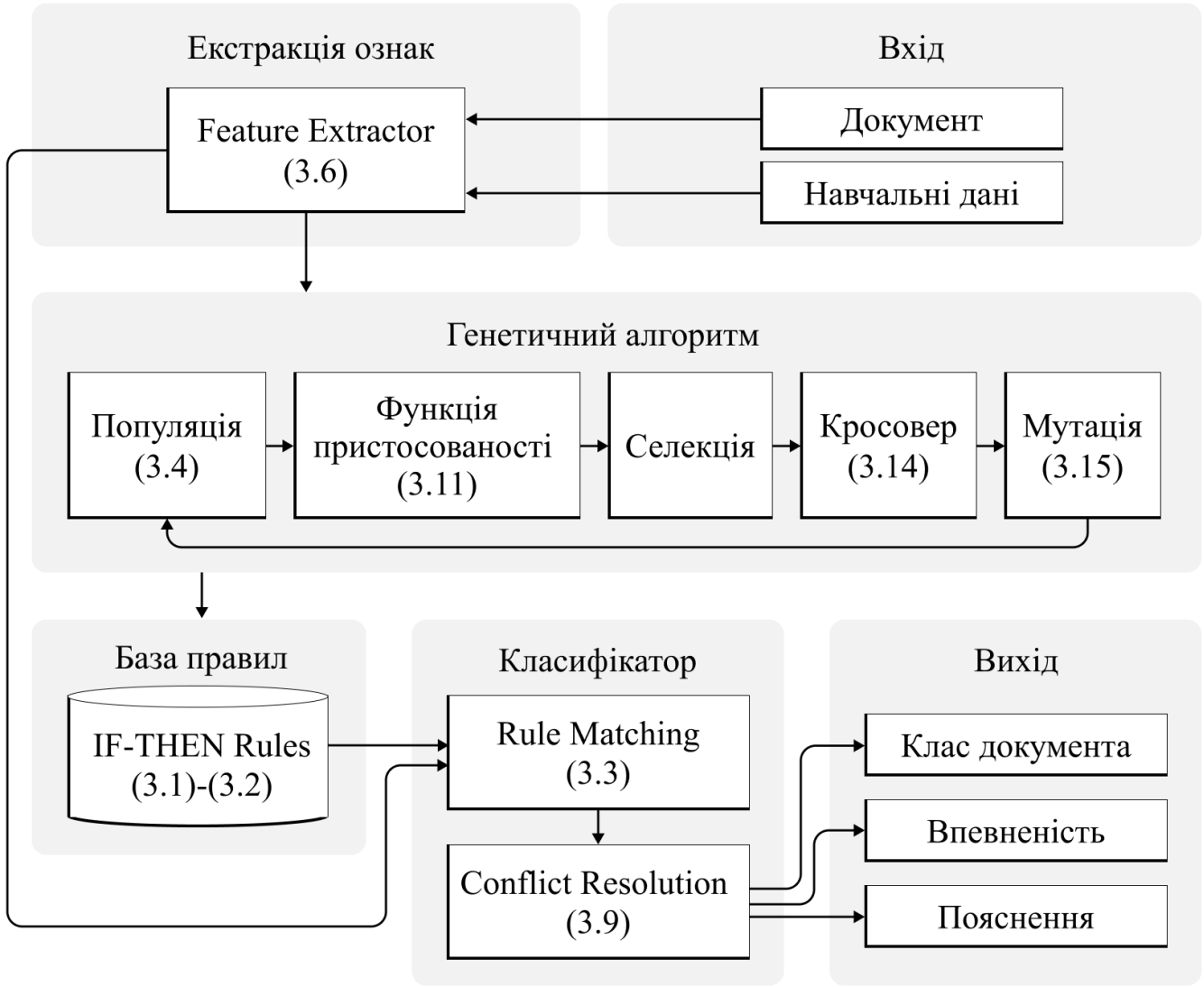


Рисунок 3.2 – Архітектура модуля класифікації документів

Модуль видобування ознак відповідає за перетворення вхідного документа у числовий вектор. Аналізатор вмісту здійснює токенізацію тексту, обчислення частотних характеристик та семантичний аналіз. Зіставник шаблонів шукає регулярні вирази, що відповідають чутливим даним. Засіб видобування метаданих отримує атрибути файлу з файлової системи або системи документообігу.

Модуль генетичного алгоритму реалізує описаний вище еволюційний процес. Цей модуль активується у режимі навчання або перенавчання класифікатора. Результатом є набір правил, що зберігається та використовується модулем класифікації.

Модуль класифікації застосовує правила до нових документів у режимі реального часу. Механізм правил перевіряє умови кожного правила для вхідного вектору ознак. Компонент диз'юнктивної агрегації за формулою (3.9) визнає документ конфіденційним, щойно спрацювало хоча б одне з активованих правил. Генератор пояснень формує текстовий опис причин рішення.

Система функціонує у двох режимах: навчання та класифікації. У режимі навчання система отримує навчальну вибірку документів з мітками, виконує генетичний алгоритм для виведення правил та зберігає результуючий класифікатор. Цей режим запускається періодично або за запитом адміністратора при зміні характеру конфіденційних даних. У режимі класифікації система працює онлайн, обробляючи потік документів у реальному часі. Для кожного документа виконується видобування ознак, застосування правил та формування рішення.

Модуль класифікації інтегрується з іншими компонентами ЗВД-системи через програмний інтерфейс. Типовий потік обробки охоплює перехоплення операції з документом агентом на кінцевій точці, передачу документа модулю класифікації, отримання класу, впевненості та пояснення, визначення дії компонентом прийняття рішень ЗВД та логування результату для аудиту та аналітики. Модуль класифікації може розгортатися централізовано на сервері ЗВД або розподілено на кінцевих точках. Централізований варіант спрощує оновлення правил, розподілений – зменшує навантаження на мережу та забезпечує роботу в офлайн-режимі.

Запропонований метод класифікації документів на основі генетичного алгоритму має певні обмеження, які необхідно враховувати при практичному застосуванні. Залежність від якості ознак є обмеженням будь-якого методу

машинного навчання. Генетичний алгоритм оптимізує правила на заданому ознаковому просторі, проте не здатний компенсувати відсутність релевантних ознак. Якщо конфіденційність документа визначається факторами, не представленими в ознаковому просторі, класифікація буде неточною. Масштабованість на дуже великі набори ознак є іншим обмеженням: при кількості ознак понад кілька тисяч простір пошуку стає занадто великим для ефективного дослідження генетичним алгоритмом. Для таких випадків рекомендується попередній відбір ознак методами фільтрації. Часова складність навчання зростає лінійно з розміром навчальної вибірки та кількістю поколінь, що може бути обмеженням для дуже великих корпоративних середовищ. Проте паралелізація оцінювання пристосованості суттєво скорочує час навчання.

Головною перевагою запропонованого методу є висока інтерпретованість результуючого класифікатора. На відміну від моделей типу “чорної скриньки”, генетичний алгоритм генерує явний набір правил, зрозумілих для людини. Результатом роботи алгоритму є набір правил виду «ЯКЩО умова 1 ТА умова 2 ТА … ТА умова k, ТО документ конфіденційний», де кожна умова формулюється мовою предметної області та має вигляд одностороннього порогового обмеження «ознака більша (менша) за поріг».

Шляхи вдосконалення методу включають застосування багатокритеріальної оптимізації з алгоритмами типу NSGA-II для одночасної оптимізації точності, повноти та складності з побудовою фронту Парето. Інтеграція з методами глибокого навчання для видобування ознак дозволила б використовувати векторні подання (embeddings) документів замість ручних ознак. Інкрементне навчання забезпечило б адаптацію до нових типів конфіденційних даних без повного перенавчання. Ці напрями розвитку є предметом подальших досліджень.

Підсумовуючи, запропонований метод класифікації документів на основі генетичного алгоритму забезпечує унікальне поєднання точності класифікації та інтерпретованості результатів. Еволюційний пошук правил класифікації дає змогу автоматично виявити найінформативніші ознаки та побудувати компактний набір правил, зрозумілих для експертів з безпеки. Функція пристосованості на основі F_1-міри зі штрафом за складність забезпечує баланс між якістю класифікації та простотою моделі. Хромосомне кодування продукційних правил IF-THEN

уможливлює застосування стандартних генетичних операторів з урахуванням структурних інваріантів. Можливість ручного коригування правил експертами є вагомою для практичного застосування у корпоративному середовищі.

Запропонована схема еволюційного синтезу класифікатора належить до елітних генетичних алгоритмів, а для таких алгоритмів збіжність до глобального оптимуму має теоретичне обґрунтування відповідно до [113].

На практиці алгоритм зупиняється не за досягненням теоретичної границі, а за прагматичним критерієм, тобто коли найкраща пристосованість не поліпшується протягом G_{stag} послідовних поколінь або вичерпано відведений обчислювальний бюджет. Збіжність контролюють емпірично за кривою найкращої пристосованості по поколіннях: щойно ця крива виходить на плато й стагнація триває G_{stag} поколінь, пошук припиняють.

Теоретична гарантія тут має асимптотичний характер. Збіжність за ймовірністю до глобального оптимуму справджується в границі нескінченної кількості поколінь; за скінченний час ГА може застрягнути в локальному оптимумі, настає передчасна збіжність, коли популяція втрачає різноманіття й концентрується навколо одного субоптимального правила. Щоб послабити цей ризик, у самому класифікаторі підтримується різноманіття популяції: на кожному поколінні приблизно десята частина найгірших особин замінюється свіжими (випадковими та евристичними), а на рівні всього пошуку застосовано мультистарт. Разом ці засоби утримують пошук від завчасного звуження й наближають практичну поведінку алгоритму до теоретично очікуваної. Для неперервних генів (порогів умов) ідеться про збіжність до як завгодно малого околу оптимальної конфігурації, а не про точне влучення в окрему точку.

Описану вище комбіновану ініціалізацію на основі статистик розподілу ознак за класами в реалізації методу доповнено трьома прийомами, що гібридизують чисту еволюцію з елементами скерованого локального пошуку. Кожен має просту мотивацію: зробити стартову популяцію змістовнішою, цілеспрямовано виправляти найгірші правила й захиститися від невдалого випадкового старту.

Перший прийом це ініціалізація жадібним послідовним покриттям. Приблизно десяту частину стартової популяції формуємо не випадково, а за жадібною стратегією «розділяй і володарюй». Для цього кожне правило вирощуємо поступово, по одній

умові. На кожному кроці додаємо ту умову, яка найбільше підвищує F1-оцінку правила на ще не покритих позитивних прикладах, тобто водночас зважає на точність і на повноту покриття. Пороги-кандидати для умов беремо як середину між середніми значеннями ознаки в позитивному й негативному класах, а також як квантилі розподілу непокритих позитивів. Щойно чергова умова перестає підвищувати оцінку, правило вважаємо завершеним, вилучаємо покриті ним позитивні приклади з розгляду й повторюємо процедуру на тих, що залишилися. Так уже в нульовому поколінні маємо невеликий набір хромосом, які відповідають реальним згусткам позитивних прикладів у даних.

Другий прийом це оператор спрямованої спеціалізації, що працює як локальне поліпшення поодиноких особин. До правила, яке дає найбільше хибних спрацювань, дописуємо додаткову умову, яка забезпечує найбільший приріст пристосованості F(g). Оскільки таку умову обирають за приростом пристосованості, вона передусім відтинає негативні приклади, на яких правило хибно спрацьовувало; водночас надмірне звуження покриття може коштувати втрати частини істинних спрацювань. Оператор застосовуємо приблизно в 15 % випадків мутації, тому що він обчислювально дорожчий за звичайну мутацію.

Третій прийом це мультистарт. Еволюцію запускаємо кілька разів незалежно, з різних випадкових початкових популяцій, а підсумковим розв'язком обирають хромосому з найкращою пристосованістю серед усіх запусків. Це класичний захист від передчасної збіжності, якщо один запуск застрягне в невдалому локальному оптимумі, інші з високою ймовірністю уникнуть цієї проблеми.

Усі три прийоми узгоджені з елітною і комбінованою ініціалізацією та підсилюють їх. Жадібне покриття покращує стартову якість, тож монотонно неспадна послідовність F_t^* стартує з вищого рівня. Спрямована спеціалізація додає до випадкового пошуку складник скерованості. Мультистарт компенсує головне обмеження елітних ГА. Він не усуває можливості передчасної збіжності в окремому запуску, але робить підсумковий результат стійкішим до неї.

Параметри методу різняться за способом визначення. Регуляризаційні коефіцієнти та ваги штрафів, що регулюють компроміс між конкурентними цілями, визначаємо сітковим пошуком на окремій валідаційній підвибірці, не задіяній у навчанні. Структурні параметри еволюційної схеми: розмір популяції, розмір турніру,

ймовірності схрещування й мутації, кількість елітних особин. Їх фіксуємо за усталеними значеннями. Коефіцієнт балансу ризику, на відміну від них, визначаємо одним із двох способів: або задаємо фіксованим значенням згідно з політикою організації, або вмикаємо у хромосому й налаштовуємо еволюційно разом з рештою генів. Зведення всіх параметрів з їхніми ролями, діапазонами та робочими значеннями подано в таблиці 3.1.

Таблиця 3.1 - Параметри методу

Параметр (формула)	Роль	Діапазон	Спосіб вибору	Робоче значення
λ (3.11)	Компроміс точність-простота через штраф $\Gamma(g)$	[0, 1]; кандидати {0; 0,02; 0,05; 0,1; 0,2}	Сітковий пошук на валідаційній підвибірці	0,05 (для однорідних задач 0,02)
λ (3.54)	Розподіл внеску контентної та поведінкової складових у R_{comb}	[0, 1]	Фіксоване значення політики або адаптивне за контекстом	Типово 0,5 або адаптивне
α (3.50)	Пригнічення частки хибних спрацювань ν_{FP}	$\alpha \geq 0$	Базова конфігурація, прийнята аналітично	0,15
β (3.50)	Згладжування коефіцієнта варіації потоку тривог ν_{CV}	$\beta \geq 0$	Базова конфігурація, прийнята аналітично	У базовій конфігурації 0
γ (3.50)	Обмеження нормованої складності $\Gamma(\chi_b)$	$\gamma \geq 0$	Базова конфігурація, прийнята аналітично	У базовій конфігурації 0
w_T (2.28)	Баланс статичної та динамічної складових довіри	[0, 1]	Обґрунтовано в розділі 2	Типово 0,6…0,8

Таким чином, розроблено метод класифікації документів за рівнем конфіденційності, у якому генетичний алгоритм синтезує компактний набір

порогових продукційних правил над контентними ознаками та метаданими. Функція пристосованості поєднує F1-міру зі штрафом за складність, що утримує модель у придатних для аудиту межах, а комбінована ініціалізація, спеціалізовані оператори та мультистарт забезпечують практичну збіжність пошуку. Кожне рішення класифікатора пояснюється переліком правил, що спрацювали, і може бути скориговане експертом без повного перенавчання.

3.2 Метод еволюційної адаптації політик за умов дрейфу концепції

Дрейф концепції означає зміну спільного розподілу вхідних ознак та міток у часі. Нехай у момент t_0 модель навчена на розподілі $P(X, Y)$, тоді в пізніший момент t_1 може статися так, що новий розподіл відрізнятиметься від початкового, що призводить до деградації якості прогнозів. Цей дрейф проявляється у кількох формах: як зміна апріорного розподілу класів $P(Y)$, зміна умовного розподілу ознак $P(X|Y)$, або найбільш проблематична зміна апостеріорного розподілу $P(Y|X)$, коли одні й ті самі ознаки починають асоціюватися з іншими класами.

У контексті ЗВД-систем дрейф концепції має численні джерела. Організаційні зміни, такі як реструктуризація підрозділів, злиття компаній або впровадження нових бізнес-процесів, суттєво змінюють шаблони легітимного обміну даними. Технологічна еволюція приносить нові канали комунікації, формати даних та інструменти співпраці, які не були враховані при початковому налаштуванні системи. Поведінкові зміни користувачів, адаптація до віддаленої роботи або перехід на нові робочі практики, створюють шаблони, які можуть хибно класифікуватися як аномальні. Нарешті, самі зловмисники адаптують свої тактики, щоб обходити наявні механізми виявлення, породжуючи так званий змагальний дрейф.

Наслідки дрейфу концепції для ЗВД-систем є двоякими та однаково небезпечними. З одного боку, дрейф призводить до зростання хибнопозитивних спрацювань, коли легітимна активність класифікується як підозріла. Це не лише створює операційні незручності та перевантажує аналітиків безпеки, але й породжує явище втоми від тривог, внаслідок якого справжні інциденти можуть бути проігноровані. З іншого боку, дрейф може маскувати нові форми зловмисної активності, що призводить до хибнонегативних результатів – пропущених витоків

даних, кожен з яких потенційно несе значні фінансові та репутаційні збитки для організації.

Аналіз характеру дрейфу концепції дозволяє виокремити кілька типових шаблонів його прояву. Раптовий дрейф характеризується різкою зміною розподілу, наприклад, при впровадженні нової корпоративної системи або злитті з іншою компанією. Поступовий дрейф відбувається повільно, коли нові шаблони поступово витісняють старі протягом тривалого періоду. Інкрементальний дрейф передбачає послідовні малі зміни, що накопичуються з часом. Рекурентний дрейф демонструє циклічні шаблони, наприклад, сезонні зміни в бізнес-активності. Кожен тип дрейфу потребує власної стратегії адаптації, що ускладнює розробку універсального рішення.

У цьому підрозділі пропонується метод автоматичної адаптації політик ЗВД-системи на основі еволюційного алгоритму. Метод враховує специфіку предметної області, зокрема обмежену доступність розмічених даних та вимогу безперервності захисту. ЗВД-система не може припиняти роботу на час адаптації, оскільки організація щомиті залишається вразливою до витоків.

Детектори ADWIN, DDM, EDDM і KSWIN стежать за потоком помилок класифікатора, а отже, потребують оперативних міток істинності й фіксують дрейф уже постфактум, коли якість просіла. Запропонований детектор як багатоознаковий статистичний тест, що виконується безпосередньо на розподілі вхідних ознак, тому зміну можна помітити ще до того, як вона проявиться в помилках, а вимога підтвердження гасить поодинокі флуктуації, які інакше давали б хибні тривоги. Адаптивні ансамблі та інкрементні моделі [57; 58; 59] пристосовуються безперервно й стійкі до дрейфу, проте разом із цим втрачають інтерпретовану структуру політики. Еволюційне перенавчання оновлює саме набір правил, а нову політику вводять у дію лише після валідації.

Для формалізації задачі адаптації визначимо структуру вхідних даних та параметри, що підлягають оптимізації. Параметризована функція ризику забезпечує можливість налаштовувати реакцію системи без перепроектування її архітектури.

Розглянемо потік подій передачі даних, де кожна подія характеризується вектором ознак x_t та метаданими m_t, що включають інформацію про актора, канал та контекст. Для кожної події ЗВД-система має обчислити показник ризику s_t та

прийняти рішення про дію a_t. Показник ризику обчислимо параметризованою функцією оцінки ризику, а рішення приймається відповідно до формули (2.36).

Задача адаптації ЗВД-політик за своєю природою є багатокритеріальною. Ідеальна політика має одночасно мінімізувати ризик пропущених витоків, скорочувати навантаження на аналітиків від хибних спрацювань, забезпечувати низьку латентність обробки, підтримувати простоту для аудиту та інтерпретації, а також демонструвати стабільну поведінку в часі. Ці критерії часто суперечать один одному: агресивніша політика знижує ризик пропущених витоків, але збільшує хибні спрацювання; складніша політика може бути точнішою, але важчою для супроводу.

Перша цільова функція J_1 оцінює ризик пропущених витоків як очікувану суму критичностей подій, що були помилково дозволені, де y_t – індикатор фактичного витоку, φ_t – коефіцієнт критичності, що враховує чутливість даних. Друга цільова функція J_2 кількісно оцінює навантаження від хибних спрацювань як кількість легітимних подій, що були помилково заблоковані або відправлені в карантин.

Формально ці дві функції задамо так:

$$J_1(\theta) = E\left[\sum_{t=1}^{T} 1\left[y_t = 1 \wedge a_t = allow\right] \cdot \varphi_t\right], \qquad (3.18)$$

$$J_2(\theta) = E\left[\sum_{t=1}^{T} 1\left[y_t = 0 \wedge a_t \neq allow\right]\right],$$

де $J_1(\theta)$ – перша цільова функція, тобто очікуваний ризик пропущених витоків;

$J_2(\theta)$ – друга цільова функція, що вимірює навантаження від хибних спрацювань;

θ – вектор параметрів політики, які підлягають оптимізації;

E – оператор математичного сподівання;

T – кількість подій у потоці, верхня межа підсумовування;

t – номер події;

y_t – фактична мітка події, одиниця для витоку і нуль для легітимної;

a_t – дія системи щодо події t;

$allow$ – дія «дозволити»;

φ_t – коефіцієнт критичності події, що враховує чутливість даних;

$\wedge$ – логічна кон’юнкція умов.

Третя цільова функція J_3 враховує вимоги до продуктивності через 95-й перцентиль латентності обробки подій, що забезпечує контроль над хвостами розподілу для дотримання угод про рівень обслуговування. Четверта цільова функція J_4 оцінює складність політики як зважену суму кількості активних правил ψ та умов у них, де $|S_l|$ – кількість предикатів у правилі l, а α_{rules} та α_{conds} – вагові коефіцієнти. П'ята цільова функція J_5 кількісно оцінює стабільність системи тривог через дисперсію кількості тривог $n_{alert}^{(w)}$ у послідовних часових вікнах. Відповідні вирази задамо так:

$$J_3 = p_{95}(\{lat_t\}),$$
$$J_4 = \alpha_{rules} \cdot \psi + \alpha_{conds} \cdot \sum_{l=1}^{\psi} |S_l|, \quad (3.19)$$
$$J_5 = Var\left[\{(n_{alert}^{(w)}\}_{w=1}^{W}\right],$$

де J_3 – третя цільова функція, тобто показник продуктивності системи;

p_{95} – 95-й перцентиль;

lat_t – латентність обробки події t;

t – номер події у потоці;

J_4 – четверта цільова функція, що оцінює складність політики;

α_{rules} – ваговий коефіцієнт за кількість правил;

ψ – кількість активних правил;

α_{conds} – ваговий коефіцієнт за кількість умов;

S_l – множина предикатів правила l;

$|S_l|$ – кількість предикатів правила l;

J_5 – п'ята цільова функція, що характеризує стабільність системи тривог;

$n_{alert}^{(w)}$ – кількість тривог у вікні w;

w – номер часового вікна;

W – кількість вікон.

Узагальнену багатокритеріальну цільову функцію формуємо як зважену суму нормалізованих критеріїв, де ваги $w_i > 0$ визначаються відповідно до пріоритетів організації та задовольняють умову нормування. Нормалізація кожного критерію відносно спостережених мінімальних та максимальних значень забезпечує порівнянність різномасштабних величин: Вектор W задамо зовнішньою політикою

організації і під час оптимізації політики він виступає сталим. Одночасна оптимізація W тим самим критерієм виродила б задачу, переносячи вагу на найлегший складник замість поліпшення політики. Задамо цю функцію так:

$$J(\theta) = \sum_{i=1}^{5} w_i \cdot \frac{J_i(\theta) - J_i^{min}}{J_i^{max} - J_i^{min}} \to min_{\theta \in \Theta}, \quad \sum_{i=1}^{5} w_i = 1, \tag{3.20}$$

де $J(\theta)$ – узагальнена багатокритеріальна цільова функція, яку мінімізують за θ;

θ – вектор параметрів політики;

Θ – простір допустимих значень параметрів;

w_i – вага і-го критерію, додатна, причому сума ваг дорівнює одиниці;

$J_i(\theta)$ – і-та цільова функція;

J_i^{min} – спостережене мінімальне значення і-го критерію, за яким його нормують;

J_i^{max} – спостережене максимальне значення і-го критерію, за яким його нормують.

Ефективність еволюційного алгоритму суттєво залежить від вибору представлення розв'язків у формі хромосом. Хромосому χ складемо з двох логічно відокремлених частин: фіксованої частини, що кодує глобальні параметри системи, та змінної частини, що представляє набір правил політики. Фіксована частина містить шість глобальних параметрів: порогові значення τ_q та τ_b, параметр затухання ρ_w, швидкість навчання η, базову інтенсивність мутації μ_0 та додаткову інтенсивність мутації при дрейфі μ_1:

$$\chi_{fixed} = [\tau_q, \tau_b, \rho_w, \eta, \mu_0, \mu_1], \tag{3.21}$$

де χ_{fixed} – фіксована частина хромосоми з глобальними параметрами системи;

τ_q – поріг карантину;

τ_b – поріг блокування;

ρ_w – коефіцієнт затухання ваг;

η – швидкість навчання;

μ_0 – базова інтенсивність мутації;

μ_1 – додаткова інтенсивність мутації при виявленні дрейфу.

Параметр ρ_w контролює швидкість згасання ваг у зваженій суміші активних політик: менші значення означають швидше згасання впливу старих політик, що доречно при раптових змінах середовища, тоді як більші значення забезпечують

плавніші переходи, доречні при поступовому дрейфі. Параметри μ_0 та μ_1 спільно визначають адаптивну схему мутації, де μ_0 задає рівень мутацій у стабільному режимі експлуатації, а μ_1 – додаткову інтенсивність при виявленні дрейфу. Змінна частина хромосоми χ_{var} кодує набір з L правил політики, кожне з яких має структуру, що включає умову спрацювання $l_{condition}$, дію l_{action} з множини $\{allow, quarantine, block, alert\}$, вагу ω_l та прапорець активності α_l. Умова правила кодується як кон'юнкція елементарних предикатів над підмножиною ознак S_l, де кожен предикат визначає обмеження на значення відповідної ознаки: для неперервних ознак це належність до інтервалу, для категоріальних – належність до множини допустимих значень. Задамо цю величину так:

$$\chi_{var} = [\psi, u_1, \dots, u_L], \quad u_l = (l_{condition}, l_{action}, \omega_l, \alpha_l), \\ l_{condition} = \wedge_{j \in S_l} P_j(x), \tag{3.22}$$

де χ_{var} – змінна частина хромосоми, що кодує набір правил політики;

ψ – кількість активних правил;

u_l – l-те правило;

L – загальна кількість правил;

$l_{condition}$ – умова спрацювання правила;

l_{action} – дія правила з множини {allow, quarantine, block, alert};

ω_l – вага правила;

α_l – прапорець активності правила;

$\wedge$ – логічна кон'юнкція, що береться за всіма $j \in S_l$;

S_l – підмножина ознак, задіяних у правилі l;

$P_j(x)$ – елементарний предикат над ознакою j для вектора x;

x – вектор ознак, для якого перевіряють предикати правила.

Повна хромосома об'єднує обидві частини у єдину структуру, де перші шість позицій займають глобальні параметри, за ними слідує кількість активних правил та самі правила. Для забезпечення коректності хромосоми визначено низку структурних інваріантів, що перевіряються та відновлюються після кожної генетичної операції: впорядкованість порогів, обмеження кількості активних правил та непорожність умов кожного активного правила. Отже, структура повної хромосоми задамо так:

$$\chi = [\tau_q, \tau_b, \rho_w, \eta, \mu_0, \mu_1, \psi, u_1, \dots, u_L], \tag{3.23}$$

де χ – повна хромосома, що об'єднує фіксовану та змінну частини;

τ_q,τ_b, ρ_w, η, μ_0, μ_1 – фіксована частина хромосоми, означена в (3.21);

ψ – кількість активних правил політики;

u_l – правило політики з номером l;

L – кількість правил у політиці.

Відстань між хромосомами, що використовується для підтримки різноманітності популяції, визначимо комбінацією евклідової відстані для неперервних параметрів та відстані Жаккара для множин правил R_1 та R_2:

$$d(\chi_1, \chi_2) = \lambda_{cont} \cdot d_{eucl}(\chi_1^{cont}, \chi_2^{cont}) + \lambda_{rule} \cdot \left(1 - \frac{|R_1 \cap R_2|}{|R_1 \cup R_2|}\right), \quad (3.24)$$

де $d(\chi_1, \chi_2)$ – відстань між хромосомами χ_1 та χ_2;

λ_{cont} – ваговий коефіцієнт неперервної частини;

d_{eucl} – евклідова відстань, неперервні (фіксовані) частини відповідних хромосом задано як χ_1^{cont} та χ_2^{cont};

λ_{rule} – ваговий коефіцієнт дискретної частини;

R_1 та R_2 – множини правил хромосом.

Через $R_1 \cap R_2$ подано спільні правила обох хромосом, $R_1 \cup R_2$ означає їх об'єднання, а вираз у дужках дає відстань Жаккара (одиниця мінус частка спільних правил). Якщо обидві хромосоми не містять правил, жаккарову складову покладаємо нульовою.

Генетичні оператори, що застосовуються до хромосом, адаптовані до гібридної структури представлення. Оператор схрещування реалізуємо як комбінацію арифметичного схрещування для неперервних параметрів фіксованої частини та одноточкового схрещування для змінної частини з правилами. Арифметичне схрещування породжує нащадка як зважену комбінацію батьківських значень, де вага обирається випадково з рівномірного розподілу на інтервалі від нуля до одиниці. Для змінної частини точка розриву обирається між правилами, забезпечуючи цілісність кожного правила у нащадка.

Оператор мутації також диференційований для різних типів генів. Для неперервних параметрів застосовуємо гаусівську мутацію з адаптивним стандартним відхиленням, що залежить від поточного режиму роботи системи. Для дискретних компонентів правил використовуємо рівномірну мутацію: зміна оператора

порівняння, розширення або звуження множини допустимих значень категоріальної ознаки, додавання або видалення предиката з умови правила. Особливу увагу приділено мутаціям, що змінюють структуру набору правил: додавання нового правила, видалення існуючого або зміна статусу активності.

Механізм відновлення інваріантів застосовуємо після кожної генетичної операції. Якщо пороги опинилися у неправильному порядку, вони переставляються та нормалізуються. Якщо кількість активних правил перевищує максимально допустиму, зайві правила деактивуються у порядку зростання ваги.

Своєчасне виявлення дрейфу концепції є передумовою ефективної адаптації. Без автоматичного детектора система або реагуватиме із запізненням (коли деградація якості стане очевидною), або адаптуватиметься надто часто, витрачаючи ресурси на реакцію на випадкові флуктуації.

Адаптивна система спирається на детектор дрейфу концепції, що формує бінарний сигнал про необхідність переходу від режиму експлуатації до режиму дослідження. Оскільки істинні мітки витоків здебільшого недоступні, детектор працює в неконтрольованому режимі, покладаючись на непрямі сигнали зміни розподілу даних та поведінки моделі. Механізм виявлення базується на моніторингу невизначеності моделі U_t, яку для події x_t оцінюємо через ентропію розподілу оцінок ризику. Максимальна ентропія досягається при $s_t = 0{,}5$, що відповідає максимальній невизначеності. В обчисленні ентропії прийнято звичну конвенцію $0 \cdot log0 = 0$. Ентропійну оцінку невизначеності задамо так:

$$U_t = H(s_t) = -s_t logs_t - (1 - s_t)\log(1 - s_t), \tag{3.25}$$

де U_t – невизначеність моделі для події t;

H – функція бінарної ентропії Шеннона;

$s_t \in [0,1]$ – показник ризику події t;

log – двійковий логарифм, за якого максимум невизначеності дорівнює одиниці.

Детектор підтримує два ковзних вікна: референтне W_{ref}, що відповідає стабільному періоду, та поточне W_{cur}. Для кожного вікна обчислюємо середні значення невизначеності, а сигнал z_k^{uncert} дрейфу формуємо на основі статистичного тесту, де σ_{ref} – стандартне відхилення невизначеності у референтному вікні, а δ –

поріг чутливості. Додатково до невизначеності моделі, детектор моніторить статистичні характеристики розподілу ознак через двовибіркову статистику Колмогорова-Смирнова D_j для кожної ознаки j, де F_{ref}^j та F_{cur}^j – емпіричні функції розподілу у відповідних вікнах.

$$z_k^{uncert} = 1\left[\frac{\bar{U}_{cur} - \bar{U}_{ref}}{\frac{\sigma_{ref}}{\sqrt{|W_{cur}|}}} > \delta\right], \quad D_j = sup_x\left|F_{ref}^j(x) - F_{cur}^j(x)\right|, \tag{3.26}$$

де z_k^{uncert} – бінарний сигнал дрейфу за невизначеністю на ітерації k;

k – номер ітерації;

$\hat{\mathrm{U}}_{cur}$ – середня невизначеність у поточному вікні;

$\hat{\mathrm{U}}_{ref}$ – середня невизначеність у референтному вікні;

σ_{ref} – стандартне відхилення невизначеності в референтному вікні;

$|W_{cur}|$ – розмір поточного ковзного вікна, з якого у знаменнику беруть квадратний корінь;

δ – поріг чутливості тесту;

D_j – двовибіркова статистика Колмогорова-Смирнова для ознаки j, взята як супремум за x різниці розподілів;

$F_{ref}^j(x)$ – емпірична функція розподілу ознаки j у референтному вікні;

$F_{cur}^j(x)$ – емпірична функція розподілу ознаки j у поточному вікні.

Знаменник $\frac{\sigma_{ref}}{\sqrt{|W_{cur}|}}$ у тесті середніх є свідомим спрощенням, що ігнорує дисперсію поточного вікна. За порівнянних розмірів і дисперсій вікон воно консервативне, а повна форма відповідає тесту Велча.

Великі значення D_j (типово порогові значення встановлюються в межах 0,05-0,15 залежно від чутливості системи) сигналізують про значущу розбіжність розподілів однієї ознаки, що може свідчити про дрейф або аномальні, подібні до атак, події у вхідних даних. У реалізації детектора порівнюють не саму статистику, а відповідне їй p-значення: ознаку j визнають дрейфовою, якщо $p_{j,k}$ не перевищує скоригованого поправкою Бонферроні порога α/m, тому наведені межі для D_j слугують орієнтиром для інтерпретації, а не робочим порогом.

Фінальний бінарний сигнал детектора z_k формуємо як диз'юнкцію сигналів від двох джерел: z_k^{uncert} (дрейф через невизначеність моделі) та z_k^{feat} (дрейф через зміни

у розподілі ознак). Таке об’єднання забезпечує виявлення різних типів дрейфу концепції: дрейф апостеріорного розподілу (впевненість моделі падає), зсув розподілу ознак (розподіл ознак змінюється), та їх комбінації.

Однак, одиничне виявлення дрейфу на одній ітерації може бути наслідком випадкових флуктуацій або атипових подій, які не відображають справжню зміну розподілу. Щоб уникнути надмірної реактивності та хибних спрацювань детектора, застосуємо механізм підтвердження через накопичення доказів. Система вимагає, щоб дрейф був виявлений послідовно протягом k_{conf} ітерацій (типово 2-5 послідовних детекцій), перш ніж активувати адаптацію. Для оцінки цієї умови обчислюється сума бінарних сигналів за останні k_{conf} ітерацій. Якщо ця сума дорівнює k_{conf}, то всі останні k_{conf} ітерацій показали дрейф, що вважається надійним свідченням істинної зміни розподілу, та сигнал підтверджується: $z_k^{conf} = 1$. В іншому випадку: $z_k^{conf} = 0$, і система залишається в режимі експлуатації, затримуючи адаптацію до отримання більших доказів. Об’єднаний сигнал детектора та правило підтвердження формалізуємо так:

$$z_k^{feat} = 1[min_j p_{j,k} \le \alpha/m], z_k = z_k^{uncert} \vee z_k^{feat}, z_k^{conf} = 1[\sum_{i=k-k_{conf}+1}^{k} z_i = k_{conf}], \quad (3.27)$$

де z_k – сукупний сигнал дрейфу на ітерації k;

z_k^{uncert} – сигнал дрейфу за невизначеністю моделі;

z_k^{feat} – сигнал дрейфу за зміщенням розподілу ознак;

∨ – логічна диз'юнкція;

z_k^{conf} – підтверджений сигнал дрейфу;

$p_{j,k}$ – p-значення двовибіркового тесту Колмогорова-Смирнова, обчислене за статистикою D_j для ознаки j на вікнах W_{ref} і W_{cur};

α – рівень значущості статистичного тесту;

m – кількість ознак, за якою рівень значущості коригують поправкою Бонферроні до порога α/m;

k_{conf} – кількість послідовних ітерацій з виявленим дрейфом, потрібна для підтвердження сигналу;

z_i – бінарний сигнал дрейфу на ітерації i.

Детектор виявляє зміни маргінальних розподілів ознак і не гарантує виявлення зміни умовного розподілу міток за незмінних ознак. Затримка адаптації на $k_{conf} - 1$ ітерацій є необхідною ціною за стабільність: система може пропустити деякі можливості оптимізації на початку дрейфу, проте уникає дорогих та дестабілізуючих адаптацій у відповідь на локальні аномалії. Вибір порогу чутливості δ визначає баланс між своєчасністю виявлення дрейфу та частотою хибних спрацювань детектора. Занадто низький поріг призводить до надмірної реактивності: система постійно переходить у режим дослідження, витрачаючи ресурси на адаптацію до шуму замість справжніх змін розподілу. Занадто високий поріг, навпаки, затримує реакцію на реальний дрейф, збільшуючи період деградації якості. Типові значення δ у діапазоні від двох до чотирьох забезпечують прийнятний компроміс для більшості сценаріїв, проте оптимальне значення залежить від характеристик конкретного середовища та може бути визначене емпірично або включене до складу хромосоми для автоматичної оптимізації.

Розмір референтного та поточного вікон також впливає на чутливість та швидкість реакції детектора. Більші вікна забезпечують стабільніші оцінки статистик, проте уповільнюють виявлення дрейфу та збільшують затримку адаптації. Менші вікна реагують швидше, проте є вразливими до випадкових флуктуацій. На практиці розмір вікна обирається з урахуванням очікуваної швидкості дрейфу та обсягу потоку подій, типово у діапазоні від кількох сотень до кількох тисяч подій.

Дрейф концепції створює дилему для адаптивних систем. У стабільні періоди система має експлуатувати накопичені знання, використовуючи політики, оптимізовані для поточного розподілу. При виявленні дрейфу система має переключитися на дослідження нових рішень, що можуть бути краще адаптовані до зміненого розподілу. Занадто консервативна стратегія призводить до затримки адаптації, тоді як занадто агресивна може дестабілізувати систему через постійні неоптимальні зміни.

Запропонований метод реалізує адаптивну стратегію перемикання між режимами через динамічне керування параметрами еволюційного алгоритму на основі бінарного сигналу детектора дрейфу $z_k \in \{0,1\}$. Коли $z_k = 0$, система перебуває у режимі експлуатації, що відповідає стабільному розподілу даних. Коли

$z_k = 1$, система переходить у режим дослідження, сигналізуючи про виявлення дрейфу концепції та необхідність адаптації до нового розподілу.

Центральним механізмом адаптації є інтенсивність мутації μ_k, що керує ймовірністю модифікації кожного гена в хромосомі під час генетичних операцій. У режимі експлуатації ($z_k = 0$) система підтримує низьку інтенсивність мутації, яка забезпечує локальну оптимізацію навколо поточних розв'язків, мінімізуючи ризик дестабілізації вже оптимізованих політик через випадкові зміни. При виявленні дрейфу ($z_k = 1$) інтенсивність мутації істотно зростає, що розширює простір пошуку еволюційного алгоритму і знаходить принципово нові структури політик, більш адекватні до зміненого розподілу даних. Таким чином, інтенсивність мутації на ітерації k визначимо як:

$$\mu_k = \mu_0 + \mu_1 \cdot z_k^{conf}, \tag{3.28}$$

де μ_0 – базова інтенсивність мутації (типово 0,01-0,05), що підтримується у режимі експлуатації;

μ_1 – додаткова інтенсивність (типово 0,10-0,30), що активується при дрейфі.

Таким чином, формула реалізує функцію перемикача: $\mu_k = \mu_0$ при $z_k^{conf} = 0$ та $\mu_k = \mu_0 + \mu_1$ при $z_k^{conf} = 1$.

Окрім інтенсивності мутації, адаптивно керуються селекційний тиск та розмір популяції. У турнірній селекції, один з найпоширеніших методів відбору батьківських особин, система обирає найкращу особину у випадково обраній підмножині розміру T (турнір). Менший розмір турніру означає менший селекційний тиск: слабші особини мають більше шансу бути вибраними, що сприяє збереженню генетичного різноманіття. Навпаки, більший турнір означає сильніший селекційний тиск: система сконцентрована на експлуатації найкращих поточних рішень, що може призвести до передчасної збіжності до локального оптимуму.

У режимі експлуатації ($z_k = 0$, стабільні дані) система використовує великий турнір T_{max} з метою швидкої збіжності та використання знайдених добрих рішень. При виявленні дрейфу ($z_k = 1$) система переключається на менший турнір T_{min}, що послаблює селекційний тиск, сприяє збереженню різноманітності та виводить еволюційний процес з потенційних локальних оптимумів, оптимальних для старого розподілу, але неоптимальних для нового. Розмір турніру на ітерації k визначимо як:

$$T_k = T_{min} + (T_{max} - T_{min}) \cdot \left(1 - z_k^{conf}\right), \quad (3.29)$$

де T_k – розмір турніру відбору на ітерації k;

T_{min} – мінімальний турнір, що вмикається в режимі дослідження при дрейфі;

T_{max} – максимальний турнір режиму експлуатації для швидкої збіжності.

$\left(1 - z_k^{conf}\right)$ дорівнює 1 при експлуатації (дрейф не виявлено) та 0 при дослідженні (дрейф виявлено), забезпечуючи перемикання між двома режимами.

Паралельно з адаптацією селекційного тиску адаптується розмір популяції N_k. Більша популяція забезпечує кращий спектр різноманіття та дає можливість еволюційному процесу паралельно досліджувати різні регіони простору розв'язків. При виявленні дрейфу розмір популяції збільшується на коефіцієнт $(1 + \gamma)$, що виділяє більше обчислювальних ресурсів на дослідження нових можливостей. Базова популяція утримується розміром N_0 (типово 50-500 особин), а в режимі дослідження розмір зростає до $N_0 \cdot (1 + \gamma)$, де коефіцієнт збільшення γ (типово 0,5-2,0) k отримуємо окремо:

$$N_k = N_0 \cdot \left(1 + \gamma \cdot z_k^{conf}\right), \quad (3.30)$$

де N_k – розмір популяції на ітерації k;

N_0 – базовий розмір популяції у стабільному режимі;

γ – коефіцієнт збільшення популяції при дрейфі;

z_k^{conf} – сигнал дрейфу зі значеннями 0 (стабільно) або 1 (дрейф).

В усіх трьох правилах (3.28), (3.29) і (3.30) використовується саме підтверджений сигнал z_k^{conf} з (3.27): поодиноке спрацювання детектора не змінює параметрів алгоритму. Розмір популяції N_k округлюється до найближчого цілого. При $z_k^{conf} = 0$ популяція залишається розміром N_0, а при $z_k^{conf} = 1$ розтягується до $N_0 \cdot (1 + \gamma)$.

Перехід між режимами супроводжується підсівом нових особин у популяцію з метою уникнення передчасної збіжності та стимулювання дослідження. Коли система переходить у режим дослідження (після підтвердження дрейфу), частина поточної популяції замінюється новими випадково згенерованими особинами. Однак, повна заміна популяції призведе до втрати всіх накопичених знань та різко деградує якість на початку дослідження. Тому використовується гібридний підхід: деяка частка

β_{inject} популяції (типово 10-30 %) замінюється новими особинами P_{random}, а решта $(1-\beta_{inject})$ вибирається з найкращих особин попередньої популяції P_{k-1}^{elite}.

Елітна група P_{k-1}^{elite} складається з найкращих індивідів попередньої ітерації, ранжованих за значенням цільової функції. При формуванні нової популяції система отримує округлену за (3.31) кількість новостворених випадкових особин, а решту місць заповнюють особини елітної групи. На практиці новостворені особини ініціалізуються рівномірно в просторі гіперпараметрів без будь-якого сприяння попереднім розв'язкам, що максимізує різноманітність. Нову популяцію на ітерації k визначаємо як об'єднання елітної групи та випадкових особин:

$$P_k = P_{k-1}^{elite} \cup P_{random}, |P_{random}| = round(\beta_{inject} \cdot N_k), \left|P_{k-1}^{elite}\right| = N_k - |P_{random}|, \quad (3.31)$$

де P_k – нова популяція на ітерації k;

P_{k-1}^{elite} – елітна група найкращих особин попередньої ітерації;

P_{random} – випадково згенеровані особини.

Частку ін'єкції задає $\beta_{inject} \in (0,1)$, а потужності підмножин у сумі дають N_k.

Цей механізм забезпечує баланс між дослідженням (нові особини) та експлуатацією (елітна група), дозволяючи системі як зберегти гарні попередні рішення, так і спробувати нові напрями пошуку, що особливо важливо при дрейфі, коли старі оптимальні рішення можуть стати неоптимальними для нового розподілу.

Критерій завершення режиму дослідження спирається на баланс між двома принципами: система не повинна витрачати ресурси на дослідження, якщо кращих рішень уже не з'являється (стабілізація), але також не повинна закінчувати дослідження надто рано, коли мутації мають мало часу на проведення змін (мінімальна тривалість). Найкраще значення цільової функції на ітерації k позначимо як J_k – це найменше значення об'єднаної цільової функції серед усіх особин в популяції на ітерації k. Для оцінки темпу покращення порівнюємо поточну якість J_k з якістю на ітерації $k-\tau$ (де τ – це зсув, типово 10-20 ітерацій), що дозволяє згладити випадкові коливання. Якщо відносне поліпшення $\frac{|J_k - J_{k-\tau}|}{J_{k-\tau}}$ впадає нижче порога ε_J (типово 0,01-0,05, що відповідає 1-5 % поліпшенню за τ ітерацій), це сигналізує про стабілізацію процесу оптимізації.

Однак, мінімальна тривалість дослідження k_{min} (типово 10-50 ітерацій) гарантує, що система проводить достатньо часу на адаптацію після виявлення дрейфу. Якщо дослідження завершить за лічені ітерації, то новостворені особини з підсіву не матимуть достатньо часу на поліпшення та інтеграцію в популяцію. Тривалість поточного режиму дослідження визначається як $k - k_{drift}$, де k_{drift} – це ітерація, на якій було підтверджено дрейф. Режим дослідження завершується тільки тоді, коли виконані обидві умови: стабілізація цільової функції та проходження мінімального часу. Таким чином, умову завершення дослідження записуємо як

$$exploration_{complete} \Leftrightarrow \frac{|J_k - J_{k-\tau}|}{J_{k-\tau}} < \varepsilon_J \wedge k - k_{drift} > k_{min}, \quad (3.32)$$

де $exploration_{complete}$ – умова завершення режиму дослідження;

J_k – найменше значення об'єднаної цільової функції серед усіх особин популяції на ітерації k;

τ – зсув порівняння, типово 10-20 ітерацій;

$J_{k-\tau}$ – найкраще значення цільової функції на ітерації k-τ;

ε_J – поріг відносного поліпшення, типово 0,01-0,05, що відповідає поліпшенню на 1-5 % за τ ітерацій;

k_{drift} – ітерація, на якій було підтверджено дрейф;

k_{min} – мінімальна тривалість режиму дослідження, типово 10-50 ітерацій.

У знаменнику критерію згідно формули (3.32) фактично використовується величина $max(|J_{k-\tau}|, \varepsilon_0)$ з малою сталою $\varepsilon_0 > 0$, що виключає ділення на нуль. Критерій фіксує відсутність відносного покращення цільової функції протягом τ ітерацій.

Повна реініціалізація популяції при кожному циклі адаптації призводить до втрати накопичених знань. Механізм архіву зберігає успішні політики минулих періодів, тож їх можна використати як відправну точку для адаптації до схожих ситуацій у майбутньому.

Ефективність еволюційної адаптації суттєво залежить від якості початкової популяції на кожному циклі дослідження. Якщо дослідження починається з повністю випадкових особин, система втрачає всі накопичені знання та мусить перезапуститися з нуля. Замість цього запропонований метод підтримує архів успішних політик (т.зв. теплий старт), що використовуються при кожному новому виявленні дрейфу. Архів

зберігає не просто хромосоми, а й контекстну інформацію про період, в якому політика була оптимальною, що дозволяє згодом оцінити релевантність архівованої політики до поточної ситуації.

Архів $P_{archive}$ формально представляється як множина кортежів, де кожна архівована політика χ_j (збережена хромосома, яка кодує набір правил ЗВД) зберігається разом із супутніми даними: значенням цільової функції J_j; часовою міткою t_j; контекстом c_j – описує дані та розподіли у час архівування. Контекст складається з трьох компонент: μ_X^j – вектор середніх значень кожної ознаки (розмір m), що характеризує типові значення у період архівування; Σ_X^j – коваріаційна матриця розміру $m \times m$, що описує взаємні залежності та варіативність ознак; π_Y^j – розподіл оцінок ризику (гістограма), що показує, який відсоток подій мав низький, середній та високий ризик. Разом ці компоненти охоплюють як маргінальні статистики (середні та дисперсії кожної ознаки), так і структурні залежності (коваріаційна матриця) та апостеріорні характеристики (розподіл оцінок ризику).

Архів обмежений у розмірі максимум M політик (типово 20-50), що дозволяє системі зберегти довгострокову пам’ять без надмірних обчислювальних витрат при пошуку релевантних політик. Архів визначимо як:

$$P_{archive} = \{(\chi_j, J_j, t_j, c_j)\}_{j=1}^{M}, \quad c_j = (\mu_X^j, \Sigma_X^j, \pi_Y^j), \tag{3.33}$$

де $P_{archive}$ – архів успішних політик;

χ_j – збережена хромосома j-ї політики, що кодує набір правил ЗВД;

J_j – значення цільової функції цієї політики;

t_j – часова мітка архівування;

c_j – контекст, який описує дані та розподіли на момент архівування;

j – номер запису в архіві;

M – гранична кількість політик в архіві;

μ_X^j – вектор середніх значень ознак у період архівування;

Σ_X^j – коваріаційна матриця ознак;

π_Y^j – розподіл оцінок ризику.

Архів містить $M_t \leq M$ записів: стала M є верхньою межею, за її досягнення витісняється найменш релевантний запис, тому множина містить щонайбільше M найрелевантніших і найякісніших політик з усіх будь-коли архівованих.

Додавання нової політики до архіву відбувається за строгим критерієм новизни та якості, щоб запобігти дублюванню подібних рішень та переповненню архіву. Політика χ додається до архіву тільки якщо виконані дві умови: перша, її якість достатня ($J(\chi) < J_{thr}$, де J_{thr} – встановлений поріг якості, типово значення 75-го перцентилю історичних значень); друга, вона достатньо відмінна від усіх наявних архівних політик ($\min_j d(\chi, \chi_j) > d_{min}$, де $d(\chi, \chi_j)$ – деяка відстань у просторі хромосом, наприклад евклідова відстань нормалізованих параметрів, а d_{min} – мінімальна допустима відстань, типово 0,2-0,5). Таким чином, архів збагачується тільки справді новими та хорошими політиками, а не варіаціями вже архівованих рішень.

Релевантність архівованої політики до поточного контексту оцінюється шляхом порівняння контекстів. Інтуїтивно, якщо поточні дані подібні до даних, за яких було архівовано політику (подібні розподіли ознак та оцінок ризику), то архівована політика, ймовірно, буде хорошою стартовою точкою для адаптації. Релевантність $\rho_r(\chi_j)$ для кожної архівованої політики j обчислюємо як експоненційну функцію відстані контекстів, де $d_{ctx}(c_j, c_{cur})$ – міра відстані між контекстом архівованої політики c_j та поточним контекстом c_{cur} (наприклад, просумована відстань між середніми значеннями та коваріаціями), а λ_c – коефіцієнт масштабу, що контролює «ширину» експоненційного ядра (типово 0,5-2,0). Більша d_{ctx} означає меншу релевантність: експоненційна функція швидко спадає, коли контексти відрізняються. Критерій додавання та релевантність записуємо так:

$$\xi(\chi) \Leftrightarrow J(\chi) < J_{thr} \wedge \min_j d(\chi, \chi_j) > d_{min},$$
$$\rho_r(\chi_j) = \exp\left(-\lambda_c \cdot d_{ctx}(c_j, c_{cur})\right), \tag{3.34}$$

де значення $\rho_r(\chi_j)$ належать інтервалу (0,1], з максимумом 1 при нульовій відстані контекстів.

Початкова популяція P_0 при теплому старті (активується на початку режиму дослідження) формується як комбінація трьох компонентів, кожна з яких грає окрему

роль у балансі між раніш знайденими рішеннями та експериментуванням з новими можливостями.

$$P_0 = P_{warm} \cup P_{mutated} \cup P_{random}, \quad (3.35)$$

де P_0 – початкова популяція теплого старту;

P_{warm} – політики, відібрані з архіву пропорційно релевантності;

$P_{mutated}$ – мутовані версії архівованих політик;

P_{random} – випадково згенеровані особини без зв'язку з архівом.

З кожного відібраного запису архіву в популяцію потрапляє лише хромосома χ_j, а критерієм відбору слугує релевантність $\rho_r(\chi_j)$ за формулою (3.34) разом зі значенням цільової функції J_j (менші значення кращі), що узгоджує напрями оптимізації.

Перший компонент P_{warm} складається з політик, відібраних безпосередньо з архіву пропорційно їх релевантності до поточного контексту. Політики з вищою релевантністю $\rho_r(\chi_j)$ мають більший вплив і вибираються частіше, тоді як політики з низькою релевантністю (дані сильно змінилися від часу архівування) вибираються рідко або взагалі не вибираються. Цей компонент забезпечує теплий старт: замість генерації з нуля, система починає з вже оптимальних або близьких до оптимальних рішень для схожого контексту.

Другий компонент $P_{mutated}$ складається з мутованих версій архівованих політик. Мутація застосовується з підвищеною інтенсивністю, щоб створити варіації навколо архівних рішень, покращуючи їх та досліджуючи сусідній простір розв'язків. На відміну від чистого копіювання з архіву, мутовані версії можуть краще адаптуватися до конкретних змін у розподілі даних.

Третій компонент P_{random} складається з випадково згенерованих особин без зв'язку з архівом. Хоча архівовані рішення та їх мутації важливі, вони можуть утримувати еволюцію у локальних оптимумах, які були оптимальні для старого розподілу, але непридатні для нового. Випадкові особини вносять різноманітність та дозволяють алгоритму дослідити принципово нові регіони простору розв'язків, які не було досліджено раніше.

Кількість особин у кожній частині популяції зазвичай задається фіксованими частками: P_{warm} становить приблизно 30-50 % новоствореної популяції, $P_{mutated}$ –

20-40 %, та P_{random} – решта 10-30 %, залежно від конкретної конфігурації. На практиці цими частками можна керувати адаптивно залежно від темпу дрейфу та історії адаптацій.

Архів служить формою довгострокової пам'яті системи, яка охоплює не кілька сотень поточних особин популяції, а весь історичний досвід функціонування, що дає системі змогу переживати рекурентні дрейфи (коли дані повертаються до раніш спостережуваних розподілів).

Перехід від однієї політики до іншої при виявленні дрейфу є потенційно ризикованою операцією у реальних системах. Нова політика, оптимізована на обмеженій вибірці нових даних під час дослідження (можливо, кілька днів або тижнів подій), може виявитися неоптимальною при розгортанні на повному потоці, коли її застосовують до мільйонів подій з більшим різноманіттям. Тому замість миттєвого переходу від старої політики до нової система використовує зважену суміш кількох активних політик, що гарантує поступову адаптацію та зменшує ризик раптової деградації.

У кожен момент часу система підтримує ансамбль з K_{ens} активних політик, які також називаються базовими класифікаторами. Для кожної політики $j = 1, \dots, K_{ens}$ система обчислює показник ризику $s_t^{(j)}$ для поточної події x_t, тобто оцінку ймовірності витоку, яку надає j-та політика окремо. Однак замість прийняття рішення на основі однієї політики, система комбінує оцінки всіх активних політик за допомогою ваг π_j, що відображають довіру до кожної політики. Ваги задовольняють два обмеження: (1) невід'ємність $\pi_j \geq 0$ для всіх j; (2) нормування $\sum_{j=1}^{K_{ens}} \pi_j = 1$, тобто ваги складаються в одиницю і можуть інтерпретуватися як ймовірності або частки впливу.

Сукупний показник ризику ансамблю s_t^{ens} обчислюємо як зважене середнє оцінок окремих політик. Політики з вищими вагами (вища довіра) більше впливають на підсумкову оцінку і задамо їх так:

$$s_t^{ens} = \sum_{j=1}^{K_{ens}} \pi_j \cdot s_t^{(j)}, \quad \sum_{j=1}^{K_{ens}} \pi_j = 1, \tag{3.36}$$

де s_t^{ens} – сукупна оцінка ризику ансамблю в момент t;

π_j – вага (довіра) j-ї політики;

$s_t^{(j)}$ – оцінка ризику j-ї політики;

K_{ens} – кількість політик в ансамблі, причому сума ваг за всіма політиками дорівнює одиниці.

Ваги ансамблю не залишаються статичними, а оновлюються за мультиплікативною схемою. Для кожної політики використовуємо миттєву функцію вартості $\tilde{C}_t(\chi_j)$, що агрегує похибку прогнозу ризику, затримку обробки та невизначеність:

$$\pi_j(t+1) = \frac{\pi_j(t) \cdot exp(-\eta_w \cdot \tilde{C}_t(\chi_j))}{Z_t}, \tilde{C}_t(\chi_j) = \lambda_{err} \cdot (s_t^{(j)} - y_t)^2 + \lambda_{lat} \cdot \tilde{\ell}_t^{(j)} + \lambda_{conf} \cdot \tilde{U}_t^{(j)}, \quad (3.37)$$

де $\pi_j(t+1)$ – оновлена вага j-ї політики;

$\pi_j(t)$ – її вага на кроці t;

j – номер політики в ансамблі;

t – номер кроку обробки подій;

η_w – чутливість оновлення ваг;

exp – експоненційна функція;

$\tilde{C}_t(\chi_j)$ – миттєва вартість політики;

χ_j – хромосома j-ї політики ансамблю;

Z_t – нормувальна константа, яка забезпечує рівність одиниці суми оновлених ваг за всіма політиками ансамблю;

λ_{err} – невід'ємна вага складника похибки прогнозу ризику;

$\left(s_t^{(j)} - y_t\right)^2$ – квадратична похибка прогнозу ризику відносно відкладеної істинної мітки;

$s_t^{(j)}$ – оцінка ризику j-ї політики для події t;

y_t – відкладена істинна мітка події t;

λ_{lat} – невід'ємна вага складника затримки обробки;

$\tilde{\ell}_t^{(j)}$ – нормована до [0,1] затримка обробки j-ї політики;

λ_{conf} – невід'ємна вага складника невизначеності прогнозу;

$\tilde{U}_t^{(j)}$ – нормована до [0,1] невизначеність прогнозу j-ї політики.

Мультиплікативне оновлення зміщує ваги на користь політик з меншою вартістю. За відсутності відкладеної мітки замість браєрівського члена допускається лише чітко обґрунтований замінник з окремим контролем калібрування прогнозів.

Для довготривалої адаптації вводиться віковий пріор. Нехай $\Delta t_j = t - t_{added}^{(j)}$ позначає вік політики j в ансамблі, а $\rho_w \in (0,1)$ задає коефіцієнт затухання. Підсумкові ваги обчислюємо, зважуючи оновлені ваги віковим пріором і нормуючи результат так:

$$\pi_j^{norm}(t+1) = \frac{\pi_j(t+1) \cdot \rho_w^{\Delta t_j}}{\sum_l \pi_l(t+1) \cdot \rho_w^{\Delta t_l}}, \tag{3.38}$$

де $\rho_w^{\Delta t_j}$ – віковий пріор, що знижує відносний вплив давніших політик: пріоритети політик різного віку різні, тому нормування їх не скорочує, а політики однакового віку розрізняються лише базовими вагами $\pi_j(t+1)$ з формули (3.37).

Пріоритет щоразу обчислюється саме від базової ваги, а не від уже завершеної, тож повторне нашарування степенів не виникає, а знаменник відновлює нормування $\sum_j \pi_j^{norm} = 1$. Механізм знижує вплив застарілих політик без їх повного видалення, зберігаючи здатність системи реагувати на рекурентні зміни розподілу. В агрегації з формули (3.36) та консервативному правилі за формулою (3.39) використовуються саме нормовані ваги $\pi_j^{norm}(t+1)$, тоді як рекурсія за формулою (3.37) на наступному кроці продовжується від базових ваг $\pi_j(t+1)$.

Для критичних рішень застосовується консервативна агрегація, коли серед політик із достатньою вагою береться максимальна оцінка ризику:

$$s_t^{cons} = \max_{j:\pi_j > \pi_{voice}} s_t^{(j)}, \tag{3.39}$$

де s_t^{cons} – консервативна (максимальна) оцінка ризику в момент t;

$s_t^{(j)}$ – оцінка ризику j-ї політики;

π_j – вага політики;

π_{voice} – поріг «голосу», тобто мінімальна вага для участі в рішенні, максимум береться лише серед політик, вага яких перевищує цей поріг.

Такий режим гарантує врахування високоризикових сигналів навіть за домінування низькоризикових оцінок у середньому.

Описані механізми формують узгоджений алгоритм еволюційної адаптації до дрейфу, зображений на рисунку 3.3. Він поєднує стабільність, інтерпретованість і здатність до самооновлення в умовах зміни розподілу даних. Він забезпечує як реактивну перебудову ваг на основі поточної якості політик, так і контроль довгострокового впливу через затухання, що разом створює стійку динаміку ансамблю.

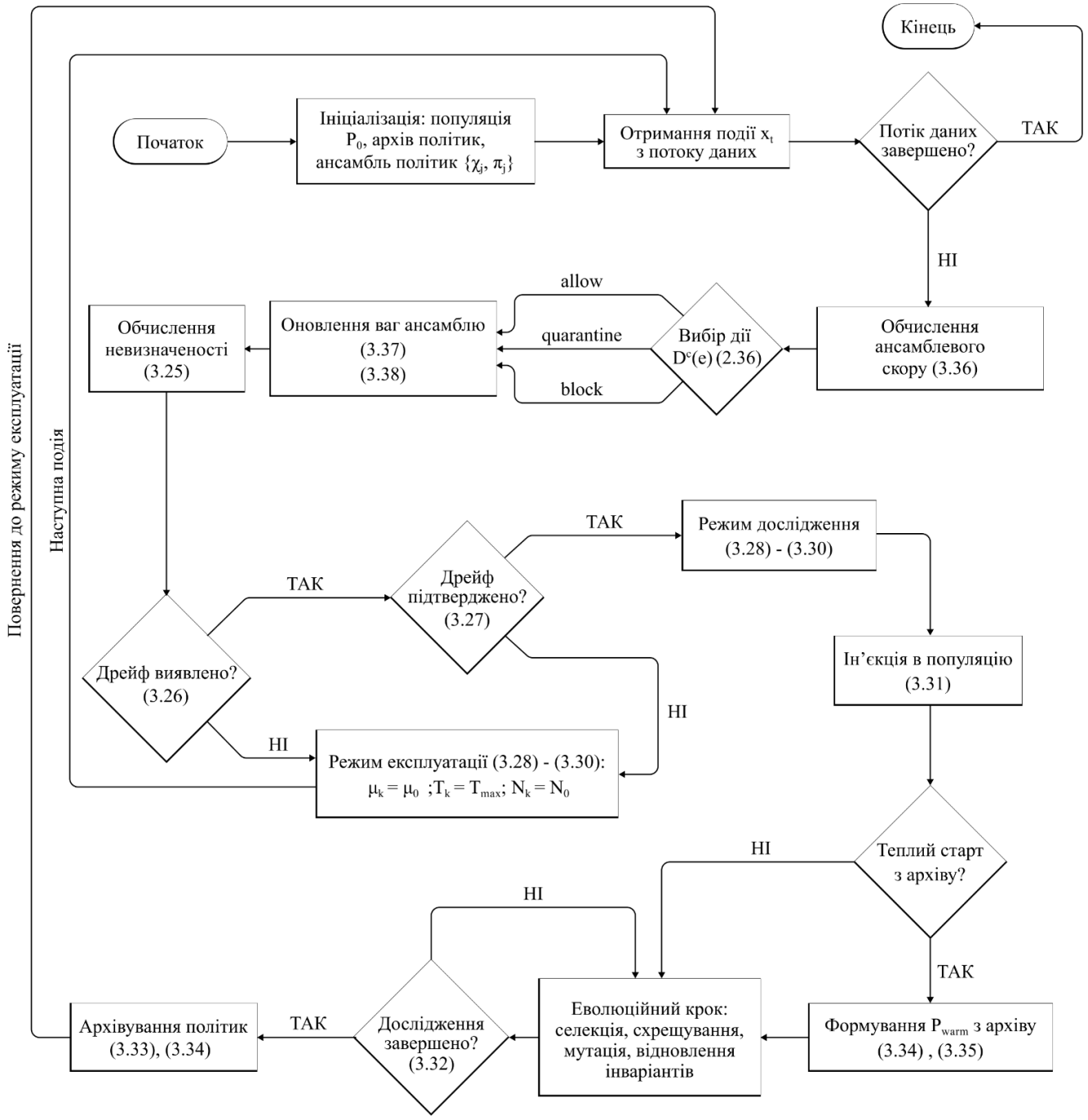


Рисунок 3.3 – Алгоритм еволюційної адаптації до дрейфу

Метод еволюційної адаптації політик має кілька системних переваг порівняно з наявними підходами до обробки дрейфу концепції. Механізм виявлення дрейфу

спирається на два незалежних джерела сигналу: моніторинг невизначеності моделі та статистичне відстеження розподілу ознак. Це забезпечує більш стабільне виявлення змін порівняно з односигнальними детекторами. Архів успішних політик із механізмом теплого старту дозволяє системі використовувати накопичений досвід при рекурентних дрейфах, релевантні архівовані політики забезпечують швидку адаптацію без пошуку рішень з нуля. Зважена суміш активних політик із мультиплікативним оновленням ваг та механізмом експоненційного затухання забезпечує плавний перехід між конфігураціями. Адаптивне керування інтенсивністю мутації (формула (3.28)) та селекційним тиском залежно від режиму роботи є оригінальним механізмом, що забезпечує автоматичний баланс між стабільністю та гнучкістю. Запропонований метод зберігає інтерпретованість результуючих політик.

Кожен розглянутий тип дрейфу співвідноситься із конкретним засобом методу, це зіставлення наведено в таблиці 3.2.

Таблиця 3.2 - Типи дрейфу концепції та механізми методу

Тип дрейфу	Визначення	Приклад у системі ЗВД	Механізм методу
Раптовий	Різка, майже миттєва зміна розподілу P(X,Y)	Реорганізація чи злиття компанії, нова корпоративна система	Детектор за двома ковзними вікнами (3.25)-(3.27); перехід у режим дослідження з підвищеною мутацією (3.28)
Поступовий	Повільне витіснення старих шаблонів новими	Поступова зміна лексики й тематики документів	Режим дослідження з ансамблем активних політик; плавне перенесення ваг (3.37)-(3.38)
Інкрементний	Послідовні дрібні зміни, що накопичуються	Повільна зміна звичок окремого користувача	Адаптивний профіль з експоненційним забуванням; вікове згасання ваг (3.38)
Рекурентний	Повернення раніше баченого розподілу	Сезонні процеси: квартальна звітність, відпустки	Архів успішних політик із контекстом (3.33)-(3.35) і теплий старт

Жоден окремий засіб не покриває всіх чотирьох сценаріїв, тому метод поєднує детектор дрейфу з режимом дослідження, ансамбль активних політик, експоненційне забування профілю й архів політик з теплим стартом.

Класична схема навчання з відкладеною тестовою вибіркою погано узгоджується з потоковою постановкою задачі. Розподіл подій у системі ЗВД змінюється з часом, тому фіксована тестова вибірка швидко застаріває й перестає відображати поточний стан середовища. Натомість для безперервних потоків застосовують почергове тестування-навчання. Кожен зразок, що надходить, спочатку класифікується наявною моделлю і лише після цього долучається до її оновлення. Так одне й те саме спостереження дає внесок і в оцінку якості, і в подальше навчання, а окрема відкладена вибірка стає непотрібною.

Перевага такого підходу для адаптивних систем у тому, що він вимірює модель саме в умовах, у яких вона працює. Кожне передбачення робиться до того, як модель «побачила» правильну відповідь, тож накопичена або ковзна оцінка точності чесно відбиває здатність системи узагальнювати на ще не баченому. Це особливо зручно для аналізу поведінки методу до і після дрейфу [114]. Зіставляючи ковзну оцінку якості на референтному та поточному вікнах, можна простежити просідання показника в момент зміни розподілу й швидкість його відновлення після переходу в режим дослідження.

Вибір метрики для потоку подій ЗВД диктує сильний дисбаланс класів. Конфіденційні події, що потребують втручання, становлять незначну частку загального трафіку. За такого співвідношення частка правильних відповідей (accuracy) втрачає інформативність: тривіальний класифікатор, що позначає кожну подію як легітимну, досягає високої точності, не виявивши жодного витоку. Міра F1 усуває цю ваду, бо погано переносить ігнорування рідкісного класу. Повнота відповідає частці виявлених витоків і пов'язана з ціною пропуску, що її формалізує цільова функція J_1, точність відбиває частку справді небезпечних подій серед позначених системою й кореспондує з J_2.

Стандартна F1 зважує повноту й точність симетрично, тоді як у задачі ЗВД вартість пропуску витоку істотно перевищує вартість хибної тривоги. Цю асиметрію формалізовано у функції вартості помилок $C_{total} = C_{FP} \cdot FP + C_{FN} \cdot FN$ (формула (2.37)), де C_{FN} значно перевищує C_{FP}, та в задачі вартісно-чутливої оптимізації

порогів (2.65)-(2.68). Тому F1 використано як зведену міру для зіставлення конфігурацій і відстеження динаміки якості в потоці, а не як безпосередній критерій оптимізації: пріоритет повноти забезпечують цільові функції J_1 і J_2.

3.3 Метод поведінкового профілювання користувачів на основі генетичного алгоритму

Методи класифікації документів та адаптації політик оперують характеристиками інформаційних об'єктів та параметрами системи, проте не враховують поведінкового контексту дій користувача, який є важливим індикатором інсайдерських загроз. Контентний аналіз фіксує факт передачі чутливих даних, однак не здатний розрізнити рутинну операцію уповноваженого працівника та аномальну активність скомпрометованого облікового запису. Налаштування детектора аномалій у реальному корпоративному середовищі з десятками поведінкових ознак, різними категоріями користувачів та мінливими робочими шаблонами вимагає підбору великої кількості параметрів, ручне налаштування цих параметрів стає непрактичним уже при масштабі в кілька сотень користувачів. Ця суперечність між потребою в індивідуалізованій конфігурації та неможливістю забезпечити її експертними зусиллями мотивує застосування генетичного алгоритму для автоматичної оптимізації параметрів поведінкового детектора.

Побудова профілю починається з реєстрації подій з журналів автентифікації, файлових операцій, мережної активності та доступу до корпоративних ресурсів. Кожна подія описується кортежем $e = (u, t, a, r, s)$, де u – ідентифікатор користувача, що виконав дію; t – часова мітка події; a – тип дії (автентифікація, файлова операція, мережне з’єднання тощо); r – ресурс або об’єкт, до якого здійснювався доступ; s – вектор контекстних атрибутів, що включає робочу станцію, IP-адресу, роль користувача, канал доступу та інші релевантні характеристики.

Така структура охоплює різнорідні джерела журналів корпоративних систем: записи Active Directory про автентифікацію, журнали файлових серверів, записи проксі-серверів та брандмауерів, записи систем контролю доступу. Контекстний вектор s забезпечує гнучкість представлення, дозволяючи включати специфічні для конкретного середовища атрибути без зміни базової структури.

Безпосередня робота з потоком елементарних подій є обчислювально неефективною і концептуально недоцільною. Поодинокі дії рідко несуть достатньо інформації для оцінки аномальності; значущими є шаблони, що проявляються в агрегованих характеристиках активності за певний період. Тому поведінка користувача аналізується у часових вікнах фіксованої тривалості $\Delta > 0$.

Множину подій користувача u у вікні з порядковим номером k визначаємо як:

$$E_{u,k} = \{e: e = (u, t, a, r, s) \wedge k\Delta \leq t < (k+1)\Delta\}, \quad (3.40)$$

де $E_{u,k}$ – множина подій користувача u у вікні з порядковим номером k;

e – окрема подія;

u – ідентифікатор користувача;

t – часова мітка події;

a – тип дії;

r – ресурс або об'єкт доступу;

s – вектор контекстних атрибутів;

Δ – тривалість часового вікна, подія потрапляє у вікно, коли її мітка t лежить у напівінтервалі від $k\Delta$ до $(k+1)\Delta$.

Вибір тривалості вікна Δ залежить від специфіки організації: для виявлення добових шаблонів зазвичай використовують $\Delta = 24$ години, для більш детального аналізу – погодинні або навіть п'ятнадцятихвилинні інтервали. Надто короткі вікна підвищують чутливість до випадкових флуктуацій, надто довгі – згладжують значущі відхилення.

На основі множини подій $E_{u,k}$ формується вектор ознак, що кількісно характеризує поведінку користувача у відповідному часовому вікні:

$$x^{(u)}(k) = \left(x_1^{(u)}(k), x_2^{(u)}(k), \ldots, x_m^{(u)}(k)\right), \quad (3.41)$$

де $\mathrm{x}^{(u)}(k) \in \mathrm{R}^m$ – вектор ознак користувача u у вікні k;

m – загальна кількість ознак у профілі;

$x_i^{(u)}(k)$ – значення i-ї ознаки.

Кожну ознаку обчислюємо через агрегаційну функцію, застосовану до відфільтрованої підмножини подій:

$$x_i^{(u)}(k) = Agg_{c_i}\left(\{e \in E_{u,k}: \tau(e) \in T_i\}\right), \quad (3.42)$$

де $x_i^{(u)}(k)$ – значення i-ї ознаки користувача u у вікні k;

Agg_{c_i} – агрегаційна функція категорії c_i (сумування, сума, середнє, максимум тощо);

e – подія;

$E_{u,k}$ – множина подій користувача у вікні, тип події e повертає $\tau(e)$;

T_i – множина типів подій, релевантних для i-ї ознаки.

Конкретний склад ознак визначається доступними джерелами журналів та аналітичними потребами організації. Типовий профіль охоплює ознаки чотирьох великих категорій: автентифікаційна активність (кількість входів, невдалі спроби, зміни паролів), файлові операції (читання, запис, копіювання, видалення з розбивкою за типами ресурсів), мережна активність (обсяг трафіку, кількість з'єднань, унікальні призначення) та доступ до корпоративних ресурсів (бази даних, внутрішні сервіси, хмарні застосунки). Детальний перелік категорій ознак наведено у таблиці 3.3.

Таблиця 3.3 - Категорії поведінкових ознак для профілювання користувачів

Категорія	Приклади ознак	Тип агрегації
1	2	3
Успішні автентифікації	Кількість входів у систему за період	Підрахунок
Невдалі автентифікації	Кількість невдалих спроб входу	Підрахунок
Автентифікації поза робочим часом	Входи у вечірні/нічні години та вихідні	Підрахунок
Унікальні робочі станції	Кількість різних пристроїв для входу	Унікальні
Читання файлів	Кількість операцій читання	Підрахунок
Запис файлів	Кількість операцій запису/створення	Підрахунок
Копіювання файлів	Кількість операцій копіювання	Підрахунок
Видалення файлів	Кількість операцій видалення	Підрахунок
Доступ до конфіденційних каталогів	Операції з позначеними ресурсами	Підрахунок
Обсяг прочитаних даних	Сумарний розмір прочитаних файлів	Сума

Кінець таблиці 3.3

1	2	3
Обсяг записаних даних	Сумарний розмір записаних файлів	Сума
Унікальні файлові розширення	Різноманітність типів файлів	Унікальні
Вихідні мережні з’єднання	Кількість з’єднань назовні	Підрахунок
Обсяг вихідного трафіку	Сумарний обсяг надісланих даних	Сума
Обсяг вхідного трафіку	Сумарний обсяг отриманих даних	Сума
Унікальні зовнішні адреси	Кількість різних зовнішніх хостів	Унікальні
З’єднання з хмарними сховищами	Доступ до Dropbox, Google Drive тощо	Підрахунок
Використання USB-пристроїв	Підключення знімних носіїв	Підрахунок
Надіслані електронні листи	Кількість надісланих повідомлень	Підрахунок
Листи із вкладеннями	Повідомлення з прикріпленими файлами	Підрахунок
Обсяг вкладень	Сумарний розмір вкладень	Сума
Унікальні зовнішні адресати	Кількість різних одержувачів поза організацією	Унікальні
Доступ до баз даних	Кількість запитів до БД	Підрахунок
Обсяг вивантажених даних з БД	Сумарний розмір результатів запитів	Сума

Детектор, що спирається на фіксовану базову лінію, почне сигналізувати про аномалії там, де насправді відбулася природна еволюція робочих шаблонів.

Фіксована базова лінія поведінки не враховує природну еволюцію робочих шаблонів користувачів. Механізм експоненційного забування дозволяє профілю адаптуватися до поступових змін, надаючи більшу вагу останнім спостереженням.

Для врахування дрейфу поведінки пропонується механізм адаптивного профілю з експоненційним забуванням. Замість обчислення статистик за всю історію спостережень, більша вага надається недавнім спостереженням, а вплив старих даних

поступово згасає. Такий підхід дозволяє профілю “слідувати” за легітимними змінами поведінки, зберігаючи при цьому чутливість до різких відхилень.

Адаптивне середнє значення i-ї ознаки для користувача u оновлюємо за рекурентною формулою так:

$$\mu_i^{(u)}(k) = \rho \cdot \mu_i^{(u)}(k-1) + (1-\rho) \cdot x_i^{(u)}(k), \quad (3.43)$$

де $\mu_i^u(k)$ – експоненційно зважене середнє і-ї ознаки користувача u після оновлення;

u – користувач, для якого будують профіль;

k – номер оновлення профілю;

$\rho \in [0,1]$ – коефіцієнт забування, більші значення коефіцієнту забування формують профіль інертнішим, а менші – його чутливішим до нових спостережень;

$\mu_i^u(k-1)$ – попереднє значення експоненційно зваженого середнього;

$x_i^u(k)$ – спостережене значення і-ї ознаки користувача u на кроці k.

Параметр ρ визначає швидкість адаптації профілю до нових даних. При $\rho \to 1$ профіль змінюється повільно, зберігаючи “пам’ять” про давні спостереження; при $\rho \to 0$ профіль швидко забуває минуле і реагує переважно на останні події. Оптимальне значення ρ залежить від динаміки конкретного середовища і є одним з параметрів, що підлягають еволюційній оптимізації.

Для нормалізації відхилень необхідно також відстежувати масштаб варіабельності ознак. Адаптивну дисперсію обчислюємо аналогічним чином:

$$(\sigma_i^{(u)}(k))^2 = \rho \cdot (\sigma_i^{(u)}(k-1))^2 + (1-\rho) \cdot (x_i^{(u)}(k) - \mu_i^{(u)}(k))^2 + \varepsilon_v, \quad (3.44)$$

де $(\sigma_i^{(u)}(k))^2$ – експоненційно зважена дисперсія після оновлення;

$\varepsilon_v > 0$ - мала регуляризаційна стала.

Оскільки в інноваційному доданку використано вже оновлене середнє, то рекурсія є консервативним варіантом класичної експоненційно зваженої оцінки дисперсії (внесок $\rho^2(1-\rho)\delta^2$ замість $\rho(1-\rho)\delta^2$ для відхилення δ від попереднього середнього), дисперсія дещо занижується, що робить детектор чутливішим, порядок обчислень принциповий: оцінка (3.45) завжди передує оновленням (3.43) і (3.44), для захисту норми від поступового «отруєння» події, що спричинили тривогу, доцільно включати до оновлення профілю зі зниженою вагою або лише після підтвердження аналітиком.

Нормалізовану міру відхилення поточного значення від профільної норми виразимо через z-оцінку (z-score) так:

$$z_i^{(u)}(k) = \frac{x_i^{(u)}(k) - \mu_i^{(u)}(k-1)}{\sigma_i^{(u)}(k-1) + \varepsilon}, \quad (3.45)$$

де $z_i^{(u)}(k)$ – стандартизоване відхилення i-ї ознаки користувача u у вікні k, обчислене за профілем попереднього вікна $\left(\mu_i^{(u)}(k-1), \sigma_i^{(u)}(k-1)\right)$;

$x_i^{(u)}(k)$ – поточне спостереження;

$\varepsilon > 0$ – мала константа чисельної стійкості (типово 10^{-6}).

Показник обчислюється до оновлення профілю, тому поточна подія не входить у власний еталон і не приглушує аномальність.

Значення $z_i^{(u)}(k)$ показує, на скільки стандартних відхилень поточне спостереження відрізняється від очікуваного для цього користувача. При $|z| < 1$ поведінка перебуває в межах типової варіабельності; значення $|z| > 2$ вказують на помірні відхилення; $|z| > 3$ зазвичай свідчать про суттєві аномалії. Важливо, що ця нормалізація є контекстуальною: однакове абсолютне відхилення матиме різну z-оцінку для користувачів з різною базовою варіабельністю, що забезпечує справедливе порівняння поведінки різних співробітників.

Обчислені z-оцінки для окремих ознак необхідно агрегувати у єдиний показник аномальності поведінки користувача. Проста сума або середнє арифметичне z-оцінок не є оптимальним рішенням з кількох причин. По-перше, різні ознаки мають різну діагностичну цінність для виявлення загроз. По-друге, від'ємні z-оцінки (поведінка "нижче норми") зазвичай менш підозрілі, ніж позитивні (перевищення норми). По-третє, деякі ознаки можуть бути шумовими або надлишковими, і їх включення погіршує якість виявлення.

Для перетворення z-оцінки у невід'ємний внесок в загальний показник аномальності застосовуємо функцію активації:

$$\varphi(z) = \max(0, z), \quad (3.46)$$

де $\varphi(z)$ – функція активації, що повертає більше зі значень 0 та z, обнуляючи від'ємні відхилення;

z – стандартизована z-оцінка відхилення ознаки.

Вибір ReLU-подібної функції обґрунтовується тим, що для більшості поведінкових ознак саме перевищення норми (більше файлових операцій, більше з'єднань, більше невдалих автентифікацій) є індикатором потенційних проблем, тоді як зниження активності рідко свідчить про зловмисні дії. У разі потреби функцію можна модифікувати для окремих категорій ознак, де значення мають симетричні відхилення.

Загальний показник аномальності для користувача u у часовому вікні k обчислюємо як зважену суму перетворених z-оцінок за активними ознаками:

$$S^{(u)}(k) = \frac{\sum_{i=1}^{m} b_i \cdot w_i \cdot \varphi\left(z_i^{(u)}(k)\right)}{\sum_{i=1}^{m} b_i \cdot w_i + \varepsilon}, \tag{3.47}$$

де $S^{(u)}(k) \geq 0$ – скалярний показник аномальності;

$b_i \in \{0,1\}$ – індикатор включення i-ї ознаки в модель;

$w_i \geq 0$ – вага ознаки, що визначає її відносну важливість;

$\varphi(\cdot)$ – функція перетворення з формули (3.46).

Нормування на суму активних ваг усуває залежність масштабу показника від кількості ознак, робить поріг спрацювання порівнянним між конфігураціями та ототожнює хромосоми, еквівалентні з точністю до масштабування ваг і порога.

Бінарний вектор $(b_1, \dots, b_m)$ реалізує механізм відбору ознак: якщо $b_i = 0$, ознака виключається з розрахунку незалежно від її ваги. Це дозволяє зменшити розмірність моделі, усунувши шумові або надлишкові ознаки. Ваги w_i забезпечують тонке налаштування відносного впливу кожної активної ознаки.

Рішення про генерацію тривоги приймаємо шляхом порівняння оцінки аномальності з пороговим значенням:

$$\delta_u(k) = 1\left[S^{(u)}(k) \geq \tau\right], \tag{3.48}$$

де $\delta_u(k) \in \{0,1\}$ – вихід детектора (1 – тривога, 0 – норма);

$\tau > 0$ – поріг спрацювання.

Поріг τ безпосередньо контролює баланс між чутливістю та специфічністю детектора. Низький поріг забезпечує високу повноту виявлення, але ціною збільшення хибних тривог. Високий поріг зменшує хибні спрацювання, але може пропускати реальні інциденти. Оптимальне значення τ залежить від співвідношення

вартості пропущеного інциденту та вартості обробки хибної тривоги, що варіюється між організаціями.

Ефективність описаного детектора залежить від правильного вибору конфігурації: які ознаки включити, як розподілити ваги, яке значення порогу встановити, з якою швидкістю адаптувати профіль. Ручне налаштування цих параметрів є трудомістким і ненадійним, особливо в умовах великого простору ознак та змінних вимог. Пропонується автоматизувати цей процес за допомогою генетичного алгоритму, який шукає оптимальну конфігурацію в просторі можливих варіантів.

Налаштування детектора аномалій охоплює вибір активних ознак, їхніх ваг, порогу спрацювання та швидкості адаптації профілю. Кодування цих параметрів у формі хромосоми дозволяє застосувати генетичний алгоритм для автоматичного пошуку оптимальної конфігурації.

Конфігурацію детектора кодуємо як хромосому, що об'єднує всі параметри в єдину структуру:

$$\chi_b = (B, W_b, \tau, \rho), \tag{3.49}$$

де χ_b – хромосома, що представляє повну конфігурацію детектора;

$B = (b_1, \dots, b_m)$ – бінарна маска активних ознак;

$W_b = (w_1, \dots, w_m)$ – вектор ваг ознак;

τ – поріг тривоги;

ρ – коефіцієнт забування в адаптивному профілі.

Така структура хромосоми поєднує дискретні компоненти (бінарна маска B) з неперервними (W_b, τ, ρ), що потребує спеціалізованих генетичних операторів. Для бінарної частини застосовується однорідне схрещування та бітова мутація; для неперервних параметрів – арифметичне схрещування та гаусівська мутація з адаптивним кроком.

Функція пристосованості визначає якість конфігурації з урахуванням множинних критеріїв. Оскільки метою є не лише максимізація точності виявлення, а й забезпечення експлуатаційної придатності, у функцію вводимо явні штрафи за небажані характеристики:

$$F(\chi_b) = F_1(\chi_b) - \alpha \cdot \nu_{FP}(\chi_b) - \beta \cdot \nu_{CV}(\chi_b) - \gamma \cdot \Gamma(\chi_b), \tag{3.50}$$

де $\mathrm{F}(\chi_b)$ – значення пристосованості конфігурації χ_b;

$F_1(\chi_b)$ – F_1-міра, обчислена на валідаційному періоді;

$\nu_{FP}(\chi_b)$ – частка хибнопозитивних спрацювань;

$\nu_{CV}(\chi_b)$ – коефіцієнт варіації потоку тривог;

$\Gamma(\chi_b)$ – нормована складність моделі;

$\alpha, \beta, \gamma \geq 0$ – коефіцієнти ваг штрафів.

Коефіцієнти α, β, γ дозволяють налаштовувати політику оптимізації відповідно до пріоритетів організації. Для середовищ з обмеженими ресурсами на реагування збільшують α для жорсткішого обмеження хибних тривог. Організації з нерівномірним навантаженням на команду безпеки підвищують β для стабілізації потоку сповіщень. Вимоги до інтерпретованості відображаються через γ.

Волатильність потоку тривог вимірюємо коефіцієнтом варіації – відношенням стандартного відхилення до середнього:

$$\nu_{CV} = \frac{\sigma_A}{\bar{A}+\varepsilon}, \tag{3.51}$$

де σ_A – стандартне відхилення кількості тривог по вікнах за період оцінювання;

$\bar{A}$ – середня кількість тривог на вікно;

ε – мала константа для уникнення ділення на нуль.

Низьке значення ν_{CV} означає рівномірний потік тривог без різких піків і провалів, що полегшує планування роботи аналітиків. Високе значення вказує на нестабільність детектора, який то генерує хвилі сповіщень, то затихає.

Складність моделі нормуємо як частку активних ознак від загальної кількості:

$$\Gamma(\chi_b) = \frac{1}{m}\sum_{i=1}^{m} b_i, \tag{3.52}$$

де $\Gamma(\chi_b) \in [0,1]$ – нормована складність;

m – загальна кількість доступних ознак.

Значення $\Gamma = 0{,}3$ означає, що модель використовує лише 30 % доступних ознак, що спрощує інтерпретацію та знижує ризик перенавчання на шумі.

Генетичний алгоритм ініціалізує популяцію випадкових хромосом і ітеративно покращує їх через цикл відбору, схрещування та мутації. На кожній ітерації (поколінні) хромосоми оцінюються функцією пристосованості, найкращі особини з більшою ймовірністю обираються для розмноження, нащадки успадковують

характеристики батьків із випадковими варіаціями. Алгоритм роботи поведінкового детектора зображено на рисунку 3.4.

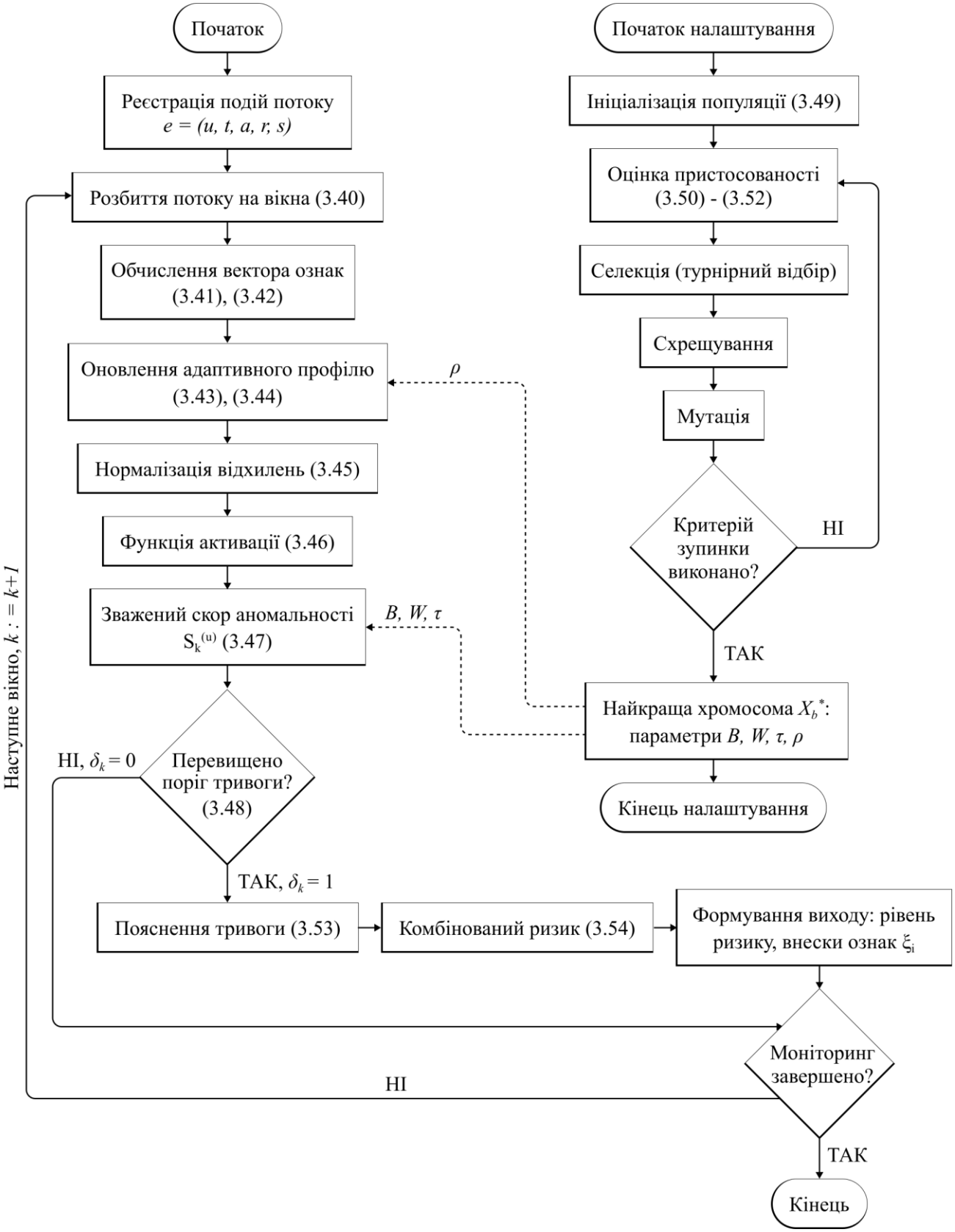


Рисунок 3.4 – Алгоритм поведінкового профілювання на основі генетичного алгоритму

Поведінковий детектор, налаштований генетичним алгоритмом, призначений не для заміни контентного аналізу, а для його доповнення. Обидва підходи виявляють різні аспекти загроз: контентний аналіз ідентифікує конфіденційні дані в потоках інформації, поведінковий – аномальні закономірності їх використання. Інтеграція цих сигналів дозволяє формувати більш обґрунтовані рішення про рівень ризику.

Одним із практичних застосувань поведінкової оцінки є пояснення спрацювань. Оскільки загальний показник є лінійною комбінацією внесків окремих ознак, для кожної тривоги можна визначити, які саме аспекти поведінки спричинили реакцію детектора:

$$\xi_i^u(k) = \frac{b_i \cdot w_i \cdot \varphi\left(z_i^u(k)\right)}{S^{(u)}(k)+\varepsilon}, \tag{3.53}$$

де $\xi_i^u(k)$ – внесок i-ї ознаки у загальний показник аномальності користувача u у вікні k і частки внеску нормуються на суму зважених активацій усіх ознак, тому їх сума дорівнює одиниці;

b_i – індикатор включення i-ї ознаки в модель зі значенням 0 або 1;

w_i – вага ознаки;

$\varphi\left(z_i^u(k)\right)$ – активоване значення z-оцінки за формулою (3.46);

$z_i^u(k)$ – сама z-оцінка i-ї ознаки;

$S^{(u)}(k)$ – загальний показник аномальності;

ε – мала стала, що запобігає діленню на нуль.

Ця інформація надає аналітику чітке пояснення: “тривога викликана перевищенням норми файлових копіювань на 3,2 стандартного відхилення (внесок 45 %) та нетиповою мережною активністю (внесок 35 %)”. Така прозорість прискорює процес розслідування та підвищує довіру до системи.

Для інтеграції з контентним аналізом пропонується комбінований показник ризику, що поєднує обидва сигнали:

$$R_{comb} = \lambda \cdot S_{cont} + (1-\lambda) \cdot S_{beh}, \tag{3.54}$$

де R_{comb} – комбінований показник ризику;

S_{cont} – нормалізований показник контентного детектора;

S_{beh} – нормалізований поведінковий показник;

$\lambda \in [0,1]$ – коефіцієнт балансу між компонентами.

Коефіцієнт λ може бути фіксованим або адаптивним залежно від контексту. Для операцій з документами високого рівня конфіденційності більша вага надається контентній оцінці; для загальних шаблонів активності – поведінковій. Комбінований показник дозволяє ранжувати інциденти за рівнем ризику та пріоритезувати реагування.

Обробка поведінкових метаданих у корпоративному середовищі базується передусім на законних інтересах роботодавця відповідно до Загального регламенту про захист даних (GDPR) [115]. Цей інтерес полягає в захисті інформаційних активів та запобіганні витокам і має бути збалансований із розумними очікуваннями працівника щодо приватності. GDPR закріплює принцип мінімізації даних. Обробці підлягає лише те, що справді потрібне для визначеної мети. На рівні національного законодавства збір і обробку персональних даних (ПД) регулює Закон України «Про захист персональних даних» [116]. Окрему роль відіграє прозорість. Працівники мають бути завчасно поінформовані про факт і межі моніторингу через внутрішні політики організації.

Низка правових вимог [115; 116] реалізована безпосередньо в методі. Профіль зберігає агреговані статистики, а не послідовність первинних дій користувача. Завдяки цьому агрегування є технічним втіленням принципу мінімізації: відновити з профілю конкретну дію в конкретний момент часу неможливо. Ідентифікатори користувачів піддають псевдонімізації, доступ до профілів розмежовують за ролями, а строк зберігання статистик обмежують і прив'язують до тривалості вікна спостереження W.

Не менш суттєво окреслити межі того, що система свідомо залишає поза увагою. Вона не аналізує вміст особистих повідомлень, не відстежує активність працівника поза корпоративними ресурсами й не досліджує приватне листування. Спостереження обмежено метаданими корпоративної активності без проникнення у зміст комунікації.

Практичне впровадження методу потребує врахування кількох аспектів. Ініціалізація профілів нових користувачів вимагає періоду накопичення даних (зазвичай три-чотири тижні) перед активацією виявлення аномалій. Сезонні закономірності (відпустки, святкові періоди, квартальна звітність) можуть потребувати окремого моделювання або тимчасового підвищення порогів. Зміни

організаційної структури (злиття відділів, масове наймання) є джерелом колективного дрейфу, який може викликати хвилі хибних тривог при недостатній швидкості адаптації профілів.

Поки історія спостережень за користувачем u надто коротка, оцінки μ_i^u і σ_i^u лишаються нестійкими, і застосування z-оцінки за формулою (3.45) до такого профілю давало б високу частку хибних тривог. Відхилення вимірювалося б відносно ще не сформованого еталона. Тому детекцію для актора доцільно вмикати не за фіксованою кількістю днів, а за досягненням статистичної надійності оцінок.

За критерій надійності зручно взяти ширину довірчого інтервалу для оцінки середнього. За N вікнами спостереження напівширина довірчого інтервалу для μ_i^u становить $z_{1-\frac{\alpha}{2}} \cdot \frac{\sigma_i^u}{\sqrt{N}}$, де $z_{1-\frac{\alpha}{2}}$ це квантиль стандартного нормального розподілу для рівня довіри (1−α). Детекція за ознакою і активується, коли ця напівширина стає достатньо малою відносно масштабу ознаки c_i. Оскільки профіль будується експоненційним зважуванням, то замість календарної кількості вікон N потрібно використовувати ефективний розмір вибірки $N_{eff} = (\Sigma w)^2/\Sigma w^2$, який за сталого ρ насичується величиною (1+ρ)/(1-ρ), тому оцінка у 20-30 вікон є орієнтовною і потребує емпіричної перевірки для конкретних значень ρ та ознак. Задамо співвідношення між ознаками так:

$$z_{1-\frac{\alpha}{2}} \cdot \frac{\sigma_i^u}{\sqrt{N}} \leq \varepsilon_{CI} \cdot c_i, \tag{3.55}$$

де $z_{1-\alpha/2}$ – квантиль стандартного нормального розподілу для рівня довіри $1-\alpha$;

α – рівень значущості;

σ_i^u – стандартне відхилення i-ї ознаки користувача u;

N – кількість вікон спостереження;

ε_{CI} – гранична відносна ширина довірчого інтервалу;

c_i – характерний масштаб i-ї ознаки, ліва частина нерівності є напівшириною довірчого інтервалу для середнього.

Звідси випливає мінімальна кількість вікон для активації за цією ознакою:

$$N \geq \left(z_{1-\frac{\alpha}{2}} \cdot \frac{\sigma_i^u}{\varepsilon_{CI} \cdot c_i}\right)^2.$$

Профіль вважаємо сформованим, а детекцію активованою, коли умова виконується для всіх (або заздалегідь визначеної частки) ознак, тобто за досягнення $N \geq N_{min}$. Конкретне значення N_{min} залежить від обраних α та ε_{CI} і від мінливості ознак, однак для типових налаштувань воно орієнтовно відповідає 20-30 вікнам спостереження, що за прийнятої тривалості вікна Δ дає згадані три-чотири тижні накопичення даних.

Поки профіль користувача не сформовано, система не залишається без захисту. Замість індивідуальних порогів застосовують консервативні загальнорольові. Запасний варіант із рольовим профілем, тобто усередненими статистиками (μ, σ) за роллю чи підрозділом користувача, що дозволяє виявляти грубі аномалії вже з перших вікон.

Обчислювальна ефективність методу забезпечується інкрементною природою оновлень. Кожне вікно потребує лише застосування рекурентних формул (3.43)-(3.44) та обчислення оцінки (3.47), без перерахунку всієї історії. Генетична оптимізація виконується періодично на основі накопичених даних і не впливає на продуктивність оперативного виявлення.

Розроблений метод поведінкового профілювання має кілька відмінних рис порівняно з наявними підходами до виявлення аномалій. Генетична оптимізація конфігурації детектора через хромосомне кодування маски активних ознак, ваг та порогу автоматизує процес, який у більшості систем поведінкової аналітики користувачів та сутностей (UEBA) виконується вручну. Механізм експоненційного забування в адаптивному профілі (формули (3.43)-(3.44)) дозволяє коректно обробляти легітимні зміни поведінки – зміну проєкту, переведення на іншу посаду, сезонні коливання активності. Функція пристосованості (3.50) містить штрафи за частоту хибних спрацювань, нестабільність потоку тривог та складність моделі, що забезпечує багатокритеріальну оптимізацію з урахуванням експлуатаційних вимог. Агрегація скорів через зважену суму (3.47) із бінарною маскою включення зберігає лінійну пояснюваність кожного рішення: для будь-якої тривоги система вказує конкретні ознаки та їхній відносний внесок. Комбінований показник ризику (3.54), що інтегрує контентну та поведінкову складові, забезпечує цілісну оцінку загрози. Інкрементальна природа оновлень профілю забезпечує обчислювальну ефективність, достатню для роботи в реальному часі з тисячами користувачів

3.4 Висновки до третього розділу

Розроблено метод класифікації документів за рівнем конфіденційності на основі генетичного алгоритму, в якому хромосома кодує набір продукційних правил виду «IF-THEN» із умовами на ознаки контентного та метаданого походження. Функція пристосованості поєднує F1-міру з штрафом за складність, що забезпечує регуляризацію та запобігає надмірному розростанню набору правил. Спеціалізовані генетичні оператори (мутація порогів умов, структурні мутації додавання й вилучення умов і правил та схрещування на рівні цілих правил) разом із комбінованою ініціалізацією популяції на основі статистик розподілу ознак за класами забезпечують ефективну збіжність алгоритму. Принципова перевага методу полягає в інтерпретованості результату: кожне класифікаційне правило може бути прочитане, верифіковане та за потреби скориговане експертом із безпеки, що відрізняє його від непрозорих моделей машинного навчання і відповідає вимогам аудитованості, закладеним в узагальненій моделі ЗВД-системи. Метод класифікації документів на основі генетичного алгоритму становить другий науковий результат дисертації.

Розроблено метод еволюційної адаптації політик безпеки в умовах дрейфу концепції, який моделює зміну спільного розподілу даних у часі та оптимізує п’ять цільових функцій: ризик пропущеного витоку, навантаження від хибних спрацювань, латентність обробки, складність політик та стабільність потоку сповіщень. Гібридна хромосома поєднує фіксовану частину порогових значень і гіперпараметрів із змінною частиною продукційних правил політики. Виявлення дрейфу реалізовано через моніторинг ентропії невизначеності моделі та критерій Колмогорова-Смирнова з механізмом підтвердження, що перемикає алгоритм між режимами експлуатації та дослідження з відповідною зміною інтенсивності мутацій, розміру турніру та складу популяції. Архів успішних політик із контекстними характеристиками дає змогу повторно використовувати раніше знайдені рішення для схожих епізодів дрейфу, а ансамблеве агрегування активних політик із мультиплікативним оновленням ваг і консервативним прийняттям рішень для критичних подій підвищує стійкість системи в перехідні періоди.

Розроблено метод поведінкового профілювання користувачів на основі генетичного алгоритму, що формує адаптивний профіль із експоненціальним забуванням шляхом рекурсивного обчислення математичного сподівання та дисперсії поведінкових ознак у фіксованих часових вікнах. Відхилення від індивідуального базового рівня нормалізуються за z-оцінкою з активацією, що враховує лише перевищення норми, а зважена оцінка аномальності агрегує внески окремих ознак з оптимізованими генетичним алгоритмом масками, вагами, порогом і параметром забування. Багатокритеріальна функція пристосованості штрафує за хибні спрацювання, нестабільність сповіщень та надлишкову складність, що спрямовує еволюційний пошук до збалансованих конфігурацій детектора. Обчислення внеску кожної ознаки в підсумкову оцінку забезпечує пояснюваність рішень для аналітиків безпеки, а інтеграція результатів контентного та поведінкового аналізу через комбіновану оцінку ризику замикає архітектуру ЗВД-системи, поєднуючи класифікацію документів із моніторингом дій користувачів. Метод еволюційної адаптації політик становить третій, а метод поведінкового профілювання користувачів – четвертий науковий результат дисертації.

Основні результати розділу опубліковані у працях [104; 105; 106; 107; 108; 112].

РОЗДІЛ 4
ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ МЕТОДІВ ВИЯВЛЕННЯ ВИТОКІВ ДАНИХ

4.1 Методика експериментального дослідження

Методику дослідження утворюють три складники: система показників, за якими оцінюють методи; набори даних, що відтворюють істотні властивості задачі виявлення витоків; програмне середовище, у якому виконано експерименти.

Якість класифікації та виявлення аномалій оцінюємо за показниками, що спираються на чотири величини матриці невідповідностей: кількість істинно позитивних рішень TP, хибно позитивних FP, хибно негативних FN та істинно негативних TN. Точність відображає частку справді конфіденційних подій серед усіх, які система позначила небезпечними:

$$Pr = \frac{TP}{TP+FP}, \tag{4.1}$$

де TP – кількість правильно виявлених конфіденційних подій;

FP – число хибних спрацювань, тобто легітимних подій, помилково позначених небезпечними.

Повнота відображає частку виявлених конфіденційних подій серед усіх, що справді є небезпечними:

$$Re = \frac{TP}{TP+FN}, \tag{4.2}$$

де FN – число пропущених конфіденційних подій.

Ці показники конкурують: посилення контролю підвищує повноту, але знижує точність, і навпаки. Точність відповідає навантаженню від хибних тривог, а повнота – ціні пропуску витоку, причому вартісну асиметрію цих помилок, де ціна пропуску значно перевищує ціну хибної тривоги, формалізовано функцією втрат (2.37). Зведеною мірою, що балансує обидва аспекти, слугує F_1 – їх гармонічне середнє:

$$F_1 = \frac{2 \cdot Pr \cdot Re}{Pr+Re}, \tag{4.3}$$

де F_1 – гармонічне середнє точності та повноти;

Pr – точність, тобто частка конфіденційних подій серед позначених системою;

Re – повнота, тобто частка виявлених витоків.

Саме F_1 обрано основним показником якості: за дисбалансу класів, типового для документообігу, де конфіденційні документи становлять меншість, частка правильних відповідей неінформативна, бо досягається тривіальним віднесенням усіх подій до легітимних. Для аналізу хибних рішень додатково використовуємо частку хибних спрацювань FPR та частку пропусків FNR:

$$FPR = \frac{FP}{FP+TN}, \quad FNR = \frac{FN}{TP+FN} = 1 - Re, \tag{4.4}$$

де FPR – частка хибних спрацювань;

FP – кількість хибнопозитивних рішень;

TN – кількість істиннонегативних рішень;

FNR – частка пропущених витоків;

FN – кількість хибнонегативних рішень;

TP – кількість істиннопозитивних рішень;

Re – повнота.

Для потокових експериментів, де модель безперервно оновлюється, застосовуємо преквенціальне оцінювання [114] за схемою «перевірка-потім-навчання»: кожен зразок спершу класифікують поточною моделлю, а лише потім використовують для її оновлення. Поточну якість у потоці вимірюємо ковзною преквенціальною точністю на вікні W:

$$Acc_{preq}(t) = \frac{1}{|W|}\sum_{i \in W} 1[\hat{y}_i = y_i], \tag{4.5}$$

де $\hat{y}_i$ – прогноз моделі для i-го зразка, зроблений до його використання в навчанні;

W – поточне вікно оцінювання.

Така оцінка відображає якість системи саме в момент надходження події й не потребує окремої тестової вибірки.

Роботу детектора дрейфу оцінюємо за точністю та повнотою виявлення точок зміни концепції, а також за затримкою виявлення – середньою кількістю зразків між істинною точкою дрейфу та підтвердженою тривогою:

$$D_{det} = \frac{1}{|A_{tp}|}\sum_{a \in A_{tp}} \big(pos(a) - pos^*(a)\big), \tag{4.6}$$

де A_{tp} – множина істинних тривог;

$pos(a)$ – позиція тривоги в потоці;

$pos^*(a)$ – позиція відповідної істинної точки дрейфу.

Інтерпретованість класифікатора вимірюємо трьома показниками: кількістю активних правил у підсумковому наборі; середньою кількістю умов у правилі; часткою позитивних рішень, які пояснюються одним-двома правилами. Менші значення перших двох і більше значення третього свідчать про компактнішу й прозорішу модель.

Набори даних, використані в експериментах, охоплюють дві сфери, які цільова система запобігання витокам даних мусить обслуговувати одночасно: розпізнавання конфіденційного вмісту в текстах і виявлення аномальної поведінки та зміни умов роботи в потоках подій. Замість штучно згенерованих колекцій усі експерименти спираються на публічно доступні реальні корпуси, що дає змогу оцінити метод в умовах, наближених до практичних, зокрема з притаманними реальним даним шумом, дисбалансом класів і нерівномірним розподілом ознак.

Для задачі класифікації документів за рівнем конфіденційності використано корпус ai4privacy PII-Masking-200k, який містить природномовні тексти з поелементною розміткою типів персональних даних [117]. Документ вважали конфіденційним, якщо він містив хоча б один критичний ідентифікатор: номер платіжної картки, банківський рахунок або IBAN, номер соціального страхування, пароль чи адресу криптогаманця; решту текстів віднесено до неконфіденційних. З повного корпусу відібрано збалансовану за складністю вибірку з 4000 документів, у якій частка конфіденційних становить близько 30 %. Такий поділ безпосередньо відтворює типовий сценарій ЗВД, коли систему цікавить не наявність персональних даних узагалі, а витік саме тих відомостей, розкриття яких завдає прямої фінансової чи правової шкоди.

Другий текстовий набір, 20 Newsgroups, є класичною колекцією повідомлень із тематичних груп новин [118] і дозволяє перевірити метод на конфіденційності іншого роду, яка визначається не окремим ідентифікатором, а темою тексту загалом. До чутливих віднесено повідомлення з груп, присвячених криптографії, медицині та володінню зброєю, тоді як нейтральні теми утворили негативний клас. Обсяг вибірки й частка позитивного класу узгоджені з попереднім набором: 4000 документів за приблизно 30 % чутливих. Тематична конфіденційність складніша для формальних

правил, тому цей корпус слугує навмисно суворою перевіркою меж інтерпретованого підходу.

Поведінкове профілювання оцінювали на корпусі CERT Insider Threat r4.2, який відтворює журнали корпоративної активності великої організації [119]. Він охоплює 1000 користувачів упродовж 501 дня, з яких 70 є інсайдерами, чиї дії позначено як зловмисні на рівні окремих днів; сукупно таких зловмисних «користувач-день» налічується 1364. Первинні журнали агрегували до рівня «користувач-день», формуючи одинадцять ознак, що описують щоденну активність, серед них кількість входів у систему, підключень USB-носіїв, операцій копіювання файлів, надісланих листів та їхній сукупний обсяг. Такий набір цінний тим, що поєднує реалістичну структуру подій з екстремальним дисбалансом класів, за якого зловмисна активність становить мізерну частку від загального обсягу, а це напряму впливає на вибір показників оцінювання.

Експерименти з виявленням дрейфу концепції спираються на потоки двох типів. Для контрольованої перевірки використано стандартні потоки на основі генератора RandomRBF бібліотеки river [122], реальні набори там, де потрібен повний контроль над властивостями задачі. Для сценарію керованості політики використано корпус ділових документів з композиційною генерацією: лексика обох класів збігається, конфіденційність визначають лише критичні ідентифікатори з валідацією контрольних сум, а кожному документу нового типу відповідає близнюкова приманка з тим самим формулюванням без валідного ідентифікатора. Для оцінки адаптації політик додатково використано потік подій, концепція якого має структуру політики ЗВД, тобто диз'юнкції кон'юнктивних умов над ознаками ризику, з відомими позиціями двох раптових і одного поступового дрейфу, у яких задано дрейф зі зміною концепту (зсув розподілу ознак разом із прив’язаним до центроїдів правилом класифікації) і точно відомими позиціями змін, що дозволяє однозначно вимірювати затримку виявлення та кількість хибних спрацювань. Реальний бік задачі представлено сенсорними потоками INSECTS з відомими точками зміни концепції [120], де дрейф має неперервний характер і супроводжується високою розмірністю ознак. Окремо для експерименту з адаптацією політик залучено потік Electricity (Elec2), що фіксує коливання попиту на електроенергію [121] і містить природну нестаціонарність, не пов'язану зі штучно вставленими змінами. Два згенеровані

ресурси доповнюють Статус наборів даних такий: корпус ai4privacy PII-Masking-200k згенеровано синтетично з подальшою ручною валідацією, потоки RandomRBF також синтетичні; реальними є корпус CERT r4.2 і потоки Elec2 та INSECTS, тож формулювання про «реальні дані» стосуються саме них. Тривоги детекторів дрейфу зіставляються зі справжніми точками змін у вікні толерантності (500, 1200 або 2000 зразків залежно від потоку). Повторні тривоги в межах вікна зараховуються тому самому дрейфу, тому наведені показники не є взаємно однозначним зіставленням і чутливі до ширини вікна. Для потокових експериментів основною метрикою є преквенціальна точність. Порівняння стратегій ведеться на тому самому потоці, що усуває вплив дисбалансу класів на відносні висновки.

Прототип реалізовано мовою Python. Класичні методи класифікації та показники якості взято з бібліотеки scikit-learn, а потокові детектори дрейфу з бібліотеки river, що забезпечує зіставлення запропонованого методу з усталеними реалізаціями базових підходів. Відтворюваність результатів гарантовано фіксацією початкових зерен генераторів випадкових чисел. Для задач класифікації застосовано стратифіковану перехресну валідацію на п'яти блоках, тоді як для потокових задач показники усереднювали за незалежними прогонами або окремими потоками, що згладжує вплив випадкових коливань конкретного запуску.

4.2 Експериментальне дослідження запропонованих методів

Поєднаємо чотири взаємопов’язані блоки експериментів: класифікацію документів за рівнем конфіденційності, поведінкове профілювання користувачів, виявлення дрейфу концепції з еволюційною адаптацією політик та наскрізний приклад функціонування системи.

Мета першого експерименту полягала в перевірці того, чи здатний класифікатор на продукційних правилах, який побудований еволюційним пошуком, зрівнятися з поширеними методами машинного навчання за якістю розпізнавання і водночас дати те, чого вони не дають, а саме прозорий та придатний для аудиту опис прийнятого рішення. Для порівняння відібрано п'ять методів, що добре зарекомендували себе в задачах текстової класифікації: випадковий ліс, градієнтний бустинг, лінійний SVM, логістичну регресію, наївний баєсів класифікатор. Усі вони,

разом із запропонованим методом, працювали в єдиному ознаковому просторі, що поєднує 300 ознак TF-IDF із шаблонними ознаками критичних ідентифікаторів, визначеними за моделлю розділу 2. Оцінювання проводилося за п'ятиблоковою стратифікованою перехресною валідацією на двох реальних корпусах: ai4privacy PII-Masking-200k [117] і 20 Newsgroups [118].

На корпусі ai4privacy конфіденційним вважався документ, що містить критичний ідентифікатор: номер платіжної картки, банківський рахунок чи IBAN, номер соціального страхування, пароль або адресу криптогаманця. Результати цього блоку зведено в таблиці 4.1. Найвищу якість показав випадковий ліс (F1 = 0,856), близькі значення дали лінійний SVM (0,850), логістична регресія (0,845) і градієнтний бустинг (0,840). Запропонований класифікатор на правилах досяг F1 = 0,826 за точності 0,904 і повноти 0,761, поступившись найкращому методу приблизно на 0,03. Цей розрив цілком очікуваний і має зрозуміле пояснення. Порогові правила добре вловлюють концентровані сигнали, як-от спрацювання шаблону картки чи номера соцстрахування, проте огрублюють слабкі, розподілені по багатьох ознаках свідчення, які ансамблеві методи агрегують поступово. Там, де рішення тримається на дрібних внесках десятків ознак TF-IDF, дискретизація в кілька умов неминуче втрачає частину інформації. Втім, ця втрата компенсується головною перевагою методу: замість непрозорого ансамблю з сотень дерев отримуємо компактний набір правил придатний до аудиту.

Таблиця 4.1 - Результати класифікації на корпусі ai4privacy (5-блокова перехресна валідація, середнє ± с.в.)

Метод	F1	Точність	Повнота	Час навчання, с
Запропонований метод (ГА-правила)	0,826 ± 0,018	0,904	0,761	30,04
Випадковий ліс	0,856 ± 0,015	0,931	0,793	1,22
Градієнтний бустинг	0,840 ± 0,019	0,925	0,770	1,12
Лінійний SVM	0,850 ± 0,009	0,911	0,796	0,01
Логістична регресія	0,845 ± 0,008	0,934	0,772	0,02
Наївний баєсів класифікатор	0,819 ± 0,008	0,889	0,761	0,00

Для 20 Newsgroups конфіденційними позначено дописи з тем, що торкаються криптографії, медицини та зброї, на противагу нейтральним групам. Як відображено в таблиці 4.2, загальний рівень якості нижчий для всіх методів: лінійний SVM досяг F1 = 0,703, наївний баєсів класифікатор 0,675, логістична регресія та випадковий ліс по 0,664, градієнтний бустинг 0,605. Запропонований метод отримав F1 = 0,591, і відставання тут помітно більше, ніж на ai4privacy. Причина криється в самій природі задачі. Тематична належність тексту не зводиться до кількох маркерів, вона проявляється через розподілену лексику, де сигнал розмазаний по сотнях слів, кожне з яких окремо малоінформативне. Такий сигнал погано стискається в кілька порогових умов, і саме тут перевага лінійних моделей, що зважують усі ознаки одночасно, стає найвідчутнішою. Прикметно, що класифікатор на ГА-правилах тяжіє до високої повноти (0,805) за низької точності (0,468). Правила спрацьовують широко, але недостатньо вибірково.

Таблиця 4.2 - Результати класифікації на корпусі 20 Newsgroups (тематична конфіденційність)

Метод	F1	Точність	Повнота	Час навчання, с
Запропонований метод (ГА-правила)	0,591 ± 0,013	0,468	0,805	20,91
Випадковий ліс	0,664 ± 0,021	0,828	0,557	1,50
Градієнтний бустинг	0,605 ± 0,020	0,873	0,464	1,61
Лінійний SVM	0,703 ± 0,022	0,767	0,650	0,02
Логістична регресія	0,664 ± 0,027	0,835	0,553	0,07
Наївний баєсів класифікатор	0,675 ± 0,019	0,736	0,627	0,00

Особливістю розробленого методу є інтерпретованість класифікатора, показники якої, отримані з аналізу структури моделей, подано в таблиці 4.3. На корпусі ai4privacy набір складався в середньому з 4,8 правила по 1,1 умови на правило, на 20 Newsgroups - з 4,6 правила по 2,2 умови. Це означає, що кожне позитивне рішення відтворюється переліком правил, що спрацювали, і людина-аудитор може простежити всю логіку класифікації за лічені хвилини. Частка позитивних рішень, які пояснюються не більш ніж двома правилами, становить 0,99

на корпусі ai4privacy та 0,98 на 20 Newsgroups. Правила прямо посилаються на семантично осмислені детектори: "IF kw_credential > 0.304 THEN confidential", "IF pat_ssn > 0.0556 THEN confidential", "IF pat_card_like > 0.0972 THEN confidential". Замість абстрактних ваг ознак модель відтворює предметну область мовою, посилаючись на детектори облікових даних, номерів соціального страхування і платіжних карток. Саме така форма пояснення потрібна для регуляторного аудиту, де важливо не лише класифікувати документ, а й обґрунтувати підставу рішення.

Таблиця 4.3 - Показники інтерпретованості класифікатора

Показник	Значення
Кількість правил у наборі (ai4privacy)	4,8
Середня кількість умов у правилі (ai4privacy)	1,1
Кількість правил у наборі (20 Newsgroups)	4,6
Середня кількість умов у правилі (20 Newsgroups)	2,2
Час навчання ГА-правил проти випадкового лісу, с	30,0 проти 1,2
Частка позитивних рішень, пояснених не більш як двома правилами (ai4privacy)	0,99
Частка позитивних рішень, пояснених не більш як двома правилами (20 Newsgroups)	0,98

Вплив параметра регуляризації λ, що штрафує складність набору правил, досліджено окремо (табл. 4.4). На корпусі 20 Newsgroups нарощування λ послідовно скорочувало модель: від 11,7 правила при $\lambda = 0{,}0$ до 5,0 при 0,02, 3,0 при 0,05 і 1,3 при 0,2, причому якість спадала повільно (F1 від 0,589 до 0,556). Іншими словами, більшість правил можна прибрати ціною лише кількох сотих F1, отримавши натомість лаконічнішу і зручнішу для перегляду модель. За результатами цього аналізу як робоче значення для цього корпусу обрано $\lambda = 0{,}02$, що поєднує майже незмінну якість із компактним набором правил.

Робочі значення коефіцієнта регуляризації відповідають конфігурації програмного прототипу: $\lambda = 0{,}05$ для корпусу ai4privacy та $\lambda = 0{,}02$ для тематичного корпусу 20 Newsgroups.

Окремої уваги заслуговує обчислювальна вартість. Навчання запропонованого методу на корпусі ai4privacy зайняло близько 30,0 с проти приблизно 1,2 с у випадкового лісу (на 20 Newsgroups відповідно 20,9 с проти 1,5 с), тобто на порядок

довше. Ця різниця, однак, не є критичною для цільового сценарію застосування. Еволюційний пошук виконується один раз під час побудови політики класифікації, після чого отриманий набір правил застосовується до нових документів практично миттєво і без повторного навчання. Іншими словами, підвищена вартість навчання платиться разово і компенсується прозорістю та стабільністю моделі на етапі експлуатації, коли інтерпретованість рішень важить більше, ніж кілька зекономлених секунд на побудову.

Таблиця 4.4 - Вплив коефіцієнта регуляризації складності λ (корпус 20 Newsgroups, 3 блоки)

λ	F1 (середнє)	F1 (с.в.)	Кількість правил
0,00	0,589	0,008	11,7
0,02	0,587	0,010	5,0
0,05	0,575	0,011	3,0
0,10	0,566	0,007	2,0
0,20	0,556	0,011	1,3

Окремий експеримент кількісно оцінює керованість політики, тобто ту властивість, заради якої обрано подання знань у формі явних правил. Змодельовано типову для практики ситуацію. У документообігу з’являється новий тип критичного ідентифікатора (міжнародний номер банківського рахунку IBAN), для якого ще немає жодного розміченого прикладу. Навчальний корпус згенеровано без IBAN-позитивів, а в тестовому періоді третину конфіденційних документів визначає саме він. За принципом близнюкових приманок кожному IBAN-документу відповідає легітимний переказ з тим самим формулюванням і 27-цифровим розрахунковим референсом замість номера рахунку, тож відрізнити класи за загальними ознаками на предмет щільності цифр неможливо.

Реакція методів на цю ситуацію показова (табл. 4.5). До набору правил, синтезованого генетичним алгоритмом, експерт дописує одне правило про спрацювання детектора IBAN. Це займає хвилини, не потребує жодного розміченого документа і миттєво піднімає повноту на документах нового типу до 1,00 за загального F1 0,847. Без втручання випадковий ліс пропускає майже чотири з п’яти нових конфіденційних документів (повнота 0,23).

Таблиця 4.5 - Реакція на появу нового типу критичного ідентифікатора (3 незалежні прогони, середнє)

Модель	F1 до втручання	F1 після втручання	Повнота на нових документах	Вартість втручання
Запропонований метод (ГА-правила) + правило експерта	0,828	0,847	1,00	0 розмічених
Випадковий ліс	0,778	0,778	0,23	без втручання

Таким чином, перший експеримент дає зважену картину можливостей методу. Класифікатор на продукційних правилах не перевершує найкращі ансамблеві та лінійні методи за сирою якістю. На ai4privacy він відстає приблизно на 0,03 за F1, на 20 Newsgroups розрив ширший через розподілену природу тематичного сигналу. Проте там, де конфіденційність визначається наявністю чітких критичних ідентифікаторів, метод дає близьку до конкурентів якість і водночас забезпечує те, чого позбавлені решта підходів. Він компактний, семантично прозорий і придатний для аудиту набір правил, кожне з яких можна перевірити та обґрунтувати, а в разі появи нового типу ідентифікаторів політику можна виправити одним правилом експерта без збирання розмітки.

Суть методу поведінкового профілювання користувачів полягає в тому, що подія стає підозрілою не сама собою, а лише у зіставленні з усталеною нормою конкретної людини, і саме персональна точка відліку дозволяє вловити відхилення, які на популяційному рівні виглядають цілком буденно.

Перевірку проведено на відкритому корпусі CERT Insider Threat r4.2 [119], який містить журнали активності тисячі користувачів упродовж 501 дня; серед них 70 інсайдерів, а сумарно зафіксовано 1364 зловмисні пари «користувач-день» за одинадцятьма ознаками поведінки (кількість входів у систему, підключення USB-носіїв, копіювання файлів, надіслані листи, їхній обсяг та інші). Профілі будувалися на ранньому відрізку періоду спостереження, до 300-го дня, а тестування відбувалося на пізнішій частині, що відтворює реалістичний сценарій, коли модель нормальної поведінки формується заздалегідь і згодом застосовується до нових даних.

Генетичний алгоритм оптимізував конфігурацію персонального детектора, а саме: маску активних ознак, їхні ваги, поріг спрацювання та коефіцієнт забування, який визначає, наскільки швидко застаріває попередня історія користувача. Оцінювання свідомо винесено на рівень користувача через ранжування за піковою денною аномалією, оскільки аналітик безпеки на практиці працює саме зі списком найпідозріліших облікових записів, а не з окремими рядками журналу. За базу порівняння взято три поширені глобальні детектори аномалій: ізоляційний ліс, локальний фактор викиду (LOF) та однокласовий SVM.

Результати зведено в таблиці 4.6 і розрив між персональним профілюванням та глобальними методами виявився суттєвим саме там, де він має практичне значення, тобто у верхній частині рейтингу. На робочому бюджеті тривог у двадцять облікових записів запропонований ГА-профіль дає точність 0,90, а на п'ятдесяти зберігає 0,54. Середня влучність за всім ранжуванням становить AP=0,421. Для порівняння, найкращий із глобальних методів, однокласовий SVM, досягає лише AP=0,145 і точності 0,35 у топ-20, ізоляційний ліс дає AP=0,114 за точності 0,25, а LOF практично не виявляє інсайдерів у верхній частині списку (точність 0,05 на двадцяти і 0,02 на п'ятдесяти найпідозріліших). Таким чином, на однаковому операційному бюджеті персональний профіль ідентифікує приблизно у два з половиною рази більше справжніх порушників, ніж найсильніша глобальна база.

Причина цієї переваги лежить у самій природі того, що кожен із підходів вважає аномальним. Глобальні детектори позначають рядки, рідкісні для всієї популяції, наприклад, незвично велика для колективу кількість надісланих листів, нетипово об'ємне копіювання файлів чи рідкісний сценарій входу. Проте інсайдер із низькою інтенсивністю дій не породжує таких колективних викидів. Його зловмисна активність цілком укладається в діапазон, який звичний для тисячі облікових записів, і тому залишається непоміченою. Персональний профіль натомість вимірює відхилення від власної усталеної норми користувача і навіть помірне за абсолютною величиною зростання USB-підключень чи копіювань стає помітним, якщо для цього конкретного користувача воно нехарактерне. Показово, що серед активних ознак ГА-профілю опинилися саме кількість входів, підключення USB, копіювання файлів, надіслані листи та їхній обсяг, тобто ті канали, якими найчастіше й реалізується витік зсередини.

Оцінюючи результати, потрібно враховувати хибне значення одного з показників. Значення ROC-AUC для всіх методів залишаються низькими (0,466 у ГА-профілю, 0,452 в ізоляційного лісу, 0,561 в однокласового SVM, 0,262 у LOF) і на перший погляд свідчать про майже випадкову якість. Насправді за екстремального дисбалансу класів, коли інсайдери складають лічені відсотки популяції, ROC-AUC стає малоінформативною метрикою. Вона усереднює якість за всім діапазоном порогів, більшість з яких для завдання пошуку кількох порушників серед тисяч користувачів не має жодного практичного сенсу. Тому придатність методу коректно оцінювати за середньою влучністю та за точністю і повнотою на реальному бюджеті тривог, а не за площею під ROC-кривою. У цій системі координат персональне профілювання виконує своє призначення. Воно виявляє саме тих низькоінтенсивних інсайдерів, яких глобальні методи систематично пропускають, хоча слід визнати й обмеження підходу, адже абсолютні значення повноти (R@50=0,39, R@100=0,44) показують, що поза верхівкою рейтингу значна частина порушників усе ще залишається невиявленою.

Таблиця 4.6 - Виявлення інсайдерів на корпусі CERT Insider Threat r4.2 (оцінювання на рівні користувача)

Метод	AP	Точність top-20	Точність top-50	Повнота top-100
Запропонований метод (ГА-профіль)	0,421	0,90	0,54	0,44
Isolation Forest	0,114	0,25	0,22	0,17
LOF	0,067	0,05	0,02	0,04
One-Class SVM	0,145	0,35	0,24	0,23

Таким чином, експеримент з поведінковим профілюванням підтвердив головну перевагу індивідуальної норми поведінки. За реалістичного бюджету тривог ГА-профіль виявляє інсайдерів у кілька разів точніше за глобальні детектори аномалій, які порівнюють користувачів із загальнопопуляційним базовим рівнем. Водночас ROC-AUC, що усереднює ранжування за всім діапазоном частки хибних тривог, не характеризує роботу на малому бюджеті тривог, тому порівняння методів коректно вести за середньою влучністю та точністю на робочому бюджеті.

Виявлення концептуального дрейфу в неперервних потоках даних порушує припущення, на якому тримається будь-яка навчена наперед політика захисту, що статистичні властивості вхідних ознак залишаються сталими в часі. Запропонований детектор перевіряє це припущення багатоознаковим тестом Колмогорова-Смирнова, який зіставляє розподіл кожної ознаки на еталонному вікні з розподілом на поточному ковзному вікні. Оскільки одночасна перевірка багатьох ознак збільшує ймовірність хибного відхилення нульової гіпотези, то рівень значущості коригується поправкою Бонферроні. Ключовим елементом детектора є правило підтвердження, внесок якого експеримент оцінює окремо. Тривога про дрейф оголошується не за першим значущим сигналом, а лише після трьох послідовних значущих спрацювань. Така логіка полягає в тому, що поодинокий викид або короткочасне коливання розподілу не повинні змушувати систему перенавчатися. Справжній дрейф залишає стійкий слід у кількох послідовних вікнах.

Порівняння з тим самим детектором без правила підтвердження та з класичними методами ADWIN, DDM, EDDM і KSWIN проводилося спершу на стандартних потоках RandomRBF, згенерованих бібліотекою river [122] (десять потоків, по чотири концепти в кожному, три відомі позиції дрейфу на потік). Точно відомі позиції дрейфу дають змогу коректно виміряти повноту, затримку та кількість хибних тривог, що видно з таблиці 4.7. На раптовому дрейфі запропонований метод досягає F1=0,918 за повної повноти R=1,000, тоді як та сама схема без підтвердження дає лише F1=0,753. Різниця пояснюється майже цілком хибними тривогами: правило підтвердження зменшує їхню кількість із 2,1 до 0,6 на потік, зберігаючи здатність виявити всі справжні зсуви. Втрати за цю точність очікувані: затримка виявлення зростає до 275 зразків, бо детектор навмисно чекає на підтвердження, перш ніж діяти. На поступовому дрейфі картина зберігається: F1=0,932 за 0,5 хибних тривог на потік.

Відрив від класичних детекторів на цих потоках за раптового дрейфу великий. Найкращий із них, ADWIN, досягає лише F1=0,387, тоді як DDM, EDDM і KSWIN тримаються в межах від 0,136 до 0,193, причому EDDM додатково страждає від украй високої кількості хибних спрацювань (близько 20,9 на потік). Причина такого розриву методологічна: ADWIN, DDM та EDDM відстежують дрейф через частоту помилок класифікатора, тобто реагують лише тоді, коли зсув розподілу вже почав псувати

якість прогнозів. Прямий тест розподілу вхідних ознак спрацьовує раніше і не залежить від того, наскільки швидко деградує конкретна модель.

Таблиця 4.7 - Виявлення дрейфу концепції на стандартних потоках RandomRBF (10 потоків, середнє)

Детектор	F1 (раптовий)	Хибних (раптовий)	F1 (поступовий)	Хибних (поступовий)
Запропонований (KS + підтвердження)	0,918	0,6	0,932	0,5
KS без підтвердження	0,753	2,1	0,848	1,6
ADWIN	0,387	0,8	0,661	0,3
DDM	0,136	1,3	0,353	0,7
EDDM	0,193	20,9	0,264	19,1
KSWIN	0,169	1,0	0,213	1,0

Проте це не стосується всіх видів потоків. Перевірка на реальних сенсорних потоках INSECTS [120], де дрейф високовимірний і фактично неперервний, а не зосереджений у кількох чітких точках, показала значно гірші результати. Тут запропонований метод дає F1=0,232, і класичні детектори на потоці помилок виявляються конкурентними або кращими, а саме: KSWIN досягає F1=0,384, DDM 0,336, ADWIN 0,259. Коли розподіл ознак змінюється постійно і в багатьох вимірах водночас, багатоознаковий тест Колмогорова-Смирнова спрацьовує майже безперервно, а сама концепція дискретної події дрейфу з чіткою позицією втрачає сенс. Навіть у цьому несприятливому режимі правило підтвердження виконує свою функцію. Воно зменшує кількість хибних тривог з 23,0 до 15,5 на варіант потоку. Отже, внесок правила підтвердження підтверджується стабільно, тоді як загальна перевага прямого тесту розподілу над методами на потоці помилок обмежена сценаріями з відносно дискретними, а не розмитими подіями дрейфу.

Сигнал детектора дрейфу набуває практичної цінності лише тоді, коли система вміє на нього реагувати, і саме це перевіряє четвертий експеримент. Порівнювалися три стратегії керування політикою класифікації: статична політика, навчена одноразово, незмінна надалі. Запропонована адаптивна схема, яка за сигналом детектора перемикається між режимом експлуатації (використання наявних правил) і режимом дослідження (перегляд і перенавчання правил) та інкрементна логістична

регресія, що безперервно оновлює ваги з кожним новим зразком. Оцінювання велося за преквенціальною точністю, тобто кожен зразок спершу класифікується, а вже потім використовується для навчання, що коректно імітує роботу системи в реальному часі. У прототипі на цих загальних реальних потоках періодичне перенавчання за сигналом детектора виконується для компактного класифікатора (дерева рішень обмеженої глибини), що ізолює внесок самого механізму керованого перенавчання. Еволюційний класифікатор правил перенавчається на потоках зі структурою політики ЗВД.

Результати на реальних потоках (таблиця 4.8) надають змогу встановити, що реакція на дрейф справді окупається. На потоці Elec2 [121] з 45312 зразків адаптивна схема досягає преквенціальної точності 0,731 проти 0,686 у статичної політики, тобто відносний виграш становить 6,6 % ціною 56 перенавчань за весь потік. На потоці INSECTS-Abr із 52848 зразків розрив ще помітніший: 0,567 проти 0,525, приріст 8,0 % лише за 16 перенавчань. Рисунок 4.1 унаочнює механізм цього виграшу на прикладі згенерованого потоку зі структурою політики ЗВД: на графіку ковзної преквенціальної точності видно, як після кожної позначеної точки дрейфу крива статичної політики просідає й далі не відновлюється, тоді як адаптивна схема реагує перенавчанням і повертає точність до попереднього рівня.

Таблиця 4.8 - Преквенціальна точність стратегій адаптації

Стратегія	Elec2	INSECTS-Abr	Потік-політика
Статична політика	0,686	0,525	0,512
Адаптивна стратегія за сигналом детектора (запропонована)	0,731	0,567	0,851
Інкрементна логістична регресія	0,864	0,724	0,739

Третій потік доповнює картину випадком, найближчим до цільової задачі. Це згенерований потік подій, концепція якого має структуру політики ЗВД, тобто є диз’юнкцією кон’юнктивних умов над ознаками ризику, з двома раптовими і одним поступовим дрейфом у відомих позиціях. На п’яти незалежних потоках адаптивна стратегія дає преквенціальну точність 0,851 ± 0,013 проти 0,512 у статичної політики, тобто приріст становить 66,3 % вихідного рівня, і, що важливіше, випереджає інкрементну логістичну регресію (0,739) в усіх п’яти потоках. Пояснення структурне:

диз’юнкцію кон’юнкцій лінійна модель принципово апроксимує лише грубо, хоч би скільки оновлень ваг вона отримала, тоді як простір пошуку генетичного алгоритму збігається з класом концепцій потоку. Отже, на потоках загального вигляду інкрементні моделі беруть гору за точністю як такою, але на потоках, чия концепція має структуру політики захисту, а саме такими є правила промислових ЗВД-систем, запропонована схема виграє і за якістю, зберігаючи водночас інтерпретований набір правил.

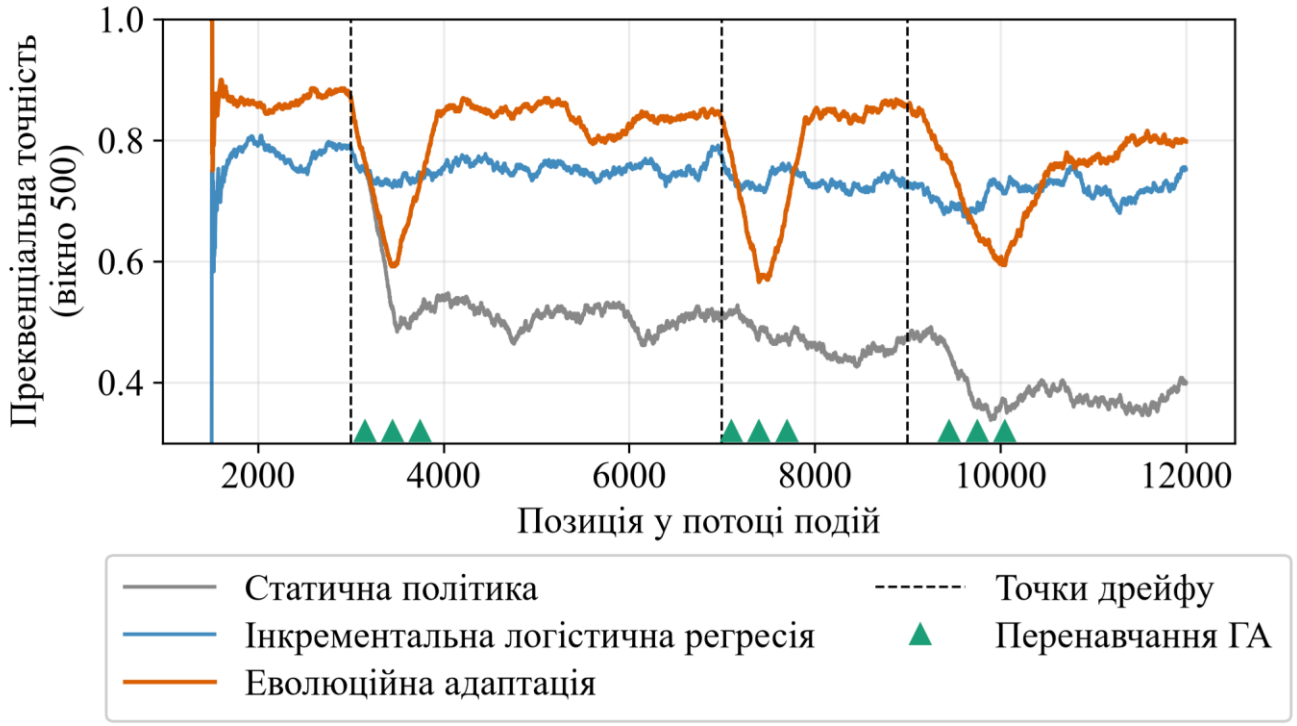


Рисунок 4.1 – Ковзна преквенціальна точність стратегій адаптації на згенерованому потоці зі структурою політики ЗВД; штриховими лініями позначено точки дрейфу, трикутниками – перенавчання ГА

На реальних потоках загального вигляду інкрементна логістична регресія демонструє найвищу сиру точність, 0,864 на Elec2 та 0,724 на INSECTS-Abr, і на перший погляд знецінює обидві попередні стратегії. Проте це порівняння оманливе, якщо зважити на призначення системи. Безперервне оновлення ваг лінійної моделі не залишає інтерпретованого набору правил, придатного для аудиту: рішення такої моделі не подаються у формі продукційних правил, які можна безпосередньо редагувати, верифікувати й оскаржувати як елементи політики. Коефіцієнти лінійної моделі пояснюють внески ознак, але редагованої політики не утворюють. Натомість

адаптивна схема зберігає компактний набір продукційних правил, і її виграш над статичною політикою досягається без відмови від прозорості. Тож вибір між двома підходами не зводиться до одного числа точності, а відображає компроміс між сирою якістю прогнозів та можливістю пояснити й перевірити кожне рішення.

Таким чином, внесок детектора окреслюється чітко, хоч і з застереженням. Правило підтвердження стабільно і суттєво зменшує кількість хибних тривог у всіх перевірених режимах, тоді як загальна перевага детектора над класичними методами прив'язана до сценаріїв з дискретними подіями дрейфу. Результат адаптації політик повніший. Адаптивне перемикання режимів дає стабільний приріст точності над статичною політикою на обох реальних потоках, а поступка інкрементній моделі за точністю як такою компенсується збереженням аудитованості, без якої система захисту конфіденційних даних втрачає практичну цінність.

На графіках функцій з рисунку 4.1 видно механізм переваги адаптивної стратегії. Після кожної позначеної штриховою лінією точки дрейфу точність статичної політики різко падає і вже не відновлюється, тоді як крива еволюційної адаптації після короткочасного провалу повертається на рівень близько 0,8-0,85 завдяки перенавчанням, моменти яких позначено трикутниками. Інкрементна логістична регресія зберігає стабільність упродовж усього потоку, проте на стаціонарних ділянках її крива проходить помітно нижче за криву адаптивної стратегії, оскільки лінійна модель лише грубо апроксимує концепцію зі структурою політики ЗВД.

Роботу системи в цілому ілюструє наскрізний приклад. Нехай співробітник надсилає зовнішньому адресатові документ, що містить рядок "payment approved, card 4808 4562 9023 5415, amount 2400 usd". Конвеєр обробки виділяє з тексту шаблонні ознаки: валідний за алгоритмом Луна номер платіжної картки та грошову суму, а також ключові слова платіжного контексту. На отриманому векторі спрацьовує правило "IF pat_card_like > 0.0972 THEN confidential", і документ дістає позначку конфіденційного, після чого модуль прийняття рішень блокує передачу і фіксує подію в журналі аудиту з переліком правил, що спрацювали.

Для повноти наведемо приклад набору правил, синтезованого генетичним алгоритмом на корпусі ai4privacy в одному з блоків перехресної валідації (умовні позначення ознак відповідають реалізації ознакового простору):

IF kw_credential > 0.304 THEN confidential

IF pat_crypto > 0.124 THEN confidential

IF pat_iban > 0.0399 THEN confidential

IF pat_ssn > 0.0556 THEN confidential

IF pat_card_like > 0.0972 THEN confidential

Таким чином, чотири блоки експериментів дають узгоджене подання спроможності розроблених методів. Класифікатор на еволюційно синтезованих правилах поступається найкращому ансамблевому методу лише приблизно на 0,03 за F1 там, де конфіденційність визначають критичні ідентифікатори, і виправляється одним правилом експерта без розмітки. Персональні поведінкові профілі виявляють інсайдерів приблизно у 2,5 раза точніше за глобальні детектори на робочому бюджеті тривог, правило підтвердження зменшує кількість хибних тривог детектора дрейфу в 3,5 раза, а еволюційна адаптація політик перевершує статичну на 6,6-8,0 % преквенціальної точності на реальних потоках і на 66,3 % на потоках зі структурою політики ЗВД.

4.3 Програмна реалізація ЗВД-системи та рекомендації щодо її застосування

Розроблені методи реалізовано у програмному комплексі запобігання витоку даних з еволюційною адаптацією на основі генетичних алгоритмів (ЗВДЕАГА), зареєстрованому як об’єкт авторського права [126]. Цей комплекс є програмними засобами. Комплекс написано мовою Python 3.12 і побудовано за модульним принципом, що безпосередньо відтворює функціональну структуру моделі ЗВД-системи. Модуль класифікації контенту реалізує ГА-класифікатор на продукційних правилах, модуль поведінкового аналізу підтримує індивідуальні адаптивні профілі користувачів, детектор дрейфу виконує багатоознаковий тест Колмогорова-Смирнова з правилом підтвердження, а модуль адаптації політик керує перемиканням між режимами експлуатації та дослідження з еволюційним перенавчанням набору правил. Окрему групу становлять модулі формування ознакового простору документів і завантаження корпусів та потоків даних, а сценарії експериментів відтворюють усі досліди цього розділу.

Прототип розгортається на виділеному сервері аналізу в сегменті засобів безпеки корпоративної мережі. Встановлення зводиться до створення віртуального середовища Python та інсталяції залежності однією командою з файлу requirements.txt. Сценарії запускаються як модулі з кореня репозиторію. У термінах моделі середовища компоненти комплексу логічно прив'язуються до точок контролю, зображених на рисунку 1.1. Модуль класифікації контенту обробляє документи, перехоплені на поштовому каналі (K1) та шлюзі безпеки (K2), модуль поведінкового аналізу споживає журнали подій робочих станцій і серверів, зокрема операції зі знімними носіями та друком (K3), а сканування сховищ (K5) постачає документи для первинного маркування. Для промислової інтеграції доцільно під'єднувати модуль класифікації до наявного проксі або поштового шлюзу через стандартні інтерфейси перехоплення вмісту, а модуль профілювання живити подіями із системи збирання журналів безпеки (SIEM).

Результати роботи комплекс зберігає у машиночитаних журналах. Кожен експериментальний сценарій формує файл результатів у форматі JSON у каталозі results/, а зведений текстовий дайджест дає змогу звірити всі числові значення розділу 4 з первинними даними. Журнал рішень класифікатора для кожної події містить перелік правил, що спрацювали, і саме це подає інтерпретованість системи аналітикові. Так, для події з підрозділу 4.2.4 журнал фіксує спрацювання правила IF pat_card_like > 0.0972 THEN confidential і рішення «блокувати», даючи точну відповідь, чому передачу зупинено. У сценарії появи нового типу ідентифікатора журнал показує, що повноту виявлення відновило саме дописане експертом правило IF pat_iban > 0 THEN confidential (таблиця 4.5). Структуру програмного комплексу, приклад журналу рішень із поясненням та коротку інструкцію користувача наведено в додатку Е, а опис відкритого репозиторію з повним вихідним кодом подано в додатку Д.

За результатами експериментів сформульовано практичні рекомендації щодо інтеграції еволюційно-адаптивних компонентів у наявні корпоративні системи безпеки. Класифікатор на продукційних правилах доцільно застосовувати там, де конфіденційність визначається розпізнаваними маркерами (платіжні реквізити, ідентифікатори особи, облікові дані) і де рішення системи мають бути придатними для аудиту й оскарження. За суто тематичної конфіденційності його варто поєднувати

з ансамблевою або лінійною моделлю у гібридній схемі, залишаючи правилам роль пояснюваного першого рубежу. Поведінкове профілювання потрібно розгортати з явним робочим бюджетом тривог, співмірним із пропускною здатністю аналітика (перші десять-двадцять користувачів у ранжуванні), а початкові профілі навчати на кількох тижнях історії подій. Детектор дрейфу з правилом підтвердження рекомендовано вмикати для потоків з відносно дискретними змінами концепції. За неперервного високо вимірного дрейфу доцільніше спиратися на моніторинг якості класифікації. Еволюційне перенавчання політик варто виконувати на окремому обчислювальному ресурсі, зберігаючи чинну політику до завершення валідації нової, що гарантує безперервність захисту, а швидкою реакцією на новий тип критичного ідентифікатора має бути дописане експертом правило, яке не потребує ні розмітки, ні перенавчання.

4.4 Обмеження дослідження

Обмеження запропонованого підходу проступають найвиразніше там, де інтерпретованість перестає бути вирішальною перевагою. На класифікації документів за тематичною ознакою (корпус 20 Newsgroups, де конфіденційність визначається належністю тексту до чутливих тем) класифікатор на продукційних правилах досягає F1 близько 0,591, тоді як лінійний метод опорних векторів сягає 0,703, а решта індуктивних моделей тримається в межах від 0,60 до 0,68 (таблиця 4.2). Схожа картина повторюється і на потоковій класифікації політик: на Elec2 адаптивна схема з перенавчанням компактного класифікатора (дерева рішень обмеженої глибини) дає преквенціальну точність 0,731 проти 0,864 в інкрементної логістичної регресії, а на INSECTS-Abr, відповідно, 0,567 проти 0,724. Різниця не випадкова: коли клас визначається розмитою комбінацією багатьох слабких ознак без чіткого порогового ідентифікатора, компактний набір із кількох правил неминуче огрублює межу рішення. Запропонований метод виграє натомість інтерпретованістю та придатністю до аудиту, і саме там, де ці властивості критичні (наприклад, коли рішення про конфіденційність має бути обґрунтованим і оскаржуваним), його поступка в точності як такій виправдана. На даних ai4privacy, де конфіденційність спирається на

присутність критичних ідентифікаторів, розрив звужується (F1 0,826 проти 0,856 у випадкового лісу), а самі правила залишаються прозорими та змістовними.

Друге обмеження стосується області застосовності детектора дрейфу. Побудований на багатоознаковому критерії Колмогорова-Смирнова (наборі одновимірних двовибіркових тестів з поправкою Бонферроні). Він виявляє дрейф розподілу вхідних ознак: на стандартних потоках RandomRBF правило з вторинним підтвердженням дає F1 0,918 за раптового і 0,932 за поступового дрейфу, з часткою хибних тривог нижчою за одну на потік, тоді як класичні детектори на потоці помилок (ADWIN, DDM, EDDM, KSWIN) залишаються далеко позаду. Проте ця перевага не переноситься на високовимірні неперервні сенсорні потоки: на реальних даних INSECTS запропонований детектор дає F1 лише 0,232, поступаючись KSWIN (0,384) та DDM (0,336). Коли розподіл ознак змінюється повільно й безперервно, а розмірність висока, то спостереження за потоком помилок класифікатора виявляється надійнішим сигналом, ніж прямий статистичний тест на вхідних ознаках. Отже, природна область застосування методу це дрейф розподілу вхідних ознак з відносно чіткими точками переходу; за неперервного дрейфу перевагою залишається лише менша кількість хибних спрацювань (правило підтвердження знижує їх з 23,0 до 15,5 на потоках INSECTS), що само по собі цінне, але вже не забезпечує переваги в повноті чи точності виявлення.

Оцінювання поведінкового профілювання має проблему через власну методологічну складність, притаманну самій задачі виявлення інсайдерів. У корпусі CERT r4.2 зловмисна активність становить частки відсотка від усього обсягу (70 інсайдерів на 1000 користувачів, 1364 зловмисні пари «користувач-день»). ROC-AUC усереднює якість ранжування за всім діапазоном частки хибних тривог і тому не характеризує операційну область малого бюджету тривог. Значення для всіх методів не перевищують 0,56 і не розрізняють їх у практично важливій зоні. Через це результати доводиться подавати на робочому бюджеті тривог, тобто на тій кількості користувачів, яку аналітик реально здатен перевірити. У цих термінах ГА-профіль показує точність 0,90 серед перших двадцяти підозрюваних і середню влучність 0,421, помітно випереджаючи Isolation Forest, LOF та One-Class SVM, проте таке порівняння прив'язане до конкретної організації, її розміру та структури активності. Перенесення цих показників на інші установи, з відмінними закономірностями

роботи й іншою базовою частотою загроз, потребує окремої перевірки, і робити з отриманих чисел висновок про універсальну ефективність було б передчасно. Слід ураховувати й асиметрію протоколу. Конфігурація ГА-профілю добирається за мітками навчального періоду, тоді як глобальні детектори працюють без міток і без відповідного налаштування, а мітка користувача обчислюється за всією часовою шкалою. Тому результат коректно трактувати як досяжну якість персоналізованого детектора, а не як порівняння за строго однакових умов.

Нарешті, обмеженням загального характеру є обчислювальна вартість еволюційного пошуку. Навчання класифікатора на правилах займає близько 30 секунд проти 1,2 секунди у випадкового лісу на тому самому корпусі ai4privacy, тобто приблизно на порядок довше, і ця різниця зумовлена самою природою популяційного пошуку з багаторазовим оцінюванням пристосованості. Для одноразового навчання чи періодичного перенавчання така вартість цілком прийнятна, однак вона обмежує застосовність методу в сценаріях із жорсткими вимогами до частоти оновлення моделі або з дуже великими обсягами навчальних даних. Ця вартість є ціною, яку метод віддає за здобуту інтерпретованість, і в кожному конкретному застосуванні співвідношення цих двох властивостей слід зважувати окремо.

4.5 Висновки до четвертого розділу

У четвертому розділі розроблені методи експериментально перевірено на відкритих корпусах ai4privacy (синтетично згенерованому і валідованому експертами) та 20 Newsgroups, реальних наборах CERT Insider Threat r4.2, Elec2 та INSECTS і стандартних синтетичних потоках RandomRBF. Генетичний класифікатор підтвердив конкурентну якість там, де конфіденційність визначається критичними ідентифікаторами (F1 = 0,826 проти 0,856 у найкращого ансамблевого методу за компактного набору із п’яти правил, кожне з яких піддається аудиту), а дописане експертом правило миттєво відновлює повноту виявлення нового типу ідентифікаторів без жодного розміченого прикладу. Поведінкове профілювання з індивідуальною нормою на робочому бюджеті тривог виявляє інсайдерів приблизно у 2,5 рази точніше за глобальні методи (точність 0,90 проти щонайбільше 0,35 у першій двадцятці підозрюваних). Правило підтвердження детектора дрейфу зменшує

кількість хибних тривог більш як утричі за повного виявлення справжніх змін на потоках з дискретними подіями дрейфу, а еволюційна адаптація політик підвищує преквенціальну точність на 6,6-8,0 % відносно статичної політики, зберігаючи інтерпретований набір правил; на потоках зі структурою політики ЗВД вона випереджає й інкрементну модель.

Водночас окреслено межі застосовності методів. За суто тематичної конфіденційності та неперервного високовимірного дрейфу перевагу мають відповідно ансамблеві моделі й детектори на потоці помилок, а еволюційний пошук потребує приблизно на порядок більшого часу навчання. Розроблений програмний комплекс ЗВДЕАГА реалізує всі чотири функціональні модулі системи, а сформульовані практичні рекомендації визначають умови доцільної інтеграції еволюційно-адаптивних компонентів у корпоративні системи безпеки.

Основні результати розділу опубліковано у працях [104; 105; 108; 112; 126].

ВИСНОВКИ

У дисертаційній роботі розв’язано актуальну науково-прикладну задачу підвищення ефективності виявлення та запобігання витокам конфіденційної інформації в корпоративних мережах шляхом розробки еволюційно-адаптивних методів класифікації документів, поведінкового профілювання користувачів та оптимізації політик безпеки на основі генетичних алгоритмів, а також програмних засобів, що реалізують ці методи у складі єдиного програмного комплексу. Основні наукові та практичні результати роботи полягають у такому.

1. Проведено системний аналіз наявних методів та платформ запобігання витокам даних, який охопив сучасні комерційні та дослідницькі ЗВД-рішення, механізми контентної інспекції, контекстного контролю та поведінкового аналізу. Встановлено, що наявні підходи переважно ґрунтуються на статичних правилах і заздалегідь визначених сигнатурах, що обмежує їхню адаптивність до нових сценаріїв витоків і призводить до високого рівня хибних спрацьовувань в умовах змінюваного корпоративного середовища. Формалізовано функціональні обмеження розглянутих систем за критеріями масштабованості, точності класифікації та здатності до самонавчання, що обґрунтувало доцільність застосування еволюційно-адаптивних методів для побудови ЗВД-систем нового покоління.

2. Удосконалено модель процесу запобігання витокам даних для корпоративних мереж, що інтегрує три взаємопов’язані компоненти: модель витоків даних, яка описує актора, об’єкт, канал передачі та ризик-модель інциденту; модель корпоративного середовища, що формалізує граф мережної інфраструктури, рольовий доступ із контекстним розширенням, потоки даних між зонами безпеки та точки контролю ЗВД; модель даних, де документ подається як кортеж вмісту, метаданих та контексту, а ознаковий простір формується на основі TF-IDF, структурованих шаблонів і метаданих. Композиція цих моделей утворює результуючу модель ЗВД-системи з чотирма функціональними модулями (класифікації контенту, поведінкового аналізу, адаптації політик та прийняття рішень), що забезпечує цілісне формальне підґрунтя для подальшої розробки еволюційних алгоритмів оптимізації.

3. Розроблено метод класифікації документів за рівнем конфіденційності на основі генетичного алгоритму, в якому правила класифікації кодуються у хромосомній формі IF-THEN, а еволюційний пошук спрямовується функцією пристосованості з регуляризацією складності правил. Запропоновано спеціалізовані генетичні оператори схрещування та мутації, адаптовані до структури правил контентного аналізу, а також процедуру жадібного послідовного покриття для формування частини початкової популяції. Експериментальна перевірка на реальному корпусі документів з персональними даними підтвердила конкурентну якість класифікації (F1 = 0,826 проти 0,856 у найкращого ансамблевого методу, тобто поступка близько 3,5 %) за компактного набору приблизно з п’яти правил; на суто тематичній задачі метод закономірно поступається ансамблевим класифікаторам, що окреслює область його застосовності, а саме контентний аналіз з вираженою шаблонною структурою. Керованість політики підтверджено окремим сценарієм: за появи нового типу ідентифікатора одне дописане експертом правило миттєво відновлює повноту виявлення без додаткового навчання, чого непрозорі моделі не забезпечують. Суттєвою перевагою методу є інтерпретованість: 98-99 % позитивних рішень пояснюється одним-двома правилами, які читаються безпосередньо в термінах політик безпеки, на відміну від ансамблів із сотень дерев.

4. Розроблено метод еволюційної адаптації політик ЗВД за умов дрейфу концепції, ключовим елементом якого є детектор змін на основі підтвердженого багатоознакового статистичного тесту. Політики безпеки кодуються у хромосомній формі з глобальними та локальними параметрами, а за сигналом детектора метод перемикається між режимами експлуатації та дослідження простору рішень з перенавчанням набору правил. На стандартних потоках з контрольованим дрейфом правило підтвердження зменшило кількість хибних тривог у 3,5 раза (з 2,1 до 0,6 на потік) за повного виявлення справжніх змін (F1 = 0,918 за раптового та 0,932 за поступового дрейфу) і випередило класичні детектори; на високовимірних сенсорних потоках з неперервним дрейфом ця перевага звужується, проте правило підтвердження і там скорочує кількість хибних тривог. Керована детектором еволюційна адаптація підвищила преквенціальну точність класифікації в нестаціонарному потоці на 6,6-8,0 % порівняно зі статичною політикою, зберігши інтерпретований набір правил, якого позбавлена суто інкрементна модель, а на

потоках зі структурою політики ЗВД адаптивна стратегія випередила статичну політику на 66,3 %, а інкрементну модель на 15 % за самою точністю (0,851 проти 0,739).

5. Розроблено метод поведінкового профілювання користувачів корпоративної мережі на основі генетичного алгоритму, який формує індивідуальні профілі за категоріями ознак активності. Для відстеження еволюції поведінки впроваджено адаптивне середнє з експоненційним забуванням, а оцінка аномальності обчислюється як зважена сума внесків ознак, де маска активних ознак, ваги, поріг спрацювання та коефіцієнт забування оптимізуються генетичним алгоритмом. На реальному корпусі інсайдерських загроз метод на робочому бюджеті тривог виявляє порушників приблизно у 2,5 раза точніше за глобальні методи виявлення аномалій (точність 0,90 проти щонайбільше 0,35 у першій двадцятці підозрюваних, середня влучність 0,421 проти 0,145) (ізоляційний ліс, локальний фактор викиду, однокласовий метод опорних векторів); перевага зумовлена вимірюванням відхилень відносно індивідуального, а не популяційного базового рівня.

6. Спроектовано та реалізовано програмний комплекс ЗВДЕАГА мовою Python із модульною архітектурою, що відповідає розробленим формальним моделям та об'єднує класифікацію контенту, поведінкове профілювання, виявлення дрейфу й адаптацію політик, саме він становить розроблені в дисертації програмні засоби виявлення і запобігання витоку даних. Експериментальне дослідження на відкритих корпусах і реальних наборах даних (синтетично згенерований ai4privacy, 20 Newsgroups, CERT Insider Threat r4.2, INSECTS, Electricity) та стандартних потоках RandomRBF підтвердило ефективність розроблених методів у чітко окреслених межах застосовності (F1 = 0,826 у класифікації за критичними ідентифікаторами, виявлення інсайдерів у 2,5 раза точніше за глобальні детектори, скорочення хибних тривог детектора дрейфу в 3,5 раза, приріст преквенціальної точності від 6,6 до 8,0 % на реальних потоках і 66,3 % на потоках зі структурою політики) й верифікувало коректність теоретичних положень дисертації. Окреслено практичні рекомендації щодо меж доцільного застосування розроблених еволюційно-адаптивних компонентів у корпоративних системах інформаційної безпеки.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Verizon Business. 2025 Data Breach Investigations Report. 2025. URL: https://www.verizon.com/business/resources/reports/2025-dbir-data-breach-investigations-report.pdf (дата звернення: 15.02.2026).

2. Cheng L., Liu F., Yao D. D. Enterprise data breach: causes, challenges, prevention, and future directions. *WIREs Data Mining and Knowledge Discovery*. 2017. DOI: 10.1002/widm.1211.

3. Periasamy J. K., Catherine A. C., Elamathi R., Subhiksha S. Data Leakage Vulnerability Assessment. *2023 Intelligent Computing and Control for Engineering and Business Systems (ICCEBS)*: Proceedings of the International Conference, 14–15 December, 2023. DOI: 10.1109/ICCEBS58601.2023.10448949.

4. IBM Security. Cost of a Data Breach Report 2024. 2024. URL: https://www.ibm.com/reports/data-breach (дата звернення: 15.02.2026).

5. Syarova S., Toleva-Stoimenova S., Kirkov A., Petkov S., Traykov K. Data Leakage Prevention and Detection in Digital Configurations: A Survey. *15th International Scientific and Practical Conference*: Proceedings of the International Conference, 27–28 June, 2024. P. 253–258. DOI: 10.17770/etr2024vol2.8045.

6. Herrera Montano I., García Aranda J. J., Ramos Diaz J. et al. Survey of Techniques on Data Leakage Protection and Methods to address the Insider threat. *Cluster Computing*. 2022. Vol. 25. P. 4289–4302. DOI: 10.1007/s10586-022-03668-2.

7. Cybersecurity & Infrastructure Security Agency. Defining insider threats. URL: https://www.cisa.gov/topics/physical-security/insider-threat-mitigation/defining-insider-threats (дата звернення: 15.02.2026).

8. Song S., Gao N., Zhang Y., Ma C. BRITD: behavior rhythm insider threat detection with time awareness and user adaptation. *Cybersecurity*. 2024. Vol. 7, Article 2. DOI: 10.1186/s42400-023-00190-9.

9. Mehrtak M., SeyedAlinaghi S., MohsseniPour M. et al. Security challenges and solutions using healthcare cloud computing. *Journal of Medicine and Life*. 2021. Vol. 14(4). DOI: 10.25122/jml-2021-0100.

10. Matthee M. H. Tagging Data to Prevent Data Leakage. 2016. URL: https://www.sans.org/white-papers/36967/ (дата звернення: 15.02.2026).

11. Beaman C., Barkworth A., Akande T., Hakak S., Khan M. K. Ransomware: Recent Advances, Analysis, Challenges and Future Research Directions. *Computers & Security*. 2021. Vol. 111. DOI: 10.1016/j.cose.2021.102490.

12. Savenko O., Sachenko A., Lysenko S., Markowsky G., Vasylkiv N. Botnet Detection Approach Based On The Distributed Systems. *International Journal of Computing*. 2020. Vol. 19(2). P. 190–198. DOI: 10.47839/ijc.19.2.1761.

13. Lysenko S., Savenko O., Bobrovnikova K. DDoS Botnet Detection Technique Based on the Use of the Semi-Supervised Fuzzy c-Means Clustering. *CEUR-WS*. 2018. Vol. 2104. P. 688–695. URL: http://ceur-ws.org/Vol-2104/paper_251.pdf.

14. Kashtalian A., Lysenko S., Savenko O., Nicheporuk A., Sochor T., Avsiyevych V. Multi-computer malware detection systems with metamorphic functionality. *Radioelectronic and Computer Systems*. 2024. P. 152–175. DOI: 10.32620/reks.2024.1.13.

15. Lysenko S., Savenko O., Bobrovnikova K., Kryshchuk A., Savenko B. Information technology for botnets detection based on their behaviour in the corporate area network. *Communications in Computer and Information Science*. 2017. Vol. 718. P. 166–181. DOI: 10.1007/978-3-319-59767-6_14.

16. Стасюк О. І., Корченко А. О. Метод виявлення аномалій, викликаних кібератаками в комп'ютерних мережах. *Захист інформації*. 2012. Т. 14, № 4. С. 127–132. DOI: 10.18372/2410-7840.14.3503.

17. Rajasekaran P., Crane S., Gens D., Na Y., Volckaert S., Franz M. CoDaRR: Continuous Data Space Randomization against Data-Only Attacks. *Asia Conference on Computer and Communications Security (ASIA CCS)*: Proceedings of the International Conference, 05–09 October, 2020. P. 494–505. DOI: 10.1145/3320269.3384757.

18. Feng H., Jiao H., Zhang C., Li Z., Qiu S. The Model and Method of Information Security Data Leakage Detection and Prevention. *International Conference on Data Science and Network Security (ICDSNS)*: Proceedings of the International Conference, 26–27 July, 2024. P. 1–5. DOI: 10.1109/ICDSNS62112.2024.10691002.

19. Agrawal G., Goyal S. J. Survey on Data Leakage Prevention through Machine Learning Algorithms. *2022 International Mobile and Embedded Technology Conference (MECON)*: Proceedings of the International Conference, 10–11 March, 2022. P. 121–123. DOI: 10.1109/MECON53876.2022.9752047.

20. Jadhav P., Chawan P. M. Data Leak Prevention System: A Survey. *International Research Journal of Engineering and Technology (IRJET)*. 2020. Vol. 07. URL: https://www.researchgate.net/publication/343391577_Data_Leak_Prevention_System_A_Survey.

21. Chen K., Yang Q., Wang J., Ma D., Wang L., Xu Z. What You See is the Tip of the Iceberg: A Novel Technique for Data Leakage Prevention. *27th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*: Proceedings of the International Conference, 08–10 May, 2024. P. 2870–2875. DOI: 10.1109/CSCWD61410.2024.10580487.

22. Sakr H., El-Afifi M. A Framework for Confidential Document Leakage Detection and Prevention. *Nile Journal of Communication and Computer Science*. 2024. DOI: 10.21608/njccs.2024.243509.1023.

23. Muppalaneni R., Inaganti A. C., Ravichandran N. AI-Enhanced Data Loss Prevention (DLP) Strategies for Multi-Cloud Environments. *Journal of Computing Innovations and Applications*. 2024. Vol. 2, No. 2. P. 1–13. URL: https://ciajournal.com/index.php/jcia/article/view/9.

24. Esther D. AI in Data Loss Prevention: Safeguarding Sensitive Data Against Unauthorized Access and Leakage. 2024. URL: https://www.researchgate.net/publication/385747343 (дата звернення: 15.02.2026).

25. Domnik J., Holland A. On Data Leakage Prevention Maturity: Adapting the C2M2 Framework. *Journal of Cybersecurity and Privacy*. 2024. Vol. 4(2). P. 167–195. DOI: 10.3390/jcp4020009.

26. SANS. A SANS 2021 Survey: Security Operations Center (SOC). URL: https://www.sans.org/white-papers/sans-2021-survey-security-operations-center-soc (дата звернення: 15.02.2026).

27. Jabeen T., Ashraf H., Khatoon A., Band S. S., Mosavi A. A Lightweight Genetic Based Algorithm for Data Security in Wireless Body Area Networks. *IEEE Access*. 2020. Vol. 8. P. 183460–183469. DOI: 10.1109/ACCESS.2020.3028686.

28. Shi G., Xiao Y., Li Y., Xie X. From Semantic Communication to Semantic-Aware Networking: Model, Architecture, and Open Problems. *IEEE Communications Magazine*. 2021. Vol. 59, No. 8. P. 44–50. DOI: 10.1109/MCOM.001.2001239.

29. Shvartzshnaider Y., Pavlinovic Z., Balashankar A., Wies T., Subramanian L., Nissenbaum H., Mittal P. VACCINE: Using Contextual Integrity for Data Leakage Detection. *The World Wide Web (WWW)*: Proceedings of the International Conference, 13–17 May, 2019. P. 1702–1712. DOI: 10.1145/3308558.3313655.

30. Oghomwen J. N. S., Kolo J., Kpaki M. S., Medinat S. M. Avoiding Confidential Information Leakage in Organizations for Sustainable Development. *International Journal of African Innovation and Multidisciplinary Research*. 2022. Vol. 1, No. 1. URL: https://mediterraneanpublications.com/mejaimr/article/view/13.

31. Huang X., Lu Y., Li D., Ma M. A Novel Mechanism for Fast Detection of Transformed Data Leakage. *IEEE Access*. 2018. Vol. 6. P. 35926–35936. DOI: 10.1109/ACCESS.2018.2851228.

32. Abdel-Fatao H., Nofong V. M., Agbeyeye L. K., Umaru Y. M., Alese B. K. A Robust Approach for Data Leakage Assessment. *Ghana Journal of Technology*. 2021. Vol. 5, No. 2. P. 48–59. URL: http://www2.umat.edu.gh/gjt/index.php/gjt/article/view/276.

33. Gupta I., Singh A. A Probability based Model for Data Leakage Detection using Bigraph. *7th International Conference on Communication and Network Security (ICCNS)*: Proceedings of the International Conference, 24–26 November, 2017. P. 1–5. DOI: 10.1145/3163058.3163060.

34. Alhindi H., Traore I., Woungang I. Preventing Data Leak through Semantic Analysis. *Internet of Things*. 2021. Vol. 14. DOI: 10.1016/j.iot.2019.100073.

35. Fadolalkarim D., Bertino E., Sallam A. An Anomaly Detection System for the Protection of Relational Database Systems against Data Leakage by Application Programs. *International Conference on Data Engineering (ICDE)*: Proceedings of the International Conference, 20–24 April, 2020. P. 265–276. DOI: 10.1109/ICDE48307.2020.00030.

36. Braghin S., Simioni M., Sinn M. DLPFS: The Data Leakage Prevention FileSystem. *Applied Cryptography and Network Security Workshops*: Proceedings of the International Conference, 20–23 June, 2022. Vol. 13285. DOI: 10.1007/978-3-031-16815-4_21.

37. Kiperberg M., Amit G., Yeshooroon A., Zaidenberg N. Efficient DLP-visor: An efficient hypervisor-based DLP. *IEEE/ACM 21st International Symposium on*

Cluster, Cloud and Internet Computing (CCGrid): Proceedings of the International Conference, 10–13 May, 2021. DOI: 10.1109/CCGrid51090.2021.00044.

38. Peneti S., Rani B. P. Data Leakage Prevention System with Time Stamp. *International Conference on Information Communication and Embedded Systems (ICICES)*: Proceedings of the International Conference, 25–26 February, 2016. DOI: 10.1109/ICICES.2016.7518934.

39. Gaidarski I., Kutinchev P. Using Big Data for Data Leak Prevention. *Big Data, Knowledge and Control Systems Engineering (BdKCSE)*: Proceedings of the International Conference, 21–22 November, 2019. DOI: 10.1109/BdKCSE48644.2019.9010596.

40. Teymourlouei H., Harris V. E. Preventing Data Breaches: Utilizing Log Analysis and Machine Learning for Insider Attack Detection. *Computational Science and Computational Intelligence (CSCI)*: Proceedings of the International Conference, 14–16 December, 2022. P. 1022–1027. DOI: 10.1109/CSCI58124.2022.00181.

41. Ávila R., Khoury R., Khoury R., Petrillo F. Use of Security Logs for Data Leak Detection: A Systematic Literature Review. *Security and Communication Networks*. 2021. DOI: 10.1155/2021/6615899.

42. Al-Shehari T., Alsowail R. A. An Insider Data Leakage Detection Using One-Hot Encoding, Synthetic Minority Oversampling and Machine Learning Techniques. *Entropy*. 2021. Vol. 23(10). DOI: 10.3390/e23101258.

43. Gupta S. D., Kumar R. waRLOCK: Countering Ransomware and Data Leak. *International Conference on Contemporary Computing and Communications (InC4)*: Proceedings of the International Conference, 15–16 March, 2024. DOI: 10.1109/INC460750.2024.10649292.

44. Rehman M., Akbar R., Omar M., Gilal A. A Systematic Literature Review of Ransomware Detection Methods and Tools for Mitigating Potential Attacks. *Communications in Computer and Information Science*. 2024. Vol. 2001. DOI: 10.1007/978-981-99-9589-9_7.

45. Sheen S., Asmitha K. A., Venkatesan S. R-Sentry: Deception based ransomware detection using file access patterns. *Computers and Electrical Engineering*. 2022. DOI: 10.1016/j.compeleceng.2022.108346.

46. Lee S.-J., Shim H.-Y., Lee Y.-R., Park T.-R., Park S.-H., Lee I.-G. Study on Systematic Ransomware Detection Techniques. *International Conference on Advanced Communication Technology (ICACT)*: Proceedings of the International Conference, 07–10 February, 2021. DOI: 10.23919/ICACT51234.2021.9370472.

47. Li W., Meng W., Parra-Arnau J., Choo K.-K. R. Enhancing Challenge-based Collaborative Intrusion Detection Against Insider Attacks using Spatial Correlation. *Dependable and Secure Computing (DSC)*: Proceedings of the International Conference, 30 January – 02 February, 2021. DOI: 10.1109/DSC49826.2021.9346232.

48. Gupta K., Kush A. A Forecasting-Based DLP Approach for Data Security. *Lecture Notes on Data Engineering and Communications Technologies*. 2021. Vol. 54. P. 1–8. DOI: 10.1007/978-981-15-8335-3_1.

49. Guevara C., Santos M., López V. Data Leakage Detection Algorithm based on Task Sequences and Probabilities. *Knowledge-Based Systems*. 2017. Vol. 120. DOI: 10.1016/j.knosys.2017.01.009.

50. Singh Shishodia B., Nene M. J. Data Leakage Prevention System for Internal Security. *International Conference on Futuristic Technologies (INCOFT)*: Proceedings of the International Conference, 25–27 November, 2022. P. 1–6. DOI: 10.1109/INCOFT55651.2022.10094509.

51. Mihailescu M. I., Nita S. L., Rogobete M., Marascu V. Unveiling Threats: Leveraging User Behavior Analysis for Enhanced Cybersecurity. *Electronics, Computers and Artificial Intelligence (ECAI)*: Proceedings of the International Conference, 29–30 June, 2023. DOI: 10.1109/ECAI58194.2023.10194039.

52. Herrera Montano I., de la Torre Díez I., García Aranda J. J. et al. Secure File Systems for the Development of a Data Leak Protection (DLP) Tool Against Internal Threats. *17th Iberian Conference on Information Systems and Technologies (CISTI)*: Proceedings of the International Conference, 22–25 June, 2022. P. 1–7. DOI: 10.23919/CISTI54924.2022.9820170.

53. Nartey C., Tchao E. T., Gadze J. D. et al. Blockchain-IoT peer device storage optimization using an advanced time-variant multi-objective particle swarm optimization algorithm. *EURASIP Journal on Wireless Communications and Networking*. 2022. No. 5. DOI: 10.1186/s13638-021-02074-3.

54. Costa G., Forestiero A., Ortale R. Rule-Based Detection of Anomalous Patterns in Device Behavior for Explainable IoT Security. *IEEE Transactions on Services Computing*. 2023. Vol. 16, No. 6. P. 4514–4525. DOI: 10.1109/TSC.2023.3327822.

55. Alhaidari F., Abu Shaib N., Alsafi M., Alharbi H. et al. ZeVigilante: Detecting Zero-Day Malware Using Machine Learning and Sandboxing Analysis Techniques. *Computational Intelligence and Neuroscience*. 2022. Vol. 2022. DOI: 10.1155/2022/1615528.

56. Kayode A. B., Dayo A. O., Uthman A. A. A Review on Distribution Model for Mobile Agent-Based Information Leakage Prevention. *Communications and Network*. 2021. Vol. 13. P. 68–78. DOI: 10.4236/cn.2021.132006.

57. Gomes H. M., Bifet A., Read J. et al. Adaptive Random Forests for Evolving Data Stream Classification. *Machine Learning*. 2017. Vol. 106. P. 1469–1495. DOI: 10.1007/s10994-017-5642-8.

58. Cano A., Krawczyk B. Kappa Updated Ensemble for Drifting Data Stream Mining. *Machine Learning*. 2020. Vol. 109. P. 175–218. DOI: 10.1007/s10994-019-05840-z.

59. Cano A., Krawczyk B. ROSE: Robust Online Self-Adjusting Ensemble for Continual Learning on Imbalanced Drifting Data Streams. *Machine Learning*. 2022. Vol. 111. P. 2561–2599. DOI: 10.1007/s10994-022-06168-x.

60. Hinder F., Vaquet V., Hammer B. One or two things we know about concept drift – A survey on monitoring in evolving environments. Part A: Detecting concept drift. *Frontiers in Artificial Intelligence*. 2024. Vol. 7. Art. 1330257. DOI: 10.3389/frai.2024.1330257.

61. Arora S., Rani R., Saxena N. A systematic review on detection and adaptation of concept drift in streaming data using machine learning techniques. *WIREs Data Mining and Knowledge Discovery*. 2024. Vol. 14, No. 4. Article e1536. DOI: 10.1002/widm.1536.

62. Bayram F., Ahmed B. S., Kassler A. From concept drift to model degradation: An overview on performance-aware drift detectors. *Knowledge-Based Systems*. 2022. Vol. 245. Article 108632. DOI: 10.1016/j.knosys.2022.108632.

63. Suárez-Cetrulo A. L., Quintana D., Cervantes A. A survey on machine learning for recurring concept drifting data streams. *Expert Systems with Applications*. 2023. Vol. 213. Article 118934. DOI: 10.1016/j.eswa.2022.118934.

64. Katoch S., Chauhan S. S., Kumar V. A review on genetic algorithm: past, present, and future. *Multimedia Tools and Applications*. 2021. Vol. 80. P. 8091–8126. DOI: 10.1007/s11042-020-10139-6.

65. Alhijawi B., Awajan A. Genetic algorithms: theory, genetic operators, solutions, and applications. *Evolutionary Intelligence*. 2024. Vol. 17. P. 1245–1256. DOI: 10.1007/s12065-023-00822-6.

66. Slowik A., Kwasnicka H. Evolutionary algorithms and their applications to engineering problems. *Neural Computing & Applications*. 2020. Vol. 32. P. 12363–12379. DOI: 10.1007/s00521-020-04832-8.

67. Murata T., Ishibuchi H. MOGA: multi-objective genetic algorithms. *International Conference on Evolutionary Computation*: Proceedings of the International Conference, 29 November – 01 December, 1995. DOI: 10.1109/ICEC.1995.489161.

68. Long Q., Wu C., Wang X., Jiang L., Li J. A Multiobjective Genetic Algorithm Based on a Discrete Selection Procedure. *Mathematical Problems in Engineering*. 2015. P. 1–17. DOI: 10.1155/2015/349781.

69. Kumar G., Ganesh A., Bhoopal N. et al. Evolutionary Algorithms for Real Time Engineering Problems: A Comprehensive Review. *Ingénierie des systèmes d'information*. 2021. Vol. 26. P. 179–190. DOI: 10.18280/isi.260205.

70. Deng W., Shang S., Cai X. et al. An improved differential evolution algorithm and its application in optimization problem. *Soft Computing*. 2021. Vol. 25. P. 5277–5298. DOI: 10.1007/s00500-020-05527-x.

71. Ahmad M. F., Mat Isa N. A., Lim W. H., Ang K. M. Differential evolution: A recent review based on state-of-the-art works. *Alexandria Engineering Journal*. 2022. Vol. 61. DOI: 10.1016/j.aej.2021.09.013.

72. Hendrick N., Torra V. Genetic Algorithms in Data Masking: Towards Privacy as a Service. *International Conference on Artificial Intelligence and Soft Computing (ICAISC)*: Proceedings of the International Conference, 21–23 June, 2021. DOI: 10.1007/978-3-030-87986-0_34.

73. Tahir M., Sardaraz M., Mehmood Z., Muhammad S. CryptoGA: A Cryptosystem based on Genetic Algorithm for Cloud Data Security. *Cluster Computing*. 2021. Vol. 24. P. 739–752. DOI: 10.1007/s10586-020-03157-4.

74. Mirjalili S. Evolutionary Algorithms and Neural Networks. 2019. DOI: 10.1007/978-3-319-93025-1.

75. Ren T., Wang X., Li Q., Wang C., Dong J., Guo G. Vulnerability Mining Technology Based on Genetic Algorithm and Model Constraint. *IOP Conference Series: Materials Science and Engineering*. 2020. Vol. 750. DOI: 10.1088/1757-899X/750/1/012128.

76. Mann Z. Á., Kunz F., Laufer J., Bellendorf J., Metzger A., Pohl K. RADAR: Data Protection in Cloud-Based Computer Systems at Run Time. *IEEE Access*. 2021. Vol. 9. P. 70816–70842. DOI: 10.1109/ACCESS.2021.3078059.

77. Niklekaj M. Security in Computer Networks: Threats, Challenges, and Protection. *Ingenious*. 2023. No. 3. DOI: 10.58944/mcne2043.

78. Ahmad S., Mehfuz S., Beg J. Cloud security framework and key management services collectively for implementing DLP and IRM. *Materials Today: Proceedings*. 2022. Vol. 62. P. 4828–4836. DOI: 10.1016/j.matpr.2022.03.420.

79. Mejri O., Yang D., Doh I. Cloud Security Issues and Log-based Proactive Strategy. *International Conference on Advanced Communication Technology (ICACT)*: Proceedings of the International Conference, 07–10 February, 2021. DOI: 10.23919/ICACT52886.2021.9728822.

80. Bederna Z., Szádeczky T. Modelling computer networks for further security research. *Security and Defence Quarterly*. 2021. Vol. 36(4). DOI: 10.35467/sdq/141572.

81. Soma V. Handling Encryption and Data Loss prevention in the Cloud-Based Systems. *Journal of Artificial Intelligence & Cloud Computing*. 2024. P. 1–5. DOI: 10.47363/JAICC/2024(3)E124.

82. Chu A. M. Y., So M. K. P. Organizational Information Security Management for Sustainable Information Systems: An Unethical Employee Information Security Behavior Perspective. *Sustainability*. 2020. Vol. 12. P. 1–25. DOI: 10.3390/su12083163.

83. Calias S. E., Caoli B., Padilla R. et al. The Impact of BYOD (Bring Your Own Device) On Network Security: A Literature Review. *Southeast Asian Journal of Science*

and Technology. 2024. Vol. 9. URL: https://sajst.org/online/index.php/sajst/article/view/303.

84. Valleru V. Cost-Effective Cloud Data Loss Prevention Strategies for Small and Medium-Sized Enterprises. *International Research Journal of Engineering and Technology*. 2024. Vol. 11, No. 05.

85. Georg-Schaffner L., Prinz E. Corporate management boards' information security orientation: an analysis of cybersecurity incidents in DAX 30 companies. *Journal of Management and Governance*. 2022. Vol. 26. P. 1375–1408. DOI: 10.1007/s10997-021-09588-4.

86. Allakonda M., Sagar K. A Survey on data security challenges in multi cloud environment. *IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)*: Proceedings of the International Conference, 09–11 July, 2021. DOI: 10.1109/CONECCT52877.2021.9622722.

87. Chowdhury R., Prince N. U., Abdullah S., Mim L. The role of predictive analytics in cybersecurity: Detecting and preventing threats. *World Journal of Advanced Research and Reviews*. 2024. DOI: 10.30574/wjarr.2024.23.2.2494.

88. Fugkeaw S., Worapaluk K., Tuekla A., Namkeatsakul S. Design and Development of a Dynamic and Efficient PII Data Loss Prevention System. *International Conference on Computing and Information Technology (IC2IT)*: Proceedings of the International Conference, 13–14 May, 2021. DOI: 10.1007/978-3-030-79757-7_3.

89. Lavrov E., Zolkin A., Aygumov T., Chistyakov M., Akhmetov I. Analysis of information security issues in corporate computer networks. *IOP Conference Series: Materials Science and Engineering*. 2021. Vol. 1047. DOI: 10.1088/1757-899X/1047/1/012117.

90. Gartner. Market Guide for Data Loss Prevention. URL: https://www.gartner.com/en/documents/6342779 (дата звернення: 15.02.2026).

91. Alsuwaie M. A., Habibnia B., Gladyshev P. Data Leakage Prevention Adoption Model & DLP Maturity Level Assessment. *International Symposium on Computer Science and Intelligent Controls (ISCSIC)*: Proceedings of the International Conference, 12–14 November, 2021. DOI: 10.1109/ISCSIC54682.2021.00077.

92. Kleij R. V., Wijn R., Hof T. An application and empirical test of the Capability Opportunity Motivation-Behaviour model to data leakage prevention in financial organizations. *Computers & Security*. 2020. Vol. 97. DOI: 10.1016/j.cose.2020.101970.

93. Forcepoint DLP. URL: https://www.forcepoint.com/product/dlp-data-loss-prevention (дата звернення: 15.02.2026).

94. FortiDLP. URL: https://www.fortinet.com/products/fortidlp (дата звернення: 18.07.2026).

95. GTB Technologies. URL: https://gttb.com/data-loss-prevention/ (дата звернення: 15.02.2026).

96. DTEX Systems. URL: https://www.dtexsystems.com/capabilities/risk-adaptive-dlp/ (дата звернення: 15.02.2026).

97. Crowdstrike Falcon Data Protection. URL: https://www.crowdstrike.com/en-us/platform/data-protection (дата звернення: 15.02.2026).

98. Cyberhaven. URL: https://www.cyberhaven.com/blog/data-detection-and-response-introduction (дата звернення: 15.02.2026).

99. Microsoft Purview. URL: https://www.microsoft.com/uk-ua/security/business/microsoft-purview (дата звернення: 15.02.2026).

100. Broadcom Symantec DLP. URL: https://www.broadcom.com/products/cybersecurity/information-protection/data-loss-prevention (дата звернення: 15.02.2026).

101. Nightfall. URL: https://www.nightfall.ai/ (дата звернення: 15.02.2026).

102. Proofpoint Human-Centric DLP. URL: https://www.proofpoint.com/uk/products/defend-data (дата звернення: 15.02.2026).

103. Skyguard. URL: https://www.skyguard.net/products/mdlp (дата звернення: 18.07.2026).

104. Віжевський П. В., Савенко О. С. Еволюційна адаптація політик DLP за умов дрейфу концепції у потокових даних. *Центральноукраїнський науковий вісник. Технічні науки*. 2025. № 12(43). С. 9–19. DOI: https://doi.org/10.32515/2664-262X.2025.12(43).2.9-19

105. Віжевський П. В., Савенко О. С. Стратегії виявлення та запобігання витокам даних у корпоративних мережах згідно генетичного алгоритму. *Information*

Technology: Computer Science, Software Engineering and Cyber Security. 2025. № 4. С. 42–56. DOI: https://doi.org/10.32782/IT/2025-4-6

106. Віжевський П. В., Кравчик Ю. В. Модель системи запобігання витоку даних з еволюційною адаптацією на основі генетичних алгоритмів. *Вісник Хмельницького національного університету. Серія: Технічні науки*. 2025. № 5. С. 429–436. DOI: https://doi.org/10.31891/2307-5732-2025-357-56

107. Віжевський П. В., Савенко О. С. Модель процесу виявлення витоків даних з еволюційною адаптацією. *Вимірювальна та обчислювальна техніка в технологічних процесах*. 2026. № 1. С. 270–277. DOI: https://doi.org/10.31891/2219-9365-2026-85-34

108. Віжевський П. В., Савенко О. С. Поведінкове профілювання користувачів та виявлення аномалій на основі генетичного алгоритму. *Системні технології*. 2026. № 1 (162). С. 148–159. DOI: https://doi.org/10.34185/1562-9945-5-162-2026-17

109. Rehida P., Savenko O., Sachenko A., Drozd A., Vizhevski P. A trust model that ensures the correctness of computing in grid computing system. *Proceedings of the 5th International Workshop on Intelligent Information Technologies and Systems of Information Security (IntelITSIS-2024)*, Khmelnytskyi, March 2024. Vol. 3675. P. 388–401. URL: https://ceur-ws.org/Vol-3675/paper28.pdf

110. Golovko I., Savenko O., Vizhevskyi P., Sachenko A. Obfuscation process with machine learning module. *2024 14th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, Athens, Greece, 2024. P. 1–7. DOI: https://doi.org/10.1109/DESSERT65323.2024.11122185

111. Golovko I., Savenko O., Vizhevskiy P., Klein O., Salem A.-B. M. Obfuscation technologies of high-level source code using artificial intelligence. Proceedings of the 1st International Workshop on Intelligent & CyberPhysical Systems (ICyberPhyS-2024), Khmelnytskyi, Ukraine, June 28, 2024. Vol. 3736. P. 324–338. URL: https://ceur-ws.org/Vol-3736/paper24.pdf

112. Sachenko A., Vizhevskyi P., Savenko O., Ostroverkhov V., Maslyyak B. Modern strategies for data leak detection and prevention in corporate networks. *Modern Data Science Technologies Doctoral Consortium (MoDaST 2025)*, Lviv, Ukraine, June 15, 2025. 2025. P. 275–292. URL: https://ceur-ws.org/Vol-4005/paper19.pdf

113. Rudolph G. Convergence analysis of canonical genetic algorithms. *IEEE Transactions on Neural Networks*. 1994. Vol. 5. P. 96–101. DOI: https://doi.org/10.1109/72.265964

114. Gama J., Sebastião R., Rodrigues P. On evaluating stream learning algorithms. *Machine Learning*. 2013. Vol. 90. P. 317–346. DOI: https://doi.org/10.1007/s10994-012-5320-9

115. GDPR. URL: https://gdpr-text.com/uk/

116. Про захист персональних даних : Закон України від 01.06.2010 № 2297-VI. URL: https://zakon.rada.gov.ua/laws/show/2297-17

117. Ai4Privacy. PII-Masking-200k: a dataset for training and evaluating PII detection and masking models. *Hugging Face*, 2024. URL: https://huggingface.co/datasets/ai4privacy/pii-masking-200k

118. Lang K. NewsWeeder: learning to filter netnews. Machine Learning *Proceedings 1995: Proceedings of the 12th International Conference on Machine Learning*, Tahoe City, USA, 1995. P. 331–339.

119. Lindauer B. Insider Threat Test Dataset. *Carnegie Mellon University*, 2020. DOI: https://doi.org/10.1184/R1/12841247.v1

120. Souza V. M. A., Reis D. M., Maletzke A. G., Batista G. E. Challenges in benchmarking stream learning algorithms with real-world data. *Data Mining and Knowledge Discovery*. 2020. Vol. 34. P. 1805–1858. URL: https://arxiv.org/abs/2005.00113

121. Harries M. SPLICE-2 comparative evaluation: electricity pricing. *Technical report, University of New South Wales*, 1999.

122. Montiel J., Halford M., Mastelini S. M. et al. River: machine learning for streaming data in Python. *Journal of Machine Learning Research*, 2021. Vol. 22, no. 110. P. 1–8.

123. De Jong K. A., Spears W. M., Gordon D. F. Using genetic algorithms for concept learning. *Machine Learning*. 1993. Vol. 13. P. 161–188. DOI: 10.1007/BF00993042.

124. Wilson S. W. Classifier fitness based on accuracy. *Evolutionary Computation*. 1995. Vol. 3, no. 2. P. 149–175. DOI: 10.1162/evco.1995.3.2.149.

125. Urbanowicz R. J., Moore J. H. Learning classifier systems: a complete introduction, review, and roadmap. *Journal of Artificial Evolution and Applications*. 2009. Article ID 736398. DOI: 10.1155/2009/736398.

126. Свідоцтво про реєстрацію авторського права на твір (програму) № 144040. Комп'ютерна програма «Програмний комплекс запобігання витоку даних з еволюційною адаптацією на основі генетичних алгоритмів (ЗВДЕАГА)» / П. В. Віжевський, О. С. Савенко. Дата реєстрації 04.03.2026. URL: https://iprop-ua.com/cr/ny8dq30b/

127. Бекетова Г., Ахметов Б., Корченко О., Лахно В. Розробка моделі інтелектуального розпізнавання аномалій і кібератак з використанням логічних процедур, які базуються на покриттях матриць ознак. *Безпека інформації*. 2016. Т. 22, № 3. С. 242–254.

128. Androulidakis I., Kharchenko V., Kovalenko A. IMECA-based Technique for Security Assessment of Private Communications: Technology and Training. Information & Security: An *International Journal*. 2016. Vol. 35, No. 1. P. 99–120. DOI: 10.11610/isij.3505.

129. Ткачук Р. Л., Сікора Л. С., Лиса Н. К., Міюшкович Ю. Г., Марцишин Р. С., Сабат В. І. Категорні моделі представлення структури і динамічного стану ієрархічних систем для виявлення факторів атак і ризиків. *Комп'ютерні технології друкарства*. 2018. Т. 40, № 2. С. 25–45.

130. Mukhin V., Pavlenko E. Adaptive Networks Safety Control. *Advances in Electrical and Computer Engineering*. 2007. Vol. 7, No. 1. P. 54–58.

131. Штовба С. Д., Мазуренко В. В., Савчук Д. А. Генетичний алгоритм вибору правил нечіткої бази знань, збалансованої за критеріями точності та компактності. *Наукові праці Вінницького національного технічного університету*. 2012. № 3. URL: http://praci.vntu.edu.ua/index.php/praci/article/view/331 (дата звернення: 24.07.2026).

132. Clark P., Niblett T. The CN2 induction algorithm. *Machine Learning*. 1989. Vol. 3, no. 4. P. 261–283. DOI: https://doi.org/10.1007/BF00116835

133. Cohen W. W. Fast effective rule induction. *Machine Learning Proceedings 1995*. San Francisco: Morgan Kaufmann, 1995. P. 115–123. DOI: https://doi.org/10.1016/B978-1-55860-377-6.50023-2

134. Bifet A., Gavaldà R. Learning from time-changing data with adaptive windowing. *Proceedings of the 2007 SIAM International Conference on Data Mining*. 2007. P. 443–448. DOI: https://doi.org/10.1137/1.9781611972771.42

135. Gama J., Medas P., Castillo G., Rodrigues P. Learning with drift detection. *Advances in Artificial Intelligence – SBIA 2004. Lecture Notes in Computer Science*. 2004. Vol. 3171. P. 286–295. DOI: https://doi.org/10.1007/978-3-540-28645-5_29

136. Baena-García M., del Campo-Ávila J., Fidalgo R., Bifet A., Gavaldà R., Morales-Bueno R. Early drift detection method. *Fourth International Workshop on Knowledge Discovery from Data Streams (ECML PKDD 2006)*. 2006. P. 77–86.

137. Raab C., Heusinger M., Schleif F.-M. Reactive soft prototype computing for concept drift streams. *Neurocomputing*. 2020. Vol. 416. P. 340–351. DOI: https://doi.org/10.1016/j.neucom.2019.11.111

ДОДАТКИ

ДОДАТОК А

Список публікацій здобувача

Наукові праці, в яких опубліковані основні наукові результати дисертації

1. Віжевський П. В., Савенко О. С. Еволюційна адаптація політик DLP за умов дрейфу концепції у потокових даних. *Центральноукраїнський науковий вісник. Технічні науки*. 2025. № 12(43). С. 9–19. DOI: https://doi.org/10.32515/2664-262X.2025.12(43).2.9-19

2. Віжевський П. В., Савенко О. С. Стратегії виявлення та запобігання витокам даних у корпоративних мережах згідно генетичного алгоритму. *Information Technology: Computer Science, Software Engineering and Cyber Security*. 2025. № 4. С. 42–56. DOI: https://doi.org/10.32782/IT/2025-4-6

3. Віжевський П. В., Кравчик Ю. В. Модель системи запобігання витоку даних з еволюційною адаптацією на основі генетичних алгоритмів. *Вісник Хмельницького національного університету. Серія: Технічні науки*. 2025. № 5. С. 429–436. DOI: https://doi.org/10.31891/2307-5732-2025-357-56

4. Віжевський П. В., Савенко О. С. Модель процесу виявлення витоків даних з еволюційною адаптацією. *Вимірювальна та обчислювальна техніка в технологічних процесах*. 2026. № 1. С. 270–277. DOI: https://doi.org/10.31891/2219-9365-2026-85-34

5. Віжевський П. В., Савенко О. С. Поведінкове профілювання користувачів та виявлення аномалій на основі генетичного алгоритму. *Системні технології*. 2026. № 1 (162). С. 148–159. DOI: https://doi.org/10.34185/1562-9945-5-162-2026-17

Праці, які засвідчують апробацію матеріалів дисертації

6. Rehida P., Savenko O., Sachenko A., Drozd A., Vizhevski P. A trust model that ensures the correctness of computing in grid computing system. *Proceedings of the 5th International Workshop on Intelligent Information Technologies and Systems of Information Security (IntelITSIS-2024)*, Khmelnytskyi, March 2024. Vol. 3675. P. 388–401. URL: https://ceur-ws.org/Vol-3675/paper28.pdf

7. Golovko I., Savenko O., Vizhevskyi P., Sachenko A. Obfuscation process with machine learning module. *2024 14th International Conference on Dependable Systems,*

Services and Technologies (DESSERT), Athens, Greece, 2024. P. 1–7. DOI: https://doi.org/10.1109/DESSERT65323.2024.11122185

8. Golovko I., Savenko O., Vizhevskiy P., Klein O., Salem A.-B. M. Obfuscation technologies of high-level source code using artificial intelligence. Proceedings of the 1st International Workshop on Intelligent & CyberPhysical Systems (ICyberPhyS-2024), Khmelnytskyi, Ukraine, June 28, 2024. Vol. 3736. P. 324–338. URL: https://ceur-ws.org/Vol-3736/paper24.pdf

9. Sachenko A., Vizhevskyi P., Savenko O., Ostroverkhov V., Maslyyak B. Modern strategies for data leak detection and prevention in corporate networks. *Modern Data Science Technologies Doctoral Consortium (MoDaST 2025)*, Lviv, Ukraine, June 15, 2025. 2025. P. 275–292. URL: https://ceur-ws.org/Vol-4005/paper19.pdf

Публікації, які додатково відображають наукові результати дисертації

10. Свідоцтво про реєстрацію авторського права на твір (програму) № 144040. Комп'ютерна програма «Програмний комплекс запобігання витоку даних з еволюційною адаптацією на основі генетичних алгоритмів (ЗВДЕАГА)» / П. В. Віжевський, О. С. Савенко. Дата реєстрації 04.03.2026. URL: https://iprop-ua.com/cr/ny8dq30b/

ДОДАТОК Б

Результати експериментів з класифікації документів за допомогою генетичного алгоритму

Результати експерименту з класифікації документів за рівнем конфіденційності: повні показники якості та обчислювальні характеристики всіх методів на обох реальних корпусах; приклади продукційних правил, синтезованих генетичним алгоритмом.

Таблиця Б.1 - Показники класифікації на корпусі ai4privacy (5-блокова перехресна валідація, середнє ± с.в.)

Метод	F1	Точність	Повнота	Навчання, с	Класиф. 1 тис., с
ГА-правила	0,826 ± 0,018	0,904	0,761	30,04	0,000
Випадковий ліс	0,856 ± 0,015	0,931	0,793	1,22	0,035
Градієнтний бустинг	0,840 ± 0,019	0,925	0,770	1,12	0,002
Лінійний SVM	0,850 ± 0,009	0,911	0,796	0,01	0,001
Логістична регресія	0,845 ± 0,008	0,934	0,772	0,02	0,001
Наївний баєсів	0,819 ± 0,008	0,889	0,761	0,00	0,001

Таблиця Б.2 - Показники класифікації на корпусі 20 Newsgroups (5-блокова перехресна валідація, середнє ± с.в.)

Метод	F1	Точність	Повнота	Навчання, с	Класиф. 1 тис., с
ГА-правила	0,591 ± 0,013	0,468	0,805	20,91	0,000
Випадковий ліс	0,664 ± 0,021	0,828	0,557	1,50	0,037
Градієнтний бустинг	0,605 ± 0,020	0,873	0,464	1,61	0,002
Лінійний SVM	0,703 ± 0,022	0,767	0,650	0,02	0,001
Логістична регресія	0,664 ± 0,027	0,835	0,553	0,07	0,001
Наївний баєсів	0,675 ± 0,019	0,736	0,627	0,00	0,001

Таблиця Б.3 - Приклади правил, синтезованих генетичним алгоритмом

Корпус	Синтезоване правило
ai4privacy	IF pat_ssn > 0.0556 THEN confidential
ai4privacy	IF pat_iban > 0.0399 THEN confidential
ai4privacy	IF pat_crypto > 0.124 THEN confidential
ai4privacy	IF pat_card_like > 0.0972 THEN confidential
ai4privacy	IF kw_credential > 0.304 THEN confidential
20 Newsgroups	IF gun > 0.0218 THEN confidential
20 Newsgroups	IF meta_digit_ratio < 0.00514 AND god < 1e-09 AND hockey < 0.0146 THEN confidential
20 Newsgroups	IF key > 0.0266 AND team < 0.00823 AND graphics < 1e-09 THEN confidential
20 Newsgroups	IF little < 0.837 AND fbi > 0.394 THEN confidential

Таблиця Б.4 - Вартість виправлення непрозорої моделі за появи нового типу ідентифікатора

Розмічених IBAN-позитивів нового типу	F1 випадкового лісу після донавчання
0 (базова модель)	0,778 ± 0,001
10	0,856 ± 0,021
50	0,841 ± 0,005
200	0,835 ± 0,003
Довідково: ГА-правила + правило експерта (0 документів)	0,847 ± 0,007
Довідково: випадковий ліс + guard-правило (0 документів)	0,930 ± 0,005

ДОДАТОК В

Результати експерименту з поведінкового профілювання користувачів

Результати експерименту з поведінкового профілювання користувачів на корпусі CERT Insider Threat r4.2 [119]: повні показники якості всіх методів, зокрема середню влучність, точність і повноту на кількох робочих бюджетах тривог та довідкове значення AUC, а також конфігурацію детектора, знайдену генетичним алгоритмом.

Таблиця В.1 - Повні показники виявлення інсайдерів на корпусі CERT r4.2 (оцінювання на рівні користувача)

Метод	AP	Точність top-20	Точність top-50	Повнота top-50	Повнота top-100	AUC
ГА-профіль	0,421	0,90	0,54	0,39	0,44	0,466
Isolation Forest	0,114	0,25	0,22	0,16	0,17	0,452
LOF	0,067	0,05	0,02	0,01	0,04	0,262
One-Class SVM	0,145	0,35	0,24	0,17	0,23	0,561

Таблиця В.2 - Конфігурація персонального детектора, знайдена генетичним алгоритмом

Параметр конфігурації	Значення
Активні ознаки профілю	кількість входів, USB-підключення, копіювання файлів, надіслані листи, обсяг листів
Поріг спрацювання τ	1,43
Коефіцієнт забування ρ	0,995
День поділу навчання/тесту	300
Період прогріву профілю, днів	30

ДОДАТОК Г

Результати експериментів з виявлення дрейфу концепції та еволюції адаптації політик

Результати експериментів з виявлення дрейфу концепції та еволюційної адаптації політик. Показники детекторів на стандартних потоках RandomRBF [122] усереднено за десятьма незалежними потоками окремо для раптового та поступового дрейфу; окремо наведено чесну перевірку на реальних сенсорних потоках INSECTS [120], де перевага методу звужується. Преквенціальна точність стратегій адаптації подано для обох реальних потоків.

Таблиця Г.1 - Детектори дрейфу на потоках RandomRBF, раптовий дрейф (середнє за 10 потоками)

Детектор	Точність	Повнота	F1	Затримка, зразків	Хибних тривог
Запропонований (KS + підтвердження)	0,860	1,000	0,918	275	0,6
KS без підтвердження	0,613	1,000	0,753	150	2,1
ADWIN	0,467	0,333	0,387	208	0,8
DDM	0,240	0,100	0,136	351	1,3
EDDM	0,129	0,433	0,193	75	20,9
KSWIN	0,275	0,133	0,169	85	1,0

Таблиця Г.2 - Детектори дрейфу на потоках RandomRBF, поступовий дрейф (середнє за 10 потоками)

Детектор	Точність	Повнота	F1	Затримка, зразків	Хибних тривог
Запропонований (KS + підтвердження)	0,885	1,000	0,932	340	0,5
KS без підтвердження	0,756	1,000	0,848	190	1,6
ADWIN	0,775	0,600	0,661	561	0,3
DDM	0,520	0,300	0,353	403	0,7
EDDM	0,193	0,500	0,264	207	19,1
KSWIN	0,333	0,167	0,213	396	1,0

Таблиця Г.3 - Детектори дрейфу на реальних потоках INSECTS (середнє за 4 варіантами)

Детектор	Точність	Повнота	F1	Затримка, зразків	Хибних тривог
Запропонований (KS + підтвердження)	0,156	0,700	0,232	740	15,5
KS без підтвердження	0,125	0,950	0,201	760	23,0
ADWIN	0,172	0,700	0,259	148	9,0
DDM	0,292	0,475	0,336	889	2,8
EDDM	0,172	0,475	0,249	560	54,8
KSWIN	0,254	0,825	0,384	50	7,5

Таблиця Г.4 - Преквенціальна точність стратегій адаптації на реальних потоках

Стратегія	Elec2	INSECTS-Abr	Перенавчань (Elec2 / INSECTS)
Статична політика	0,686	0,525	нема
Еволюційна адаптація	0,731	0,567	56 / 16
Інкрементальна лог. регресія	0,864	0,724	посерійно

Таблиця Г.5 - Преквенціальна точність на потоках зі структурою політики ЗВД (5 незалежних потоків)

Потік	Статична	Еволюційна адаптація	Інкрементальна
Потік 1	0,573	0,864	0,756
Потік 2	0,520	0,863	0,741
Потік 3	0,550	0,857	0,748
Потік 4	0,442	0,838	0,747
Потік 5	0,474	0,832	0,704
Середнє ± с.в.	0,512 ± 0,048	0,851 ± 0,013	0,739 ± 0,018

ДОДАТОК Д

Вихідний програмний код

Вихідний програмний код, використаний під час проведення експериментальних досліджень, розміщено у відкритому репозиторії GitHub:

Віжевський П. В. DLPEA: Data Leak Prevention with Evolutionary Algorithms. GitHub. URL: https://github.com/sp4rd4/DLPEA (дата звернення: 17.07.2026).

На рисунку Д.1 наведено структуру репозиторію з вихідним програмним кодом і файлами, необхідними для відтворення експериментальних досліджень.

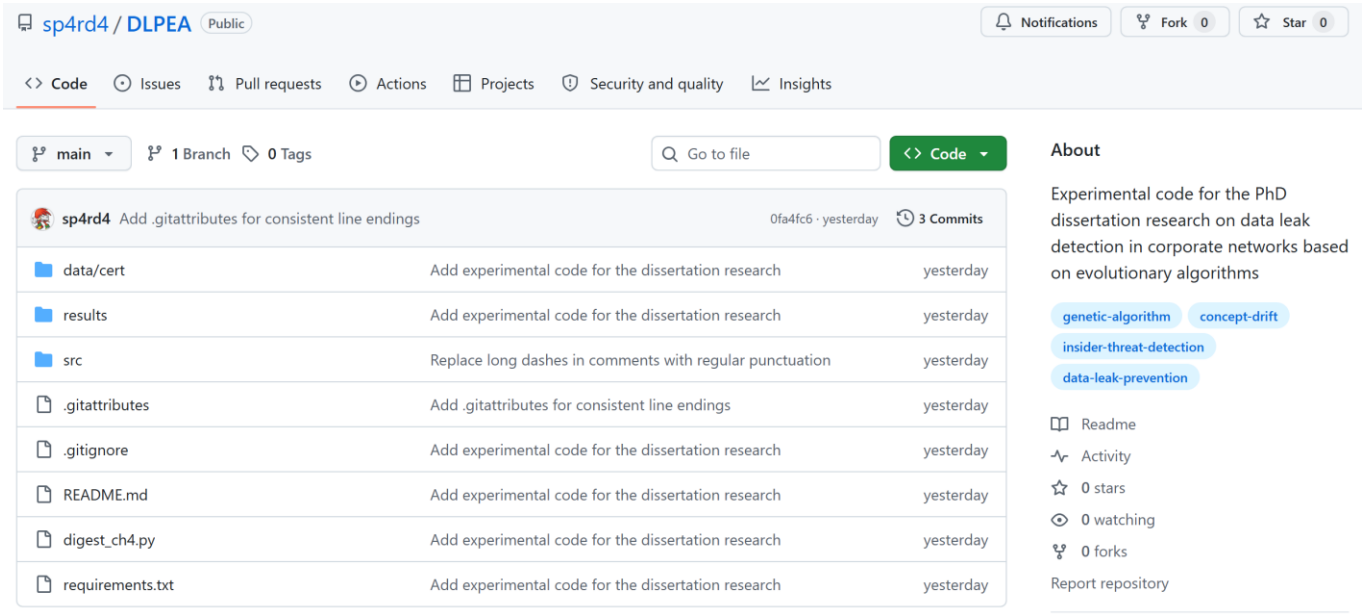


Рисунок Д.1 – Структура репозиторію GitHub із вихідним програмним кодом

Репозиторій об'єднує програмні компоненти, що реалізують методи виявлення витоку даних: класифікацію документів за рівнем конфіденційності на основі генетичного алгоритму над продукційними правилами (параграф 3.1), виявлення дрейфу концепції та еволюційну адаптацію політик безпеки (параграф 3.2), поведінкове профілювання користувачів (параграф 3.3), а також сценарії експериментів з розділу 4. Програмний код забезпечує завантаження і підготовку наборів даних, формування ознакового простору документів і поведінкових ознак, навчання та оцінювання класифікаторів, виявлення дрейфу на потоках даних і преквенціальне оцінювання стратегій адаптації політик.

Структура репозиторію така:

– src/ga_classifier.py – класифікатор на основі генетичного алгоритму над продукційними правилами IF-THEN (підрозділ 3.1);

– src/datasets.py – формування ознакового простору документів (TF-IDF, шаблони критичних ідентифікаторів, ключові слова, метадані), генератор контрольованого PII-корпусу для сценарію керованості та потоків-політик;

– src/ai4privacy_corpus.py – завантаження реального корпусу ai4privacy PII-Masking-200k і розмітка документів за наявністю критичних ідентифікаторів;

– src/behavior.py – адаптивний поведінковий профіль користувача та генетична оптимізація його конфігурації (підрозділ 3.3);

– src/cert_features.py – формування ознак «користувач-день» із журналів набору CERT Insider Threat r4.2;

– src/drift.py – детектор дрейфу концепції на основі тестів Колмогорова-Смирнова з корекцією Бонферроні та правилом підтвердження, обгортки базових детекторів ADWIN, DDM, EDDM, KSWIN (підрозділ 3.2);

– src/policy_adapt.py – преквенціальне оцінювання статичної, адаптивної та інкрементної стратегій керування політикою класифікації (підрозділ 3.2);

– src/real_streams.py – завантаження реальних потоків Electricity (Elec2) та INSECTS з позиціями змін концепції з першоджерела;

– src/synth_streams.py – генератори стандартних потоків RandomRBF з раптовим і поступовим дрейфом;

– src/run_e1.py, src/run_e1_real.py, src/run_e1_patch.py – сценарії експерименту з класифікації документів: перехресна валідація запропонованого класифікатора проти п'яти базових методів на корпусах ai4privacy та 20 Newsgroups і сценарій керованості з додаванням одного експертного правила ;

– src/run_e2_cert.py – сценарій експерименту з ідентифікації інсайдерів на наборі CERT r4.2 ;

– src/run_e3_synth.py, src/run_e3_real.py – сценарії експерименту з виявлення дрейфу концепції на потоках RandomRBF та INSECTS ;

– src/run_e4_real.py, src/run_e4_policy.py – сценарії експерименту з еволюційної адаптації політик на реальних потоках і потоках-політиках ;

– src/make_e4_plot.py, src/make_e4_real_plot.py – побудова кривих ковзної преквенціальної точності;

– data/cert/README.md – інструкція щодо розміщення файлів набору CERT Insider Threat r4.2;

– results/ – збережені результати експериментів (e1_real.json, e1_patch.json, e2_cert.json, e3_synth.json, e3_real.json, e4_real.json, e4_policy.json, e4_prequential.png), з яких сформовано таблиці розділу 4 та додатки Б, В, Г;

– digest_ch4.py – формування зведеного текстового дайджесту результатів (results/DIGEST_ch4.txt) для звіряння значень, наведених у розділі 4;

– README.md – опис призначення репозиторію, структури програмного забезпечення, залежностей і порядку відтворення експериментів;

– requirements.txt – перелік бібліотек Python, необхідних для запуску програмного коду.

Для запуску програмного забезпечення рекомендовано використовувати Python 3.12 і встановити залежності командою pip install -r requirements.txt (бібліотеки NumPy, SciPy, scikit-learn, matplotlib, river, datasets). Сценарії експериментів запускаються з кореня репозиторію як модулі Python, наприклад командою python -m src.run_e1_real; в операційній системі Windows перед запуском встановлюється змінна середовища PYTHONUTF8=1.

Корпуси ai4privacy PII-Masking-200k [117] і 20 Newsgroups [118], а також потоки Electricity [121] та INSECTS [120] завантажуються автоматично під час першого запуску відповідних сценаріїв; потоки RandomRBF [122] генеруються програмно. Набір CERT Insider Threat r4.2 [119] не включено до репозиторію через обсяг (близько 5 ГБ); його файли розміщуються в каталозі data/cert/ відповідно до інструкції data/cert/README.md. Усі експерименти використовують фіксовані зерна генераторів випадкових чисел, що забезпечує відтворюваність результатів, які наведені у розділі 4.

ДОДАТОК Е

Структура програмного забезпечення та інструкція користувача

У цьому додатку описано логічну структуру програмного комплексу ЗВДЕАГА [126], показано, як журнал результатів подає інтерпретованість рішень, і наведено коротку інструкцію користувача. Повний вихідний код комплексу розміщено у відкритому репозиторії, опис якого подано в додатку Д.

Е.1 Структура програмного комплексу

Комплекс організовано у чотири логічні рівні. Рівень даних відповідає за постачання матеріалу експериментів: завантажувачі реальних корпусів і потоків (ai4privacy PII-Masking-200k, 20 Newsgroups, CERT Insider Threat r4.2, Electricity, INSECTS), генератори стандартних потоків RandomRBF та конструктор ознакового простору документів, який поєднує TF-IDF-терми, шаблонні ознаки з валідацією за алгоритмом Луна, ключові слова та метадані. Рівень методів містить чотири модулі, що відповідають функціональним модулям моделі ЗВД-системи з підрозділу 2.4: ГА-класифікатор на продукційних правилах (файл src/ga_classifier.py, метод підрозділу 3.1), детектор дрейфу з правилом підтвердження та еволюційну адаптацію політик (файли src/drift.py і src/policy_adapt.py, методи підрозділу 3.2), а також індивідуальні поведінкові профілі з генетичною оптимізацією конфігурації (файл src/behavior.py, метод підрозділу 3.3). Рівень сценаріїв відтворює експерименти підрозділів 4.2.1-4.2.3 (файли src/run_e1_real.py, src/run_e1_patch.py, src/run_e2_cert.py, src/run_e3_synth.py, src/run_e3_real.py, src/run_e4_real.py, src/run_e4_policy.py), а рівень результатів акумулює вихідні дані в каталозі results/ у форматі JSON разом зі зведеним дайджестом DIGEST_ch4.txt, з якого сформовано таблиці розділу 4 та додатків Б, В, Г.

Взаємодію рівнів організовано за конвеєрним принципом: документ або подія передачі даних проходить конструктор ознак, після чого модуль класифікації обчислює рішення за активним набором правил; журнали подій користувачів агрегуються в добові вектори ознак і зіставляються з індивідуальним профілем; потік подій паралельно проходить детектор дрейфу, який за підтвердженого сигналу ініціює еволюційне перенавчання набору правил. Таку саму структуру зафіксовано в описі програмного комплексу в авторському свідоцтві [126].

Е.2 Приклад журналу результатів та інтерпретація рішень

Інтерпретованість рішень забезпечено формою самої моделі. Підсумком навчання є явний перелік продукційних правил, який зберігається в журналі результатів і читається безпосередньо в термінах політик безпеки. Нижче наведено фрагмент зведеного дайджесту результатів (файл results/DIGEST_ch4.txt) для корпусу ai4privacy:

```
GA-rules: F1=0.826±0.013 P=0.791 R=0.765 fit=20.93s
GA інтерпретованість: правил=4.8 умов/правило=1.1
Приклади правил GA-rules:
  IF kw_credential > 0.304 THEN confidential
  IF pat_card_like > 0.0972 THEN confidential
  IF pat_ssn > 0.0556 THEN confidential
  IF pat_crypto > 0.124 THEN confidential
```

Кожен рядок правила є повним поясненням відповідного позитивного рішення. Наприклад, для події наскрізного прикладу з підрозділу 4.2.4 система блокує передачу саме через спрацювання правила IF pat_card_like > 0.0972 THEN confidential, тобто через перевищення порога шаблонної ознаки номера платіжної картки; аналітик бачить цю причину безпосередньо в журналі рішень. В експерименті з керованості (файл results/e1_patch.json) журнал фіксує повноту виявлення документів з новим типом ідентифікатора до та після втручання експерта. Одне дописане правило IF pat_iban > 0 THEN confidential відновлює повноту виявлення до 1,00, тоді як випадковий ліс без додаткової розмітки залишається на рівні 0,23; таке саме відновлення дає і зовнішнє guard-правило над випадковим лісом, тож перевага правил полягає у втручанні всередині самої моделі, а не в недосяжності ефекту іншими засобами (таблиця 4.5, таблиця Б.4).

Е.3 Інструкція користувача

Для роботи з комплексом потрібні Python 3.12 або новіший і бібліотеки NumPy, SciPy, scikit-learn, matplotlib, river та datasets; їх перелік зафіксовано у файлі requirements.txt. Встановлення виконується у три кроки: клонувати репозиторій, створити віртуальне середовище командою python -m venv .venv і активувати його, встановити залежності командою pip install -r requirements.txt. В операційній системі Windows перед запуском слід встановити змінну середовища PYTHONUTF8=1, оскільки сценарії виводять повідомлення українською.

Корпуси ai4privacy і 20 Newsgroups та потоки Electricity й INSECTS завантажуються автоматично під час першого запуску відповідних сценаріїв; потоки RandomRBF генеруються програмно. Набір CERT Insider Threat r4.2 через обсяг (близько 5 ГБ) розміщується вручну в каталозі data/cert/ згідно з інструкцією data/cert/README.md.

Сценарії запускаються з кореня репозиторію як модулі Python: python -m src.run_e1_real (класифікація документів на реальних корпусах), python -m src.run_e1_patch (сценарій експертного правила), python -m src.run_e2_cert (виявлення інсайдерів), python -m src.run_e3_synth та python -m src.run_e3_real (детектори дрейфу), python -m src.run_e4_real та python -m src.run_e4_policy (адаптація політик). Кожен сценарій записує результати у відповідний файл каталогу results/, повторний запуск перезаписує їх; команда python digest_ch4.py формує зведений дайджест для звіряння з таблицями розділу 4. Усі експерименти використовують фіксовані зерна генераторів випадкових чисел, тому результати відтворюються; повний прохід усіх сценаріїв на настільному процесорі триває кілька годин. Щоб додати експертне правило до навченого класифікатора, достатньо повторити крок сценарію run_e1_patch: створити правило з умовою на відповідну ознаку (наприклад, pat_iban > 0) і додати його до списку правил навченої моделі, після чого класифікатор одразу застосовує оновлену політику без перенавчання.

ДОДАТОК Ж

Акти впровадження

«Затверджую»
Директор ТОВ «ІТТ»
Сімогук В.С.
«__» червня 2026 р.

АКТ
про впровадження результатів дисертаційної роботи
Віжевського Петра Володимировича за темою
«Метод та засоби виявлення витоку даних в корпоративних мережах згідно еволюційних алгоритмів»

Результати дисертаційної роботи аспіранта кафедри комп'ютерної інженерії та інформаційних систем Хмельницького національного університету Віжевського П.В. впроваджені в ТОВ «ІТТ».

У процесі впровадження методів та засобів виявлення і запобігання витокам конфіденційної інформації в корпоративній мережі ТОВ «ІТТ». були використані результати, отримані Віжевським П.В. особисто:

1) метод поведінкового профілювання користувачів корпоративної мережі, який будує індивідуальний профіль активності кожного користувача та виявляє відхилення від його персональної норми, що дало змогу фіксувати низькоінтенсивні спроби несанкціонованого виведення даних, непомітні для звичайних засобів виявлення аномалій;

2) метод еволюційної адаптації політик запобігання витокам даних за умов дрейфу концепції, що поєднує статистичний детектор зміни характеру даних із правилом підтвердження сигналу та автоматичним переналаштуванням правил, завдяки чому ефективність захисту підтримується без постійного ручного коригування політик безпеки.

Упровадження зазначених результатів дозволило виявляти аномальну поведінку користувачів, характерну для спроб несанкціонованого виведення даних, та підтримувати ефективність політик безпеки в умовах зміни бізнес-процесів і характеру інформаційних потоків без постійного ручного втручання адміністраторів.

Цей акт не є підставою для фінансових розрахунків.

Заступник директора Іванов О.В.

Системний адміністратор Веремеєнко В.А.

ДОВІДКА

про впровадження результатів дисертаційної роботи
аспіранта кафедри комп'ютерної інженерії та інформаційних систем
Хмельницького національного університету Віжевського Петра Володимировича
за темою «Метод та засоби виявлення витоку даних в корпоративних мережах
згідно еволюційних алгоритмів»

Результати дисертаційної роботи аспіранта кафедри комп'ютерної інженерії та інформаційних систем Хмельницького національного університету Віжевського П.В. впроваджені в ПП «Авіві».

У процесі впровадження методів та засобів виявлення і запобігання витокам конфіденційної інформації в корпоративній мережі ПП «Авіві». були використані результати, одержані Віжевським П.В. особисто:

1) метод класифікації документів за рівнем конфіденційності на основі генетичного алгоритму, який подає правила розпізнавання у прозорій формі IF-THEN, що дало змогу автоматично виявляти й маркувати документи з персональними та фінансовими даними й обґрунтовувати кожне рішення системи під час аудиту інцидентів;

2) програмний комплекс прототипу системи запобігання витокам даних (DLP), реалізований мовою Python, який було використано для контролю інформаційних потоків та виявлення спроб передачі конфіденційної інформації за межі захищеного периметра.

Упровадження зазначених результатів в ПП «Авіві» дозволило підвищити рівень захищеності конфіденційної інформації на 5-8 %, що обробляється в корпоративній мережі підприємства, автоматизувати виявлення документів з конфіденційними даними та знизити ризик їх несанкціонованого виведення в умовах впливів зловмисного програмного забезпечення та комп'ютерних атак.

Ця довідка не є підставою для фінансових розрахунків.

29.06.2026 р.

Директор ПП «Авіві» _____ В. В. Аскеров

ПРИВАТНЕ ПІДПРИЄМСТВО «АВІВІ» • м. Хмельницький • avivi 35605257

«Затверджую»
В.о. ректора Хмельницького
національного університету
Віктор ЛОПАТОВСЬКИЙ
«26» червня 2026 р.

АКТ

про впровадження результатів дисертаційної роботи
Віжевського Петра Володимировича за темою
«Метод та засоби виявлення витоку даних в корпоративних мережах згідно еволюційних алгоритмів»

Ми, комісія в складі: декана факультету інформаційних технологій, професора кафедри комп'ютерної інженерії та інформаційних систем, д.т.н., професора Говорущенко Т. О. (голова комісії), завідувача кафедри комп'ютерної інженерії та інформаційних систем, д.ф., доцента Павлової О. О., доцента кафедри комп'ютерної інженерії та інформаційних систем, к.т.н., доцента Капустян М. В., склали цей акт про те, що результати дисертаційної роботи Віжевського П.В. впроваджено та використовуються в освітньому процесі Хмельницького національного університету на кафедрі комп'ютерної інженерії та інформаційних систем при викладанні навчальних дисциплін «Безпека та захист комп'ютерних систем» та «Методології забезпечення якості, надійності, гарантоздатності та безпеки комп'ютерних систем та мереж».

При викладанні цих дисциплін використовуються наступні матеріали досліджень, одержані Віжевським П.В.:

а) удосконалена комплексна формальна модель системи запобігання витокам даних у корпоративній мережі, яка узгоджено поєднує модель витоків даних, модель корпоративного середовища та модель даних, що дало змогу системно підійти до проєктування архітектури засобів контролю інформаційних потоків;

б) метод класифікації документів за рівнем конфіденційності на основі генетичного алгоритму, який подає правила розпізнавання у явній формі IF-THEN, що дало змогу автоматично виявляти й маркувати документи з персональними та фінансовими даними й обґрунтовувати кожне рішення системи під час аудиту інцидентів.

Упровадження зазначених результатів надало можливість оновити зміст лекційних, практичних та лабораторних занять із дисциплін «Безпека та захист

комп'ютерних систем» та «Методології забезпечення якості, надійності, гарантоздатності та безпеки комп'ютерних систем та мереж», доповнити їх сучасними підходами до побудови систем запобігання витокам даних на основі еволюційних алгоритмів і підвищити рівень практичної підготовки здобувачів вищої освіти зі спеціальності F7 «Комп'ютерна інженерія».

Говорущенко Т. О.
Павлова О. О.
Капустян М. В.