УДК 004.056.5 **DOI:10.15330/itee.2024.1.03**

М. М. ПАВЛЮК

АНАЛІЗ МЕРЕЖЕВОГО ТРАФІКУ ЯК ЗАСІБ ДЕТЕКТУВАННЯ КІБЕРЗАГРОЗ

Вступ

Розвиток широкосмугового інтернету та web-технологій спричинив значне зростання мережевого трафіку, в тому числі через активну роботу різних пристроїв, багато з яких не мають ефективного захисту від мережевих атак. Недостатній захист у мережевому просторі загрожує безпеці конфіденційної інформації, впливаючи на здоров'я, добробут та активи громадян.

Зростання кіберзагроз у зв'язку з поширенням Інтернету речей та збільшенням кількості інтернет-пристроїв зумовлює потребу в розробці ефективних програмних рішень для аналізу мережевого трафіку, здатних виявляти складні атаки та адаптуватися до нових загроз. Актуальність теми. Зростання кіберзагроз у зв'язку з поширенням Інтернету речей та збільшенням кількості інтернет-пристроїв зумовлює потребу в розробці ефективних програмних рішень для аналізу мережевого трафіку, здатних виявляти складні атаки та адаптуватися до нових загроз.

Мета роботи: дослідження сучасного стану методів виявлення кібератак через аналіз мережевого трафіку, оцінка ефективності статистичних методів та технологій машинного навчання, встановлення залежностей, що сприяють точності ідентифікації загроз у реальному часі, та розробка програмного забезпечення, що сприяє оптимізації витрат на кіберзахист.

Для досягнення мети роботи визначено наступні завдання:

- аналіз типів кіберзагроз та їх характеристик, огляд систем і алгоритмів безпеки для виявлення загроз через аналіз трафіку;
- вибір та застосування методів статистичного аналізу та машинного навчання для аналізу даних і розробка моделей виявлення кіберзагроз;
- розробка, валідація та тестування програмного забезпечення в реальному часі, включаючи оцінку якості та практичної застосовності.

Оцінка сучасного стану об'єкту дослідження. Кібербезпека активно еволюціонує з використанням штучного інтелекту та машинного навчання, включно з нейронними мережами, для ефективного аналізу мережевого трафіку та виявлення аномалій.

Світові тенденції розв'язання поставлених проблем. Глобальні методи кіберзахисту рухаються до створення гібридних систем, що інтегрують активні та пасивні методи безпеки для адаптивного реагування на кіберзагрози в реальному часі.

Дослідження базується на поєднанні статистичних методів та інструментів машинного навчання для аналізу мережевого трафіку та ідентифікації кіберзагроз. Застосовано аналіз аномалій з використанням Гаусових моделей та PCA, методи класифікації, а також рекурентні нейронні мережі та автокодувальники. Для ідентифікації складніших патернів використовуються згорткові нейронні мережі. Дослідження базується на аналізі історичних та синтетичних датасетів, зокрема UNSW-NB15 та CICIDS2017.

Системи та алгоритми безпеки, що використовують аналіз трафіку

Системи виявлення вторгнень (IDS) та системи запобігання вторгненням (IPS) є ключовими компонентами в архітектурі кібербезпеки [1].

IDS - система або програмне забезпечення, призначене для моніторингу мережі та систем на наявність зловмисних активностей чи порушень політик безпеки [1]. Вони ідентифікують підозрілі дії, відстежують мережевий трафік та системні журнали, порівнюючи їх з базою відомих атак або аномальних патернів. Після виявлення потенційного вторгнення IDS генерує сповіщення для адміністраторів мережі або безпеки.

IPS - є розширенням попереднього типу систем, яке не лише виявляє підозрілі активності, але й вживає заходів для їх блокування або запобіганнях [1]. IPS активно аналізує та фільтрує мережевий трафік на основі визначених політик безпеки та відомих підписів атак. Вони можуть автоматично відкидати пакети даних, блокувати IP-адреси атакувальної сторони або вживати інших заходів для запобігання виконання шкідливих дій у мережі.

Обидві системи є важливими для забезпечення превентивного захисту від різноманітних кіберзагроз, забезпечуючи раннє виявлення та реагування на потенційні вторгнення.

Наведемо деякі приклади статистичних моделей, моделей машинного навчання та штучних нейронних мереж, що використовуються для виконання класів задач з виявлення кіберзагроз.

Для виявлення аномалій у мережевому трафіку можна використовувати різні статистичні моделі, моделі машинного навчання та штучні нейронні мережі (рис. 1) [2, 3]. Кожен клас моделей має свої особливості та застосування.

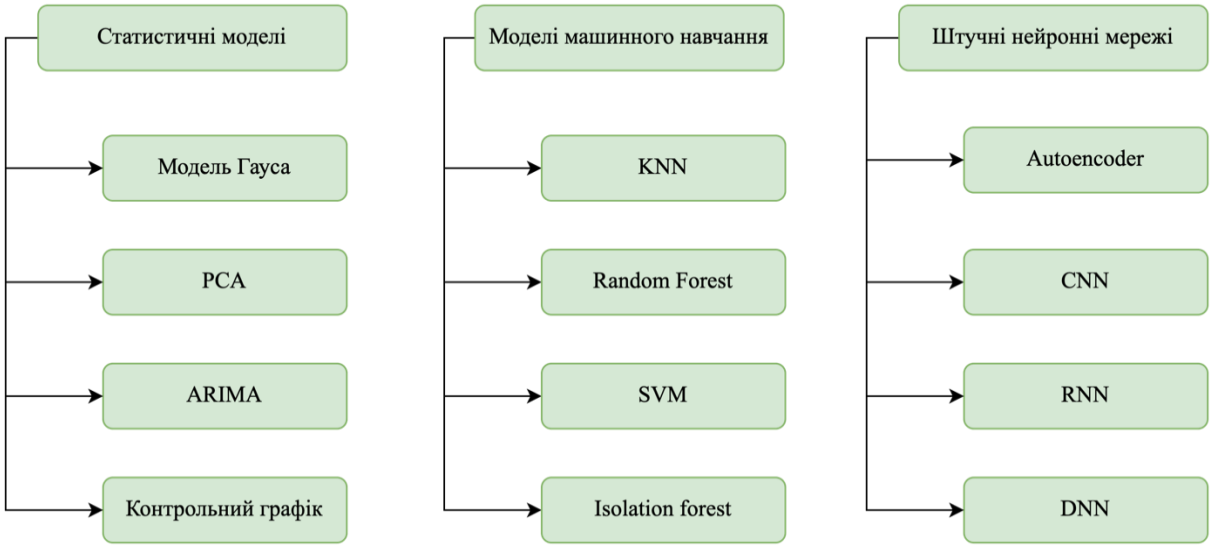

Рис. 1. Моделі аналізу трафіку (авторське опрацювання)

Кінцевою метою дослідження є створення сервісу, що інтегрується з системами фільтрації мережевих пакетів, наприклад, Iptables та Nftables, та дозволяє перейти від статичних налаштувань таких систем, до динамічних, що керуються моделями машинного навчання та штучним інтелектом. Реалізація модулів збору даних та аналізу даних з використанням ШІ доцільна у вигляді сервісів операційної системи Linux, як найбільш зручного для практичного використання варіанту. Виходячи з цього, основним завдання є створення концепції, алгоритмізація та програмна реалізація модуля штучного інтелекту, що виконує динамічні налаштування одного з UNIX-фаєрволів на базі аналізу трафіку.

Розглянуто більш детально послідовність робіт із розробки, проєктування, створення та тестування запропонованого модуля. З огляду на уже згадані статті, у яких здійснено дослідженням використання машинного навчання та штучних нейронних мереж в галузі аналізу трафіку, зроблено висновок, що дослідження проводяться в наступній послідовності (рис. 2):

а) Робота з наявними датасетами. На цьому початковому етапу використовуються доступні у відкритих джерелах датасети або такі, що надаються за запитом. Також у багатьох дослідженнях використовуються датасети, що вперше з'явились у різноманітних конкурсах науковців. Робота проводиться наступним чином:

1) проводиться попередній аналіз даних, вивчаються основні значущі ознаки та закономірності;

2) з використанням цих доступних даних будуються необхідні моделі, проводиться їх налаштування та порівняння, робляться висновки відносно доцільності та якості використання тієї чи іншої моделі;

3) проводиться більш точне налаштування та вивчення поведінки моделей, що показали найкращий результат на попередньому етапі;

б) робота з синтетичними датасетами. На цьому етапі дослідники переходять до генерування синтетичних датасетів з використанням спеціалізованого програмного забезпечення, що використовується для симуляції атак різних типів;

в) тестування створеного програмного забезпечення на тестових наборах даних та синтетичних джерелах даних про трафік, що дозволяє симулювати різноманітні типи атак ті виконувати коректність роботи програмного забезпечення у різних сценаріях використання;

г) тестування та вивчення поведінки програмного забезпечення при симулюванні атак в реальних мережах.

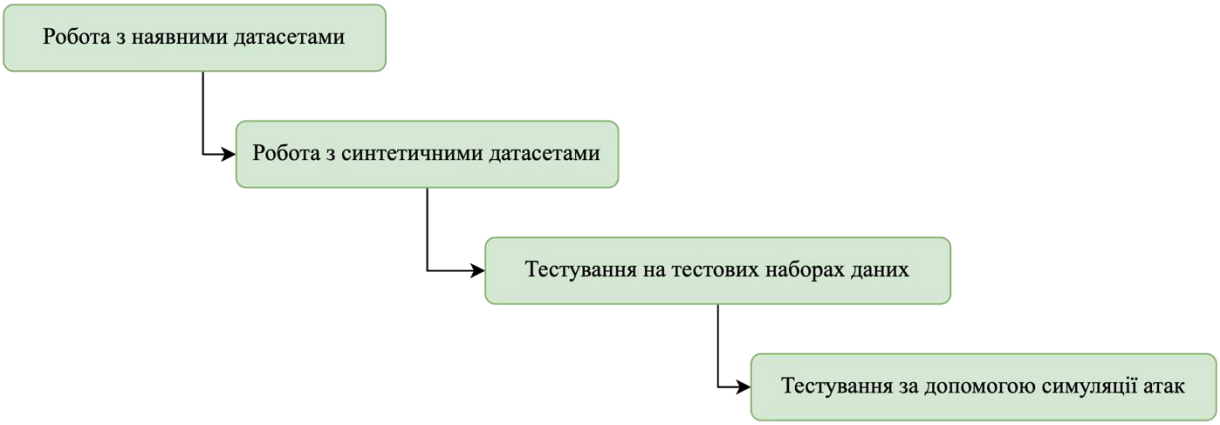

Рис. 2. Послідовність проведення дослідження (авторське опрацювання)

Проаналізовано, що численні дослідження закінчуються на етапі вивчення поведінки моделей у віртуальних середовищах чи на згенерованих датасетах, що пояснюється двома причинами:

1) спеціалізоване програмне забезпечення дозволяє створювати мережі різної топології та симулювати різні типи атак, створення яких в реальному середовищі пов'язано з чималими затратами;

2) наявні спеціалізовані датасети (історичні дані за великі періоди) містять більше патернів, змін, типів атак, аніж може симулювати дослідник у віртуальному чи реальному середовищі.

Базуючись на вище наведених міркуваннях, дослідження сплановано наступним чином:

- аналіз актуальних історичних датасетів, вибір актуального та найбільш доцільного в межах мети дослідження;
- аналіз обраного датасету з метою виділення залежностей, ранжування важливості характеристик трафіку;
- створення та тренування моделей машинного навчання та ШІ, тренування та оцінка побудованих моделей на обраному на попередніх кроках датасетові;
- тестування обраної моделі в реальному середовищі.

Дослідження наявних датасетів

Практичну частину роботи було організовано у вигляді трьох послідовних етапів: проаналізовано відкриті набори даних, здійснено моделювання атак із використанням спеціалізованого програмного забезпечення та проведено імітацію атаки в умовах реальної локальної мережі.

Опрацювання відкритих датасетів стало ключовим етапом дослідження з погляду формування та оцінювання поведінки моделей. Це дало змогу проаналізувати значні масиви зібраного мережевого трафіку, що відзначався достовірністю, різноманітністю та репрезентативністю. Завдяки цьому було виявлено особливості структури даних і закономірності, які необхідно враховувати під час побудови моделей. Використання лише спеціалізованого програмного забезпечення для симуляції атак або їх короткочасної імітації в реальному часі не дозволило б отримати настільки повний і різнорідний набір даних порівняно з інформацією, що накопичувалася дослідниками протягом тривалого часу та проходила попередню підготовку для подальшого аналізу.

Попри значний обсяг і різноманітність наявних наборів даних, було встановлено, що не всі з них відповідають цілям дослідження. Для застосування методів аналізу часових рядів будь-який датасет було попередньо трансформовано у відповідний формат. Для задач класифікації мережевого трафіку найбільш придатними було визначено KDD-99 (зокрема його модифікацію NSL-KDD), UNSW-NB15 та CAIDA DDoS Attack 2007. Для виявлення аномалій у протокольних характеристиках доцільними виявилися KDD-99 (NSL-KDD), UNSW-NB15 та CICIDS2017.

У результаті основну увагу було зосереджено на модифікованих версіях KDD-99, зокрема NSL-KDD, а також на датасеті UNSW-NB15. Додатково CAIDA DDoS Attack 2007 було використано для навчання моделей виявлення DDoS-атак, тоді як CICIDS2017 було залучено для дослідження методів пошуку аномалій.

Набір даних KDD Cup 1999 Data було розглянуто як один із базових стандартів для оцінювання систем комп’ютерної безпеки у задачах виявлення вторгнень та аналізу аномалій мережевого трафіку. Він містив широкий спектр змодельованих атак, зокрема переповнення буферів, несанкціонований доступ до даних та інші типи кібератак у контексті мережевої взаємодії. Вперше цей датасет було застосовано в межах змагання KDD Cup 1999, яке стало одним із перших масштабних заходів у галузі аналізу даних для задач кібербезпеки [4, 5].

Незважаючи на те, що контекст атак у цьому наборі даних частково втратив актуальність через розвиток сучасних методів здійснення атак, його аналіз не було проігноровано. Особливу увагу було приділено модифікованим версіям датасету, які було створено з метою усунення недоліків початкової вибірки та підвищення її збалансованості.

Датасет містив понад один мільйон унікальних записів і 42 ознаки. Цільовою змінною виступав тип атаки: було представлено 22 різновиди атак, об’єднані у чотири групи. Окремо було передбачено мітку normal, яка відповідала нормальному мережевому трафіку. Структурування атак за групами було наведено в таблиці 1.

Таблиця 1

Типи атак та їхні групи

DDoS Denial of Service	Відмова в обслуговуванні.	back, land, neptune, pod, smurf. teardrop
R2L Remote to Local	Несанкціонований доступ з віддаленого комп'ютера	ftp_write, guess_passwd, imap, multihop, phf, spy, warezclient, warezmaster
U2R User to Root	Несанкціонований доступ до привілеїв локального суперкористувача (root)	buffer_overflow, loadmodule, perl, rootkit
Probe	Спостереження та інші дослідження	ipsweep, nmap, portsweep, satan

Першими та наочними проблемами датасету є велика кількість ознак та незбалансованість класів [6, 7]. Остання ілюструється діаграмою на рис. 3.

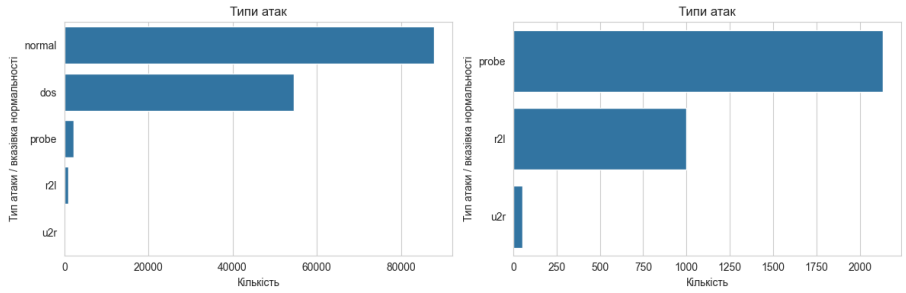

Рис. 3. Кількість атак за типами

Як модифіковану версію базового набору даних було розглянуто NSL-KDD - більш збалансовану повторну вибірку, сформовану на основі KDD-99. У цій версії було зменшено кількість дубльованих записів та приділено особливу увагу прикладам, які з високою ймовірністю могли бути неправильно класифіковані моделями, навченими на оригінальному KDD-99 [8, 9].

Разом із тим, головним обмеженням KDD-99 залишався його застарілий характер, що проявлявся у специфіці та типах атак, притаманних мережевому середовищу кінця 1990-х років. Попри це, датасет було визнано важливим етапом розвитку підходів до аналізу мережевого трафіку та оцінювання систем виявлення вторгнень.

У процесі експериментів було побудовано класифікаційні моделі на попередньо обробленому датасеті без зменшення кількості ознак. Отримані результати засвідчили високу точність роботи моделей. Порівняння основних метрик показало, що як у бінарній, так і в багатокласовій постановці задачі класифікатори продемонстрували високі значення F1-міри на тренувальних і тестових вибірках.

На підставі отриманих результатів було зроблено висновок, що подальше підвищення точності класифікації не є першочерговим завданням. В умовах обмежених обчислювальних ресурсів більш актуальною виявилася проблема забезпечення високої швидкодії алгоритмів.

У зв’язку з цим акцент у роботі було зміщено на оптимізацію використання ресурсів, зокрема на зменшення обчислювальної складності моделей за мінімальної втрати точності прогнозування. Одним із підходів до досягнення цієї мети було застосування методів зниження розмірності простору ознак.

Проведений аналіз головних компонент продемонстрував, що внесок окремих ознак у процес класифікації є нерівномірним. Результати застосування методу PCA представлено на рис. 4.

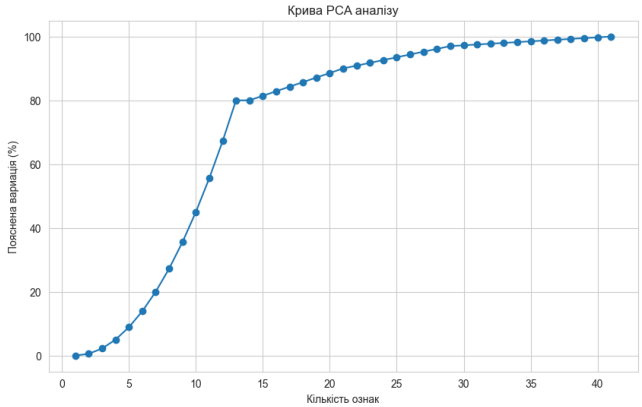

Рис. 4. Аналіз головних компонент

Зменшення кількості ознак дозволило підвищити швидкість обробки даних, що є критично важливим для систем, орієнтованих на функціонування в режимі реального часу.

Другим ключовим набором даних було визначено UNSW-NB15, розроблений Університетом Нового Південного Уельсу (Австралія) для задач виявлення вторгнень і аналізу мережевого трафіку. Цей датасет охоплював сучасніші типи атак і приклади нормального трафіку, що зробило його доцільним для побудови та тестування моделей детекції.

Водночас було враховано й певні обмеження UNSW-NB15, серед яких - відносно менша різноманітність атак, недостатня документованість окремих аспектів формування вибірки та проблеми зі збалансованістю класів. З огляду на це його не було використано як єдину основу для навчання моделей, а застосовано у поєднанні з іншими наборами даних, визначеними в межах цього розділу.

Проєктування архітектури системи та інфраструктури обробки даних

Під час розроблення системи було враховано суперечливі вимоги до обчислювальних ресурсів. З одного боку, алгоритми машинного навчання та нейронні мережі потребували значної обчислювальної потужності, особливо при роботі з великими таблицями даних, що містили сотні тисяч записів і десятки ознак. З іншого боку, практична придатність створеного програмного забезпечення безпосередньо залежала від його ресурсоємності: чим нижчими були вимоги до апаратного забезпечення, тим ширшими ставали можливості його застосування.

У зв’язку з цим було розмежовано три основні процеси:

1) збирання та підготовку даних для навчання моделей (включно з використанням великих відкритих датасетів);

2) безпосереднє навчання моделей;

3) експлуатацію вже навчених моделей у реальному середовищі.

Для організації потокового збору даних та опрацювання наявних датасетів було розгорнуто кластер на базі Apache Spark із використанням Spark Streaming та Hadoop. Навчання моделей і глибокий аналіз даних було виконано із залученням хмарних сервісів Google, що дозволило використовувати графічні процесори для обчислень у випадках, коли це було технічно доцільно. Зокрема, TensorFlow було орієнтовано на використання GPU, тоді як Scikit-learn та XGBoost переважно використовували ресурси центрального процесора.

Для тестування програмного забезпечення з інтегрованими навченими моделями було застосовано одноплатний комп’ютер Raspberry Pi 4 з 4 ГБ оперативної пам’яті та тактовою частотою процесора 1,5 ГГц, що дозволило оцінити можливість роботи системи на пристроях обмеженої потужності.

Реалізована система збору та обробки мережевого трафіку складалася з таких компонентів (рис. 5):

1) Filebeat і Fluentd - як агенти збору журналів подій із серверів, маршрутизаторів і міжмережевих екранів;

2) Apache Kafka - для централізованого приймання потокових даних від агентів;

3) Spark Streaming - для агрегації, попередньої обробки та фільтрації потоку даних;

4) Hadoop Distributed File System (HDFS) - для зберігання накопиченої інформації;

5) Apache Spark - для подальшого аналізу даних і формування вибірок для навчання математичних моделей та нейронних мереж;

6) Apache ZooKeeper - для координації та синхронізації роботи компонентів кластера.

Запропонована архітектура була доцільною у випадках збору власних даних або обслуговування локальної мережі, що включала понад 10 комп’ютерів і характеризувалася інтенсивним та різнорідним трафіком. Використання такого кластера забезпечувало не лише централізований моніторинг, а й можливість періодичного донавчання моделей на оновлених даних.

Як альтернативний сценарій було розглянуто використання програмного забезпечення з уже навченими моделями на малопотужних пристроях. Для цього варіанта було визначено мінімальний часовий інтервал аналізу (вікно спостереження), який становив приблизно 60 секунд. Було встановлено, що обсяг даних, накопичений за цей період, є достатнім для виявлення короткотривалих атак, зокрема DDoS-активності.

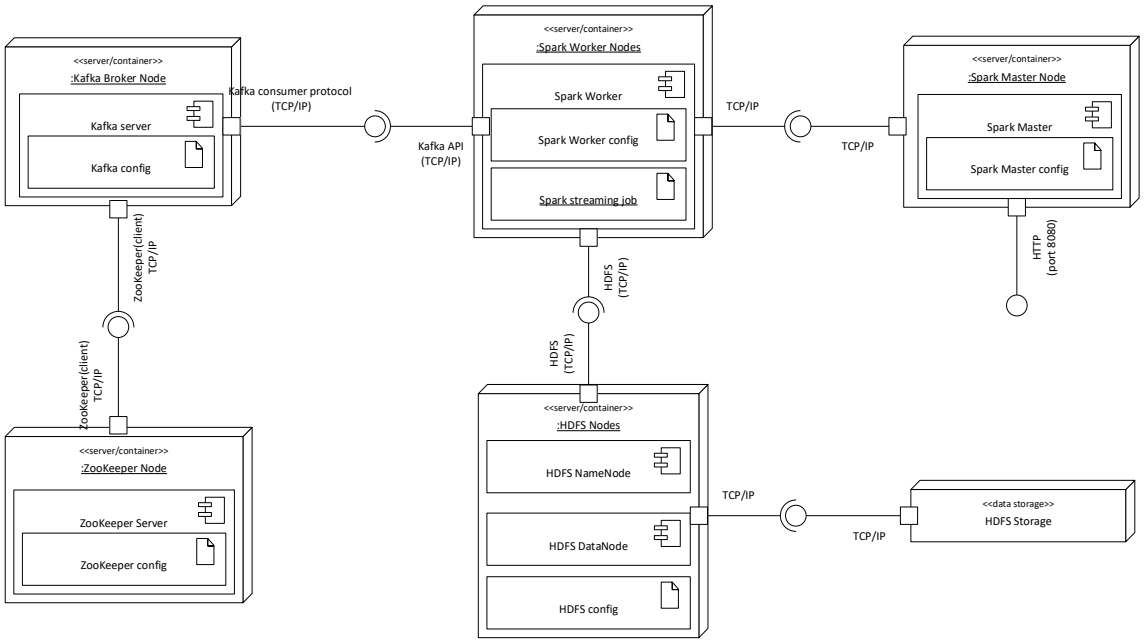

Рис. 5. Схема кластеру збору даних

Розроблене рішення відрізняється комплексним підходом, що інтегрує аналіз аномалій та класифікацію трафіку в одній системі, використовуючи машинне навчання для адаптації до нових загроз. Відмінність від традиційних IDS/IPS-систем полягає в поєднанні статистичних, поведінкових моделей та автокодувальників з CNN для виявлення як відомих, так і нових загроз. Рішення автоматично оновлює правила фаєрвола, адаптуючись до мережевих умов і знижуючи потребу в ручному налаштуванні, що робить його придатним для малих підприємств та IoT систем. Воно також стійке до загроз, орієнтованих на обхід стандартних захисних механізмів, забезпечуючи виявлення прихованих кіберзагроз.

Розроблене програмне рішення з відкритим кодом та низькими системними вимогами, що робить його доступним для малих підприємств та IoT систем, з можливістю спільноти вносити покращення завдяки MIT ліцензії. Програмне забезпечення пройшло тестування та дослідну експлуатацію в локальному тестовому середовищі та в тестовій локальній мережі. Розроблене рішення не потребує комерційних ліцензій та знижує витрати на відновлення після атак завдяки швидкому виявленню та блокуванню загроз, що робить його економічно привабливим. Запропоноване рішення для виявлення кіберзагроз надає ефективний та доступний інструмент захисту мережевого простору. Використання сучасних методів машинного навчання та аналізу аномалій сприяє зниженню хибнопозитивних спрацювань та стимулює спільноту спеціалістів до постійного вдосконалення системи.

Подальший розвиток передбачає вдосконалення алгоритмів класифікації, вдосконалення та збільшення різноманітності навчальних датасетів, що застосовуються в роботі, а також тестування в умовах реальних загроз. Можлива також інтеграції з системами інтелектуального аналізу поведінки.

Висновки

У ході дослідження було розроблено та апробовано методи й алгоритми виявлення кіберзагроз на основі аналізу мережевого трафіку із застосуванням підходів штучного інтелекту та машинного навчання. Отримані результати підтвердили їх високу ефективність у задачах детектування атак.

Проведені експерименти показали, що моделі забезпечували високі показники точності класифікації. Водночас в умовах обмежених обчислювальних ресурсів ключовим фактором стала не стільки подальша оптимізація точності, скільки підвищення швидкодії обробки даних. У зв'язку з цим акцент було зміщено на оптимізацію обчислювальних процесів із мінімальною втратою якості прогнозування.

Основним напрямом такої оптимізації визначено зменшення обчислювальної складності моделей, що дозволило забезпечити оперативну обробку значних обсягів трафіку та своєчасне реагування на потенційні загрози. З цією метою було застосовано методи зниження розмірності даних, зокрема аналіз головних компонент (PCA), що сприяло скороченню кількості ознак без істотного погіршення результатів класифікації.

Розроблене програмне забезпечення було орієнтовано на функціонування в режимі реального часу та адаптацію до змінних умов мережевого середовища. Такий підхід забезпечив можливість його використання для захисту як серверних систем, так і пристроїв із обмеженими ресурсами, підвищуючи загальний рівень стійкості мережі до кібератак

Список літератури:

1. D. Regalado, S. Harris, A. Harper, C. Eagle, J. Ness, B. Spasojevic, R. Linn та S. Sims, Gray Hat Hacking The Ethical Hacker's Handbook, Fourth Edition, McGraw Hill, 2022.
2. A. Divekar, M. Parekh, V. Savla, R. Mishra, and M. Shirole, "Benchmarking datasets for Anomaly-based Network Intrusion Detection: KDD CUP 99 alternatives," in *2018 IEEE 3rd Int. Conf. Comput., Communication Secur. (ICCCS)*, Kathmandu, Oct. 25–27, 2018. IEEE, 2018. doi: https://doi.org/10.1109/cccs.2018.8586840.
3. P. Singh, J. J. P, A. Pankaj, and R. Mitra, "Edge-Detect: Edge-Centric Network Intrusion Detection using Deep Neural Network," in *2021 IEEE 18th Annu. Consum. Commun. Netw. Conf. (CCNC)*, Las Vegas, NV, USA, Jan. 9–12, 2021. IEEE, 2021. doi: https://doi.org/10.1109/ccnc49032.2021.9369469.
4. T. Chen and C. Guestrin, "XGBoost," in *KDD '16: 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, San Francisco California USA. New York, NY, USA: ACM, 2016. doi: https://doi.org/10.1145/2939672.2939785.
5. T. Hu and T. Song, "Research on XGboost academic forecasting and analysis modelling," *J. Phys.: Conf. Ser.*, vol. 1324, p. 012091, Oct. 2019. doi: https://doi.org/10.1088/1742-6596/1324/1/012091.
6. M. R. Machado, S. Karray, and I. T. de Sousa, "LightGBM: an Effective Decision Tree Gradient Boosting Method to Predict Customer Loyalty in the Finance Industry," in *2019 14th Int. Conf. Comput. Sci. Educ. (ICCSE)*, Toronto, ON, Canada, Aug. 19–21, 2019. IEEE, 2019. doi: https://doi.org/10.1109/iccse.2019.8845529.
7. W. Richert та L. P. Coelho, Building Machine Learning Systems with Python, Packt, 2013.
8. T. T. Teoh and Z. Rong, *Artificial Intelligence with Python*. Singapore: Springer Singap., 2022. doi: https://doi.org/10.1007/978-981-16-8615-3.
9. H. Jones, An Essential Beginners Guide to Artificial Neural Networks and Their Role in Machine Learning and Artificial Intelligence, CreateSpace Independent Publishing Platform, 2018.

Надійшла до редколегії 26.11.2024

Відомості про автора:

Павлюк Михайло Михайлович – студент кафедри комп'ютерних наук та інформаційних систем, Прикарпатський національний університети імені Василя Стефаника / Vasyl Stefanyk Precarpathian National University, Україна, email: pavliuk.mykhailo.dev@gmail.com; ORCID: https://orcid.org/0009-0002-1072-1912