МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра кібербезпеки

Лущевський Борис Леонідович

Алгоритми машинного навчання для виявлення та прогнозування атак на мережеву інфраструктуру / Machine Learning Algorithms for Detection and Prediction of Network Infrastructure Attacks

спеціальність: 125 – Кібербезпека

освітньо-професійна програма – Кібербезпека

Кваліфікаційна робота

Виконав студент групи
КБм -21
Б.Л. Лущевський

Науковий керівник
к.т.н., доцент Яцків Н.Г.

Кваліфікаційну роботу
Допущено до захисту:

«____» _____ ___2023 р.

Завідувач кафедри
_____ **В.В.Яцків**

ТЕРНОПІЛЬ - 2023

АНОТАЦІЯ

Кваліфікаційна робота на тему «Алгоритми машинного навчання для виявлення та прогнозування атак на мережеву інфраструктуру» на здобуття освітнього ступеня «Магістр» зі спеціальності 125 «Кібербезпека» освітньо-професійної програми «Кібербезпека» написана обсягом 64 сторінки і містить 28 ілюстрацій, 9 таблиць, 2 додатки та 30 джерел за переліком посилань.

Метою кваліфікаційної роботи є розробка системи виявлення вторгнень на основі машинного навчання.

Проведено дослідження IDS та IPS системи виявлення вторгнень, що дало можливість здійснити оцінку ефективності виявлення вторгнень. Досліджено індикатори атак, створені штучним інтелектом на основі аналізу мережевого трафіку.

Проаналізовано можливості Splunk Machine, щодо побудови моделы виявлення вторгнень та аналізу нетипової поведінки мережевого трафіку. Виконано дослідження побудови класифікаторів атак для системи виявлення вторгнень побудованої на основі класифікаторів.

Зпроектовано систему виявлення вторгнень на основі ML та здійснено вибір набору даних для навчання. Виконано семплювання проти дисбалансу класів та оцінка значущості та відбір ознак та скорочення ознакового простору та налаштування моделі. Проведено тестування побудованої моделі та здійснено оцінку її ефективності.

Ключові слова: АЛГОРИТМ МАШИННОГО НАВЧАННЯ, SPLUNK MACHINE, IDS, IPS.

ABSTRACT

The qualifying work on the topic "Machine Learning Algorithms for Detection and Prediction of Network Infrastructure Attacks" for the Master's degree in the specialty 125 "Cybersecurity" of the educational and professional program "Cybersecurity" is written in the volume of 64 pages of the appendix and 30 sources according to the list of references .

The purpose of the qualification work is to develop an intrusion detection system based on machine learning.

A study of IDS and IPS of the intrusion detection system was conducted, which made it possible to evaluate the effectiveness of intrusion detection. Attack indicators created by artificial intelligence based on network traffic analysis were studied.

The possibilities of Splunk Machine in building an intrusion detection model and analyzing atypical behavior of network traffic are analyzed. Research on the construction of attack classifiers for an intrusion detection system built on the basis of classifiers has been carried out.

An ML-based intrusion detection system was designed and a training dataset was selected. Sampling against class imbalance and significance assessment and feature selection and feature space reduction and model tuning were performed. The built model was tested and its effectiveness was evaluated.

Keywords: MACHINE LEARNING ALGORITHM, SPLUNK MACHINE, IDS, IPS.

ЗМІСТ

ВСТУП

Актуальність роботи. Алгоритми машинного навчання відіграють ключову роль у виявленні та прогнозуванні атак на мережеву інфраструктуру, оскільки ці атаки можуть бути вкрай складними та змінюватися в часі. Мережева інфраструктура генерує велику кількість даних. Алгоритми машинного навчання здатні ефективно обробляти і аналізувати великі обсяги даних, виявляючи аномалії та високоризикові патерни.

Мета роботи полягає в дослідженні можливостей штучного інтелекту щодо виявлення вразливостей мережевої інфраструктури.

Для досягнення даної мети ставились наступні завдання:

- дослідити IDS та IPS системи виявлення вторгнень;
- дослідити індикатори атак, створені штучним інтелектом;
- проаналізувати можливості Splunk Machine;
- виконати дослідження побудови класифікаторів атак;
- виконати проектування системи виявлення вторгнень на основі ML;
- здійснити вибір набору даних для навчання;
- семплювання проти дисбалансу класів та оцінка значущості та відбір ознак;’
- виконати скорочення ознакового простору та налаштування моделі;
- виконати тестування побудованої моделі.

Об’єкт дослідження – процеси навчання штучного інтелекту та побудови моделі опрацювання ознак.

Предмет досліджень – алгоритми машинного навчання на ознові відібраних ознак для виявлення вторгнень в мережеву інфраструктуру.

Методи дослідження базуються на методах та алгоритмах машинного навчання.

Наукова новизна одержаних результатів визначається наступним чином:

– Формалізовано та побудовано модель штучного інтелекту для виявлення шаблонів вторгнень в мережеву інфраструктуру.

Практична цінність одержаних результатів полягає в тому, що:

– реалізовано програмне забезпечення для виявлення шаблонів шкідливого трафіку в мережевій інфраструктурі.

Публікації та апробація до магістерської роботи.

1. Лущевський Б.Л., Алгоритм машинного навчання для виявлення та прогнозування атак на мережеву інфраструктуру. Збірник матеріалів проблемно-наукової міжгалузевої конференції «Автоматизація та комп'ютерно – інтегровані технології» (АКІТ -2023), Тернопіль, 2023. 109 -112 с.

2. Лущевський Б.Л., Николишин В.І., Дзядик В.А., Алгоритми машинного навчання для виявлення та прогнозування атак на мережеву інфраструктуру. Збірник матеріалів науково-практичної конференції молодих вчених, аспірантів та студентів «Кібербезпека та комп'ютерно – інтегровані технології» (КБКІТ -2023), Тернопіль, 2023. 17-21 с.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 IDS та IPS системи виявлення вторгнень

Сама назва IDS говорить сама за себе. Це пристрій або програмна реалізація, призначена для виявлення шкідливої активності. Ця шкідлива активність може бути в мережі, але може бути і в системі, і часто в файловій системі[1].

IDS намагається ідентифікувати порушення політики. Однією з особливостей IDS є те, що IDS зазвичай створює звіти. І не завжди обов'язково блокує, як це робить брандмауер. Він, безумовно, може блокуватися, і в цьому випадку більше говоримо про систему запобігання вторгненням, коли вона починає відкидати пакети та забороняти трафік. Але всі системи мають одну спільну рису — можливість повідомляти. Перш ніж почнемо вдаватися до механіки роботи IDS, наведемо кілька прикладів різних способів роботи IDS. Розглянуті міжмережеві екрани, які багато в чому схожі на замок на дверях. Авторизований трафік може проходити вільно, і для авторизації у фізичному світі потрібен ключ. Трафік, який не може бути дозволений, зупиняється. Так само працюють міжмережові екрани: або пропускають трафік, або його блокують. Так само може працювати і IDS. Таким чином, він може запобігати та зупиняти атаки, а також виявляти їх, що призводить до наступної аналогії. Однією з функцій, які мають багато систем виявлення вторгнень, є ведення журналу пакетів. Досить просто спостерігати за всім, що відбувається навколо нього, та зберігати інформацію для подальшого аналізу. Це не зупинить атаку, і хтось повинен буде повернутися та проаналізувати дані, які збираються. Так само, як люди переглядають відеозаписи, системним адміністраторам потрібно буде переглянути їх і подивитися, чи зможуть вони виявити будь-які шкідливі шаблони.

Це один із режимів, в якому можна побачити роботу систем виявлення вторгнень. Наведемо цю аналогію, коли говоритимемо про реєстрацію пакетів. Подивимося на інший приклад. Системи виявлення вторгнень часто

працюють у режимі аналізатора, тобто шукають шкідливі дії. Подібно до аналогії з протоколюванням пакетів або відеокамерою, це не зупинить атаку. Але що зробить сніффер, то це підніме тривогу. Це аналог датчика безпеки. Коли він спостерігає за поведінкою, якої не повинно бути, він повідомить когось про це. Тоді треба зробити якісь дії, але це не зупинить події[2]. Це важливо, і частково тому, що іноді IDS не в змозі запобігти виникненню інциденту. Наприклад, він може бути не в змозі зупинити зміну вмісту файлу на сервері кимось з правами адміністратора, але він може підняти прапор і сказати: «Гей, можливо це те, що вам слід вивчити докладніше, просто щоб бути впевненим, що це було зроблено якимось шкідливим процесом».

Продовжимо і розберемося, чим IDS відрізняється від брандмауера.

Одна річ, яка об'єднує IDS і брандмауер, полягає в тому, що вони обидва пов'язані з безпекою інформаційних систем. Часто для безпеки, пов'язаної з мережею, IDS також може переглядати файлові системи та системні конфігурації. Коли ми дивимося на файли, ми говоримо про брандмауер між зловмисником і цільовою системою, яка, наприклад, є внутрішньою мережею. Брандмауери часто існують, щоб спробувати обмежити доступ між мережами. Вони там, щоб спробувати запобігти вторгненню.

IDS набагато краще оцінює передбачувані вторгнення, як тільки вони відбуваються. Вони також дуже добре перехоплюють вторгнення, що відбуваються через брандмауер у мережі. Деякі IDS схожі на брандмауери, тому що вони враховуватимуть такі речі, як мережеві комунікації, але вони також можуть бути складнішими. Подібно до брандмауера, IDS може розташовуватися в декількох місцях інфраструктури. Наприклад, він може знаходитися всередині цієї внутрішньої мережі, контролюючи внутрішню мережу. Це може означати, що трафік проходить через нього та система веде себе як простий хост в мережі. Система виявлення вторгнень може просто перебувати на хості як хост-система виявлення вторгнень. Одна з речей, яка допомагає прояснити роль IDS, це коли ми думаємо про те, скільки всього відбувається за брандмауером. Розглянемо сучасну мережу та подумаємо про всі речі, які ховаємо за брандмауером. Одна з речей, які варто приховувати, це

сервери. В багатьох сучасних мережах є ціла купа таких мереж із внутрішніми мережевими сегментами, і також маємо такі речі, як бази даних. Бази даних є бажаною метою для зловмисників, тому що саме там лежать дані[3].

Маємо багато клієнтських машин, і неминуче є потреба захисту від зловмисних програм, як антивірус. Але це не обов'язково завадить зловмиснику скомпрометувати одну з цих машин і використовувати її як точку опори для подальшого доступу до інших ресурсів. Наприклад, модифікована шкідлива прошивка, яка походить від принтера, і все це знаходиться за брандмауером. Які можливості для виявлення шкідливої активності принтера щодо інших цілей у мережі. Смартфони є ще одним все більш популярним вектором атаки. Це знають маленькі комп'ютери, які є у кожного в кишені і часто підключені до тієї ж мережі, що інші пристрої. Це не просто мобільні пристрої, такі як смартфони, оскільки ера Інтернету речей означає, що ми отримуємо всі види інших пристроїв, підключених до мережі, і ми також бачимо багато вразливостей у них. Яким є захист від шкідливої активності з пристроїв IoT. Неминуче, що багато з них є бездротовими, тому ми маємо бездротові базові станції. Це приваблива ціль для зловмисника. Якщо вони можуть отримати доступ через бездротову мережу, це може поставити їх прямо за брандмауер. Навіть якщо видалимо всі мережеві компоненти та повернемось до простих фізичних носіїв, який буде захист, якщо щось шкідливе буде введено в середу через USB. Це дуже відомий вектор атаки. Антивірус на ПК може відреагувати, коли щось шкідливе починає взаємодіяти з серверами або базою даних у мережі, але що буде в змозі визначити цю ненормальну поведінку за брандмауером і підняти тривогу[4].

Саме тут система виявлення вторгнень відіграє унікальну роль. Те, що не може бути виконане одним брандмауером. Розглянемо ще кілька фактів, пов'язаних з IDS, перш ніж рухатися далі. Одна річ, яку можна побачити, коли справа доходить до IDS це те, що вона згадується як аналізатор пакетів. Сніффер пакетів походить від аналогії з камерою безпеки. Спостерігати за тим як проходить трафік, записувати його, але не обов'язково зупиняти його від будь-яких шкідливих дій. У цьому була суть, і в цьому їхня відмінність від

брандмауерів. Вони, звичайно, не збираються його зупиняти, але вони часто мають можливість попередити адміністратора. Однією з найважливіших особливостей IDS є можливість запису подій та фрагментів діяльності, щоб можна було повернутися та відтворити її пізніше[5]. Існують продукти IDS, які, наприклад, реєструють всі пакети через мережу. Він може нічого не ідентифікувати під час запису, і можуть відбуватися атаки, яких раніше не було. Системний адміністратори навіть не знали, що шукати, але коли ці пакети перехоплюються, і особливо після виявлення нових шаблонів атак, тоді є можливість повернутися і перевірити. Чи це запис цих необроблених пакетів, як це може робити сніффер, або реєструвати події, важливою частиною IDS є реєстрація інформації. IDS може виявити та повідомити, і це буде дуже пасивна система. Вона нічого не зупинить, вона просто дивитиметься на те, що відбувається навколо нього, або вона може повністю заблокувати діяльність шкідливого хоста[6].

Основна відмінність полягає в тому, що коли ми говоримо про запобігання, ми говоримо про бар'єр, щось, що зупинить шкідливий трафік. Він може, як і раніше, мати інші функції IDS, такі як реєстрація трафіку і подій, але він буде превентивно зупиняти атаки.

Однією з найбільш фундаментальних реалізацій IDS є система виявлення вторгнень на основі сигнатур, в якій відомі сигнатури як шкідливі шаблони. Іншими словами, це чорний список того, що вважаємо шкідливим. У нас є клієнт і клієнт буде надсилати запити через IDS. IDS проаналізує цей трафік і звернеться до бази даних сигнатур. У двох словах сигнатура являє собою великий набір шаблонів, і роль IDS полягає в тому, щоб зіставити трафік з відомими шкідливими шаблонами, а потім зробити запит про те, чи цей трафік проходить на хост, для якого призначалися дані. У такому випадку IDS може відігравати роль IPS, якщо вона відкидає пакети з шкідливим вмістом, або вона може реєструвати їх для подальшого перегляду та пропускання трафіку, незважаючи ні на що. Одна річ, яка завжди є ризиком, називається хибними спрацьовуваннями[7].

Було розумно заздалегідь отримати сигнатуру, ввести її в IDS і усунути вразливість, що лежить в основі. Але є проблема із підписами. Якщо думати про життєвий цикл підпису, була розроблена атака, і підпис підривав розширення TLS Heartbeat у протоколі SSL. Після того, як атака була встановлена та виявлено, що це ризик, люди почали готувати підписи, і в результаті системи були захищені. Сигнатуру було впроваджено у системи виявлення вторгнень.

1.2 Бездротова система запобігання вторгненням (WIPS)

Враховуючи широкомовну природу радіохвиль, які не обмежені стінами будівель, бездротові корпоративні мережі схильні до постійних атак з боку зловмисників. Саме тому питанню бездротової мережної безпеки слід приділяти особливу увагу[8].

Для вирішення проблем безпеки в бездротових локальних мережах, багато організацій розгортають або планують розгортання бездротових систем запобігання вторгненням (WIPS - Wireless Intrusion Prevention System). Вони призначені для моніторингу бездротової активності та визначення/запобігання спробам внутрішніх та зовнішніх мережевих вторгнень. Основуючи свій аналіз на канальному та фізичному рівнях мережної моделі OSI, цей інструмент дозволяє організаціям успішно ідентифікувати та захищати свої мережі від несанкціонованих точок доступу, атак на бездротові мережі та атак типу “відмова в обслуговуванні”[9].

Чим більшого поширення набувають бездротові мережі підприємств, тим витонченішими стають атаки на них з метою заволодіння даними. У відповідь безліч організацій виконує розгортання WIPS рішень для контролю політики заборони Wi-Fi (no-wireless policy), виявлення та запобігання атакам або проникненню в мережу. Ці рішення дають удосконалені можливості моніторингу та звітності для виявлення нападів на WLAN інфраструктуру та одночасного запобігання безлічі видів атак, перш ніж вони встигнуть вплинути на роботу мережі.

На схемі, що на рисунку 1.1 показані основні загрози, яким може наражатись корпоративна бездротова мережа.

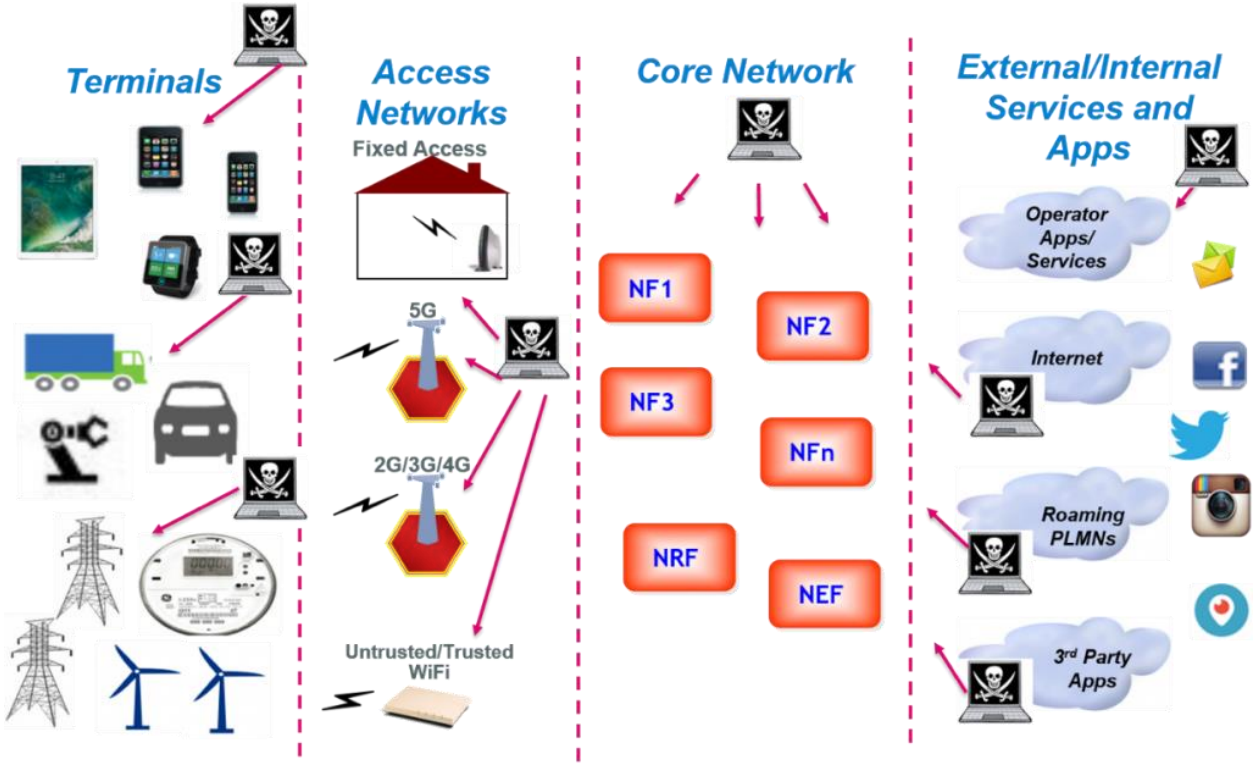


Рисунок 1.1 - Загрози бездротової мережі.

Як і в провідних мережах, у бездротових рішеннях для реалізації захисту існують компоненти, які опосередковано або безпосередньо виконують функції безпеки.

Контролер бездротових рішень виконує стандартні функції керування бездротовою мережею (точками доступу), а також реалізує додаткові можливості автентифікації користувачів[10].

Сканери мережевої безпеки нагадують собою точки доступу, але призначені виключно для мережного моніторингу та передачі на контролер чи IPS. Як правило, при побудові WIPS рішення спільно з бездротовою мережею співвідношення сканерів мережної безпеки стосовно точок доступу приймається як 1:4 або 1:5.

Точки доступу служать як станції для підключення пристроїв, але в деяких WIPS-рішеннях можуть одночасно виступати сканерами мережевої безпеки[11].

Пристрій Wireless IPS аналізує дані сканерів мережної безпеки та точок доступу та дає команди контролеру для запобігання вторгненням у разі їх виявлення. У складі WIPS-рішення може бути представлений окремий Wireless IPS пристрій або IPS у складі бездротового контролера (рисунок 1.2).

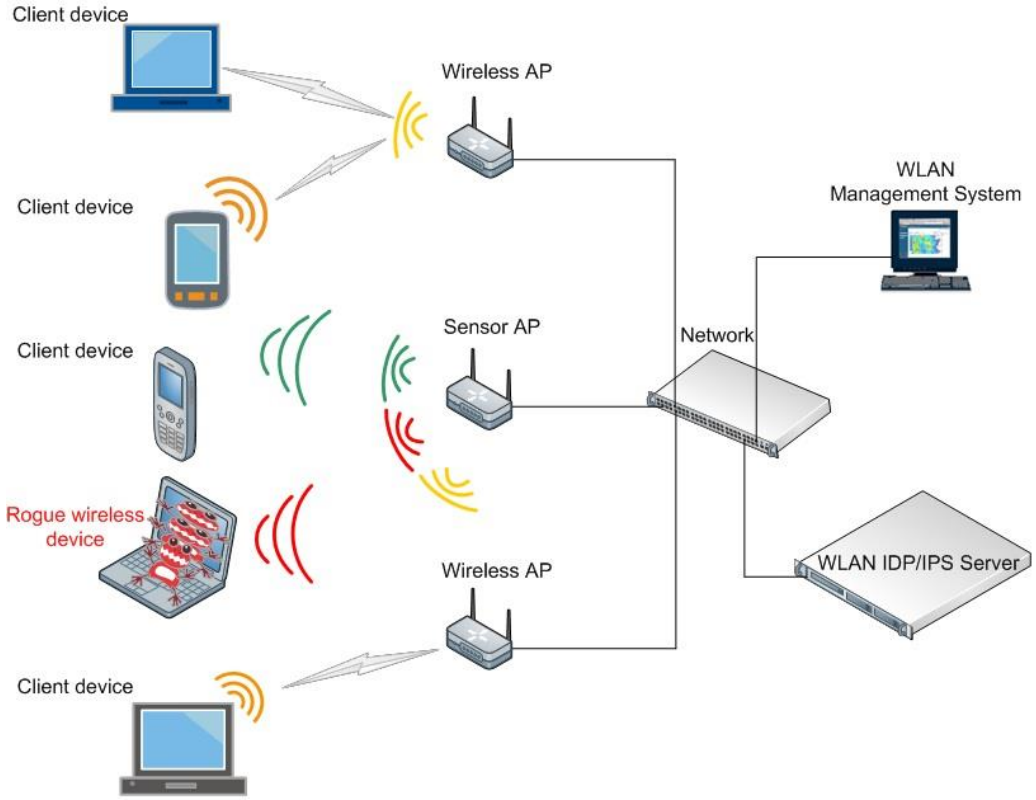


Рисунок 1.2 - Структура WIPS-мережі.

Організаціям є кілька варіантів впровадження WIPS-систем. Вони включають:

- Оверлейну (накладену) модель.

- Інтегровану (вбудовану) модель.

- Гібридну модель.

Оверлейна (накладена) модель — використовує спеціальні сенсори та систему керування для створення оверлейної WIPS-мережі над існуючою WLAN. Ця модель передбачає збільшення організацією її існуючої WLAN інфраструктури шляхом впровадження до неї спеціальних бездротових сенсорів або Air Monitors (AMs). AMs впроваджуються в існуючу WLAN як прості точки доступу, можуть розміщуватися на стінах або стелі та отримувати живлення

через PoE[12]. На відміну від точок доступу “Access Points” (APs), AMs зазвичай є пасивними пристроями, які контролюють довкілля наявність ознак атаки чи інший небажаної бездротової активності (рисунок 1.3).

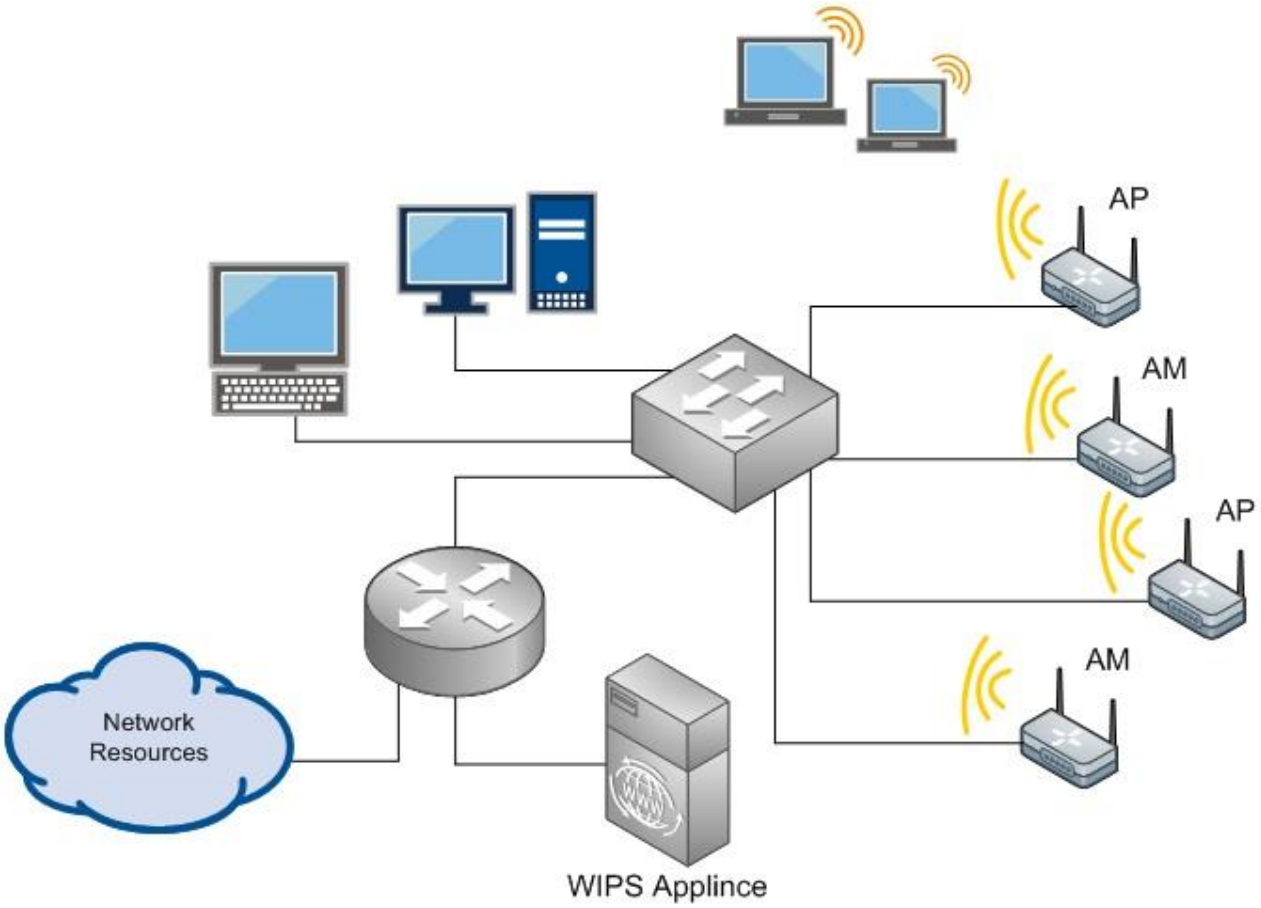


Рисунок 1.3 - Приклад оверлейної WIPS-мережі

Інтегрована (вбудована) модель — використовує одну консоль управління для WLAN та WIPS управління та здатна реалізувати оверлейну модель там, де діє політика заборони Wi-Fi. У цій моделі мережі організація посилює існуючу WLAN мережу AP/AM пристроями. Такі точки доступу є відповідальними за підключення клієнта до інфраструктури мережі, а також для аналізу бездротового трафіку з метою виявлення атак та інших небажаних активностей[13]. Ця модель часто є менш дорогою порівняно з накладеною моделлю, оскільки організації використовують одне й те саме апаратне обладнання для обслуговування користувачів та моніторингу бездротових мереж у радіусі дії, без залучення додаткових пристроїв (рисунок 1.4).

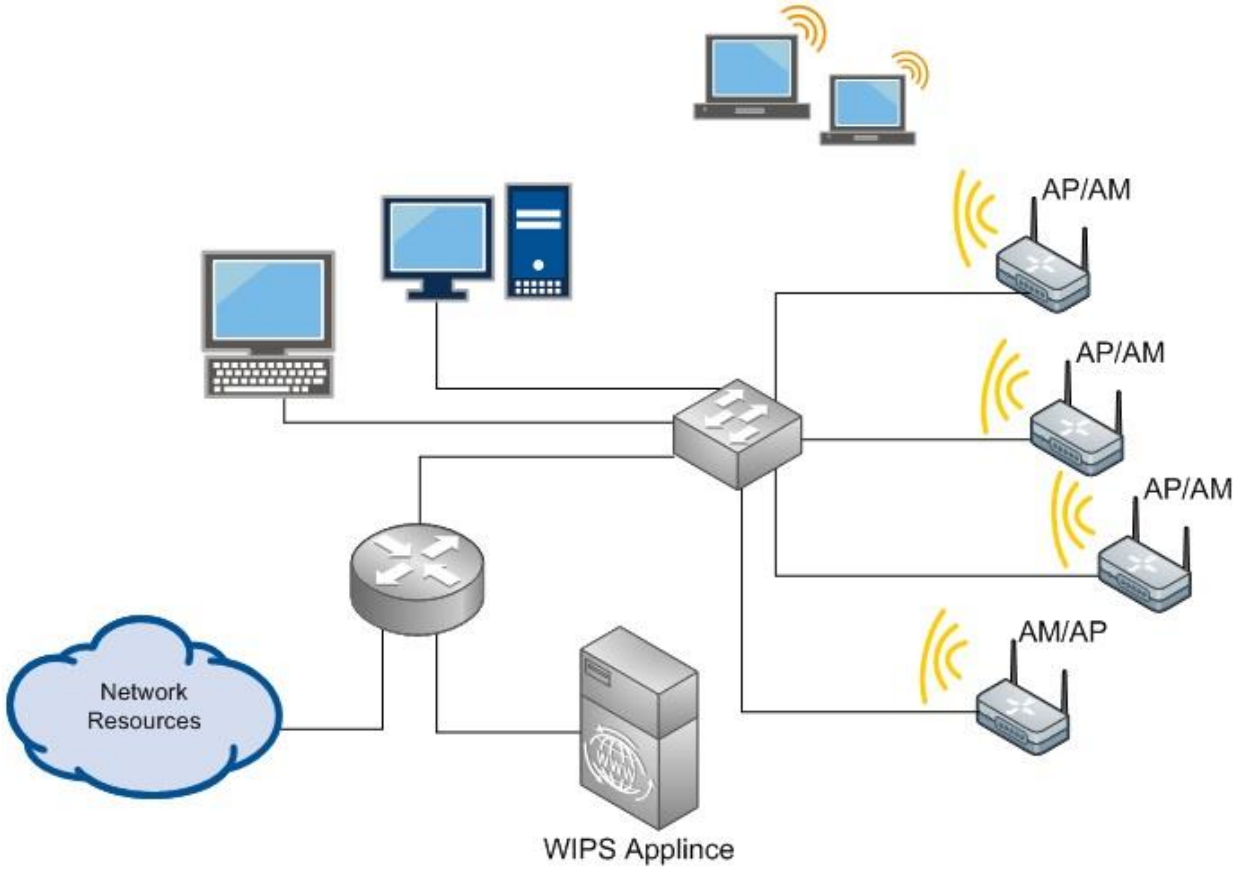


Рисунок 1.4 - Приклад інтегрованої WIPS-мережі

Гібридна модель моніторингу використовує сильні сторони двох попередніх моделей. Ця модель використовує переваги двох підходів, описаних раніше, для виявлення та запобігання бездротовим вторгненням. Так, компанії можуть використовувати APs та збільшити захист спеціальними AMs пристроями або розгорнути спеціальну моніторинг-інфраструктуру, що містить виключно AMs пристрої[14]. У цьому випадку аналіз, що виконується централізованим контролером, подібний до того, що використовується в накладеній моделі, а не при розгортанні інтегрованої моделі, де відбувається аналіз інформації від звичайних точок доступу (рисунок 1.5).

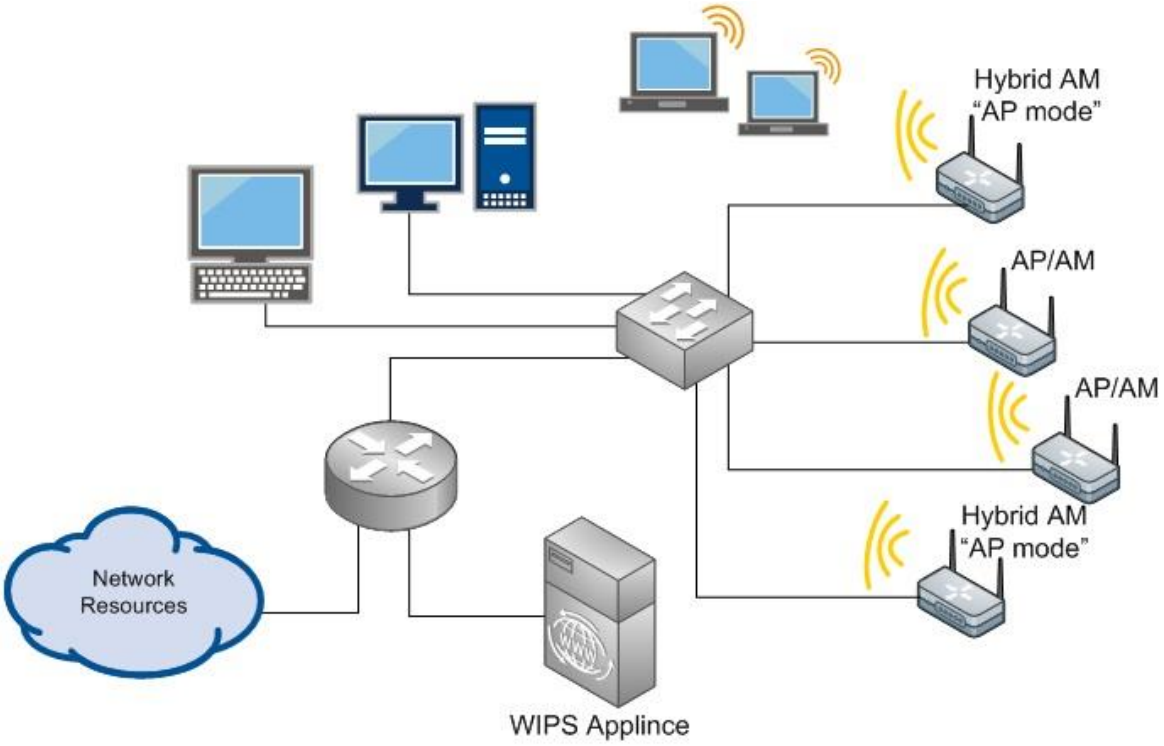


Рисунок 1.5 - Приклад гібридної мережі WIPS

Розгляд варіантів розгортання показує, що інтегрована модель покликана бути більш гнучкою, має підвищену радіочастотну видимість, знижену вартість володіння в порівнянні з оверлейною моделлю. А також чіткими механізмами аналізу та потужними механізмами запобігання вторгненням[15].

Гібридна модель пропонує ряд переваг над інтегрованою та накладеною, такі як велика гнучкість розгортання, сфокусованість на механізмах аналізу, більш комплексний метод виявлення атак та потужний механізм їх запобігання (придушення).

Існує безліч механізмів побудови WIDS/WIPS рішень, у кожному з яких є свої сильні та слабкі сторони. У цілому нині, гібридний підхід передбачає явні переваги проти альтернативними моделями. Тим не менш, для максимізації переваг інтегрованої моделі існуючі постачальники повинні бути ретельно вивчені для впевненості в тому, що вони надають необхідні можливості використання цих переваг та ефективно реагують на події внутрішніх та зовнішніх вторгнень[16].

1.3 Індикатори атак, створені штучним інтелектом

Індикатори атак, розроблені штучним інтелектом, покращують наявний рівень захисту (зображено на малюнку 1.6) за допомогою хмарного машинного навчання та виявлення загроз у реальному часі. Ці дані використовуються для аналізу подій під час функціонування програми та динамічної передачі індикаторів атак до сенсорів. Клієнт CrowdStrike подалі співставляє створені AI індикатори атак (інформацію про події у поведінці) з локальними подіями та даними файлів, щоб оцінити рівень опасности вводу-виводу. Штучний інтелект та наявні рівні сенсорної захисту взаємодіють асинхронно, враховуючи існуюче машинне навчання на основі сенсорів та наявні індикатори.

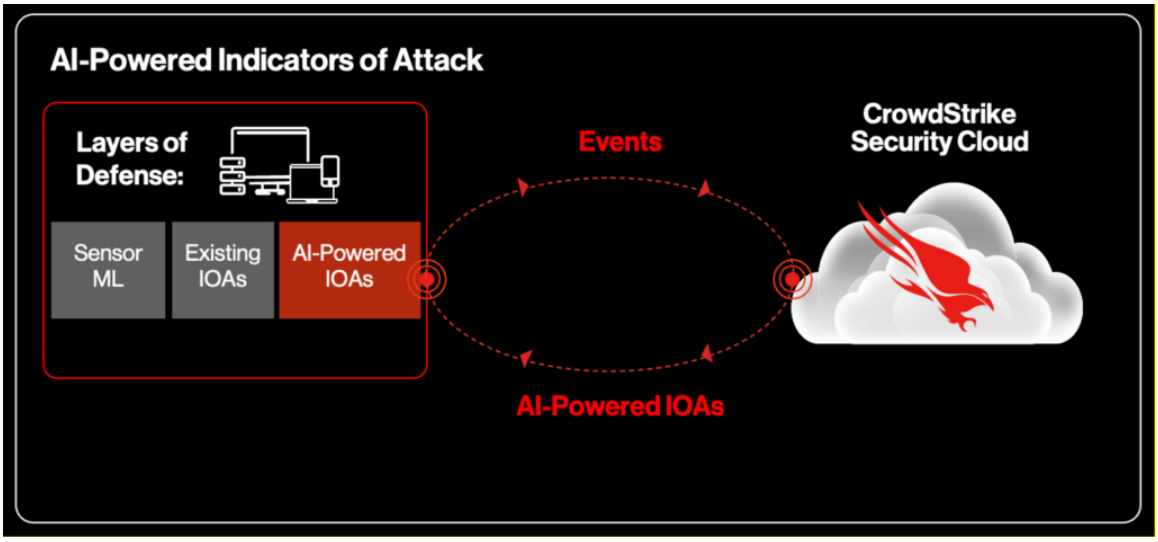


Рисунок 1.6 - IOA на базі штучного інтелекту, що згенеровані хмарними моделями машинного навчання, заснованими на багатій телеметрії CrowdStrike Security Cloud

Серед ереваг IOA на основі штучного інтелекту можна виділити ряд положень.

Виявлення нових класів загроз вдвічі швидше: позволяє бути кроком попереду зловмисників, передбачати зміну їх методів та забезпечувати активний локальний захист, що співпрацює з наявними рівнями захисту[17].

Автоматизована профілактика атак за допомогою точного виявлення: хмарні моделі штучного інтелекту обмінюються індикаторами атак із сенсором CrowdStrike Falcon в режимі реального часу, щоб припинити атаки, незалежно від конкретного шкідливого програмного забезпечення або інструментів.

Зменшення кількості помилкових спрацьовувань та підвищення продуктивності: IOA на основі штучного інтелекту, озброєні експертним досвідом та масштабовані в хмарі, синтезують інформацію від всесвітньо відомої команди пошуку загроз CrowdStrike, щоб зменшити кількість помилкових спрацьовувань та максимізувати продуктивність аналітиків у великих масштабах.

IOA (індикатори атак на основі подій) визначаються як послідовності спостережень, що вказують на активну або триваючу спробу злому системи, дозволяючи аналітикам встановити, як зловмисники спочатку отримують доступ до мережі і визначити їхню мотивацію або цілі. Це надає можливість скласти повну картину атаки, незалежно від типу шкідливого програмного забезпечення або інструментів[18].

IOA також використовує розширений аналіз поведінки для моделювання та прогнозування дій зловмисника, підвищуючи рівень захисту від майбутніх атак.

Використання IOA дає багато переваг порівняно з використанням лише індикаторів компрометації (Indicators Of Compromise, IOC). По-перших, IOA дозволяє виявляти атаки до або під час їхнього розвитку, забезпечуючи превентивний та проактивний захист. У випадку використання лише IOC, організація завжди реагує на атаки після їхньої вже появи (реактивний підхід)[19].

Рисунок 1.7 - Індикатори компрометаціїта показники атаки CrowdStrike

По-друге, зосереджуючись на зловмисних мотивах, а не на конкретних шкідливих програмах або інструментах, які використовуються, IOA дозволяють клієнтам адаптуватися до нових класів атак і змін методів зловмисної діяльності. Наприклад, це включає атаки без використання шкідливого програмного забезпечення або безфайлові атаки, які становлять 62% атак за останній рік[20].

Зрештою, оскільки IOA є універсальними за своєю природою, їх можна аналізувати паралельно (забезпечуючи ефективність обчислень та масштабування), і вони не потребують так частого оновлення, як підходи на основі сигнатур (як в разі IOC)[21].

Дотепер процес створення IOA в основному покладався на прикладний досвід всесвітньо відомих мисливців за загрозами, результатом чого стали складні та високоточні індикатори. Але для того, щоб клієнти могли виявляти загрози завтрашнього дня, процес виявлення та класифікації активних атак повинен бути швидшим за кіберзловмисників, не втрачаючи при цьому неймовірної точності IOA, створених експертами. Для досягнення цього результату CrowdStrike об'єднала людський досвід та машинне навчання для розширення можливостей створення IOA та підвищення якості створених експертами IOA, зберігаючи високу точність[22].

Використовуючи потужність CrowdStrike Security Cloud для навчання цих зразків на хмарній платформі CrowdStrike Falcon, моделі машинного навчання можуть аналізувати величезні обсяги аналізу загроз з неперевершеною швидкістю, масштабом та точністю[23]. Впроваджуючи потужність хмарного машинного навчання у процес створення IOA, клієнти продовжують отримувати вигоду від проактивних, високоякісних сигналів, наданих IOA, тепер зі швидкістю та масштабом хмари[24].

З моменту впровадження у виробництво, хмарні моделі машинного навчання точно ідентифікували понад 20 нових шаблонів індикаторів, які пізніше підтвердили експерти та застосували на платформі Falcon для автоматизованого виявлення та запобігання. Нижче розглянемо два приклади

виявлених тактик противника, які призвели до створення нових IOA для корисних навантажень після експлуатації та атак за допомогою PowerShell.

Корисне навантаження після експлуатації: Це код, який зловмисник передає хосту після отримання вихідного доступу. IOA на основі штучного інтелекту ідентифікують ці корисні навантаження, поєднуючи вихідні дані статичної моделі штучного інтелекту сенсора Windows з файлом, який запускається, і зі знаннями, доступними тільки через CrowdStrike Security Cloud. Ця інформація включає в себе інформацію про походження процесу та методи його запуску, що дозволяє досягти неймовірної точності виявлення та надавати докладні індикатори, що перевищують результати, досягнуті традиційними статичними або поведінковими методами[25].

Атаки з використанням PowerShell: Зловмисники часто використовують PowerShell для доставки шкідливого коду або виконання шкідливих дій, коли не застосовуються індикатори компрометації (IOC). Ці види атак важко ідентифікувати, і традиційні методи на основі сигнатур легко обхід. Використовуючи моделі глибокого навчання для автоматичного виділення найбільш релевантних частин коду зі сценаріїв PowerShell, можна ідентифікувати безфайлові загрози, які керуються PowerShell, і захистити системи від них.

Машинне навчання залишається критично важливим інструментом для виявлення нових закономірностей у даних та проведення глибокого аналізу поведінки для розуміння намірів та цілей зловмисників. Компанія CrowdStrike, лідер у галузі хмарного захисту кінцевих точок, хмарних робочих навантажень, ідентифікації та даних, має намір продовжувати використовувати сукупну міць штучного інтелекту та хмарних технологій для підвищення ефективності захисту, протидії методам роботи зловмисників та надання клієнтам допомоги у припиненні атак[26].

2 ДЕТЕКТУВАННЯ ТА КЛАСИФІКАЦІЯ МЕРЕЖЕВИХ АТАК ЗА ДОПОМОГОЮ SPLUNK MACHINE

2.1 Використання Splunk Learning Toolkit

У сучасних умовах впровадження цифрових технологій у різні галузі економіки, цифровізації державного управління, сфер охорони здоров'я, освіти та науки, зростання кількості інтернет-послуг та мобільних пристроїв, що використовуються, стають все більш актуальними питання забезпечення безпеки систем стільникового зв'язку. Стає дедалі важче виявляти численні і складні загрози кібербезпеки у міру розвитку та розширення джерел та методів реалізації кібератак. Класичні підходи виявлення мережевих атак, які значною мірою покладаються на статичне зіставлення, такі як сигнатурний аналіз, чорні списки або шаблони регулярних виразів, обмежені в гнучкості та є малоефективними для раннього виявлення аномалій та оперативного реагування на інциденти інформаційної безпеки. Для вирішення цієї проблеми пропонується використання алгоритмів машинного навчання, які можуть забезпечити нові підходи та вищі показники виявлення шкідливої активності в мережі.

В роботі будемо використовувати платформу аналізу даних Splunk Enterprise з використанням розширення та додатковий інструментарій машинного навчання Splunk Machine Learning Toolkit для створення, навчання, тестування та перевірки класифікатора мережевих атак. Продуктивність запропонованої моделі була оцінена з використанням чотирьох алгоритмів машинного навчання, таких як дерево рішень (a decision tree), метод опорних векторів (a support vector machine), випадковий ліс (a random forest) та подвійний випадковий ліс (a double random forest). Експериментальні результати показують, що це використані алгоритми машинного навчання можуть ефективно використовуватися виявлення мережевих атак, а метод подвійного випадкового лісу має найкращу точність виявлення атак типу «відмова у обслуговуванні».

Впровадження цифрових технологій у різні галузі економіки, цифровізація сфер державного управління, освіти та охорони здоров'я, зростання кількості інтернет-сервісів та мобільних пристроїв, що використовуються споживачами, з одного боку, та зростання кількості джерел і методів реалізації кібератак в останні роки, з іншого боку, ставлять перед операторами мобільного бездротового зв'язку завдання розвитку з урахуванням проблем кібербезпеки. Системи стільникового зв'язку постійно еволюціонують, покращуючи мережеву архітектуру, інтерфейси та протоколи, підвищуючи пропускну спроможність передачі даних. Але еволюція призводить і до появи нових уразливостей, які можуть використовуватися для реалізації атак як на мережу доступу, так і на базову мережу, викликаючи відмову в обслуговуванні чи можливість виконання шкідливого функціоналу. Все це призводить до необхідності розробки надійних систем виявлення мережевих атак та механізмів швидкого реагування на інциденти кібербезпеки [27]. Прикладом такого підходу є системи управління інформацією та подіями безпеки (Security Information and Event Management, SIEM), які стають важливою компонентою екосистеми мережевої безпеки поряд з міжмережевими екранами, системами запобігання вторгненням та виявлення вторгнень (Intrusion Prevention System / Intrusion Detection System, IPS/ID), рішеннями щодо статичного та динамічного аналізу шкідливих файлів [28].

Раніше нами було спроектовано програмний комплекс для оперативного центру інформаційної безпеки підприємства(ОЦІБ) [3]. У складі програмного комплексу реалізовано SIEM-систему для моніторингу та розслідування інцидентів інформаційної безпеки з візуалізацією інтерактивними аналітичними панелями (дашбоардами), які формуються на основі збору та аналізу різноманітних даних з різних програмно-апаратних засобів, реалізованих в ОЦІБ. Серед різних доступних варіантів SIEM рішень з відкритим вихідним кодом та ліцензійних продуктів вибрано Splunk Enterprise Security (ES), побудований на основі Splunk R Enterprise [28]. Splunk R Enterprise - це провідна в галузі платформа операційної аналітики, яка дозволяє аналізувати машинні дані, що може розширити можливості усунення несправностей,

підвищити продуктивність та покращити стан безпеки інформаційно-комунікаційної інфраструктури компанії [15]. У цій роботі представлені результати досліджень з реалізації класифікатора мережевих атак на основі алгоритмів машинного навчання з використанням додаткового інструментарію Splunk Machine Learning Toolkit [16], еталонної бази даних UNSW-NB15 [17], інтерпретатора мови Python та пошукових команд мови SPL (Splunk Processing Language). Отримано оцінки точності класифікації мережевих атак з використанням чотирьох алгоритмів машинного навчання (дерево рішень, метод опорних векторів, випадковий ліс та подвійний випадковий ліс). Експериментальні результати показали ефективність використання алгоритмів машинного навчання для детектування нормальної та аномальної мережевих активностей, а також для класифікації мережевих атак

2.2 Дослідження існуючих рішень методів детектування кібератак

В останні роки зростає кількість досліджень, присвячених вдосконаленню методів виявлення мережевих атак, з них низка досліджень з використання машинного навчання виявлення аномалій в мережі.

В оглядовій роботі [8] описано безліч методів виявлення вторгнень для боротьби з загрозами мережевої безпеки, які можна загалом розділити на системи виявлення вторгнень на основі сигнатур та системи виявлення вторгнень на основі аномалій.

Автори показали, що традиційні підходи на основі сигнатур перевіряють мережеві пакети і намагаються зіставити їх з базою даних хеш відомих атак. Але ці методи не можуть ідентифікувати атаки нульового дня, таргетовані атаки, поліморфні варіанти шкідливого програмного забезпечення (ВПЗ). Аналіз мережевих пакетів, ВПО на основі аномалій виявився більш ефективним для детектування таких складних кібератак завдяки тому, що розпізнавання аномальної активності користувача не залежить від бази даних сигнатур [29].

Методи детектування кібератак на основі аномалій автори роботи [30] поділяють на три групи: засновані на статистиці, засновані на знаннях та

засновані на машинному навчанні. Ці класи методів разом із прикладами їх підкласів представлені малюнку 2.1.

У рамках першого підходу проводиться збір та перевірка кожного запису даних у наборі елементів з подальшою побудовою статистичної моделі нормального поведінки користувача. За другого підходу дослідники ідентифікують нормальні та аномальні специфікації протоколів та екземпляри мережевого трафіку. Для реалізації останньої групи методів створюються шаблони нормального трафіку та аномальних ситуацій та поведінки користувачів, на яких проводиться навчання, тестування класифікаторів на основі методів машинного навчання шаблонів із наборів навчальних даних.

Автори огляду детальніше розглянули кілька досліджень з використанням методів машинного навчання для виявлення атак нульового дня та показали, що існуючі підходи можуть мати проблеми з генеруванням та оновленням інформації про нові атаки і дають велику кількість помилкових спрацьовувань або низьку точність. Це пов'язано з тим, що більшість існуючих методів машинного навчання навчаються на основі наборів даних DARPA/KDD99, зібраних у 1999 році, які не включають нові дії шкідливих програм. Таким чином, існує потреба в нових і повних наборах даних, що містять широкий Спектр дій шкідливого ПЗ.

Далі подано короткий огляд досліджень, у яких автори використовували той або інший метод машинного навчання для виявлення аномалій у мережі чи мережевих атак певного типу. У роботі [20] виявлення атак типу «відмова у обслуговуванні» (Denial of Service, DoS) використовується модель, заснована на збиранні даних файлів логування подій на платформі Spark, та класифікація мережевих атак за методом випадкового лісу (A random forest) з точністю 99,2%. Автори показали, що модель може використовуватись для обробки великомасштабних потоків DNS-запитів, що корисно для практичного використання.

Автори роботи [21] запропонували підхід для виявлення DoS-атак на основі інфраструктури великих даних Запропоноване рішення включає модулі збору мережного трафіку в реальному часі та модуль виявлення. Модуль збору

мережевого трафіку збирає та витягує особливості трафіку, використовуючи модель мікропакетної обробки. Модуль виявлення використовує алгоритм класифікації Spark-ML на основі методу випадкового лісу виявлення DoS-атак на мережевому трафіку. Для оцінки точності виявлення алгоритм оцінювався та порівнювався на наборах даних NSL-KDD та UNSW-NB15.

Схема детектування та класифікація мережевих атак за допомогою Splunk Machine Learning Toolkit Detection Systems, AIDS [22] приведена на рисунку 2.1.

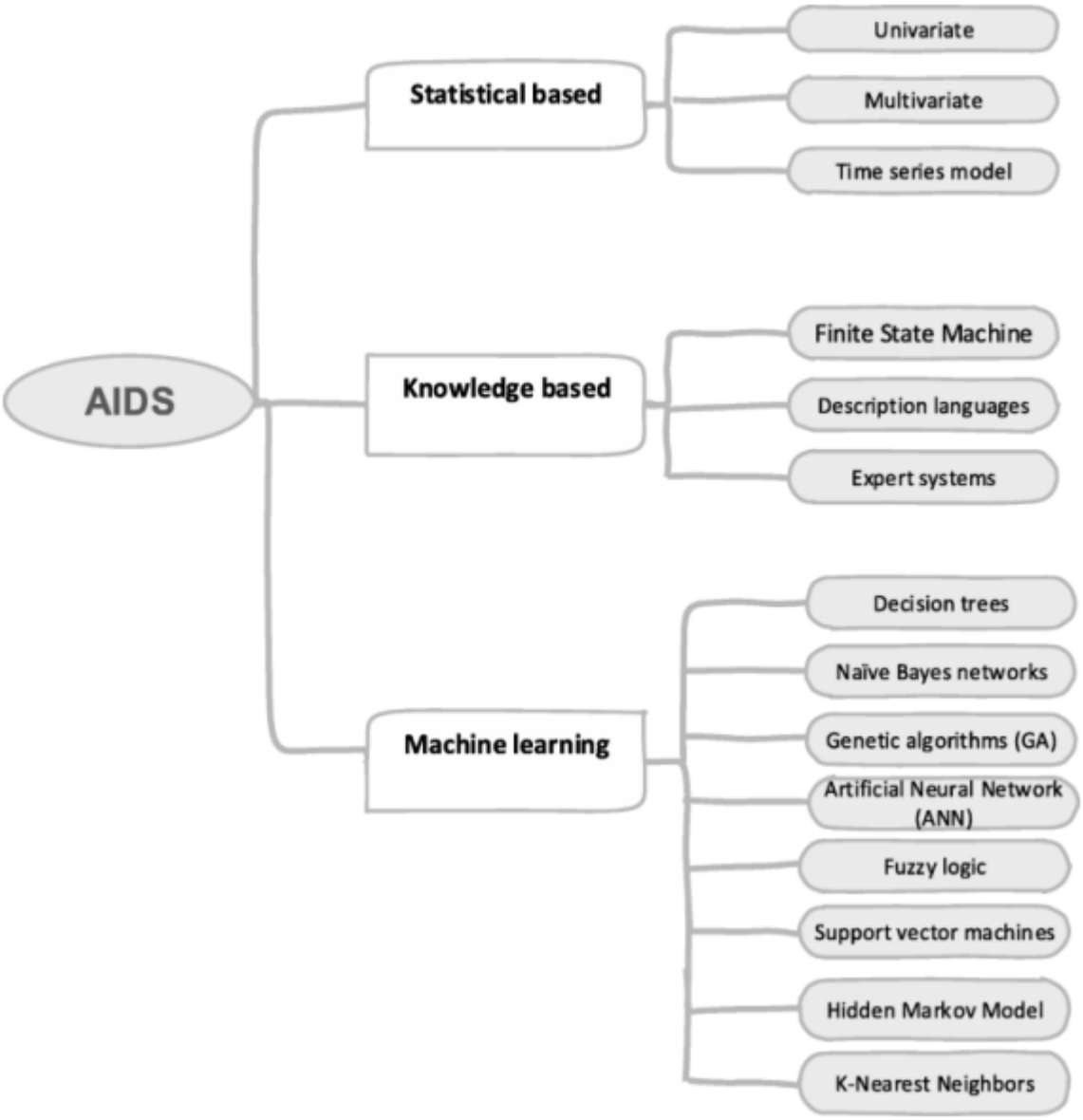


Рисунок 2.1 - Класифікація систем детектування кібератак на основі аномалій (Anomaly-based Intrusion)

Рівень виявлення досягнув рівня точності 99,95% і 98,75%, відповідно, для двох наборів даних.

У роботі [23] запропоновані моделі глибокого навчання для виявлення та зниження ризику DDoS-атак, націлених на централізований контролер у програмновизначуваній мережі, з використанням рекурентної нейронної мережі з довгою короткостроковою пам'яттю (Long short-term memory, LSTM) та згортковою нейронною мережею neural network, CNN). Точність класифікації DDoS-атак згорткової нейронної мережею була невисокою, 66%. Модель LSTM дала точність 89,63%, що перевищує показники результативності 86,85% і 82,61% для класичних методів машинного навчання - методу опорних векторів (a support vector machine SVM) та наївного класифікатора байєсівського (Naive Bayes classifier) відповідно.

Автори роботи [24], використовуючи два підходи до машинного навчання (випадковий ліс та багатошаровий персептрон) та платформу великих даних (Spark ML), змогли з точністю 99,5% виявити атаку типу «відмова в обслуговуванні» в режимі реального часу за кілька мілісекунд .

Резюмуючи короткий огляд розроблених систем, можна зробити припущення, що застосування сучасних методів машинного навчання спільно з технологіями збору машинних даних у SIEM-системах на платформі операційної аналітики може бути ефективним підходом для розробки систем детектування аномалій та класифікації мережевих атак з метою раннього розпізнавання на інциденти кібербезпеки.

Вихідні дані та методика досліджень. У цьому розділі представлено методику досліджень з проектування, реалізації та тестування інтелектуальної системи класифікації аномального та нормального мережевого трафіку на основі застосування традиційних алгоритмів машинного навчання та нового методу подвійного випадкового лісу. Для тестування вищевказаної методики проведено експериментальні дослідження з детектування аномалій для штучно згенерованого DNS та найбільш відомих категорій кібератак. Для реалізації експериментів була використана платформа для аналізу даних Splunk Enterprise з використанням розширення Machine Learning Toolkit [25]. Ця бібліотека

містить понад 30 найпоширеніших алгоритмів машинного навчання з відкритим вихідним кодом мовою Python. Підготовка даних виконана на сервері Splunk за допомогою вбудованої мови Search Processing Language (SPL). Для виконання SPL-скриптів із набором даних використовується інтепретатор мови Python. Для проведення експерименту були обрані наступні алгоритми машинного навчання, як такі, що зустрічаються у пов'язаних роботах, що дозволяє провести в подальшому порівняння експериментальних результатів з результатами інших дослідників:

- Дерево рішень (a decision tree, DT);
- випадковий ліс (a random forest, RF);
- метод опорних векторів (a support vector machine, SVM).

Дані методи є класичними, їх опис наведено у великій кількості підручників та наукових статтях, наприклад, у роботах [25, 26].

Також в експерименті використано метод подвійного випадкового лісу (a double random forest, DRF) як модифікований метод випадкового лісу. У роботі [26] авторами показано, що точність класифікації методу випадкового лісу можна поліпшити, якщо використовувати ансамбль різноманітних дерев, ніж дерева з мінімальним розміром вузлів. Для цього було запропоновано новий метод початкового завантаження навчальної вибірки на кожному вузлі у процесі створення дерева замість початкового завантаження на кореневому вузлі, як у класичному методі випадкового лісу. Експериментальні дослідження розпізнавання зображень рукописних цифр від 0 до 9 показують, що запропонований метод DRF набагато перевершує класичні ансамблеві методи класифікації [27]. Тому варто було спробувати використати алгоритм DRF з урахуванням особливостей задач класифікації мережевих атак.

2.3 Дослідження побудови класифікаторів атак

В якості навчальної та тестової бази даних використана база даних UNSW-NB15, яка була створена в лабораторії Cyber Range ACCS, з

використанням генератора аномального мережевого трафіку, що створюється 9 видами мережевих атак (див. таблиця 1).

Згідно з описом авторів роботи [20] для захоплення пакетів мережного трафіку обсягом 100 ГБ був застосований програмний інструментарій tcpdump, який також використовувався для поділу трафіку на фрагменти 1000 МБ. Далі з pcap-файлів з використанням програмного забезпечення Argus створено 4 CSVфайли, що містять ключові характеристики як надійні ознаки (атрибути) для класифікації кількох типів атак. Виділено 49 ключових характеристик (атрибутів для класифікації атак), які поділені на декілька груп: основні (Basic), контентні (Content) та тимчасові (Time), як це показано в таблиці 2.1.

Таблиця 2.1 - Розподіл записів за категоріями атак у базі даних UNSW-NB15

Типи мережевих атак	Кількість записів	Description
Normal (нормальний трафік)	2 218 761	природні дані нормального трафіку
Fuzzer (трафік, створюваний програмою фаззером)	24 246	спроба призупинити роботу мережі шляхом передачі випадково згенерованих даних
Analysis (трафік, створюваний програмою аналізатором)	2 677	трафік містить різні атаки сканування портів, html-файлів та проникнення спам
Backdoors (бекдори)	2 329	техніка, при якій механізм безпеки системи потай обходиться для доступу до комп'ютера або його даних
DoS (атаки типу «відмова в обслуговування», в тому числі розподілені)	16 353	зловмисна спроба зробити сервер або мережевий ресурс недоступний для користувачів

продовження таблиці 2.1

Exploits (атаки з використанням експлойтів)	44 525	зловмисник знає про проблему безпеки в операційній системі, використовує ці знання, застосовуючи вразливість
Generic (атака проти шифрів)	215 481	загальний метод працює проти всіх блокових шифрів (із заданим розміром блоку та ключа), без обліку структури блочного шифру
Reconnaissance (пасивні атаки з метою розвідки)	13 987	містить усі спроби проникнення в мережу, які можуть імітувати атаки зі збором інформації
Shellcode (двійковий шкідливий код)	1 511	невеликий фрагмент коду, який використовується в якість корисного навантаження при експлуатації вразливості у програмному забезпеченні
Worms (шкідливий код типу «хробак»)	174	шкідливий код, який копіює себе, щоб поширитись на інші комп'ютери

Приклад основних характеристик наведено у таблиці 2.2. Записи файлу UNSWNB15_1.csv використані як навчальна вибірка, а файлу UNSW-NB15_2.csv – як тестова.

Схема експерименту представлена на рисунку 2.2. Вибірка потрібної категорії мережевих атак здійснювалася за допомогою команди Splunk "search":

search attack_cat=Normal OR attack_cat=Dos,

де останній параметр вказує на тип атаки (Таблиця 2.2), у прикладі використаний параметр Dos – атака типу «відмова у обслуговуванні» (Denial of Service, DoS-атака).

На етапі передобробки перетворюються нечислові поля (рядки або символи) в числові, використовуючи одноразове кодування, зважаючи на те, що алгоритми машинного навчання працюють із числовими даними.

Проводиться нормування даних, в результаті дані запису мережевого трафіку перетворюються на числову матрицю з числових значень атрибутів, що подаються на алгоритм класифікації за одним із 4-х методів машинного навчання.

Таблиця 2.2 - Приклади атрибутів потоків

№ атрибуту	Позначення	Тип даних	Опис
1	srcip	N (номінальний)	IP Адреса джерела (Source IP address)
2	sport	I (ціле число)	Номер порту джерела (Source port number)
3	dstip	N	IP Адреса одержувача (Destination IP address)
4	dsport	I	Номер порту одержувача (Destination port number)
5	proto	N	Протокол транзакції (Transaction protocol)
29	stime	T (мітка часу)	Час початку запису (record start time)
34	synack	F(число з плаваючою комою)	Година між пакетами SYN та SYN_ACK у TCP The time between the SYN and the SYN_ACK packets of the TCP
48	attack_cat	N	Назва категорії атаки (Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, Worms) див. Таблицю 1
49	Label	B (двійковий)	0 для запису нормального трафіку та 1 для запису трафіку при мережевій атаці

Форму запиту пошуку даних з індексованого дата сету та застосування до нього певного алгоритму машинного навчання наведено в наступному прикладі:

index=nb15_test|search attack_cat=Normal OR attack_cat=Dos|fields label, ackdat, ct_dst_ltm, ct_dst_sport_ltm, ct_dst_src_ltm, ct_flw_http_mthd, ct_ft_

ct_src_dport_ltm, ct_src_ltm, ct_srv_dst, ct_srv_src, ct_state_ttl, dbytes, dinpkt, djit,

dload, dloss, dmean, dpkts, dtcpb, dttl, dur, dwin, is_ftp_login, is_sm_ips_ports, proto, rate, response_body_len, sbytes, service, sinpkt, sjit, sload, sloss, smean, spkts, state, synack, tcprtt, trans_depth |apply DRF_DOS

Виявлення аномалій або детектування певного виду мережевих атак відноситься до завдань бінарної класифікації, для оцінки якої використовуються наступні: матриця помилок (a confusion matrix, CM), акуратність (accuracy), точність (a precision), повнота (a recall) та F-міра.

Таблиця 2.3 представляє вигляд матриці помилок класифікатора кібератак, яка може бути використана для оцінки продуктивності алгоритмів машинного навчання. Кожен стовпець матриці представляє екземпляри у прогнозованому класі, а кожен рядок представляє екземпляри у реальному класі.

Таблиця 2.3 - Матриця помилок завдання класифікації атак

Actual Class	Predicted Class	
Class	Normal	Attack
Normal	True negative (TN)	False Positive (FP)
Attack	False Negative (FN)	True positive (TP)

Ефективність алгоритму класифікації зазвичай оцінюються на основі наступних 4 стандартних показників, формули для обчислення яких представлені нижче на основі введених позначень у матриці помилок (таблиця 2.3). Показник Accuracy (акуратність чи правильність, точність

класифікації) визначається як частка правильних результатів стосовно всіх можливих варіантів передбачення, яка досягається класифікатором, як показано на рисунку 2.2.

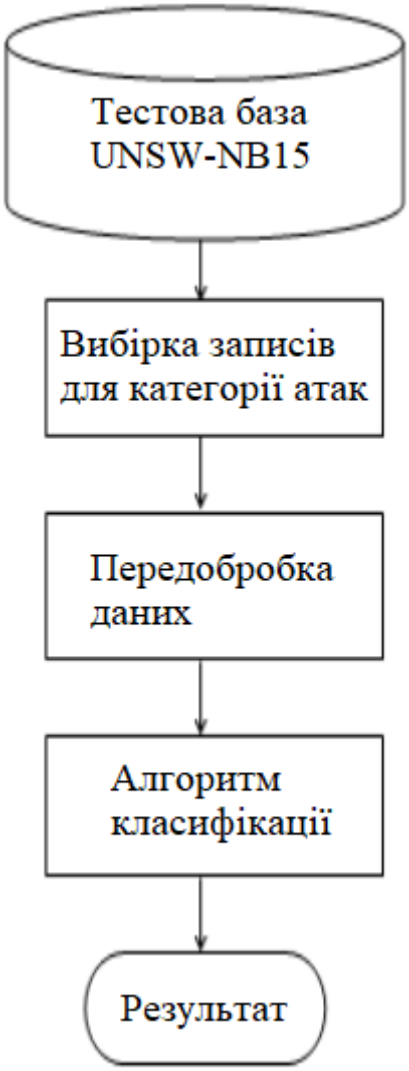


Рисунок 2.2 - Схема експерименту з класифікації атак із використанням алгоритмів машинного навчання

$$\text{Accuracy}=\frac{TP+TN}{TP+FP+FN+TN} \tag{2.1}$$

Метрика Precision (прецензійність, точність класифікації) визначається як частка правильно визначених екземплярів класів мережевих атак, і при цьому дійсно є трафіками мережевих атак:

$$\text{Precision} = \frac{\text{TP}}{TP + FP} \tag{2.2}$$

Чутливість (повнота) методу класифікації Recall, або True Positive Rate (TPR), розраховується як відношення кількості правильно передбачених атак до загального кількості атак:

$$\text{Recall} = \frac{\text{TP}}{TP + FN} \tag{2.3}$$

Якщо всі мережеві вторгнення правильно класифіковані, то показник TPR дорівнює 1, що дуже рідко для реальних систем виявлення вторгнень. Значення TPR частіше прагне 1.

Також використовується спосіб об'єднання кількох критеріїв в один агрегований критерій якості F1-score (F-мера) як середнє гармонійне значення точності та чутливості:

$$\text{F1} - score = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.4}$$

F-міра досягає максимуму при чутливості та точності, рівними одиниці, і близька до нуля, якщо один із аргументів близький до нуля.

Усі формули для обчислення описаних вище метрик ефективності алгоритму машинного навчання реалізовано у бібліотеці scikit-learn інструменту Machine Learning Toolkit, так само, як і три алгоритми машинного навчання, що використовуються (дерево рішень DT, випадковий ліс RF, метод опорних векторів SVM). Для реалізації класифікації з використанням нового методу – подвійний випадковий ліс DRF – мовою Python був розроблено додатковий плагін (програмний модуль) SPL-DRF.

При реалізації модуля SPL-DRF з метою підвищення швидкодії виконується паралельна побудова дерев та паралельне обчислення прогнозів за

допомогою параметр *n_jobs*. Якщо *n_jobs=k*, тоді обчислення поділяються на k завдань, які та виконуються на k ядрах машини (рисунок 2.3).

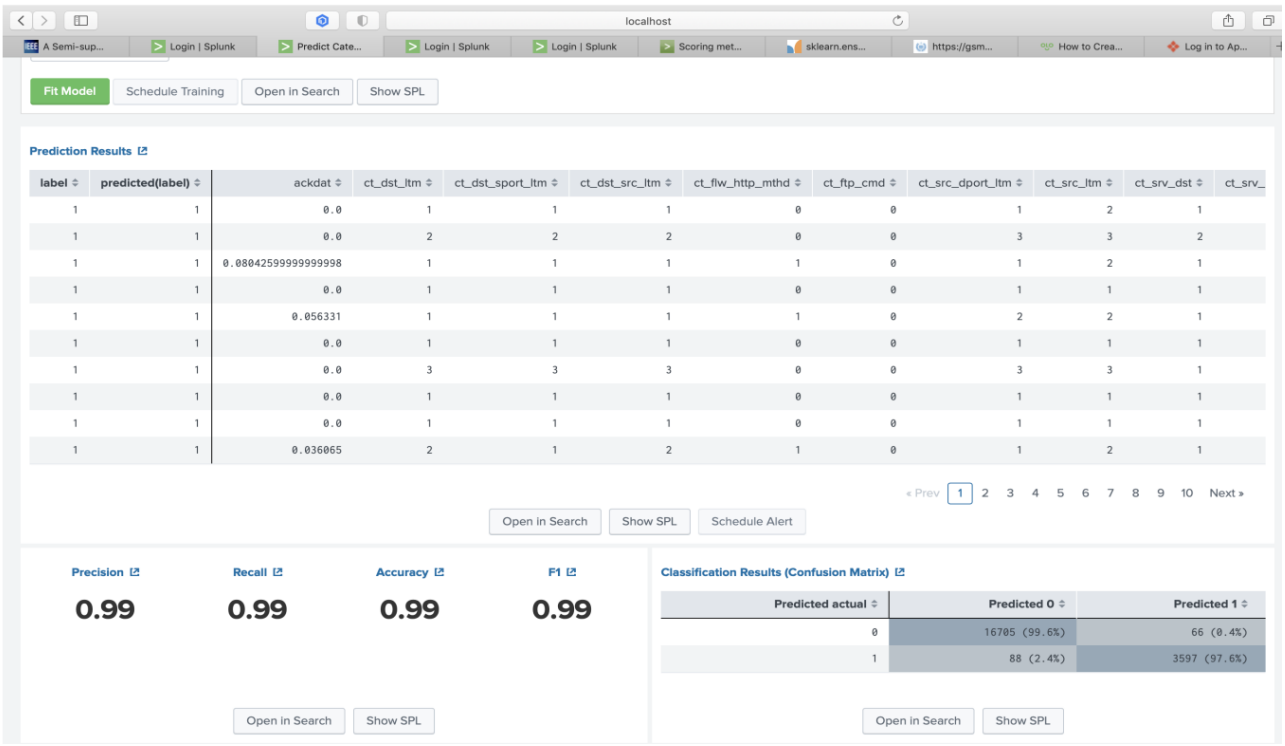


Рисунок 2.3 - Результати класифікації трафіку для детектування DoS-атак за допомогою алгоритму RF

Якщо *n_jobs*= -1, тоді використовуються всі ядра, наявні на машині. Таким чином, отримано значне прискорення під час побудови великої кількості дерев або коли побудова одного дерева потребує значної кількості часу у великих наборах даних.

На рисунках 2.4-2.6 представлені скріншоти з результатами тестування класифікатора нормальних та аномальних мережевих трафіків з бази даних UNSW-NB15B та з використанням алгоритмів DT, RF, SVM відповідно. У прикладах представлені дані для атак типу відмова в обслуговуванні (Denial of Service, DoS).

Як видно з рисунків 2.4-2.6, у нижній лівій частині представлені обчислені значення метрик ефективності алгоритмів у класифікації. Використання подання з двома розрядами після коми не дозволяє точно визначити, чи є відмінність у значеннях різних метрик. У правій нижній частині

скріншотів представлені матриці помилок кожного алгоритму класифікації. Використовуючи ці дані, можна повторно зробити обчислення за формулами (2.1)-(2.4) і уточнити значення всіх метрик, використовуються у машинному навчанні.

У таблиці 2.4 зведено всі експериментальні результати з поданням 4-х розрядів після коми, отримані після зміни параметрів алгоритмів.

Таблиця 2.4 - Порівняння результатів класифікації під час використання різних алгоритмів машинного навчання

Алгоритм машинного навчання	Accuracy	Precision	Recall	F1-score
Випадковий ліс (RF)	0,9925	0,9948	0,9961	0,9954
Дерево рішень (DT)	0,9885	0,9932	0,9927	0,9930
Машина опорних векторів (SVM)	0,6985	0,8370	0,6945	0,7329
Подвійний випадковий ліс (DRF)	0,9984	0,9987	0,9993	0,9990

На рисунку 2.4 приведено результати класифікації трафіку для детектування атак за допомогою алгоритму DT.

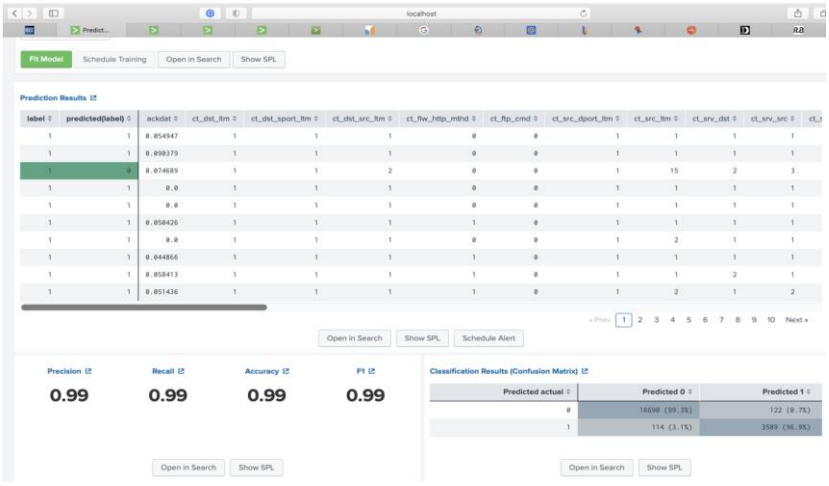


Рисунок 2.4 - Результати класифікації трафіку для детектування DoS-атак за допомогою алгоритму DT

Результати класифікації трафіку для детектування DoS-атак за допомогою алгоритму SVM приведені на рисунку 2.5.

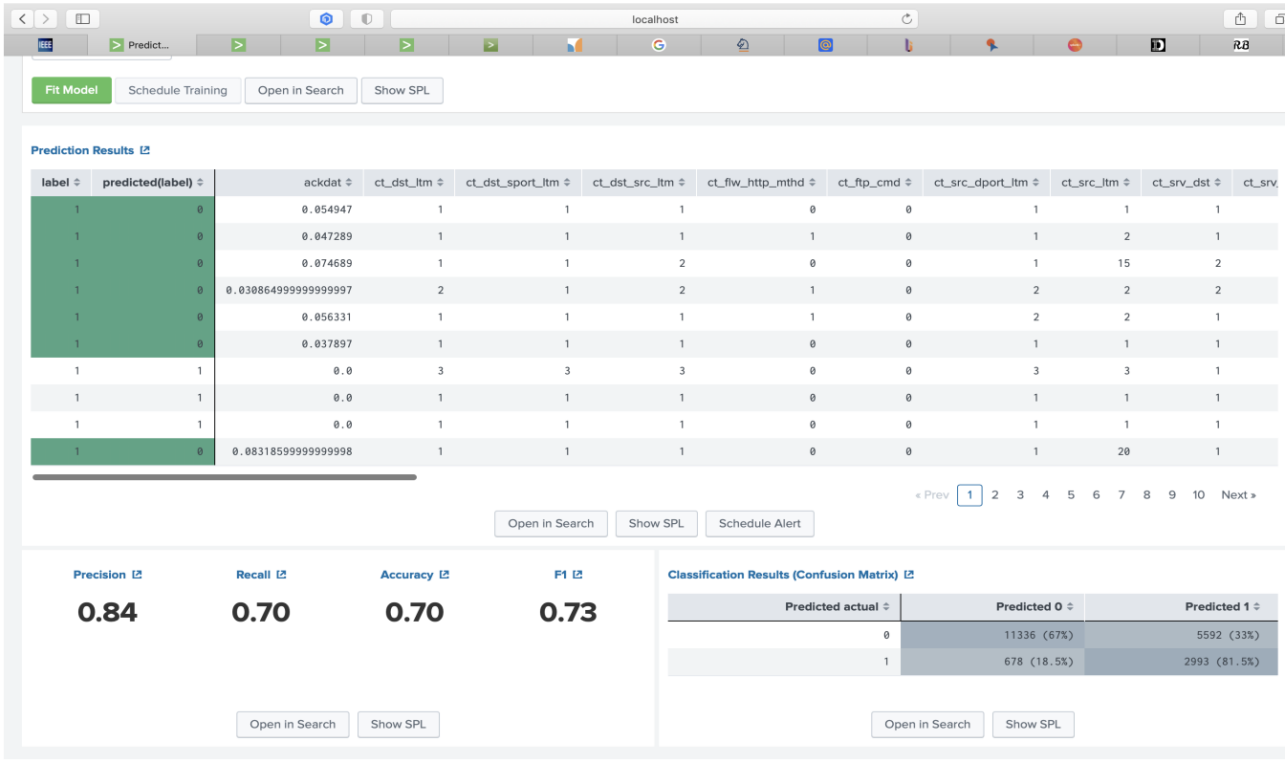


Рисунок 2.5 - Результати класифікації трафіку для детектування DoS-атак за допомогою алгоритму SVM

Проведемо оцінку точності класифікації атак за пропонованими алгоритмами опрацювання даних.

Таблиця 2.5 – Оцінки точності класифікації DoS-атак.

Алгоритм машинного навчання	Accuracy	Precision	Recall	F1-score
Випадковий ліс (RF)	0,9925	0,9948	0,9961	0,9954
Дерево рішень (DT)	0,9885	0,9932	0,9927	0,9930
Машина опорних векторів (SVM)	0,6985	0,8370	0,6945	0,7329
Подвійний випадковий ліс (DRF)	0,9984	0,9987	0,9993	0,9990

На малюнку 2.6 представлені оцінки точності класифікації DoS-атак з використанням розробленого програмного модуля SPL-DRF та бази даних UNSW-NB15B.

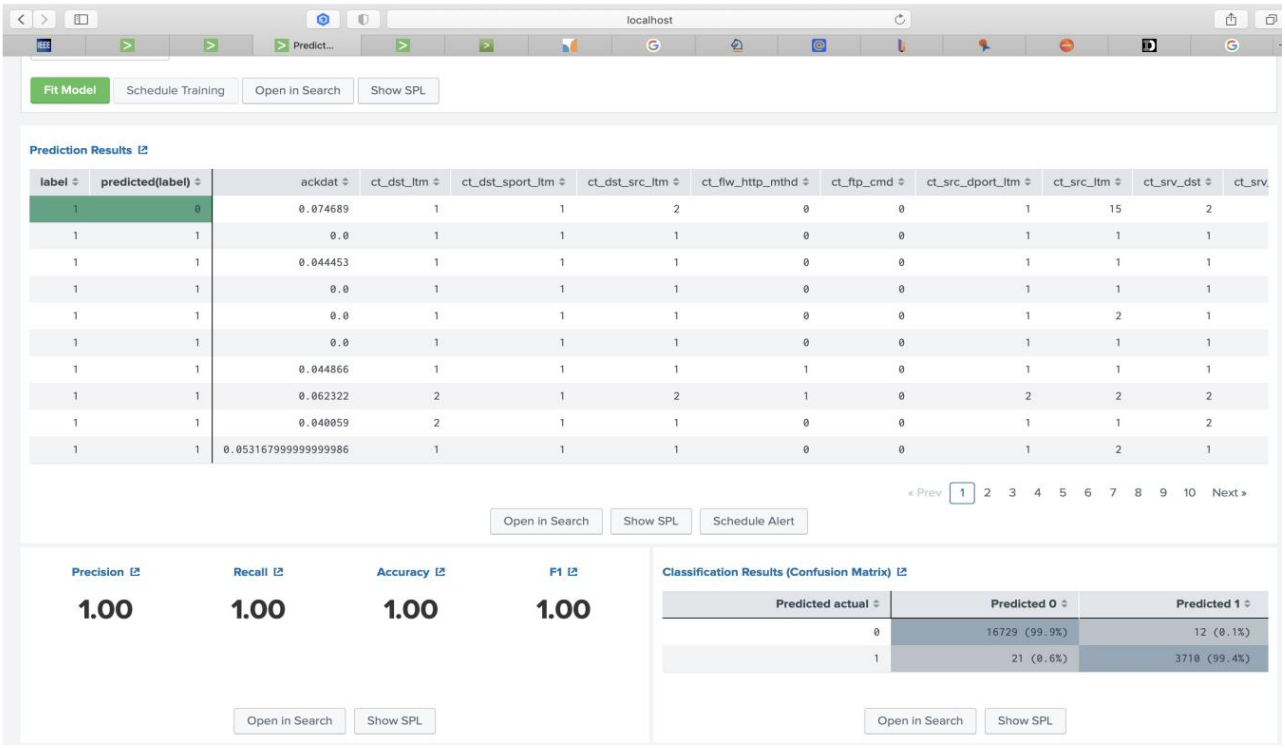


Рисунок 2.6 - Результати класифікації трафіку для детектування DoS-атак за допомогою алгоритму DRF

Експериментальні результати тестування навченого класифікатора на базі даних UNSW-NB15B показують, що найкращі результати з детектування Dos атак за точністю класифікації та параметром помилкового спрацьовування дає алгоритм «випадковий ліс» у порівнянні з іншими використаними класичними методами машинне навчання. Результати для методу RF також добре узгоджуються з оцінками точності класифікації Dos-атак 99,2% та 98,75%, отриманими авторами робіт [18] та [20] відповідно. Отримані оцінки точності класифікації DoS-атак із використанням реалізованого методу подвійного випадкового лісу (a double random forest, DRF) та бази даних UNSW-NB15B показують покращення порівняно з немодифікованим шляхом випадкового лісу.

Представлені результати досліджень із застосування сучасних методів машинного навчання спільно з технологіями збору машинних даних на платформі операційної аналітики Splunk Enterprise Security. Показано, що запропонований підхід для розробки систем детектування аномалій та класифікації мережевих атак є ефективним для завдань раннього розпізнавання та оперативного реагування на інциденти кібербезпеки.

Отримано оцінки продуктивності (у метриках завдань класифікації та розпізнавання) моделі детектування аномалій з використанням класичних алгоритмів машинного навчання (дерево рішень, метод опорних векторів, випадковий ліс). Експериментальні результати показують, що всі використані алгоритми машинного навчання можуть ефективно використовуватися для детектування нормальної та аномальної мережевих активностей, а також для класифікації мережевих атак.

Розроблено новий алгоритм побудови дерев рішень із використанням усіх даних навчальної вибірки кожному проміжному вузлі, включаючи кореневий вузол. Це дозволяє будувати більш розгалужені ансамблі, ніж у методі випадкового лісу, який будує окремі дерева, використовуючи початкове завантаження даних лише на кореневий вузол, при цьому на вхід подається тільки частина інформації, що є випадковою вибірку з навчальної бази. Розширено функціональні можливості інструменту Splunk Machine Learning Toolkit, додаткового програмного модуля (плагіна) на Bulletin of L.N.

Детектування та класифікація мережевих атак за допомогою Splunk Machine Learning Toolkit мовою Python для реалізації подвійного випадкового лісу Для забезпечення швидкодії обробки ансамблів дерев великого розміру використані можливості мови SPL для паралельного побудови дерев та паралельного обчислення прогнозів.

Експериментальні результати з детектування аномалій та класифікації мережевих атак показали, що модифікований метод випадкового лісу має найкращу точність виявлення атак типу «відмова в обслуговуванні в порівнянні з використаними класичними алгоритмами машинного навчання.

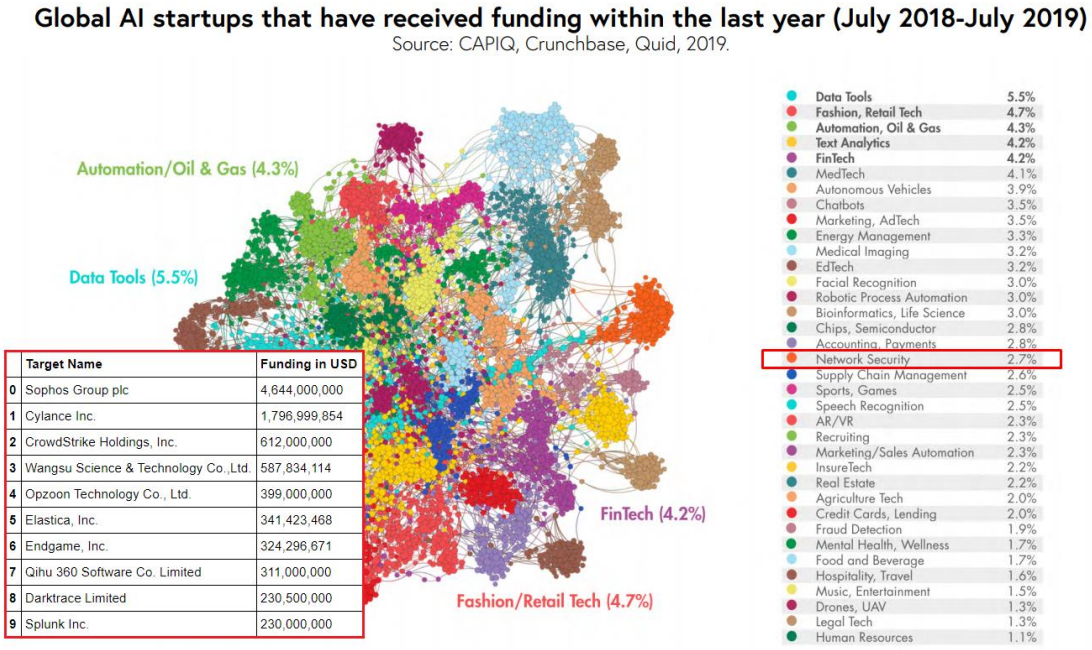
Планується провести експерименти з навченою моделлю класифікації мережевих атак на основі реальних даних аналізу мережевого трафіку, що збирається з усіх пристроїв у програмному комплексі для ОЦІБ.

3 СИСТЕМА ВИЯВЛЕННЯ ВТОРГНЕНЬ НА ОСНОВІ МАШИННОГО НАВЧАННЯ

3.1 Проектування системи виявлення вторгнень на основі ML

Так, сьогодні всі знають про машинне навчання. Розумна колонка та онлайн-кінотеатр вгадують ваш настрій та близько до ідеального пропонують у рекомендаціях наступний трек/фільм. Коли ви телефонуєте на «гарячу лінію» банку, важко зрозуміти, хто відповідає – робот чи жива людина. Безпілотні автомобілі на дорогах загального призначення.

Важко уявити розповсюдження технологій AI / ML у проектах та додатках інформаційної безпеки. В звіті зі Стенфорда «AI Index 2019 Report» , 220 сторінок про сучасний стан справ у галузі штучного інтелекту.



	Target Name	Funding in USD
0	Sophos Group plc	4,644,000,000
1	Cylance Inc.	1,796,999,854
2	CrowdStrike Holdings, Inc.	612,000,000
3	Wangsu Science & Technology Co.,Ltd.	587,834,114
4	Opzoon Technology Co., Ltd.	399,000,000
5	Elastica, Inc.	341,423,468
6	Endgame, Inc.	324,296,671
7	Qihu 360 Software Co. Limited	311,000,000
8	Darktrace Limited	230,500,000
9	Splunk Inc.	230,000,000

Рисунок 3.1 - Використання штучного інтелекту в проектах та галузях

Залучення інвестицій провідними світовими розробниками засобів захисту інформації із застосуванням технологій штучного інтелекту

Система виявлення атак, що розробляється, покликана не замінити, а лише доповнити сигнатурний аналізатор з метою підвищення загальної ефективності системи, особливо щодо раніше невідомих атак.

Послідовність кроків при проектуванні. Проектування системи виявлення вторгнень на основі ML буде складатись з наступних кроків:

1. Вибір набору даних навчання системи виявлення комп'ютерних атак.
2. Попередня обробка даних.
3. Семплювання проти дисбалансу класів.
4. Оцінка значущості та відбір ознак.
5. Скорочення ознакового простору.
6. Вибір моделі.
7. Налаштування та навчання моделі.
8. Тестування та апробація.

Приведемо схему навчання з учителем, як це показано на рисунку 3.2.

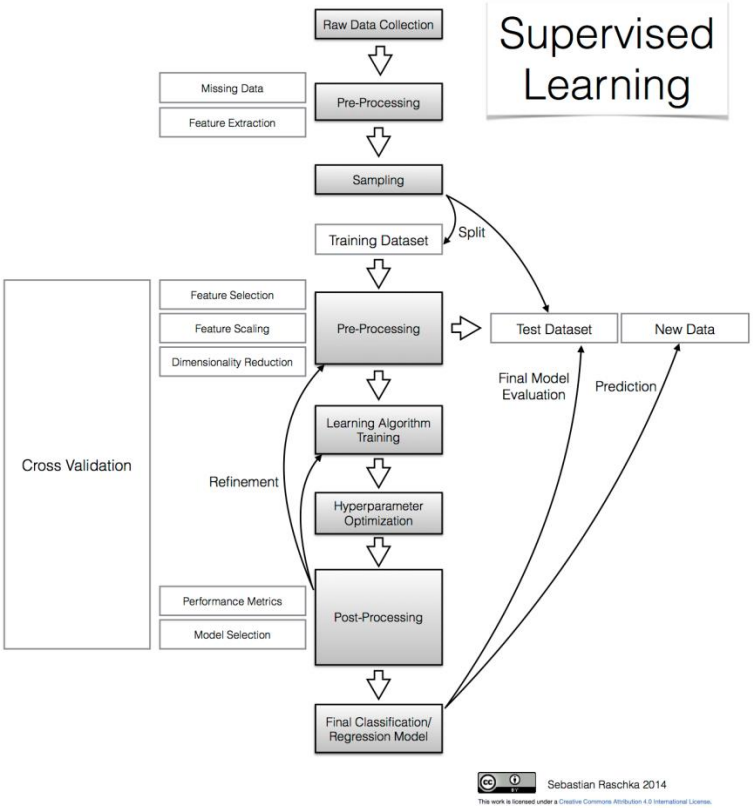


Рисунок 3.2 - Типова схема навчання з учителем

3.2 Вибір набору даних для навчання

Для навчання системи виявлення атак серед доступних публічних наборів даних (DARPA1998, KDD1999, ISCX2012, ADFA2013 та інших) обрано один із найбільш актуальних (на момент початку дослідження) – «Intrusion Detection Evaluation Dataset» CICIDS2017 . Розробник – Canadian Institute for Cybersecurity. Набір даних CICIDS2017 підготовлений за результатами аналізу мережевого трафіку в ізольованому середовищі, де моделювалися дії 25 легальних користувачів, а також шкідливі дії порушників.

Набір об'єднує понад 50 Гб «сирих» даних у форматі PCAP і включає 8 попередньо оброблених файлів у форматі CSV, що містять розмічені сесії з виділеними ознаками в різні дні спостереження. Короткий опис файлів та кількісний склад набору даних наведено в таблицях нижче (таблиця 3.1-3.2).

Таблиця 3.1 - Опис файлів набору даних CICIDS2017

№	Назва файлу	Різновиди атак
1	Monday-WorkingHours.pcap_ISCX.csv	Benign (обычный трафик)
2	Tuesday-WorkingHours.pcap_ISCX.csv	Benign, FTP-Patator, SSH-Patator
3	Wednesday-workingHours.pcap_ISCX.csv	Benign, DoS GoldenEye, DoS Hulk, DoS Slowhttptest, DoS slowloris, Heartbleed
4	Thursday-WorkingHours-Morning-WebAttacks.pcap_ ISCX.csv	Benign, Web Attack - Brute Force,Web Attack - Sql Injection, Web Attack - XSS
5	Thursday-WorkingHours-Afternoon-lnfilteration.pcap_ ISCX.csv	Benign, Infiltration
6	Friday-WorkingHours-Morning.pcap_ISCX.csv	Benign, Bot
7	Friday-WorkingHours-Afternoon-PortScan.pcap_ISCX.csv	Benign, PortScan
8	Friday-WorkingHours-Afternoon-DDos.pcap_ISCX.csv	Benign, DDoS

Приведемо кількісні показники для обраного набору даних в таблиці 3.2.

Таблиця 3.2 - Кількісний склад набору даних CICIDS2017

№	Тип запису	Кількість записів
1	BENIGN	2359087
2	DoS Hulk	231072
3	PortScan	158930
4	DDoS	41835
5	DoS GoldenEye	10293
6	FTP-Patator	7938
7	SSH-Patator	5897
8	DoS slowloris	5796
9	DoS Slowhttptest	5499
10	Bot	1966
11	Infiltration	36
12	Heartbleed	11
13	Web Attack - Brute Force	1507
14	Web Attack - XSS	652
15	Web Attack - SQL Injection	21

Приклад одного запису з набору даних CICIDS2017 містить декілька записів. Кожен запис відповідає мережевої сесії та характеризується 85 ознаками, такими як на рисунку 3.3.

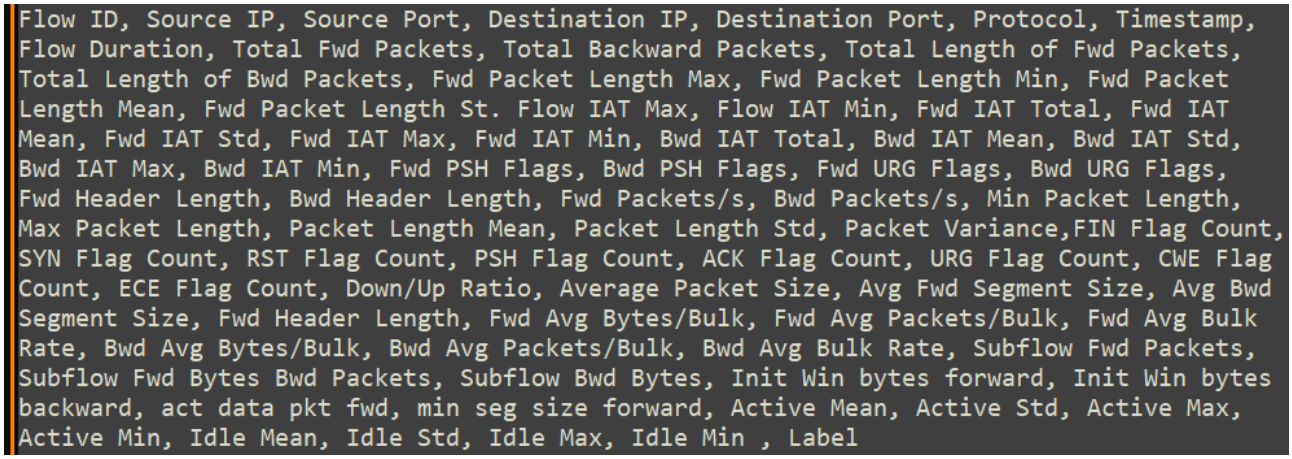


Рисунок 3.3 - Приклад одного запису з набору даних CICIDS2017

Наведемо приклад одного з записів, які будемо аналізувати.

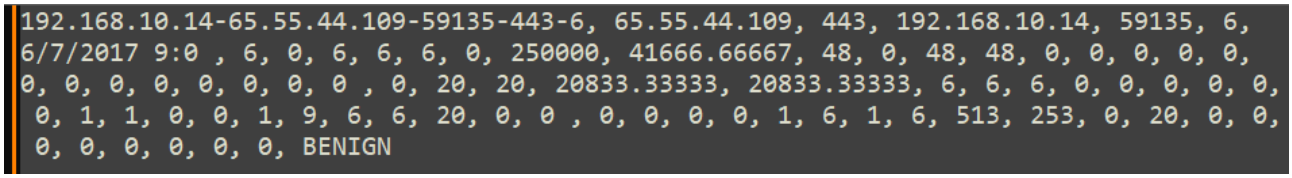


Рисунок 3.4 – Заповнений запис з таблиці вибірки

Хороші дані – обов'язкова умова побудови хорошого класифікатора.

В оглядах набору даних CICIDS2017 (Intrusion2017 , Panigrahi2018 , Sharafaldin2018) відзначалися проблеми дисбалансу класів, складної файлової структури, пропуску значень. Ці зауваження будемо сприймати як некритичні.

В процесі аналізу виникли запитання щодо акуратності розмітки набору даних CICIDS2017, тому розберемо весь pipeline – починаючи від сніфера та передобробки мережевих сесій, закінчуючи моделлю машинного навчання та тестуванням у реальній мережі.

3.3 Попередня обробка даних

Відправною точкою для проведення власних експериментів з датасетом CICIDS2017 послужило дослідження Kahraman Kostas "Anomaly Detection in Networks Using Machine Learning" . При спробі відтворення цього дослідження

було виявлено розбіжності у результатах, та були виявлені помилки у коді автора.

Для скорочення часу обчислень у навчальній вибірці залишили єдиний клас атак – веб-атаки (Brute Force, XSS, SQL Injection). Для цього було підготовлено вибірку «WebAttacks» на основі обробки файлу Thursday-WorkingHours-Morning-WebAttacks.pcap_ISCX.csv з набору даних CICIDS2017. Набір WebAttacks включає 458968 записів, з яких 2180 відносяться до веб-атак, решта – до нормального трафіку.

Таке рішення спрощує завдання та знижує якість підсумкових висновків: багатокласова класифікація звелася до бінарної класифікації, значно зменшився розмір навчальної вибірки.

Етапи попередньої обробки набору даних CICIDS2017 та підготовки підвиборки WebAttacks містять наступні кроки:

1. Усунення ознаки "Fwd Header Length.1" (ознаки "Fwd Header Length" та "Fwd Header Length.1" є ідентичними).

2. Видалення записів з null значеннями ідентифікатора сесії Flow ID (з 458968 записів після видалення залишилося 170366 записів).

3. Заміна нечислових значень ознак Flow Bytes/s, Flow Packets/s значеннями -1.

4. Заміна невизначених значень (NaN) та нескінченних значень значеннями -1.

5. Приведення рядкових значень ознак Flow ID, Source IP, Destination IP, Timestamp до числових значень методом label encoding.

6. Кодування відповідей у навчальній вибірці відповідно до правила: 0 – «ні атаки», 1 – «є атака».

Зберігаємо Jupyter блокноти на Github у репозиторії ml-cybersecurity , а посилання даємо на Google Colaboratory – тоді код готовий до запуску прямо у браузері. Вихідний код у Google Colaboratory приведений в додатку А.

3.3 Пошук та виправлення помилок у датасеті.

На етапі попередньої обробки даних було виявлено похибки в ознаковому просторі (як мінімум, підозрілою видалася наявність двох різних ознак з попарно однаковими значеннями). І вирішено виконати перевірку: взяти «сирий» трафік CICIDS2017, виділити в ньому мережеві сесії та сформувати свій датасет. Отриманий датасет мав збігтися з датасетом CICIDS2017.

Обробляємо pcap файл із записаним трафіком власним сніффером, виділяємо сесії та ознаки, порівнюємо з датасетом Thursday-WorkingHours-Morning-WebAttacks.pcap_ISCX.csv, намагаємося знайти та виправити розбіжності. Процес пошуку розбіжностей приведений на рисунку 3.5.

	Flow ID	Source IP	Source Pc	Destination	Destination Pc	Protoc	Timestamp	Flow Duratic	Total Fwd Packe	Total Backward Packe	Total Length of Fwd Packe	Total Length of Bwd Packe
2	192.168.10.3-192.168.10.50-389-33898-6	192.168.10.50	33898	192.168.10.3	389	6	06.07.2017 8:59	113095465	48	24	9668	10012
3	192.168.10.3-192.168.10.50-389-33904-6	192.168.10.50	33904	192.168.10.3	389	6	06.07.2017 8:59	113473706	68	40	11364	12718
5	192.168.10.14-65.55.44.109-59135-443-6	192.168.10.14	59135	65.55.44.109	443	6	06.07.2017 8:59	60261928	9	7	2330	4221
8	192.168.10.14-65.55.44.109-59135-443-6	65.55.44.109	443	192.168.10.14	59135	6	06.07.2017 9:00	48	1	1	6	6
25	192.168.10.3-192.168.10.19-389-32791-6	192.168.10.19	32791	192.168.10.3	389	6	06.07.2017 9:00	487	11	4	172	326
26	192.168.10.3-192.168.10.19-88-41567-6	192.168.10.19	41567	192.168.10.3	88	6	06.07.2017 9:00	1206	10	7	3150	3152
27	192.168.10.3-192.168.10.19-389-32792-6	192.168.10.19	32792	192.168.10.3	389	6	06.07.2017 9:00	27779	17	11	3450	6654
28	192.168.10.3-192.168.10.19-88-41569-6	192.168.10.19	41569	192.168.10.3	88	6	06.07.2017 9:00	1133	9	6	3150	3152
29	192.168.10.3-192.168.10.19-389-32794-6	192.168.10.19	32794	192.168.10.3	389	6	06.07.2017 9:00	118034439	106	61	17896	27110
66	192.168.10.3-192.168.10.19-88-41567-6	192.168.10.19	41567	192.168.10.3	88	6	06.07.2017 9:00	62	1	3	0	12
67	192.168.10.3-192.168.10.19-88-41569-6	192.168.10.19	41569	192.168.10.3	88	6	06.07.2017 9:00	75	1	4	0	12
71	192.168.10.3-192.168.10.19-389-32792-6	192.168.10.19	32792	192.168.10.3	389	6	06.07.2017 9:00	75	1	4	0	24
74	192.168.10.3-192.168.10.19-389-32791-6	192.168.10.19	32791	192.168.10.3	389	6	06.07.2017 9:00	79	1	4	0	12
76	192.168.10.3-192.168.10.19-389-32798-6	192.168.10.19	32798	192.168.10.3	389	6	06.07.2017 9:00	396	10	4	165	326
77	192.168.10.3-192.168.10.19-389-32798-6	192.168.10.19	32798	192.168.10.3	389	6	06.07.2017 9:00	2	2	0	7	0
78	192.168.10.3-192.168.10.19-88-41574-6	192.168.10.19	41574	192.168.10.3	88	6	06.07.2017 9:00	1086	9	6	3150	3152
79	192.168.10.3-192.168.10.19-3268-44766-6	192.168.10.19	44766	192.168.10.3	3268	6	06.07.2017 9:00	116446503	72	47	13380	15294
96	192.168.10.3-192.168.10.19-389-32798-6	192.168.10.3	389	192.168.10.19	32798	6	06.07.2017 9:00	50	4	0	12	0

Рисунок 3.5 - Подробиці пошуку помилок

Вибрана сесія для аналізу та виконано відбір пакетів. Проведемо аналіз сесії: Flow ID = "192.168.10.14-65.55.44.109-59135-443-6", Source IP = "65.55.44.109". У канадських дослідників два останні пакети виділено в окрему сесію. Уважно переглянемо вихідні дані із сніффера і знайдемо підтвердження у методі addPacket .

Що потрібно врахувати в нашому сніфері:

1. При появі пакета з прапором FIN у напрямку forward для відтворення експерименту потрібно завершити поточну сесію та створити нову. Щоб два останні пакети FIN ACK і ACK потрапили до другої сесії разом, умову переривання сесії потрібно доповнити: кількість пакетів у сесії має бути більшою за 1.

2. Завершення сесії з таймера, 120 секунд (хоча в readme сказано про 600 секунд).

Для тієї ж сесії у прямому напрямку зафіксовано один пакет ("Total Fwd Packets" = 1), при цьому загальна довжина переданих пакетів у прямому напрямку "Total Length of Fwd Packets" = 6. За даними Wireshark, довжина пакета = 0. Виявлена різниця у 6 байт. Спускаємося від TCP до Ethernet і виявляємо "невраховані" 6 байт у вигляді доповнення (padding) кадру Ethernet. Спірна ситуація, чи потрібно включати ці 6 байт у довжину TCP пакета. Проведена перевірка зображена на рисунку 3.6.

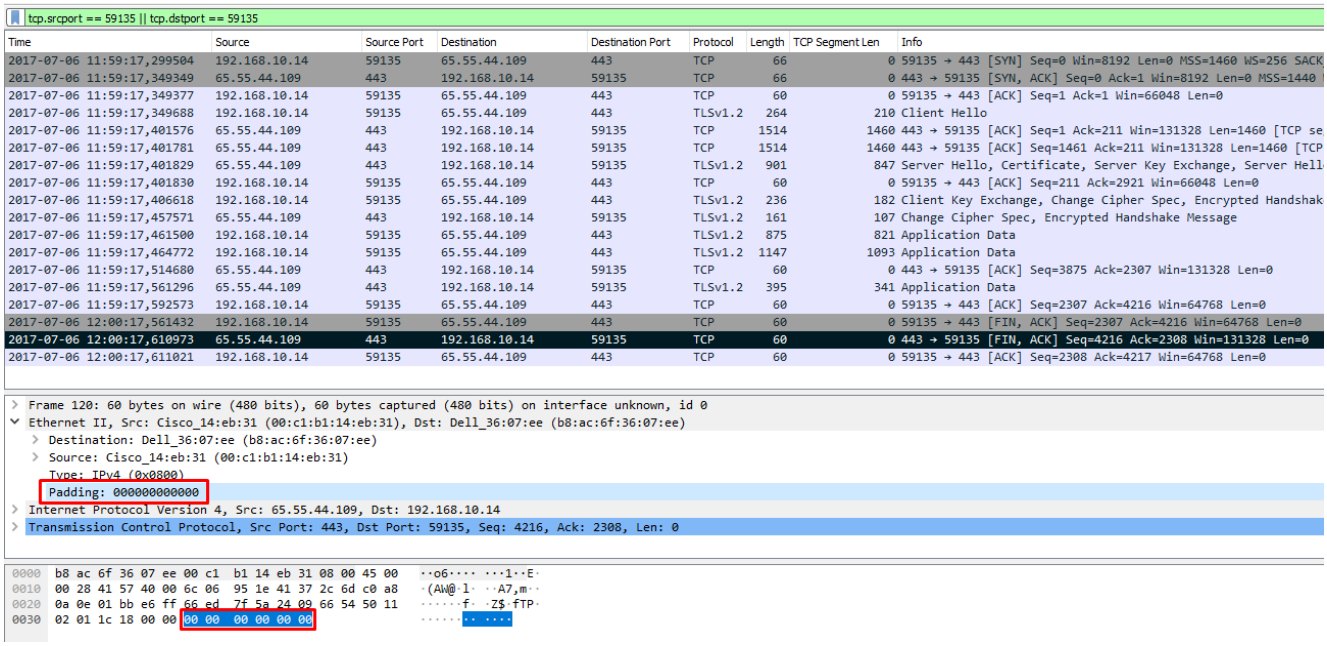


Рисунок 3.6 - Перевірка довжини пакету у Wireshark

Перевіряємо решту ознак для цієї сесії, збігаються всі значення, крім Average Packet Size = 9. Як при двох пакетах по 6 байт отримати значення 9, не ясно. У цьому Packet Length Mean = 6, збігається.

Починаємо перевіряти інші сесії, і виявляється, що часто трапляються невеликі розбіжності в ознаках: Packet Length Mean, Packet Length Std, Packet Length Variance, Average Packet Size, Average Fwd Segment Size, Average Bwd Segment Size. Розбір перехоплених пакетів (відновлення вихідних доданків за значеннями середніх) показує, що при спрацьовуванні таймууту сесії довжина пакета, що відкидається, помилково враховується в статистиці(рисунок 3.7).

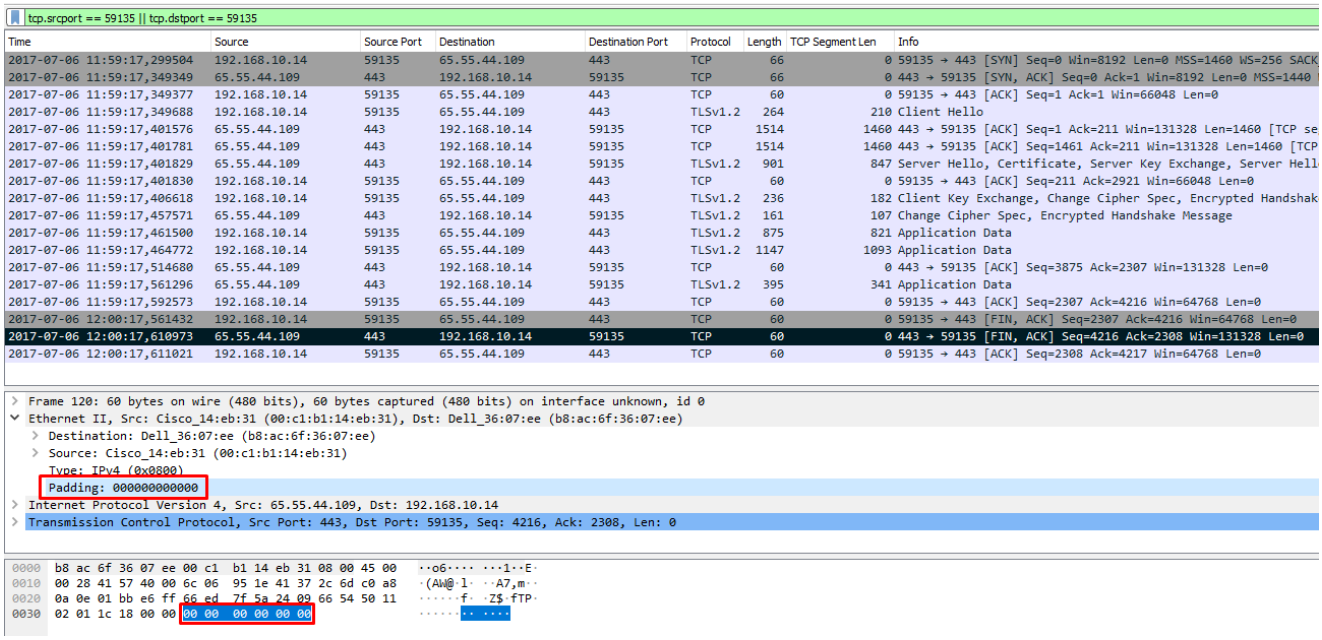


Рисунок 3.7 – Розбіп перехоплених пакетів у Wireshark

Після усунення більшості розбіжностей у тестованому датасеті та датасеті CICIDS2017 можна рухатися далі: тепер зрозуміло, як обчислюються значення ознак із записаного трафіку.

3.4 Семплювання проти дисбалансу класів та оцінка значущості та відбір ознак

Підготовлена підвибірка «WebAttacks» є збалансованою: за загальної кількості записів 170366 клас «не атаки» об'єднує 168186 екземплярів, клас «є атака» – 2180 екземплярів. Для усунення дисбалансу класів підійде метод випадкового семплювання (субдискретизація, undersampling), що полягає у видаленні випадково вибраних екземплярів класу немає атаки. Цільове співвідношення кількості екземплярів класів «немає атаки» і «є атака» обрано 70%/30%.

Попередньо з ознакового простору були виключені ознаки "Flow ID", "Source IP", "Source Port", "Destination IP", "Destination Port", "Protocol", "Timestamp" у припущенні, що ознаки "форми" (відповідні статистикам мережного трафіку) є більш значущими для загального випадку. Крім того,

ознаки адресації, що виключаються, можуть бути відносно легко підроблені зловмисником і не повинні враховуватися при навчанні.

Аналіз значущості ознак я виконав за допомогою вбудованого механізму методу sklearn.ensemble.RandomForestClassifier (атрибут feature_importances_).

Перші результати оцінки значущості показали сильний взаємозв'язок ознак Init_Win_bytes_backward, Init_Win_bytes_forward з міткою класу в навчальній вибірці, що може свідчити про допущені похибки при формуванні набору даних. Зазначені ознаки було виключено з ознакового простору.

Підсумкові результати аналізу значимості представлені малюнку нижче, список обмежений першими двадцятьма ознаками(рисунок 3.8).

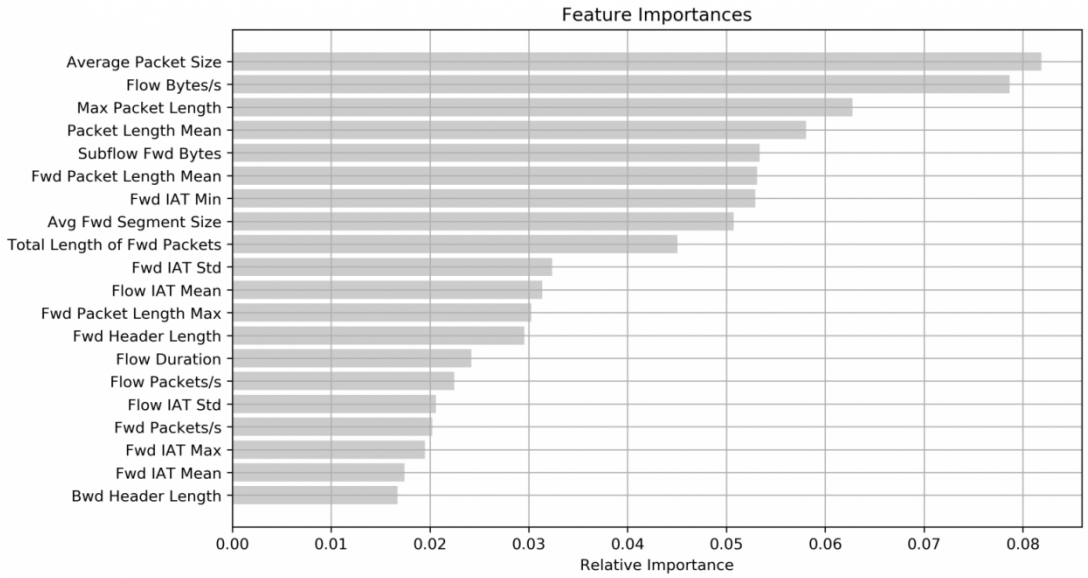


Рисунок 3.8 – Результати оцінки значущості ознак

Додаткові експерименти показали, що можна побудувати досить точний класифікатор, спираючись на одну єдину ознаку – або Init_Win_bytes_backward, або Init_Win_bytes_forward.

3.5 Скорочення ознакового простору

На наступному малюнку представлена кореляційна матриця з лінійними коефіцієнтами кореляції (коефіцієнтами кореляції Пірсона), розрахованими

всім пар двадцяти найбільш значущих ознак. Насиченість кольору заливки пропорційна значенню коефіцієнта кореляції(рисунок 3.9).

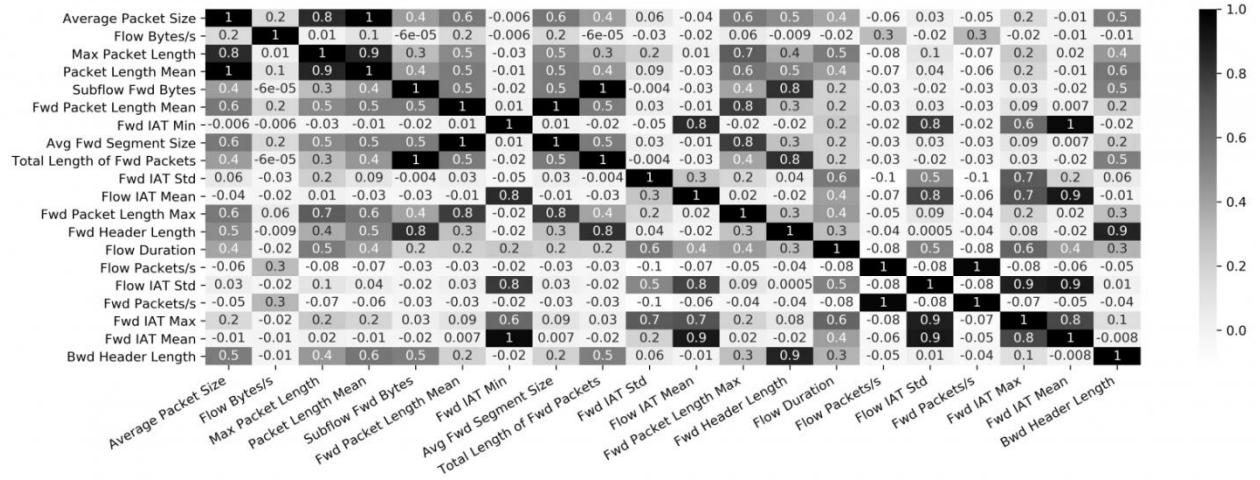


Рисунок 3.9 – Результати кореляційного аналізу двадцяти найбільш значимих ознак

Кореляційний аналіз показав сильну залежність між парами ознак:

1. "Average Packet Size" та "Packet Length Mean".
2. Subflow Fwd Bytes і Total Length of Fwd Packets.
3. "Fwd Packet Length Mean" та "Avg Fwd Segment Size".
4. "Flow Duration" та "Fwd IAT Total".
5. "Flow Packets/s" та "Fwd Packets/s".
6. "Flow IAT Max" та "Fwd IAT Max".

За результатами кореляційного аналізу з ознакового простору були виключені такі ознаки: Packet Length Mean, Subflow Fwd Bytes, Avg Fwd Segment Size, Fwd IAT Total, Fwd Packets/s, Fwd IAT Max.

Після виключення ознак із найменшою значимістю ознаковий простір було скорочено до об'єднання 10 ознак:

1. Average Packet Size, середня довжина поля даних пакета TCP/IP (далі – довжина пакета).
2. "Flow Bytes/s", швидкість потоку даних.
3. Max Packet Length, максимальна довжина пакета.

4. "Fwd Packet Length Mean", середня довжина переданих у прямому напрямку пакетів.

5. "Fwd IAT Min", мінімальне значення міжпакетного інтервалу (IAT, inter-arrival time) у прямому напрямку.

6. "Total Length of Fwd Packets", сумарна довжина переданих у прямому напрямку пакетів.

7. Fwd IAT Std, середньоквадратичне відхилення значення міжпакетного інтервалу в прямому напрямку пакетів.

8. "Flow IAT Mean", середнє значення міжпакетного інтервалу.

9. "Fwd Packet Length Max", максимальна довжина переданого у прямому напрямку пакета.

10. "Fwd Header Length", сумарна довжина заголовків переданих у прямому напрямку пакетів.

3.6 Вибір та налаштування моделі

На етапі вибору моделі я взяв 10 найпоширеніших моделей машинного навчання та оцінив їхню якість на підбірці WebAttacks.

Список із 10 моделей

Для порівняння було обрано наступні моделі (алгоритми) машинного навчання (у дужках вказується скорочене позначення та відповідна реалізація моделі зі складу пакету scikit-learn):

1. Метод k найближчих сусідів (KNN, sklearn.neighbors.KNeighborsClassifier).

2. Метод опорних векторів (SVM, sklearn.svm.SVC).

3. Дерево рішень (CART, алгоритм навчання CART, sklearn.tree.DecisionTreeClassifier).

4. Випадковий ліс (RF, sklearn.ensemble.RandomForestClassifier).

5. Модель адаптивного бустингу над вирішальним деревом (AdaBoost, sklearn.ensemble.AdaBoostClassifier).

6. Логістична регресія (LR, sklearn.linear_model.LogisticRegression).

7. Байєсовський класифікатор (NB, sklearn.naive_bayes.GaussianNB).

8. Лінійний дискримінантний аналіз (LDA, sklearn.discriminant_analysis.LinearDiscriminantAnalysis).

9. Квадратичний дискримінантний аналіз (QDA, sklearn.discriminant_analysis.QuadraticDiscriminantAnalysis).

10. Багатошаровий персептрон (MLP, sklearn.neural_network.MLPClassifier).

Якість відповідей класифікаторів (моделей) порівнювалося з використанням наступних метрик:

- частка правильних відповідей (accuracy);
- точність (precision, наскільки можна довіряти класифікатору);
- повнота (recall, скільки об'єктів класу «є атака» визначає класифікатор);
- F1-міра (F1-measure, гармонійне середнє між точністю та повнотою).

Оцінка якості класифікаторів проводилася на збалансованому та передопрацьованому підборі веб-атак WebAttacks набору даних CICIDS2017 (співвідношення нормального та аномального трафіку 70% / 30%, 20 найбільш значущих ознак). У таблиці нижче наведено отримані значення метрик якості, усереднені за результатами 5 ітерацій крос-валідації.

Таблиця 3.3 - Результати оцінки якості десяти класифікаторів.

Модель (алгоритм)	Acuracy	Precision	Recall	F1	Час виконання, с
KNN	0,971	0,942	0,961	0,969	4,57
SVM	0,705	0,669	0,036	0,602	176,04
CART	0,975	0,973	0,946	0,969	1,53
RF	0,971	0,978	0,943	0,970	1,14
AdaBoost	0,978	0,962	0,965	0,973	23,40
LR	0,955	0,939	0,914	0,963	15,80
Naive Bayes	0,722	0,520	0,956	0,754	0,47
LDA	0,939	0,921	0,872	0,941	2,23
QDA	0,872	0,978	0,597	0,949	1,28
MLP	0,904	0,921	0,912	0,776	93,83

Найкращі результати очікувано продемонстрували моделі (алгоритми) KNN, CART, RF, AdaBoost, LR. З огляду на мінімальний час виконання застосування моделі «випадковий ліс» (RF) для вирішення поставленого завдання є обґрунтованим вибором.

Отже, за основу взято модель на кшталт «випадковий ліс», реалізація в scikit-learn – RandomForestClassifier .

Серед настроюваних гіперпараметрів моделі були обрані такі: кількість дерев у лісі (n_estimators), мінімальна кількість об'єктів в одному аркуші дерева (min_samples_leaf), максимальна глибина дерева (max_depth), максимальна кількість ознак для одного дерева (max_features).

Ступінь квазіоптимальності параметрів моделі оцінювався значенням F1-заходи. Проведений експертний аналіз доповнив результатами вбудованого методу оптимізації параметрів GridSearchCV бібліотеки scikit-learn, підсумкові значення параметрів моделі «випадковий ліс» вийшли наступні:

```
RandomForestClassifier(
    bootstrap=True, class_weight=None, criterion='gini',
    max_depth=17, max_features=10, max_leaf_nodes=None,
    min_impurity_decrease=0.0, min_impurity_split=None,
    min_samples_leaf=3, min_samples_split=2,
    min_weight_fraction_leaf=0.0, n_estimators=50,
    n_jobs=None, oob_score=False, random_state=1, verbose=0,
    warm_start=False)
```

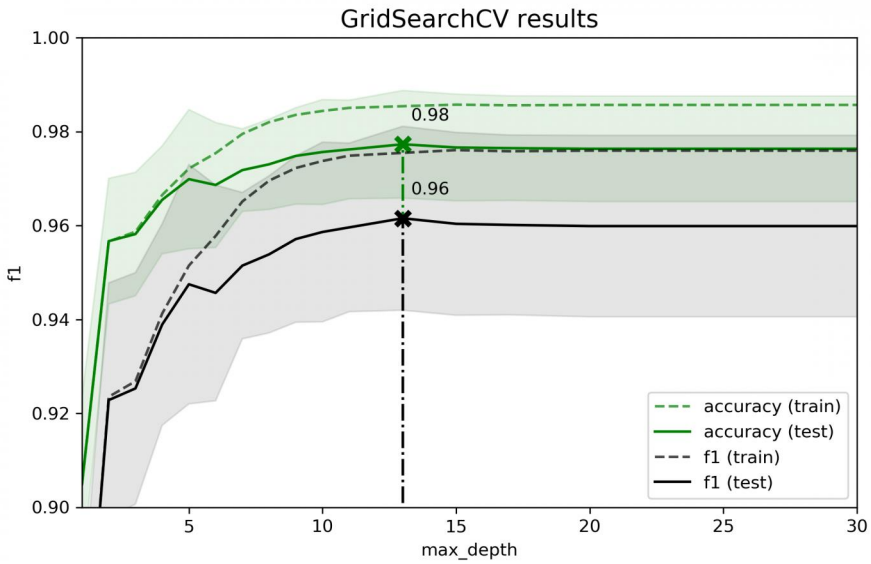


Рисунок 3.10 - Приклад налаштування моделі RandomForestClassifier

Приклад результатів підбору одного гіперпараметра (max_depth) при фіксованих значеннях інших гіперпараметрів (n_estimators, min_samples_leaf, max_features) представлений на малюнку нижче у вигляді залежності метрики якості (F1-заходи) від значення параметра (max_depth).

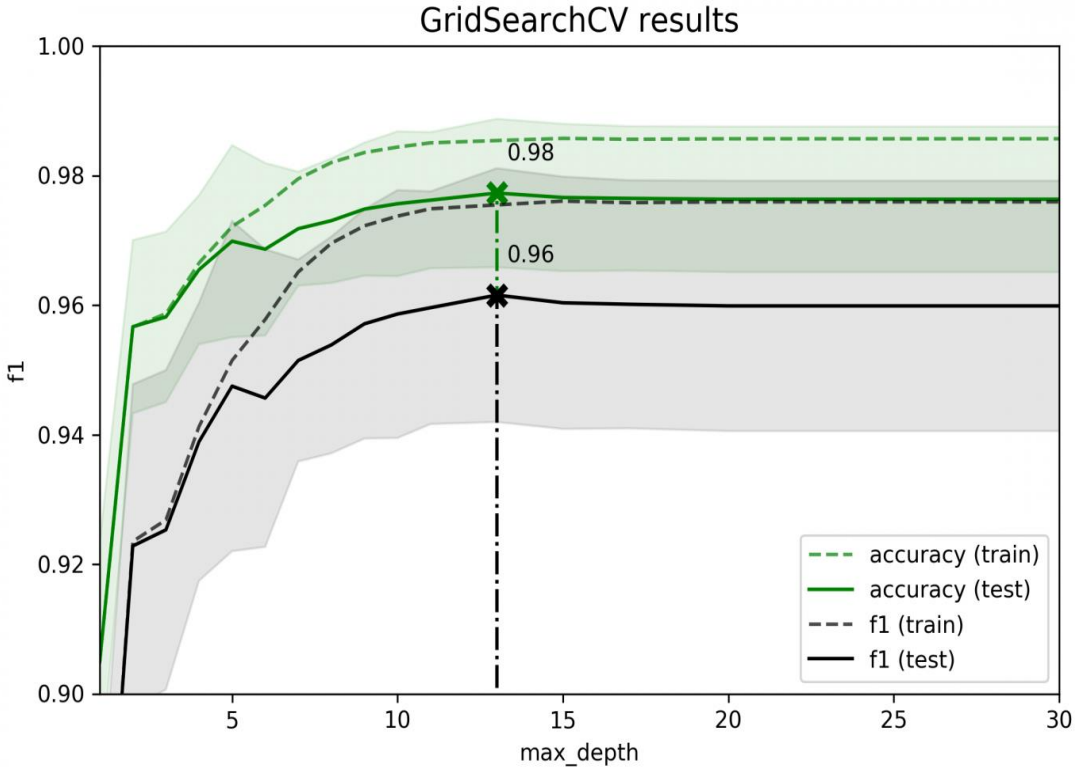


Рисунок 3.11 - Залежність F1-заходи моделі від параметра max_depth

3.7 Тестування та апробація

Налаштована та навчена модель RandomForestClassifier на тестовій вибірці дозволила отримати оцінку повноти (recall) 0.961 та F1-заходи 0.971 (запуск №1 у протоколі експерименту, див. таблицю нижче). Досягнутий результат свідчить про можливість підвищення точності моделі за рахунок квазіоптимального підбору гіперпараметрів (результати дослідження Kahraman Kostas – recall 0.94 та F1-мера 0.94, результати авторів CICIDS2017 – recall 0.97 та F1-мера 0.97).

Для апробації моделі на реальній мережній інфраструктурі розроблено мережевий аналізатор – сніфер (C#). Аналізатор дозволяє перехопити мережний трафік і з використанням алгоритмів реконструкції TCP сесій вільно розповсюджуваних програмних продуктів Wireshark і TCP Session Reconstruction Tool виділити окремі сесії. Для кожної збереженої сесії сніффер на основі алгоритму CICFlowMeter виділяє ознаки та таким чином формує набір даних.

Як веб-додаток, що атакується, використовувалася розроблена консоль адміністратора безпеки (PHP) з єдиним включеним модулем авторизації, що функціонує під керуванням веб-сервера Apache. Схема стенду тестування приведена на рисунку 3.12.

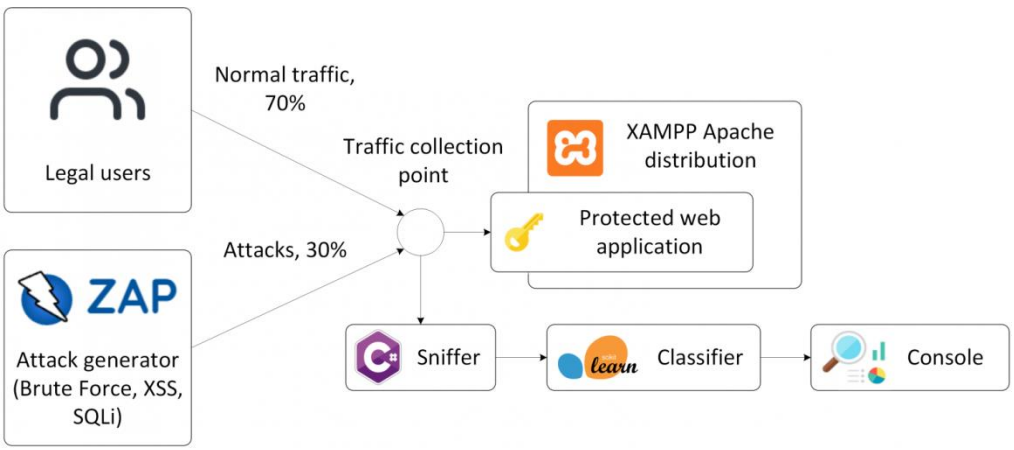


Рисунок 3.12 - Схема стенду тестування

Нормальний трафік відповідав запитам легальних користувачів на підключення до консолі адміністратора та авторизацію. Шкідливий (аномальний) трафік моделювався програмним засобом OWASP ZAP і включав три типи атак: Brute Force, XSS, SQL Injection. Співвідношення нормального та аномального трафіку в реальному тестовому наборі даних склало 70%/30% (Рисунок 3.13).

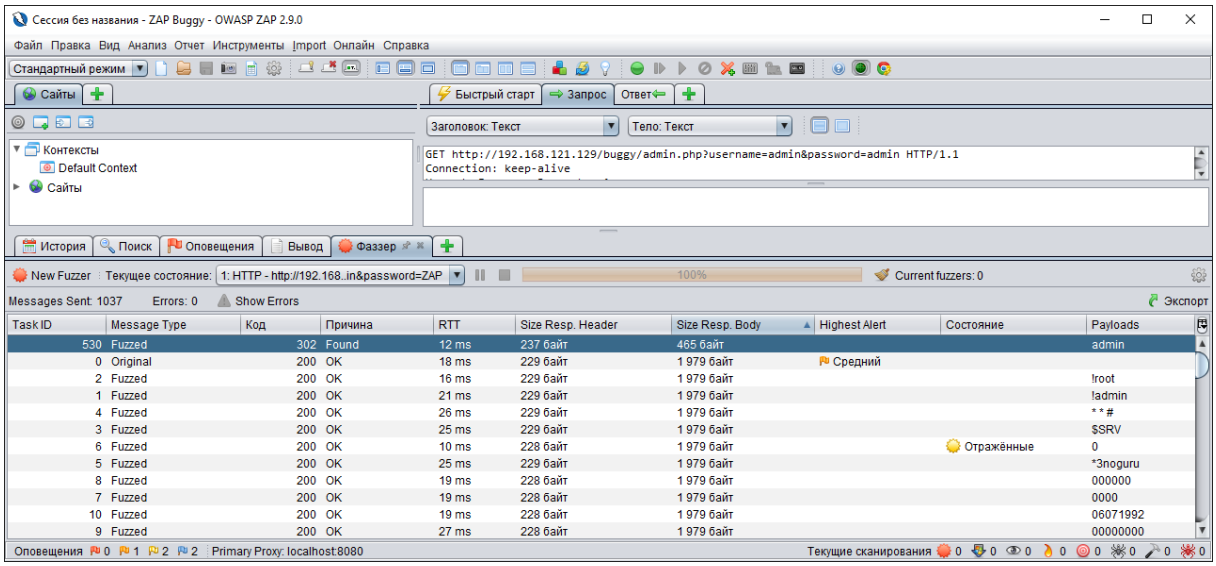


Рисунок 3.13 - Проведені експерименти на сформованому наборі даних

Проведені експерименти на сформованому наборі даних (запуски №2, №3 у протоколі експерименту) показали неможливість застосування моделі, навченої на наборі даних CICIDS2017, з наступних причин:

1. Аналіз навчальної вибірки показує, що характер комп'ютерних атак, що моделюються, у дослідженні авторів набору даних CICIDS2017 відрізняється від реального. Так, атаки типу Brute Force є в сесіях з максимальними швидкостями до 10 Кбіт/с, що не відповідає випадкам застосування автоматизованих засобів перебору паролів.

2. Серед десяти ознак з найбільшою значимістю чотири ознаки - Flow Bytes/s (швидкість потоку даних), Fwd IAT Min (мінімальне значення міжпакетного інтервалу в прямому напрямку), Flow IAT Std (середньоквадратичне відхилення значення міжпакетного інтервалу), Flow IAT Mean» (середнє значення міжпакетного інтервалу) безпосередньо залежать від

фізичної структури мережі, в якій проводиться збір мережевого трафіку, а також налаштувань мережевого обладнання. У навчальному наборі даних сесії з ознаками веб-атак записані з низькими значеннями швидкості потоку та високими значеннями міжпакетних інтервалів, що не відповідає характеристикам реальної інфраструктури мережі (мережа Ethernet 100 Мбіт/с).

Хороший датасет повинен відповідати певним вимогам. Автори CICIDS2017 мають роботу , в якій перераховано 11 таких вимог. Головне з цього: забезпечити різноманітність мережного обладнання, комп'ютерів та операційних систем у тестовій інфраструктурі, різноманітність потоків мережевого трафіку за різними напрямками, різноманітність протоколів та типів атак, розмітити дані для атак та для чистого трафіку. Код побудови моделі приведено в додатку А.

Поставимо ще одне завдання – уточнити можливість побудови евристичного аналізатора і грубо оцінити його точність. Не претендуючи на збір надякісного датасету, оскільки це завдання цілих інституцій.

План збору датасету:

1 етап. Запис pcap файлів, очищення. При збиранні «брудного» трафіку змінюємо параметри фазера і ставимо паузи під час фазингу, щоб розірвати сесії та збільшити їх кількість у датасеті. При збиранні «чистого» трафіку моделюємо різні дії користувача.

2 етап. Подача pcap файлів на вхід сніфера та виділення ознак. Об'єднання всіх розмічених записів в один датасет.

Запуск №2. Модель була навчена на вибірці WebAttacks набору даних CICIDS2017 (трафік збирався в одній мережі). Після цього я протестував модель на реальному трафіку в іншій мережі, що відрізняється швидкістю та іншими характеристиками першої. І отримано незадовільну якість – значення F1-заходи 0.064.

Оцінка обчислювальної складності проводилася непрямим способом: розроблений серед Jupyter Notebook макет системи виявлення веб-атак запускався на персональному комп'ютері (процесор Intel Core i5-2300 CPU @ 2300 ГГц, ОЗУ 8 Гб) у режимі виявлення. Тестовий набір даних містив близько

70000 записаних сесій, час виявлення становив 0,74669 с. Таким чином, швидкість виявлення веб-атак оцінюється величиною близько 100 000 сесій в секунду.

Таблиця 3.4 - Протокол експерименту

Експеримент/Характеристика	Запуск 1	Запуск 2	Запуск 3
Етап навчання моделі			
Використовуваний набір даних	Збалансована та передопрацьована подібірка веб-атак WebAttacks набору даних CICIDS2017. 7267 записів, з них 5087 екземплярів класу «немає атаки» та 2180 екземплярів класу «є атака».		Сформований набір даних, що відповідають реальному мережному трафіку
Навчальна моель	70% записів використовуваного набору даних		70% записів набору даних
Ознаковий простір	1. Average Packet Size 2. Flow Bytes/s 3. Max Packet Length 4. Fwd Packet Length Mean 5. FwdlATMin 6. Total Length of Fwd Packets 7. FwdlATStd 8. Flow IAT Mean 9. Fwd Packet Length **Max** 10. Fwd Header Length		1. Flow Packets/s 2. Flow IAT Max 3. Bwd Packet Length Min 4. Flow Duration 5. Flow IAT Mean 6. Flow IAT Std 7. Average Packet Size 8. Fwd Packet Length Max 9. Total Packets 10. Fwd Header Length
Етап тестування моделі			
Тестова вибірка	30% записів використовуваного набору даних. Тестова і навчальна вибірка немає перетинів.	100% записів сформованого набору даних, що відповідають реальному мережевому трафіку	30% записів використовуваного набору даних. Тестова та навчальна вибірка не мають перетинів.
Значення метрик якості			
Accuracy	0.983	0.456	0.858
Precision	0.982	0.812	0.812
Recall	0.961	0.033	0.966
Fl	0.971	0.064	0.882

Завершено експеримент із розробкою моделі «випадковий ліс» для вирішення завдання виявлення комп'ютерних атак. Модель навчена на публічному наборі даних CICIDS2017 та протестована в реальних умовах.

Налаштування параметрів вибраного класифікатора RandomForestClassifier пакету scikit-learn дозволило на тестовій вибірці отримати оцінку повноти (recall) 0.961 та F1-міри 0.971 для набору даних CICIDS2017 та 0.966 та 0.882 відповідно для сформованого.

Головний висновок експерименту: методи машинного навчання практично застосовуються для виявлення комп'ютерних атак.

1. Характер комп'ютерних атак, що моделюються, при зборі навчального набору даних відрізнявся від реального.

2. Частина значимих ознак безпосередньо залежить від фізичної структури мережі, у якій проводився збір мережного трафіку, і навіть налаштувань мережного устаткування.

Ідеально – навчати модель на наборі даних, розміченому на основі аналізу мережевого трафіку в мережі, що захищається. При використанні попередньої *в* іншій мережі моделі (проблема transfer learning) обов'язковим є відповідність фізичної структури мережі і мережі, в якій навчалася модель, а також налаштувань мережевого обладнання.

ВИСНОВКИ

Проведено дослідження IDS та IPS системи виявлення вторгнень, що дало можливість здійснити оцінку ефективності виявлення вторгнень.

Досліджено індикатори атак, створені штучним інтелектом на основі аналізу мережевого трафіку.

Проаналізовано можливості Splunk Machine, щодо побудови моделы виявлення вторгнень та аналізу нетипової поведінки мережевого трафіку.

Виконано дослідження побудови класифікаторів атак для системи виявлення вторгнень побудованої на основі класифікаторів.

Зпроектовано систему виявлення вторгнень на основі ML та здійснено вибір набору даних для навчання.

Виконано семплювання проти дисбалансу класів та оцінка значущості та відбір ознак та скорочення ознакового простору та налаштування моделі.

Проведено тестування побудованої моделі та здійснено оцінку її ефективності.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Zhijun W. et al. Low-rate DoS attacks, detection, defense, and challenges: a survey //IEEE Access. -2020. -T. 8. -P. 43920-43943.

2. Hristov M. et al. Integration of Splunk Enterprise SIEM for DDoS Attack Detection in IoT //2021 IEEE 20th International Symposium on Network Computing and Applications (NCA). -IEEE, 2021. -P. 1-5.

3. Moustafa N., Slay J. UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSWNB15 network data set) //2015 military communications and information systems conference (MilCIS). -IEEE, 2015. -P. 1-6.

4. Khraisat, A., Gondal, I., Vamplew, P. et al. Survey of intrusion detection systems: techniques, datasets and challenges // Cybersecur. 2019. -Vol.2. - №. 1. - P. 1-22.

5. Liguo Chen, Yuedong Zhang, Qi Zhao, Guanggang Geng, ZhiWei Yan, Detection of DNS DDoS Attacks with Random Forest Algorithm on Spark // Procedia Computer Science. -2018. -Volume 134.- P. 310-315.

6. Gao Y. et al. A distributed network intrusion detection system for distributed denial of service attacks in vehicular ad hoc network //IEEE Access. -2019. -T. 7. -P. 154560-154571.

7. Gadze J. D. et al. An investigation into the application of deep learning in the detection and mitigation of DDOS attack on SDN controllers //Technologies. -2021. -T. 9. -№. 1. -P. 14.

8. Awan M. J. et al. Real-time DDoS attack detection system using big data approach //Sustainability. -2021. -T. 13. -№. 19. -P. 10743.

9. Han, S., Kim, H. & Lee, YS. Double random forest// Mach Learn. -2020. -P. 1569-1586. *https* : ==*doi:org*=10:1007=*s*10994 - 020 - 05889 – 1

10. Moustafa N., Slay J. UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSWNB15 network data set) //2015 military communications and information systems conference (MilCIS). -IEEE, 2015. -P. 1-6.

11. Symantec, "Internet security threat report 2017," April, 7017 2017, vol. 22/ [Електронний ресурс].

URL:https://www.symantec.com/content/dam/symantec/docs/reports/istr-22-2017-en.pdf

12. Breach_LeveL_Index. (2017, November). Data breach statistics. [Електронний ресурс]. URL: http://breachlevelindex.com/

13. Australian. (2017, November). Australian cyber security center threat report 2017. [Електронний ресурс]. URL: https://www.cyber.gov.au/acsc/viewall-content/reports-and-statistics/acsc-threat-report-2017

14. R. Singh, P.-K. Mannepalli, “Survey on Feature Reduction Techniques of Intrusion Detection System”. International Journal of Engineering Research in Current Trends (IJERCT) ISSN: 2582-5488, Volume-2 Issue-3, Jun 2020

15. H.-J. Liao, C.-H. Richard Lin, Y.-C. Lin, and K.-Y. Tung, "Intrusion detection system: a comprehensive review," J Netw Comput Appl, vol. 36, no. 1, pp. 16–24, 2013a/01/01/ 2013

16. Khraisat A, Gondal I, Vamplew P (2018) An anomaly intrusion detection system using C5 decision tree classifier. In: Trends and applications in knowledge discovery and data mining. Springer International Publishing, Cham, pp 149–155

17. Kreibich C, Crowcroft J (2004) Honeycomb: creating intrusion detection signatures using honeypots. SIGCOMM Comput Commun Rev 34(1):51– 56

18. Roesch M (1999) Snort-lightweight intrusion detection for networks. In: Proceedings of the 13th USENIX conference on system administration. Seattle, Washington, pp 229–238

19. Vigna G, Kemmerer RA (1999) NetSTAT: a network-based intrusion detection system. J Comput Secur 7:37–7278

20. C. R. Meiners, J. Patel, E. Norige, E. Torng, and A. X. Liu, "Fast regular expression matching using small TCAMs for network intrusion detection and prevention systems," presented at the Proceedings of the 19th USENIX conference on security, Washington, DC, 2010

21. Lin C, Lin Y-D, Lai Y-C (2011) A hybrid algorithm of backward hashing and automaton tracking for virus scanning. IEEE Trans Comput 60(4):594– 601

22. Butun I, Morgera SD, Sankar R (2014) A survey of intrusion detection systems in wireless sensor networks. IEEE Communications Surveys & Tutorials 16(1):266–282

23. A. Alazab, M. Hobbs, J. Abawajy, and M. Alazab, "Using feature selection for intrusion detection system," in 2012 international symposium on communications and information technologies (ISCIT), 2012, pp. 296–301

24. Creech G, Hu J (2014a) A semantic approach to host-based intrusion detection systems using Contiguousand Discontiguous system call patterns. IEEE Trans Comput 63(4):807–819

25. Ye N, Emran SM, Chen Q, Vilbert S (2002) Multivariate statistical analysis of audit trails for host-based intrusion detection. IEEE Trans Comput 51(7):810–820

26. Bhuyan MH, Bhattacharyya DK, Kalita JK (2014) Network anomaly detection: methods, systems and tools. IEEE Communications Surveys & Tutorials 16(1):303–336

27. L. Chao, S. Wen, and C. Fong, "CANN: an intrusion detection system based on combining cluster centers and nearest neighbors," Knowl-Based Syst, vol. 78, pp. 13–21, 4// 2015

28. S. Elhag, A. Fernández, A. Bawakid, S. Alshomrani, and F. Herrera, "On the combination of genetic fuzzy systems and pairwise learning for improving detection rates on intrusion detection systems," Expert Syst Appl, vol. 42, no. 1, pp. 193–202, 1// 201579

29. Buczak AL, Guven E (2016) A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Communications Surveys & Tutorials 18(2):1153–1176

30. N. Walkinshaw, R. Taylor, and J. Derrick, "Inferring extended finite state machine models from software executions," Empirical Software Engineering, journal article vol. 21, no. 3, pp. 811–853, June 01 2016.