COMPUTER SYSTEMS AND NETWORKS

Vol. 7, No. 2, 2025

DOI: https://doi.org/10.23939/csn2025.02.217

УДК 004.7

СИСТЕМА ВИЯВЛЕННЯ АНОМАЛІЙ ТА МОНІТОРИНГУ ТРАФІКУ В КОМП'ЮТЕРНИХ МЕРЕЖАХ

О. В. Хома, І. Ю. Юрчак, А. О. Хіч

Національний університет "Львівська політехніка",
кафедра систем автоматизованого проектування
E-mail: Oleh.Khoma.kn.2021@lpnu.ua, Iryna.Y.Yurchak@lpnu.ua, Andrii.O.Khich@lpnu.ua

Розглянуто проблему виявлення аномалій у мережному трафіку та запропоновано комплексне рішення для підвищення рівня кібербезпеки організацій різного масштабу. Здійснено порівняльний аналіз наявних систем моніторингу та виявлення аномалій, включаючи як відкриті рішення, так і комерційні продукти. На основі проведеного дослідження обґрунтовано вибір технологічного стека для реалізації системи виявлення аномалій у мережному трафіку, зокрема оптимальне поєднання бібліотек Python для збору та обробки мережного трафіку, підготовки даних, застосування алгоритмів машинного навчання та візуалізації результатів.

Розроблено повнофункціональну систему з п'ятьох основних компонентів: модуля збору мережного трафіку на базі PyShark, сервісу обробки пакетів для агрегації потоків, модуля виявлення аномалій із використанням алгоритмів машинного навчання, API-модуля на базі FastAPI та вебінтерфейсу користувача на React. Система забезпечує високу точність виявлення потенційних загроз, оптимізацію використання обчислювальних ресурсів та зручний аналіз стану мережі.

Проведено експериментальне дослідження на наборі даних UNSW-NB15, що містить понад 2,5 мільйона записів мережних з'єднань. Здійснено порівняльний аналіз трьох алгоритмів машинного навчання: випадковий ліс, метод опорних векторів та логістична регресія. Випадковий ліс продемонстрував найкращі результати з точністю 94 % на тестовій вибірці та площею під ROC-кривою 0,99, значно перевищуючи показники альтернативних алгоритмів.

Особливістю запропонованого підходу є інтеграція методів машинного навчання з комплексним аналізом різноманітних параметрів мережного трафіку, що дає змогу значно підвищити точність виявлення аномалій та мінімізувати кількість хибних спрацьовувань порівняно з наявними рішеннями.

Ключові слова: аналіз даних, виявлення аномалій, кібербезпека, машинне навчання, мережна безпека, мережний трафік, python, random forest.

Вступ

У сучасному світі комп'ютерні мережі стали критичною інфраструктурою для організацій всіх масштабів, забезпечуючи функціонування бізнес-процесів, комунікацій та доступ до інформаційних ресурсів. Однак разом із розвитком мережних технологій зростає й кількість та складність кібератак, що становлять серйозну загрозу для безпеки інформаційних систем.

Сучасний стан систем виявлення аномалій та моніторингу мережного трафіку характеризується значною різноманітністю підходів та технологій. Найпоширенішими є традиційні системи виявлення

вторгнень (Intrusion Detection System, IDS) та системи запобігання вторгненням (Intrusion Prevention System, IPS), які базуються на сигнатурному аналізі; системи аналізу мережного трафіку (Network Traffic Analysis, NTA), що застосовують методи машинного навчання; платформи управління інформаційною безпекою (Security Information and Event Management, SIEM); та спеціалізовані рішення для моніторингу продуктивності мережі (Network Performance Monitor, NPM).

Технічні виклики в цій галузі пов'язані зі зростанням обсягів мережного трафіку, ускладненням мережних інфраструктур через впровадження гібридних та хмарних середовищ та еволюцією кібератак, що стають дедалі складнішими та витонченішими.

Проблема виявлення аномалій та моніторингу трафіку в комп'ютерних мережах набуває особливої актуальності в умовах стрімкого розвитку цифрових технологій та зростання кількості кібератак. Сучасні кіберзагрози характеризуються високою складністю, стрімкою еволюцією та здатністю обходити традиційні механізми захисту, що робить їх виявлення критично важливим завданням для забезпечення безпеки інформаційних систем.

Кардинальна відмінність запропонованого підходу полягає в інтеграції методів машинного навчання з комплексним аналізом різноманітних параметрів мережного трафіку. На відміну від існуючих рішень, які зосереджуються переважно на окремих аспектах мережної активності, розроблена система враховує широкий спектр характеристик з'єднань – від базових показників (IP-адреси, порти, протоколи) до складних статистичних метрик (варіації часу між пакетами, особливості TCP-з'єднань, розміри вікон та інші). Це дає змогу значно підвищити точність виявлення аномалій та мінімізувати кількість хибних спрацьовувань.

Особливої актуальності статті надає зростаюча потреба в системах, які поєднують ефективність виявлення аномалій з доступністю для широкого кола організацій та зручністю використання спеціалістами різного рівня кваліфікації. Розробка такої системи з відкритою архітектурою та зрозумілим користувацьким інтерфейсом дасть змогу подолати бар'єр між складними технологіями аналізу даних та практичними потребами мережних адміністраторів.

Наведено результат створення повнофункціональної програмної системи, яка дасть змогу ефективно виявляти різноманітні типи аномалій у мережному трафіку, надаватиме інформацію про стан мережі та виявлені загрози, а також забезпечуватиме можливість оперативного реагування на потенційні порушення кібербезпеки.

Огляд літературних джерел

Проблема виявлення аномалій у мережному трафіку є однією з ключових у сфері кібербезпеки та мережного адміністрування, оскільки від її вирішення залежить своєчасне виявлення атак, зловмисних дій та порушень у роботі інформаційних систем. Упродовж останніх років запропоновано низку підходів, що охоплюють як класичні методи машинного навчання, так і сучасні глибинні нейронні мережі.

Зокрема, у праці [1] досліджено застосування згорткових нейронних мереж для виявлення аномалій у трафіку з використанням датасету UNSW-NB15. Автори показали, що CNN дає змогу досягти високої точності класифікації завдяки автоматичному витягуванню ознак з вхідних даних. Подібний напрям розвинуто в статті [2], автори запропонували вдосконалену автоенкодерну архітектуру для побудови системи виявлення вторгнень та довели перевагу глибокого навчання над класичними моделями, зокрема опорних векторів та логістичною регресією.

Інший підхід запропонували автори [3], які поєднали автоенкодери з LSTM- та CNN-архітектурами у гібридну модель, що дала змогу підвищити точність класифікації та зменшити кількість хибних спрацьовувань. Важливим напрямом також є оптимізація класичних алгоритмів: у праці [4] удосконалено метод випадкового лісу для багатокласового виявлення аномального трафіку.

Сучасні тенденції зосереджені на розробці методів, здатних працювати з обмежено розміченими або зашифрованими даними. Автори статті [5] запропонували фреймворк SAFE на основі самоконтрольованого підходу, який перетворює табличні дані мережного трафіку на зображення для подальшого аналізу за допомогою автоенкодера. У праці [6] висвітлено використання графових

нейронних мереж для самонавчального витягування ознак трафіку, що дозволило враховувати топологію зв'язків між вузлами. Автори статті [7] розробили гібридну ансамблеву модель, що орієнтована на баланс між точністю та захистом приватності даних. У статті [8] запропоновано підхід до інтерпретації результатів виявлення аномалій у зашифрованому трафіку на основі SHAP-метрик, що підвищує прозорість роботи моделей.

Аналіз наукових праць свідчить про активний розвиток напряму виявлення аномалій у мережному трафіку. Проте залишаються проблеми зменшення кількості хибнопозитивних спрацьовувань, підвищення здатності працювати із зашифрованими потоками та інтеграції методів машинного навчання у повнофункціональні системи моніторингу. Це обґрунтовує актуальність подальших досліджень у цій сфері.

Проблема виявлення аномалій у мережному трафіку

Аномалії мережного трафіку – це відхилення від нормальної, очікуваної поведінки мережі, які можуть свідчити про наявність загроз, технічні несправності або неефективні налаштування мережного обладнання.

Основна складність виявлення аномалій полягає у необхідності точного визначення "нормальної" поведінки мережі та автоматичної ідентифікації значних відхилень від цієї норми. Це комплексне завдання, що ускладнене кількома ключовими факторами.

Нормальна поведінка мережі не є статичною. Вона змінюється з часом відповідно до корпоративної активності, робочих графіків, сезонних коливань попиту на послуги та еволюції мережної інфраструктури. Така динамічність суттєво ускладнює формування статичних моделей нормальної поведінки.

Сучасні мережі характеризуються надзвичайно високим обсягом та різноманітністю трафіку. Корпоративна мережа середнього підприємства може генерувати терабайти даних щодня, що містять мільйони мережних з'єднань різних типів. Обробка та аналіз такої кількості інформації в режимі реального часу потребує значних обчислювальних ресурсів та ефективних алгоритмів.

Проблема виявлення аномалій у мережному трафіку постійно загострюється з кількох взаємопов'язаних причин. Зростаюча складність мережних інфраструктур є одним із ключових факторів. Сучасні організації впроваджують гібридні та хмарні середовища, програмно-визначені мережі, віртуалізацію мережних функцій та контейнеризовані додатки. За даними дослідження [8], приблизно 80 % підприємств вже мають стратегію гібридної хмари, а за прогнозами IDC, до 2027 року 65 % найбільших світових компаній використовуватимуть мультихмарну інфраструктуру через партнерські екосистеми, що суттєво розширює площу потенційних атак та ускладнює моніторинг [9, 10].

Паралельно з цим відбувається еволюція кіберзагроз. Методи кібератак постійно вдосконалюються, стаючи більш цілеспрямованими та складним. Кібератаки Постійна Серйозна Загроза (Advanced Persistent Threats, APT) можуть залишатися непоміченими в мережі протягом місяців, повільно переміщуючись між системами та збираючи дані.

Збільшення обсягів мережного трафіку також суттєво ускладнює проблему. За даними Cisco Visual Networking Index (VNI), щорічний глобальний трафік досягнув 4,8 зеттабайт у 2022 році, що втричі перевищує показники попередніх років. Такий швидкий ріст ускладнює моніторинг та аналіз трафіку, створюючи умови, де аномалії легше приховати [11].

Поширення IoT-пристроїв створює додаткові виклики. Інтеграція мільярдів під'єднаних пристроїв до корпоративних мереж створює нові вектори атак та генерує специфічні типи трафіку, які важко аналізувати традиційними методами.

Своєчасне виявлення аномалій у мережному трафіку є критично важливим для сучасних організацій. Насамперед це питання захисту від економічних втрат. За даними дослідження IBM "Cost of a Data Breach Report 2024", середня вартість витоку даних для компаній сягає 4,88 мільйона доларів, що на 10 % більше порівняно з 2023 роком [12]. Ефективне виявлення аномалій дає змогу запобігти значним фінансовим збиткам, пов'язаним з порушеннями безпеки. Крім того, публічне розкриття інформації про порушення безпеки може суттєво вплинути на репутацію компанії.

Не менш важливим є забезпечення безперервності бізнес-процесів. Мережні аномалії, навіть якщо вони не пов'язані з кібератаками, можуть призводити до збоїв у роботі критичних сервісів та застосунків. За оцінками Gartner, середня вартість IT-простою становить $5,600 за хвилину, що в перерахунку на годину може становити від $140,000 до $540,000 залежно від галузі та розміру бізнесу, а для великих підприємств ця сума може сягати понад $1 мільйон за годину [13].

Оптимізація використання мережних ресурсів є додатковою перевагою. Своєчасне виявлення та реагування на аномалії в мережному трафіку дає змогу оптимізувати використання пропускної здатності мережі та зменшити навантаження на мережне обладнання, що призводить до суттєвої економії коштів на інфраструктуру. Підвищення пропускної здатності мережі, зменшення затримок та покращення якості сервісів – це особливо критично для організацій, що надають онлайн-послуги, де затримки в мілісекунди можуть призводити до втрати клієнтів.

Залежно від походження мережні аномалії можна розділити на зловмисні та незловмисні. Зловмисні аномалії є результатом навмисних дій кіберзлочинців і охоплюють різноманітні типи кібератак. Такі аномалії спрямовані на порушення конфіденційності, цілісності та доступності інформаційних ресурсів організації. До цієї категорії належать атаки на відмову в обслуговуванні, спроби несанкціонованого доступу, розповсюдження шкідливого програмного забезпечення та інші форми зловмисної активності.

Незловмисні аномалії виникають внаслідок технічних несправностей, помилок конфігурації, збоїв програмного забезпечення або непередбачених змін у характері використання мережі. Хоча такі аномалії не пов'язані зі зловмисними намірами, вони також можуть суттєво впливати на продуктивність мережі та якість надання послуг. Прикладами незловмисних аномалій можуть бути перевантаження мережних сегментів через некоректні налаштування маршрутизації, проблеми в роботі DNS-серверів або раптове збільшення трафіку через певні бізнес-події.

Найпоширенішими типами зловмисних мережних аномалій є атаки відмови в обслуговуванні (DoS) та розподілені атаки відмови в обслуговуванні (DDoS). Ці атаки спрямовані на порушення доступності сервісів шляхом їх перевантаження або виснаження ресурсів. DDoS-атаки характеризуються раптовим збільшенням вхідного трафіку, зазвичай з великої кількості джерел, що призводить до вичерпання ресурсів систем. Сучасні DDoS-атаки стають дедалі складнішими, поєднуючи різні типи трафіку та адаптуючись до захисних механізмів.

Сучасні підходи до виявлення аномалій в мережному трафіку

Фундаментальним підходом до виявлення аномалій залишається статистичний аналіз. Його суть полягає у встановленні базових характеристик нормального трафіку та ідентифікації відхилень від них. Сучасні статистичні методи вийшли далеко за межі простого порівняння з фіксованими пороговими значеннями.

Адаптивні порогові методи автоматично коригують граничні значення на основі поточних спостережень та історичних даних. Замість статичних порогів використовуються динамічні моделі, що враховують сезонність, денні цикли та тренди у мережному трафіку. Наприклад, система може автоматично збільшувати допустимий поріг активності під час робочих годин і зменшувати його вночі або у вихідні дні.

Багатовимірний статистичний аналіз розглядає взаємозв'язки між різними шаблонами мережного трафіку. Такі підходи дають змогу виявляти складні аномалії, які не проявляються в окремих частинах даних, але помітні під час розгляду кореляцій між ними. Прикладами таких методів є аналіз головних компонент (PCA) та метод Махаланобіса для оцінки відстаней у багатовимірному просторі ознак.

Для виявлення мережних аномалій потужним інструментом стало машинне навчання, завдяки своїй здатності автоматично виявляти приховані закономірності в даних. Також застосовуються методи глибокого навчання завдяки їхній здатності автоматично витягувати складні ознаки з необроблених даних.

Для виявлення локальних шаблонів в трафіку залучають згорткові нейронні мережі (CNN), що ефективно працюють з часовими рядами мережних даних.

Для виявлення аномалій, що розвиваються з часом, застосовують рекурентні нейронні мережі (RNN), особливо їх варіанти LSTM (Long Short-Term Memory) та GRU (Gated Recurrent Unit), що спеціалізуються на моделюванні послідовних даних. Вони здатні враховувати часові залежності в мережному трафіку.

Глибокі автоенкодери навчаються стискати дані про нормальний трафік до низьковимірного представлення, а потім відновлювати їх. Аномалії виявляються через високу помилку реконструкції, оскільки модель не була навчена відтворювати аномальні зразки. Варіаційні автоенкодери (VAE) розширюють цю концепцію, моделюючи ймовірнісні розподіли нормальних даних.

Поведінковий аналіз фокусується на виявленні відхилень від встановлених шаблонів поведінки користувачів, систем та мережних сервісів. Цей підхід передбачає створення профілів нормальної активності для різних сутностей мережі та ідентифікацію відхилень від цих профілів.

Профілювання користувачів базується на аналізі типових шаблонів доступу до ресурсів, часу активності, використовуваних сервісів та обсягів генерованого трафіку. Відхилення від встановленого профілю може вказувати на компрометацію облікового запису або інсайдерську загрозу.

Поведінковий аналіз систем досліджує типові шаблони взаємодії між різними компонентами мережі. Це дає змогу виявляти аномальну активність, таку як нетипові з'єднання між системами, що раніше не взаємодіяли, або зміни в стандартних шаблонах комунікації.

Аналіз мережних протоколів зосереджується на виявленні відхилень від стандартної поведінки протоколів. Такий підхід ефективний для ідентифікації спроб використання нестандартних протоколів або спроб обходу мережних фільтрів.

Також сучасні системи виявлення аномалій часто використовують гібридні підходи, що поєднують переваги різних методів. Інтеграція сигнатурного аналізу з поведінковими методами дає змогу ефективно виявляти як відомі загрози, так і нові або модифіковані атаки. Комбінація методів машинного навчання з експертними правилами дозволяє збалансувати автоматизацію з людською експертизою.

Ансамблеві методи, що поєднують прогнози кількох моделей, демонструють вищу точність та стійкість порівняно з окремими алгоритмами. Вони можуть інтегрувати різнорідні джерела даних, такі як аналіз пакетів чи системні журнали.

Постановка задачі

Розробка ефективної системи виявлення аномалій у мережному трафіку потребує ретельного підходу до вибору інструментів та технологій, які відповідатимуть поставленим цілям та обмеженням проєкту. Цей процес має базуватися на чітко визначених критеріях, що враховують як функціональні, так і нефункціональні вимоги до кінцевого продукту.

Одним із найкритичніших параметрів для систем виявлення аномалій є продуктивність обробки мережного трафіку. Сучасні корпоративні мережі генерують гігабіти даних за секунду, і система має аналізувати цей трафік з мінімальною затримкою для своєчасного виявлення потенційних загроз.

Для забезпечення необхідного рівня продуктивності потрібно враховувати швидкість обробки пакетів, яка визначає кількість мережних пакетів, які система може проаналізувати за одиницю часу. Для корпоративних мереж середнього розміру необхідна пропускна здатність від 1 до 10 Гбіт/с. Обрані технології мають забезпечувати можливість оптимізації алгоритмів для досягнення максимальної швидкості обробки з урахуванням доступних обчислювальних ресурсів.

Час відгуку системи критично важливий для виявлення загроз у режимі реального часу. Затримка між надходженням підозрілого трафіку та генерацією відповідного сповіщення має бути мінімальною – у межах кількох секунд або навіть мілісекунд залежно від типу загрози. Інструменти для розробки мають підтримувати асинхронну обробку даних та ефективний паралелізм.

Ефективність використання ресурсів визначає, наскільки оптимально система використовує доступні обчислювальні потужності, пам'ять та сховище даних. Враховуючи безперервний характер моніторингу мережного трафіку, надмірне споживання ресурсів може призвести до значних витрат. Обрані технології мають забезпечувати баланс між функціональністю та ресурсоємністю.

Ефективна система виявлення аномалій повинна адаптуватися до змін у мережній інфраструктурі та зростання обсягів трафіку без необхідності повного переосмислення архітектури. Масштабованість є ключовим критерієм, що гарантує довгострокову цінність розробленого рішення.

Горизонтальна масштабованість передбачає можливість збільшення продуктивності системи шляхом додавання нових обчислювальних вузлів. Обрані технології мають підтримувати розподілену архітектуру з ефективними механізмами розподілу навантаження та синхронізації стану між компонентами. Це особливо важливо для обробки мережного трафіку, оскільки дає змогу розподілити аналіз різних сегментів мережі на окремі обчислювальні ресурси.

Вертикальна масштабованість стосується можливості підвищення продуктивності системи шляхом збільшення потужності окремих компонентів, наприклад, шляхом додавання процесорів, пам'яті або використання спеціалізованого обладнання для обробки даних. Вибрані технології мають ефективно використовувати доступні ресурси та демонструвати лінійне зростання продуктивності під час розширення апаратного забезпечення.

Адаптивність до змін навантаження є важливим аспектом масштабованості, оскільки мережний трафік характеризується значними коливаннями інтенсивності протягом дня. Система має динамічно розподіляти ресурси відповідно до поточного навантаження, забезпечуючи баланс між продуктивністю та ефективністю використання ресурсів.

Основною метою системи є точне виявлення аномалій у мережному трафіку, тому критерії, пов'язані з ефективністю алгоритмів аналізу, відіграють вирішальну роль у виборі інструментів та технологій.

Збалансована точність виявлення передбачає мінімізацію як хибнопозитивних, так і хибнонегативних результатів. Хибнопозитивні спрацьовування (помилкове виявлення аномалій у нормальному трафіку) призводять до надмірного навантаження на команду безпеки та зниження довіри до системи. Хибнонегативні результати (пропуск реальних аномалій) становлять безпосередню загрозу безпеці мережі. Обрані алгоритми машинного навчання мають забезпечувати оптимальний баланс між цими двома типами помилок.

Архітектура системи виявлення аномалій

Система складається з п'яти основних компонентів, які взаємодіють між собою через чітко визначені інтерфейси. Модуль збору мережного трафіку захоплює пакети з мережного інтерфейсу та передає їх до модуля обробки пакетів. Цей модуль агрегує окремі пакети у логічні потоки та обчислює статистичні характеристики, після чого зберігає результати у базі даних (рис. 1).

Модуль виявлення аномалій періодично опитує базу даних для отримання необроблених записів, застосовує до них натреновану модель машинного навчання та оновлює записи з результатами класифікації. API-модуль забезпечує доступ до обробленої інформації через REST-інтерфейс, надаючи дані для вебінтерфейсу користувача.

Взаємодія між компонентами відбувається асинхронно, що дає змогу кожному модулю працювати у власному ритмі. Збір трафіку та обробка пакетів відбуваються в режимі реального часу, тоді як виявлення аномалій може працювати з певними інтервалами, обробляючи дані пакетами для оптимізації продуктивності.

Опис основних модулів та їх призначення

Модуль збору мережного трафіку реалізовано у класі PacketCollector і відповідає за захоплення пакетів з мережного інтерфейсу. Цей компонент використовує бібліотеку PyShark для взаємодії з мережними адаптерами та отримання детальної інформації про кожен пакет. Модуль забезпечує безперервний моніторинг мережної активності та передачу захоплених пакетів до наступного рівня обробки.

Сервіс обробки пакетів перетворює потік окремих пакетів у структуровані мережні потоки. Цей модуль відстежує стан TCP-з'єднань, агрегує статистичну інформацію про передачу даних та визначає моменти завершення потоків. Цей компонент обробляє як TCP, так і UDP трафік, враховуючи специфіку кожного протоколу.

Модуль виявлення аномалій інтегрує натреновану модель машинного навчання в загальну архітектуру системи. Цей компонент періодично обробляє накопичені дані, застосовуючи до них алгоритм класифікації та позначаючи підозрілі потоки як аномальні. Модуль працює у пакетному режимі, що робить його використання обчислювальних ресурсів ефективним.

API-модуль забезпечує програмний інтерфейс для доступу до даних системи. Він надає різноманітні кінцеві точки для отримання статистичної інформації, виявлених аномалій та аналітичних даних для візуалізації.

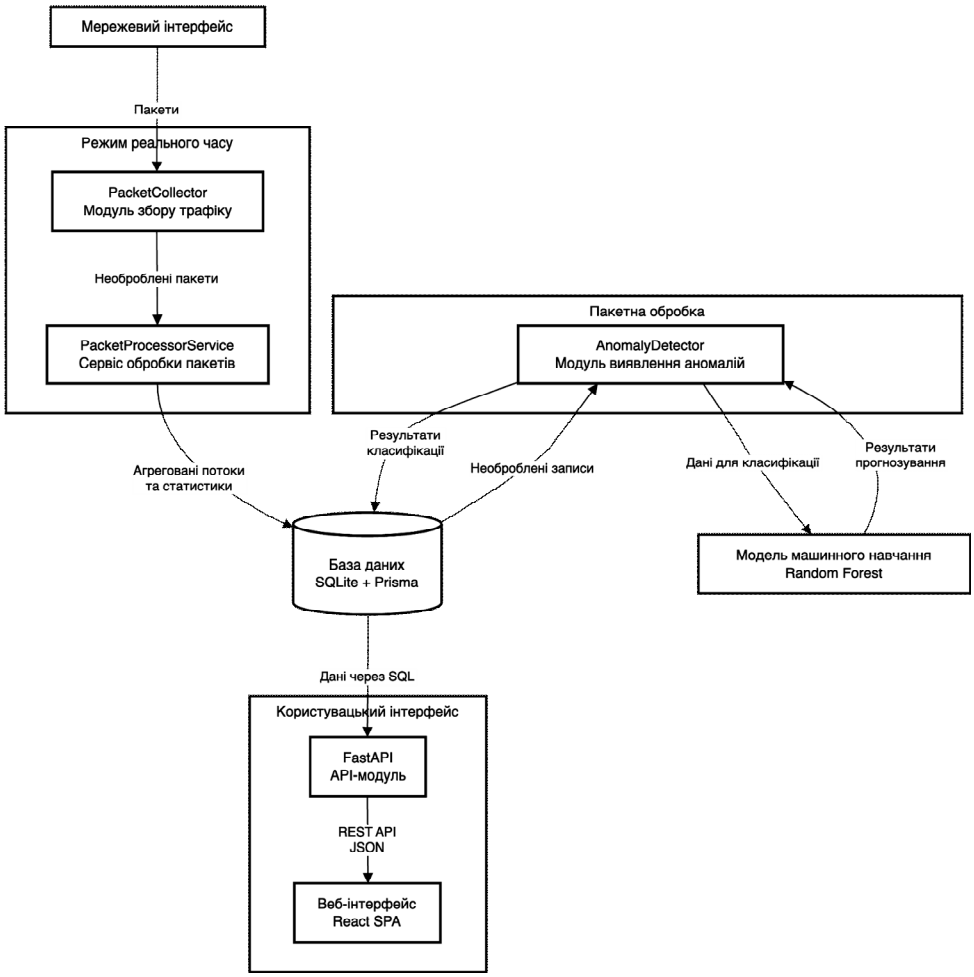

Рис. 1. Загальна схема системи та взаємодії компонентів

Вебінтерфейс користувача реалізовано як односторінковий додаток, що взаємодіє з API для відображення інформації про стан мережі. Інтерфейс містить інтерактивні графіки, таблиці з детальною інформацією про потоки та панелі моніторингу для оперативного контролю ситуації.

Технологічний стек

Як систему управління базами даних обрано SQLite з використанням ORM Prisma. SQLite забезпечує достатню продуктивність для системи середнього масштабу, не потребує окремого серверного процесу та спрощує розгортання системи. Prisma надає зручний, надійний та типізований інтерфейс для роботи з базою даних та автоматичну генерацію схеми.

FastAPI обрано для реалізації API через його високу продуктивність, автоматичну генерацію документації та підтримку асинхронного програмування без потреби додаткового налаштування. Ця бібліотека забезпечує швидку розробку надійних API з мінімальною кількістю коду, що значно пришвидшує процес самої розробки.

Для машинного навчання використовується scikit-learn як основна бібліотека, що надає широкий спектр алгоритмів з уніфікованим інтерфейсом. Додатково залучаються NumPy та Pandas для ефективної обробки числових даних та роботи з часовими рядами.

Схема потоку даних у системі

Потік даних у системі починається із захоплення пакетів мережним інтерфейсом. Модуль PacketCollector безперервно отримує пакети та передає їх до PacketProcessorService для обробки. На цьому етапі відбувається витягування ключової інформації із заголовків протоколів та початкове групування пакетів за характеристиками з'єднань.

PacketProcessorService агрегує пакети у логічні потоки, відстежуючи стан кожного з'єднання та обчислюючи статистичні характеристики. Коли потік вважається завершеним за певними критеріями, його дані передаються до бази даних через NetworkFlowRepo. На цьому етапі кожен потік отримує унікальний ідентифікатор та позначається як необроблений.

AnomalyDetector періодично сканує базу даних на наявність необроблених записів. Виявлені записи передаються до ModelService, що завантажує натреновану модель та застосовує її для класифікації. Результати класифікації записуються назад у базу даних, позначаючи кожен потік як нормальний або аномальний.

API-модуль отримує запити від вебінтерфейсу та виконує відповідні запити до бази даних. Дані обробляються та агрегуються відповідно до потреб конкретної кінцевої точки, після чого повертаються у JSON-форматі. Вебінтерфейс отримує ці дані та відображає їх у вигляді графіків, таблиць та інших візуальних елементів.

Підготовка та аналіз навчальних даних

Для навчання та тестування системи виявлення аномалій обрано набір даних UNSW-NB15, який розробили дослідники Університету Нового Південного Уельсу [14]. Цей датасет являє собою сучасну колекцію мережного трафіку, що містить як нормальну активність, так і різноманітні типи кібератак. На відміну від застарілих наборів даних UNSW-NB15 містить реалістичні сучасні атаки та відображає актуальні шаблони мережного трафіку.

Обраний набір даних містить понад 2,5 мільйона записів мережних з'єднань, кожне з яких характеризується 49 різними ознаками. Ці ознаки охоплюють базові характеристики мережних з'єднань, статистичні показники потоків даних, часові характеристики та додаткові метрики, що дають змогу детально аналізувати поведінку мережного трафіку. Датасет містить дев'ять категорій атак: Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode та Worms, що забезпечує широке покриття різних типів кіберзагроз.

Вибір та інженерія ознак

Процес вибору ознак базувався на аналізі їх релевантності для задачі виявлення аномалій та обчислювальної ефективності системи. З початкових 49 ознак було відібрано 22 найзначущіші характеристики, які забезпечують оптимальний баланс між точністю виявлення та швидкодією системи.

До вибраних ознак увійшли основні характеристики з'єднань, такі як тривалість (duration), протокол (protocol), сервіс (service) та стан з'єднання (state). Статистичні характеристики потоків наведено кількістю пакетів у кожному напрямку (spkts, dpkts), обсягом переданих даних (sbytes, dbytes), швидкістю передачі (rate), навантаженням на відправника та отримувача (sload, dload). Часові характеристики містять інтервали між пакетами (sinpkt, dinpkt) та варіації часових затримок (sjit, djit) (рис. 2).

	duration	protocol	service	state	spkts	dpkts	sbytes	dbytes	rate	sload	...	sjit	djit	swin	dwin
0	0.121478	TCP	-	FIN	6	4	258	172	74.087490	14158.9420	...	30.177547	11.830604	255	255
1	0.649902	TCP	-	FIN	14	38	734	42014	78.473370	8395.1120	...	61.426933	1387.778300	255	255
2	1.623129	TCP	-	FIN	8	16	364	13186	14.170161	1572.2719	...	17179.586000	11420.926000	255	255
3	1.681642	TCP	FTP	FIN	12	12	628	770	13.677108	2740.1790	...	259.080170	4991.784700	255	255
4	0.449454	TCP	-	FIN	10	6	534	268	33.373825	8561.4990	...	2415.837600	115.807000	255	255

Рис. 2. Візуалізація перших п'яти записів з набору даних UNSW-NB15

Додатково до аналізу додано характеристики TCP-з'єднань, такі як розміри вікон (swin, dwin), базові послідовності TCP (stcpb, dtcpb), час кругового обороту TCP (tcprtt) та середні розміри пакетів (smean, dmean). Ці ознаки дають змогу виявляти аномалії на рівні транспортного протоколу та ідентифікувати нетипову поведінку TCP-з'єднань.

Попередня обробка та очищення даних

Попередня обробка даних розпочалася з видалення записів з відсутніми значеннями для забезпечення цілісності датасету. Хоча датасет UNSW-NB15 характеризується високою якістю, деякі записи можуть містити неповну інформацію, що могло б негативно вплинути на процес навчання моделі.

Категоріальні ознаки потребують спеціальної обробки для перетворення текстових значень у числовий формат, придатний для алгоритмів машинного навчання. Для ознак protocol, service та state застосовано Label Encoder з бібліотеки scikit-learn. Цей підхід привласнює кожному унікальному значенню категоріальної змінної числовий код, зберігаючи водночас інформацію про всі можливі варіанти для подальшого використання в системі.

Процес кодування категоріальних ознак потребує збереження кодувальників для забезпечення постійності між навчанням та застосуванням моделі. Кожен з них зберігається окремо, що дає змогу правильно обробляти нові дані з тими самими категоріями, які використовувалися під час навчання.

Числові ознаки потребують нормалізації через значні відмінності в масштабах різних характеристик. Наприклад, кількість байтів може вимірюватися мільйонами, а тривалість з'єднання часто становить долі секунди. Для нормалізації застосовано MinMaxScaler, який масштабує всі числові ознаки до діапазону від 0 до 1. Такий підхід зберігає розподіл даних та забезпечує рівний внесок всіх ознак у процес навчання моделі.

Аналіз матриці кореляцій

Для виявлення взаємозв'язків між ознаками та потенційними проблемами мультиколінеарності побудовано матрицю кореляцій Пірсона. Це квадратна таблиця, яка відображає коефіцієнти кореляції між кожною парою змінних у наборі даних, де коефіцієнт кореляції Пірсона вимірює силу лінійного зв'язку між двома змінними і може приймати значення від -1 до +1, де -1 означає повну негативну кореляцію, 0 – відсутність лінійного зв'язку, а +1 – повну позитивну кореляцію. Мультиколінеарність – це статистичний феномен, за якого дві або більше незалежних змінних демонструють сильну лінійну залежність між собою, що може призводити до нестабільності моделей машинного навчання, зниження їх інтерпретованості та появи проблем з узагальненням на нових даних. Візуалізація матриці здійснювалася за допомогою теплової карти за допомогою бібліотеки Seaborn (рис. 3).

Аналіз матриці кореляцій виявив кілька груп сильно корельованих ознак. Найвищу кореляцію (0,98) демонструють ознаки swin та dwin, що представляють розміри TCP-вікон у різних напрямках з'єднання. Це логічно, оскільки розміри вікон часто синхронізуються в межах одного TCP-з'єднання. Також високу кореляцію (0,97) показують ознаки sbytes та dbytes, що відображають обсяги переданих даних у кожному напрямку.

Значний зв'язок (0,78) демонструють базові послідовності TCP stcpb та dtcpb, що також є природним через взаємозалежність TCP-параметрів у межах одного з'єднання. Помірну кореляцію (0,69) показують ознаки варіації часових затримок sjit та djit, що відображає схожість у поведінці джитера в обох напрямках передачі. Джиттер – це варіація затримки доставки пакетів у мережі, яка вимірює нестабільність часових інтервалів між надходженням послідовних пакетів і є важливим показником якості мережного з'єднання.

Цікавим спостереженням є порівняно низька кореляція між середніми розмірами пакетів smean та dmean (0,43–0,45), що вказує на можливу асиметрію в розмірах пакетів різних напрямків залежно від типу трафіку та застосунків.

Для виявлення критично високої кореляції встановлено поріг у 0,9, щоб ідентифікувати три пари ознак з потенційними проблемами мультиколінеарності: swin-dwin (0,98), sbytes-dbytes (0,97) та частково stcpb-dtcpb (0,78). Попри виявлення цих високих кореляцій, рішення про видалення ознак не приймалося автоматично, оскільки навіть корельовані ознаки можуть нести унікальну інформацію для задачі виявлення аномалій.

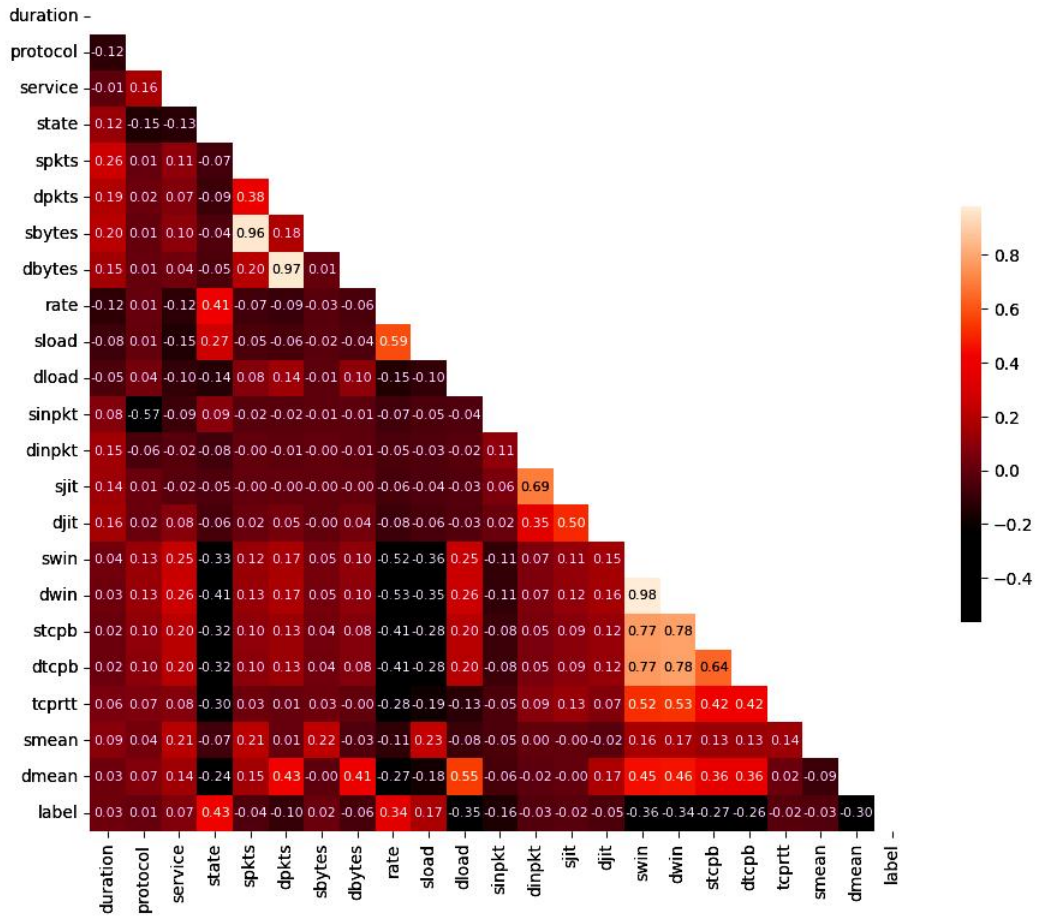

Рис. 3. Графічне зображення матриці кореляцій Пірсона

Розділення на навчальну та тестову вибірки

Для забезпечення об'єктивної оцінки продуктивності моделі набір даних розділено на навчальну та тестову вибірки у співвідношенні 70 % до 30 % відповідно. Розділення здійснювалося з використанням стратифікованої вибірки – методу, що гарантує збереження пропорцій між класами в обох підмножинах.

Стратифікована вибірка важлива для наборів даних з нерівномірним розподілом класів, якими часто характеризуються дані про кібербезпеку і обраний набір даних не є винятком. Такий підхід забезпечує репрезентативність як навчальної, так і тестової вибірок. Це підвищує надійність оцінки продуктивності моделі.

Аналіз та навчання моделі

Класифікатор випадкового лісу (Random Forest Algorithm, RFA) належить до ансамблевих методів машинного навчання, що базуються на принципі об'єднання множини простих моделей для створення більш потужного та стабільного класифікатора [15]. Алгоритм створює велику кількість дерев рішень, кожне з яких навчається на випадковій підвибірці даних та використовує випадковий набір ознак для кожного розділення. Фінальне рішення приймається шляхом голосування всіх дерев, що значно зменшує ризик перенавчання та підвищує узагальнювальну здатність моделі.

Також перевагою алгоритму випадкового лісу є його стійкість до шуму в даних та здатність ефективно працювати з великими наборами ознак без необхідності їх попереднього відбору. Алгоритм автоматично оцінює важливість кожної ознаки, що дає змогу краще зрозуміти, які характеристики мережного трафіку найбільше впливають на виявлення аномалій. Випадковий ліс демонструє високу продуктивність на практичних задачах. Має порівняну простоту в налаштуванні гіперпараметрів.

Метод опорних векторів (Support Vector Machine, SVM) – алгоритм, що знаходить оптимальну гіперплощину для розділення класів у багатовимірному просторі ознак [16]. Метод особливо ефективний для задач з високою розмірністю та може використовувати різні ядрові функції для роботи з нелінійно розділюваними даними. Алгоритм демонструє хорошу узагальнювальну здатність, може ефективно працювати навіть з порівняно невеликими наборами даних.

Логістична регресія (Logistic Regression) є лінійним алгоритмом, що використовує логістичну функцію для моделювання ймовірності належності до певного класу. Попри свою простоту, цей алгоритм часто демонструє хороші результати на практичних задачах, забезпечує високу інтерпретованість результатів. Завдяки швидкості навчання та прогнозування логістична регресія добре підходить для систем, що працюють з даними в реальному часі.

Детальний опис алгоритму випадкового лісу

Як було описано раніше, випадковий ліс будує ансамбль з великої кількості дерев рішень, де кожне дерево навчається на вибірці з оригінального навчального набору. Ця вибірка створюється шляхом випадкового відбору зразків з поверненням, що означає, що деякі зразки можуть потрапити до вибірки кілька разів, а інші взагалі не увійти до неї. Цей процес забезпечує різноманітність між окремими деревами ансамблю.

Також до різноманітності на рівні даних алгоритм застосовує випадковість на рівні ознак. За кожного розділення вузла дерева розглядається лише випадкова підмножина всіх доступних ознак, що запобігає домінуванню найбільш інформативних ознак та сприяє створенню більш збалансованих дерев. Кількість ознак, що розглядаються на кожному кроці, зазвичай становить квадратний корінь від загальної кількості ознак для задач класифікації.

Процес прогнозування у випадковому лісі відбувається шляхом агрегації результатів всіх дерев ансамблю. Для задач класифікації кожне дерево голосує за певний клас, а фінальне рішення приймається на основі принципу більшості. Завдяки такій схемі отримані прогнози є стабільними, а вплив окремих помилкових рішень зменшується.

Однією з особливостей випадкового лісу є можливість оцінки важливості ознак. Алгоритм обчислює внесок кожної ознаки у зменшення неоднорідності вузлів дерев, що дає змогу ранжувати ознаки за їх значущістю для задачі класифікації. Ця інформація може бути використана для подальшого відбору ознак або інтерпретації результатів.

Процес навчання та оптимізація гіперпараметрів

Навчання моделі випадкового лісу розпочинається з ініціалізації базової конфігурації з 100 деревами рішень, мінімальною кількістю зразків для розділення вузла у 10 та необмеженою глибиною дерев. Параметри обрано на основі попереднього досвіду та рекомендацій для подібних задач класифікації.

Для знаходження оптимальної конфігурації моделі використано клас Grid Search для пошуку найкращої комбінації гіперпараметрів. Простір пошуку містив варіацію кількості дерев від 100 до 300, максимальну глибину дерев від 10 до необмеженої та мінімальну кількість зразків для розділення від 2 до 10. Хороші результати продемонструвала модель із кількістю дерев 300, мінімальною кількістю зразків 5 та необмеженою глибиною. Ці гіперпараметри застосовано для створення фінальної моделі, яка буде збережена для подальшого використання.

Результати дослідження

Проведення експериментів та оцінка продуктивності

Більш детальний аналіз результатів класифікації показав збалансовану продуктивність для обох класів. Для нормального трафіку модель досягла точності 0,91, повноти 0,92 та F1-міри 0,91. Для аномального трафіку результати такі: точність 0,95, повнота 0,95 та F1-міра 0,95.

- Точність характеризує частку правильно ідентифікованих позитивних прикладів серед всіх прикладів, які модель класифікувала як позитивні. Інакше кажучи, вона відповідає на питання "Скільки з тих, що модель назвала аномаліями, дійсно є аномаліями?".

- Повнота вимірює частку правильно ідентифікованих позитивних прикладів від загальної кількості реальних позитивних прикладів у наборі даних, відповідаючи на питання “Скільки з усіх реальних аномалій модель змогла виявити?”.
- F1-міра – гармонічне середнє між точністю та повнотою, забезпечуючи збалансовану оцінку продуктивності моделі, важливу для задач з нерівномірним розподілом класів.

У табл. 1 наведено основні метрики продуктивності моделі випадкового лісу. Алгоритм продемонстрував точність 0,99 на навчальній та 0,94 на тестовій вибірках, що свідчить про відсутність перенавчання (табл. 1).

Детальний аналіз класифікації показав, що для нормального трафіку точність становила 0,91, тоді як для аномального трафіку точність досягла 0,95 (табл. 2).

Таблиця 1

Основні метрики продуктивності моделі випадкового лісу

Метрика	Значення
Оцінка на навчальних даних	0,99
Оцінка на тестових даних	0,94
Точність	0,94
Оцінка ROC	0,99

Таблиця 2

Детальні метрики класифікації випадкового лісу по класах

Клас	Точність	Повнота	F1-міра
Нормальний	0,91	0,92	0,91
Аномалія	0,95	0,95	0,95

Аналіз ROC-кривої підтвердив хорошу якість моделі. ROC-крива є графічним представленням залежності між часткою істинно позитивних результатів та часткою хибнопозитивних результатів при різних порогах класифікації, що дає змогу оцінити здатність моделі розрізняти класи незалежно від конкретного порогу прийняття рішення.

Ця крива є корисною для задач бінарної класифікації. Вона демонструє, як зміна порогу класифікації впливає на баланс між чутливістю моделі до виявлення аномалій та кількістю хибних спрацьовувань. Площа під ROC-кривою (AUC) становила 0,99, що наближається до ідеального значення 1.0. Крива демонструє швидке зростання істинно позитивних спрацьовувань за мінімального рівня хибнопозитивних. Високий показник AUC вказує на здатність моделі ефективно ранжувати зразки за ймовірністю належності до класу аномалій (рис. 4).

Порівняльний аналіз з альтернативними алгоритмами

Для об’єктивної оцінки ефективності обраного підходу проведено порівняльне тестування з двома альтернативними алгоритмами: методом опорних векторів та логістичною регресією. Експериментальне дослідження включало налаштування гіперпараметрів для кожного алгоритму та детальний аналіз їх продуктивності на ідентичних наборах даних.

Метод опорних векторів налаштовано з радіальним базисним ядром (RBF kernel) та параметром регуляризації C = 1.0. Використання RBF-ядра дає змогу алгоритму працювати з нелінійно розділюваними даними. Параметр gamma встановлено у режимі 'scale', що автоматично адаптує ядрову функцію до розмірності даних. Для подолання проблеми нерівномірного розподілу класів застосовано збалансовані ваги класів (class_weight='balanced').

Процес навчання SVM виявився більш ресурсоємним порівняно з випадковим лісом, особливо на великих наборах даних. Алгоритм продемонстрував точність 0,84 як на навчальній, так і на тестовій вибірках, що свідчить про відсутність перенавчання. Детальний аналіз класифікації показав цікаву особливість: для нормального трафіку точність становила 0,73, тоді як для аномального трафіку точність досягла 0,94 (табл. 3–4).

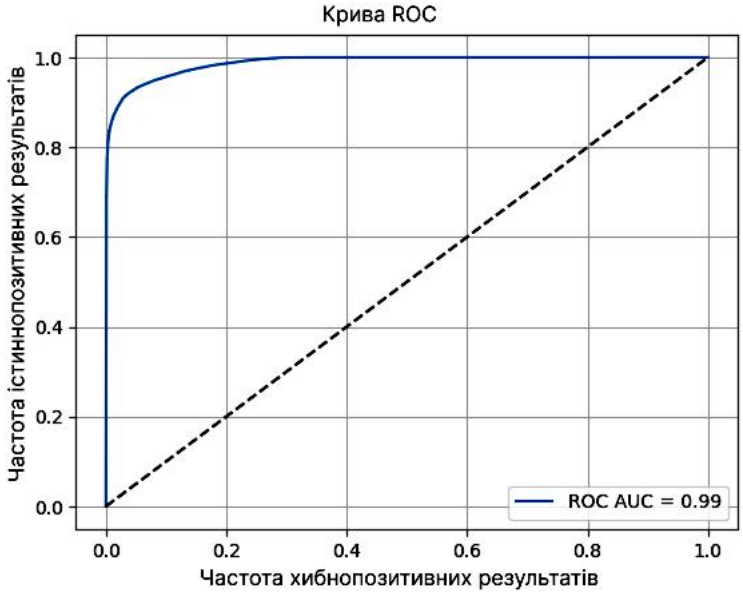

Рис. 4. Графічне зображення ROC-кривої для випадкового лісу

Таблиця 3

Основні метрики продуктивності моделі методу опорних векторів

Метрика	Значення
Оцінка на навчальних даних	0,84
Оцінка на тестових даних	0,84
Точність	0,84
Оцінка ROC	0,91

Таблиця 4

Детальні метрики класифікації методу опорних векторів по класах

Клас	Точність	Повнота	F1-міра
Нормальний	0,73	0,90	0,80
Аномалія	0,94	0,81	0,87

Така асиметрія в результатах вказує на схильність SVM до консервативної класифікації аномалій, що може бути як перевагою (менше хибних спрацьовувань), так і недоліком (пропуск частини реальних загроз). Площа під ROC-кривою для SVM становила 0,91, що є хорошим результатом, проте помітно поступається випадковому лісу (рис. 5).

Для логістичної регресії проведено систематичну оптимізацію гіперпараметрів за допомогою Grid Search. До експериментування входила варіація параметра регуляризації C від 0.1 до 100, типи регуляризації (L1 та L2), різна кількість ітерацій. Оптимальною конфігурацією виявилася комбінація C = 100, L1-регуляризація, вирішувач 'saga' та максимум 1000 ітерацій.

L1-регуляризація сприяє автоматичному відбору ознак, обнуляючи коефіцієнти менш важливих змінних. Вирішувач 'saga' забезпечує ефективне навчання для великих наборів даних та підтримує різні типи регуляризації.

Логістична регресія продемонструвала точність 0,83 на тестовій вибірці, що є найнижчим результатом серед всіх досліджених алгоритмів. Аналіз по класах виявив схожу до SVM тенденцію: нормальний трафік класифікувався з точністю 0,72, тоді як аномальний трафік досягав точності 0,91. Площа під ROC-кривою становила 0,91, що відповідає результату SVM (табл. 5–6, рис. 6).

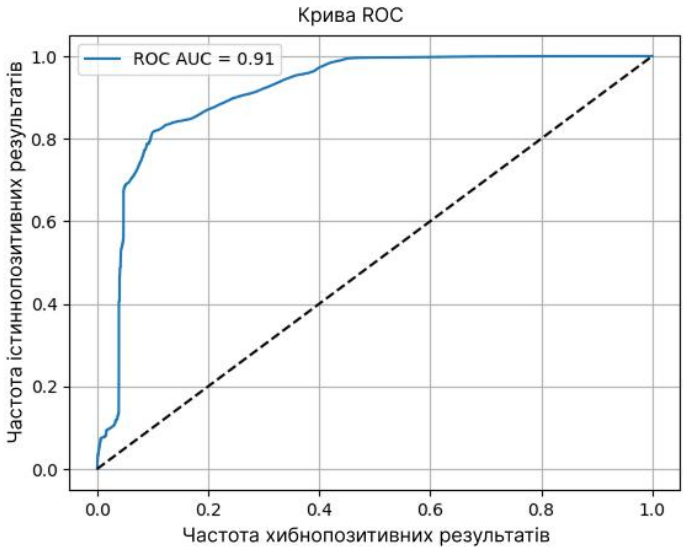

Рис. 5. Графічне зображення ROC-кривої для методу опорних векторів

Таблиця 5

Основні метрики продуктивності моделі логістичної регресії

Метрика	Значення
Оцінка на навчальних даних	0,83
Оцінка на тестових даних	0,83
Точність	0,83
Оцінка ROC	0,91

Таблиця 6

Детальні метрики класифікації логістичної регресії по класах

Клас	Точність	Повнота	F1-міра
Нормальний	0,72	0,85	0,78
Аномалія	0,91	0,81	0,86

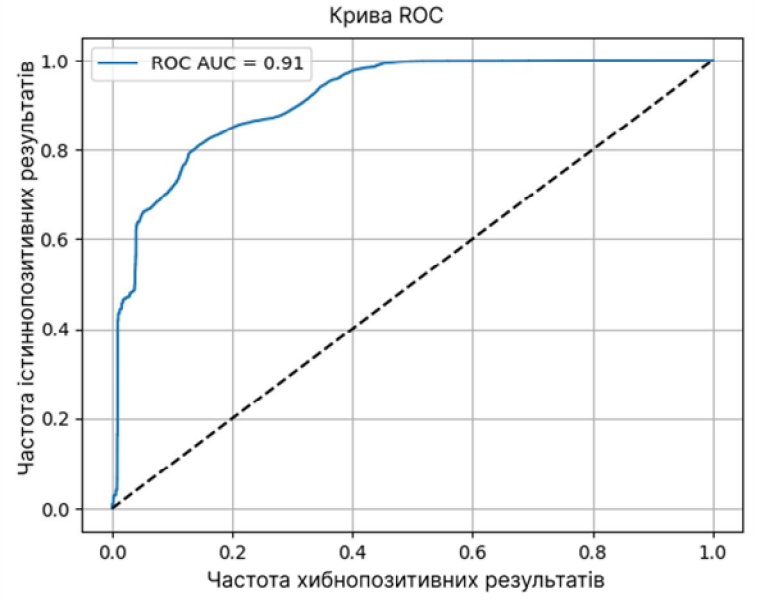

Рис. 6. Графічне зображення ROC-кривої для логістичної регресії

Основною перевагою логістичної регресії є швидкість навчання та прогнозування, а також висока інтерпретованість результатів. Коефіцієнти моделі можуть бути легко проаналізовані для розуміння внеску кожної ознаки у фінальне рішення. Однак лінійна природа алгоритму обмежує його здатність виявляти складні нелінійні взаємодії між ознаками.

Порівняння результатів

Комплексне порівняння трьох алгоритмів демонструє явні переваги випадкового лісу. З точки зору загальної точності випадковий ліс досягає 0,94, що на 10 % та 11 % вище за SVM (0,84) та логістичну регресію (0,83) відповідно. Площа під ROC-кривою для випадкового лісу (0,99) значно перевищує показники конкурентів (0,91 для обох), що свідчить про кращу здатність до ранжування зразків за ймовірністю належності до класів (табл. 7–8).

Таблиця 7

Порівняльний аналіз основних метрик продуктивності алгоритмів машинного навчання

Модель	Оцінка на навчальних даних	Оцінка на тестових даних	Точність	Оцінка ROC
Випадковий ліс	0,99	0,94	0,94	0,99
SVM	0,84	0,84	0,84	0,91
Логістична регресія	0,83	0,83	0,83	0,91

Таблиця 8

Порівняльний аналіз детальних метрик класифікації алгоритмів машинного навчання по класах

Модель	Нормальний			Аномалія		
	Точність	Повнота	F1-міра	Точність	Повнота	F1-міра
Випадковий ліс	0,91	0,92	0,91	0,95	0,95	0,95
SVM	0,73	0,90	0,80	0,94	0,81	0,87
Логістична регресія	0,72	0,85	0,78	0,91	0,81	0,86

Позитивною рисою випадкового лісу є збалансованість результатів між класами. Якщо SVM та логістична регресія демонструють значну асиметрію в точності для різних класів, випадковий ліс забезпечує високі показники для обох типів трафіку: точність 0,91–0,95 та повнота 0,92–0,95.

З практичного погляду випадковий ліс також виявляється найбільш привабливим через стійкість до перенавчання, можливість роботи з великими наборами ознак без попереднього відбору та порівняну простоту налаштування гіперпараметрів. Алгоритм також забезпечує оцінку важливості ознак.

Час навчання випадкового лісу займає проміжне положення між швидкою логістичною регресією та ресурсоємним SVM, що робить його оптимальним компромісом між продуктивністю та ефективністю.

Однією з ключових характеристик систем виявлення аномалій у комп'ютерних мережах є затримка між появою аномальної активності та її ідентифікацією. У випадку використання алгоритму випадкового лісу ця затримка формується на кількох етапах: агрегація мережного трафіку, витягування ознак, їх попередня обробка та безпосереднє проходження через ансамбль дерев. У середньому час інференсу для одного потоку даних становить від 1 до 10 мілісекунд на сучасних CPU, однак загальна end-to-end затримка, що містить агрегацію та обробку трафіку, сягає 100–300 мс для окремих потоків і може досягати 0,5–2 секунд у разі масової пакетної обробки.

Важливим аспектом є баланс між точністю класифікації та затримкою. Алгоритм випадкового лісу демонструє високу якість розпізнавання на наборі UNSW-NB15, однак ці показники досягаються завдяки збільшенню кількості дерев у моделі, що безпосередньо впливає на час обчислення. Змен-

шення глибини та кількості дерев скорочує час реакції, проте призводить до часткового зменшення точності. У практичних сценаріях, особливо в мережах із високою пропускною здатністю, постає необхідність у пошуку компромісу між продуктивністю й якістю розпізнавання.

Оптимізувати цей баланс можна кількома шляхами. По-перше, використання інкрементальної обробки потоків з коротшими інтервалами агрегації зменшує затримку виявлення. По-друге, застосування апаратного прискорення (GPU, FPGA) дає змогу паралелізувати обчислення у великій кількості дерев без суттєвої втрати часу. По-третє, інтеграція випадкового лісу з методами попереднього фільтрування або легшими моделями (наприклад, логістичною регресією для швидкої оцінки й подальшого передавання підозрілих потоків у випадковий ліс) забезпечує баланс між швидкістю та точністю. Використання оптимізованих бібліотек Python (Scikit-learn, XGBoost) та стиснених представлень ознак також надає змогу суттєво скоротити середній час реакції системи.

Висновки

Реалізовано систему виявлення аномалій та моніторингу мережного трафіку, включно з проєктуванням архітектури, підготовкою навчальних даних, розробкою та тестуванням моделі машинного навчання, створенням бази даних та API, а також реалізацією користувацького інтерфейсу.

Розроблена модульна архітектура системи забезпечує ефективну взаємодію між п’ятьма основними компонентами: модулем збору мережного трафіку на базі PyShark, сервісом обробки пакетів для агрегації потоків, модулем виявлення аномалій з інтегрованою моделлю машинного навчання, API-модулем на FastAPI та вебінтерфейсом на React. Асинхронна взаємодія між компонентами дає змогу кожному модулю працювати в оптимальному режимі, забезпечуючи обробку даних в режимі реального часу.

До комплексної підготовки навчальних даних на базі датасету UNSW-NB15 входив ретельний відбір 22 найбільш значущих ознак з початкових 49 для забезпечення оптимального балансу між точністю виявлення та швидкодією системи. Застосування Label Encoder для категоріальних змінних та MinMaxScaler для числових ознак забезпечило правильну підготовку даних для алгоритмів машинного навчання. Аналіз матриці кореляцій виявив важливі взаємозв’язки між ознаками та допоміг ідентифікувати потенційні проблеми мультиколінеарності.

Експериментальне дослідження трьох алгоритмів машинного навчання продемонструвало явні переваги класифікатора випадкового лісу, який досягнув точності 94 % на тестовій вибірці та площі під ROC-кривою 0,99, значно перевищивши показники методу опорних векторів (84 % точності) та логістичної регресії (83 % точності). Збалансованість результатів випадкового лісу між класами (точність 0,91–0,95 та повнота 0,92–0,95) підтверджує його придатність для практичного застосування в системах виявлення аномалій.

Алгоритм випадкового лісу є ефективним інструментом для виявлення аномалій у мережному трафіку завдяки високій точності та стійкості до шумів, проте його застосування в реальному часі обмежується затримками, що залежать від умов обробки. Правильне налаштування моделі та архітектури системи дає змогу досягти прийнятного компромісу між швидкістю реакції та надійністю виявлення аномалій в мережному трафіку, що є критично важливим для сучасних систем кіберзахисту.

Особливістю запропонованого підходу є інтеграція методів машинного навчання з комплексним аналізом різноманітних параметрів мережного трафіку, завдяки чому значно підвищилась точність виявлення аномалій та мінімізувалась кількість хибних спрацьовувань порівняно з наявними рішеннями. Система демонструє високу практичну цінність через поєднання точності виявлення аномалій, зручності використання та можливості масштабування.

Створено комплексне рішення, що забезпечує підвищення точності виявлення потенційних загроз, оптимізацію використання обчислювальних ресурсів під час обробки мережного трафіку, скорочення часу від моменту виникнення аномалії до її виявлення та надання зручного інструментарію для аналізу стану мережі. Розроблений підхід готовий для практичного застосування в

організаціях різного масштабу та може слугувати основою для подальшого розвитку в напрямі адаптивних моделей, здатних автоматично пристосовуватися до змін у мережному середовищі, та інтеграції різноманітних джерел даних для формування цілісного уявлення про стан мережної безпеки.

Продуктивність системи виявлення аномалій може бути суттєво підвищена завдяки інтеграції глибоких нейронних мереж, використанню розподілених платформ для обробки даних та оптимізації обчислювальних моделей. Оптимальний баланс між точністю та затримкою досягається шляхом каскадного застосування різних класів алгоритмів, апаратного прискорення та архітектурних рішень на основі потокової обробки даних.

Список літератури

1. Vibhute A., Khan M., Patil C.H., Gaikwad S.V., et al. Network anomaly detection and performance evaluation of Convolutional Neural Networks on UNSW-NB15 dataset. Procedia Computer Science, Vol. 235, 2024, pp. 2227–2236. DOI: https://doi.org/10.1016/j.procs.2024.04.211.

2. Thiyam R., Dey D. An improved deep autoencoder-based network intrusion detection system with enhanced performance. International Journal of Internet Technology and Secured Transactions, v. 13(3), January 2024, pp. 270–290. DOI:10.1504/IJITST.2024.136658.

3. Yankun Xue, Chunying Kang, Hongchen Yu HAE-HRL: A network intrusion detection system utilizing a novel autoencoder and a hybrid enhanced LSTM-CNN-based residual network. Computers & Security, Volume 151, April 2025, 104328, ISSN 0167-4048. DOI: https://doi.org/10.1016/j.cose.2025.104328.

4. Zhang C, Zhang M, Yang G, Xue T, Zhang Z, Liu L, Wang L, Hou W, Chen Z. Three-Way Selection Random Forest Optimization Model for Anomaly Traffic Detection. Electronics. 2023; 12(8):1788. DOI: https://doi.org/10.3390/electronics12081788.

5. Li E., Shang Z., Gungor O., Rosing T. LI, Elvin, et al. SAFE: Self-Supervised Anomaly Detection Framework for Intrusion Detection. arXiv preprint arXiv:2502.07119, 2025.

6. Guerra L., Chapuis T., Duc G., Mozharovskyi P., Nguyen V-T. Self-Supervised Learning of Graph Representations for Network Intrusion Detection. arXiv:2509.16625, 2025.

7. Liu S., Zhao Z., He W., Wang J., Peng J., Ma H. Privacy-Preserving Hybrid Ensemble Model for Network Anomaly Detection. arXiv:2502.09001, 2025.

8. Singh K., Kashyap A., Cherukuri A.K. Interpretable Anomaly Detection in Encrypted Traffic Using SHAP with Machine Learning Models. arXiv:2505.16261, 2025.Cloud Computing Statistics / Market.us. URL: https://scoop.market.us/cloud-computing-statistics (Accessed: September 25, 2025).

9. IDC Research Report / Worldwide IT Industry 2025 Predictions. URL: https://my.idc.com/getdoc.jsp?containerId=US51736824 (Accessed: September 25, 2025).

10. Cisco Predicts More IP Traffic in the Next Five Years than in the History of the Internet / Cisco Newsroom. URL: https://newsroom.cisco.com/c/r/newsroom/en/us/a/y2018/m11/cisco-predicts-more-ip-traffic-in-the-next-five-years-than-in-the-history-of-the-internet.html (Accessed: September 25, 2025).

11. IBM Report: Escalating Data Breach Disruption Pushes Costs to New Highs / IBM Newsroom. URL: https://newsroom.ibm.com/2024-07-30-ibm-report-escalating-data-breach-disruption-pushes-costs-to-new-highs (Accessed: September 25, 2025).

12. The Cost of IT Downtime / The 20 Blog. URL: https://www.the20.com/blog/the-cost-of-it-downtime (Accessed: September 25, 2025).

13. The UNSW-NB15 Dataset. URL: https://research.unsw.edu.au/projects/unsw-nb15-dataset (Accessed: September 25, 2025).

14. Random Forest, Explained: A Visual Guide with Code Examples. URL: https://medium.com/data-science/random-forest-explained-a-visual-guide-with-code-examples-9f736a6e1b3c (Accessed: September 25, 2025).

15. Support Vector Machine (SVM) Algorithm. URL: https://www.geeksforgeeks.org/machine-learning/support-vector-machine-algorithm/.

16. Logistic Regression in Machine Learning. URL: https://www.geeksforgeeks.org/machine-learning/understanding-logistic-regression/ (Accessed: September 25, 2025).

ANOMALIES DETECTION AND TRAFFIC MONITORING SYSTEM IN COMPUTER NETWORKS

O. V. Khoma, I. Yu. Yurchak, A. O. Khich

Lviv Polytechnic National University,
Department of "Computer Design Systems"

The paper addresses the problem of anomaly detection in network traffic and proposes a comprehensive solution to enhance the level of cybersecurity for organizations of various scales. A comparative analysis of existing monitoring and anomaly detection systems has been carried out, including both open-source solutions and commercial products. Based on the conducted research, the technological stack for implementing a network traffic anomaly detection system has been justified, particularly the optimal combination of Python libraries for collecting and processing network traffic, data preparation, applying machine learning algorithms, and visualizing the results.

A fully functional system consisting of five main components has been developed: a network traffic collection module based on PyShark, a packet processing service for flow aggregation, an anomaly detection module using machine learning algorithms, an API module based on FastAPI, and a user web interface developed in React. The system ensures high accuracy in detecting potential threats, optimized use of computational resources, and convenient analysis of the network state.

An experimental study was conducted on the UNSW-NB15 dataset, which contains over 2.5 million records of network connections. A comparative analysis of three machine learning algorithms was performed: Random Forest, Support Vector Machine, and Logistic Regression. Random Forest demonstrated the best results with 94 % accuracy on the test set and an area under the ROC curve of 0.99, significantly outperforming the alternative algorithms.

A key feature of the proposed approach is the integration of machine learning methods with comprehensive analysis of various network traffic parameters, which significantly improves anomaly detection accuracy and minimizes the number of false positives compared to existing solutions.

Keywords: anomaly detection, cybersecurity, data analysis, machine learning, network traffic, network security, python, random forest.