УДК 004.056.5 **DOI:10.30837/rt.2024.2.217.04**

М.С. КАВЕЦЬКИЙ, О.В. СЄВЄРІНОВ, канд. техн. наук., Р.Ю. ГВОЗДЬОВ, А.О. СМІРНОВ, канд. техн. наук.

ВИКОРИСТАННЯ МАШИННОГО НАВЧАННЯ ДЛЯ КЛАСИФІКАЦІЇ АТАК ТИПУ DOS/DDOS

Вступ

Актуальність роботи проявляється у необхідності виявлення та протидії атакам типу DOS/DDOS, що є серйозною загрозою для сучасних інформаційних систем. Аналіз показав, що ці кібератаки призводять до значних економічних збитків та перерв у роботі мережевих сервісів, підкреслюючи важливість розробки ефективних методів їх виявлення [1, 2]. Сьогодні набирає популярність використання машинного навчання для класифікації різного роду атак. В роботі пропонується використовувати метод дерева прийняття рішень для покращення подібних класифікаторів. Дослідження потенціалу дерев прийняття рішень в цьому контексті вказує на перспективність таких моделей для забезпечення безпеки мереж та запобігання інформаційним загрозам у сучасному цифровому середовищі. Необхідність побудови більш точних класифікаторів для виявлення атак типу DOS/DDOS визначається швидкою еволюцією кіберзагроз та їх все більш складним характером. Точні класифікатори, такі як ті, що базуються на деревах прийняття рішень з оптимально підібраними гіперпараметрами, є ключовим елементом в забезпеченні ефективного захисту мережевих ресурсів і даних користувачів від сучасних кіберзагроз.

Дерева прийняття рішень є одним з методів машинного навчання для класифікації. Алгоритм дерев прийняття рішень легко використовувати за допомогою мови програмування Python та наявності відповідних бібліотек. В Python одним з найпопулярніших пакетів для реалізації алгоритмів дерева прийняття рішень є бібліотека scikit-learn. Вона забезпечує простий і зручний інтерфейс для побудови, навчання та оцінки моделей дерев прийняття рішень, що допомогає швидко та ефективно проводити експерименти з виявлення та протидії кібератакам, а також порівнювати отримані результати, використовуючи загальноприйняті метрики.

Вибір методу дерев прийняття рішень був мотивований його здатністю до адаптації до складних класифікаційних задач та потенціалом у вдосконаленні результатів через правильне налаштування параметрів моделі та відбір оптимального датасету. Використання методу дерев прийняття рішень показало значні покращення у точності виявлення атак, а також в повноті класифікації, що підкреслює його перевагу у контексті обраної проблеми. Також відзначається важливість правильного вибору параметрів моделі та датасету для досягнення найкращих результатів у виявленні атак типу DOS/DDOS. Висновки дослідження підкріплені порівняльним аналізом з іншими методами з інших досліджень, що підтверджує перевагу обраного методу в даному контексті.

Метою роботи є проведення порівняльного аналізу методу дерев прийняття рішень для побудови класифікаторів виявлення атак типу DOS/DDOS. В результаті порівняння побудованих класифікаторів було підтверджено поставлену гіпотезу, а також побудована модель, яка має кращі характеристики з усіх досліджуваних робіт.

Загальні відомості про атаки DOS/DDOS

DoS (Denial of Service) та DDoS (Distributed Denial of Service) атаки є одними з найпоширеніших форм кібератак, що спрямовані на виведення з ладу мережевих ресурсів шляхом перевантаження їх запитами. Вони можуть призвести до недоступності сервісів для легітимних користувачів. DoS атака здійснюється з одного джерела і спрямована на перевантаження цільового сервісу або мережі, використовуючи великі обсяги трафіку або експлуатуючи уразливості системи. DDoS атака є розподіленою, що здійснюється з багатьох

джерел (часто ботнетів), координовано спрямованих на ціль, що робить її більш важкою для виявлення та блокування.

Для виявлення атак типу DoS/DDoS машинне навчання використовується для аналізу комплексних мережевих даних з метою автоматичного виявлення аномальної активності, яка може свідчити про потенційні загрози. Основна перевага цих методів полягає у їх здатності розпізнавати як відомі, так і невідомі типи атак, що є критично важливим для протидії кіберзагрозам, що постійно змінюються.

Машинне навчання для виявлення DoS/DDoS атак базується на аналізі різних параметрів мережі, таких як обсяги трафіку, швидкість передачі даних, час між пакетами, типи запитів і відповідей, а також інші характеристики. Моделі навчаються на історичних даних, де вони виявляють закономірності та шаблони нормальної мережевої активності. Після тренування моделі стають здатними автоматично виявляти відхилення від цих норм, що може свідчити про атаку. Наприклад, модель може виявити збільшену кількість запитів від одного джерела або надмірне використання ресурсів, що є типовими ознаками DoS/DDoS атак. Після виявлення аномалій модель може приймати рішення щодо подальших дій, таких як блокування атакуючих IP-адрес або активація додаткових захисних механізмів.

Аналіз наукових робіт

Автори роботи [3] побудували декілька моделей для виявлення DDOS/DOS атак на базі алгоритмів RF, SVM, NB. І хоча їх задача не була досягти дуже гарної точності (точність RF=0.86, NB=0.79, SVM=0.79), вони описали датасет та провели експеримент з вибором найбільш корисних характеристик з нього. Запевняють, що складність алгоритму класифікації залежить від кількості характеристик в датасеті та кількості даних. Проте, в результаті вони використовували датасет CSICIDS2017, який має 84 унікальні характеристики та обрали всього лише 10 з них. В ході виконання цієї роботи з'явилося припущення, що поганий результат точності класифікації в роботі [3] може бути через вибір занадто малої кількості характеристик. Власне, в цій роботі було обрано більшу кількість характеристик та побудовано на їх основі модель класифікації, яка має більшу точність по всім параметрам (accuracy, recall, precision). Ці метрики наведені далі, а числові результати роботи [3] детально описані в розділі з результатами та порівняннями.

В роботі [4] було використано власноруч зібраний датасет. І хоча методи для виявлення веб-атак були майже такіж самі (SVM, KNN, ANN, NB) як і в роботі [3], але наведене дослідження корисне саме їх експериментом. Суть в тому, що вони будували різні моделі на своєму датасеті, в ході експерименту обираючи різні характеристики з цього датасету. Вони прийшли до висновку, що надійність класифікатору можна сказати визначається ще перед тренуванням, коли обирається набір характеристик, які будуть використовуватися під час тренування. Найбільша точність, яка вийшла в них після вибору певних ознак стала 98.3 % на алгоритмі KNN, проте без вибору ознак вона була 95.67 %. Це на 2.63 % точності менше. І хоча вони отримали непогану точність 98.3 %, проте обрані методи для класифікації, а також результат можна вважати не дуже задовільним в порівнянні з отриманим під час використання дерев прийняття рішень в цій роботі, а також іншого, більш великого, датасету. По іншим метрикам типу Sensitivity, Recall також є просадка, що дає більш загальну картину точності моделі.

Дослідження [5] використовує датасет CICIDS2018, який є оновленою версією CICIDS2017 та містить більший спектр атак. Хоча це не грає ролі для даного дослідження, бо з того датасету беруться дані тільки для атак DOS/DDOS, які є в обох версіях датасетів. Автори роботи «підготували» (опрацювали) для тренування датасет (ці етапи також розглядаються в даній роботі), нормалізували його, та натренували багато різних класифікаторів на різних алгоритмах таких як RF, SVM, KNN, DecisionTree (дерева прийняття рішень), XGBoost. В цій роботі зазначається, що всі алгоритми використовували під час тренування параметри за замовчуванням. Проте, для більшої ефективності моделі необхідно змінювати

ці параметри та підбирати більш оптимальні, щоб результат був краще. Тому і хоча в ході експерименту в них найкраще показав себе алгоритм RF (точність 0.9913), проте зазначається, що в ході експерименту з підбором характеристик з датасету, найбільший приріст у точності дав алгоритм дерев прийняття рішень (DecisionTree), який дав найкращий приріст у точності з 0.9554 (для датасету без обирання певних характеристик) до 0.9895 – тобто, приріст у точності після обрання кращих характеристик з датасету склав 0.0341, де алгоритм RF посів друге місце з приростом всього лише 0.0278. Тож, можна зробити припущення, що алгоритм дерев прийняття рішень для подібної задачі може навіть перевершити результати всіх інших досліджень. Також варто зауважити, що автори дослідження [5] використовували під час тренування параметри за замовчуванням, що є певним недоліком даної роботи, який можна покращити та побудувати модель кращої якості по всім параметрам.

Загалом роботи, які використовують алгоритми машинного навчання для виявлення атак типу DOS/DDOS, дають певне поле для вдосконалення в плані якості моделей. Найбільш гарно показали себе моделі з роботи [5], які давали найбільшу точність. Проте, також ця робота стимулює спробувати покращити результати шляхом використання алгоритму дерев прийняття рішень, бо саме цей алгоритм згідно з цією роботою показав найкращий приріст в точності серед всіх інших. Також сама по собі реалізація має свої недоліки під час тренування, які можна прибрати та побудувати більш точну модель класифікатора. Розглянемо алгоритми, які використовувалися в минулих роботах, а також опис алгоритму дерев прийняття рішень, який був досліджений. Наведено переваги з точки зору теорії даного методу серед інших.

Аналіз алгоритмів машинного навчання в порівнянні з деревом рішень

Розглянемо основний узагальнений принцип дії алгоритмів машинного навчання, які використовувалися в роботах [3 – 5], з їх принципом дії, перевагами та недоліками в розрізі з використаним в цій роботі алгоритмом дерев прийняття рішень.

Принцип дії алгоритму Random Forest можна навести так: це ансамбельний метод, який використовує кілька дерев рішень для класифікації або регресії. Кожне дерево обчислює прогноз, і потім за допомогою голосування або середнього значення результатів дерев формується кінцевий прогноз моделі. Порівнюючи зі звичайним деревом прийняття рішень Random Forest використовує більше одночасно побудованих дерев, що зменшує ризик перенавчання, що може виникнути у випадку глибоких дерев рішень. Це дозволяє отримувати більш стабільні та точні результати, особливо коли даних багато і вони мають велику кількість ознак. Однак його складно інтерпретувати, він довго навчається та потребує багато ресурсів.

Наступний метод Support Vector Machine (SVM) – це метод для розділення класів шляхом знаходження гіперплощини, яка максимально відокремлює дані одного класу від іншого. SVM шукає оптимальну гіперплощину, яка максимізує маржу (відстань між найближчими точками обох класів) і мінімізує помилки класифікації. Порівнюючи з деревом рішень SVM використовує математичні методи для знаходження оптимальної гіперплощини, тоді як Decision Tree розділяє дані на основі порогових значень ознак. SVM може дати кращі результати у випадках, коли дані мають складну неоднорідність, і коли кількість ознак більша за кількість вибірок. Однак може бути чутливий до масштабування ознак і об'єму даних, а також вимагає чіткого налаштування параметрів ядра і регуляризації.

Наївний Баєсівський класифікатор Naive Bayes (NB) використовує теорему Баєса для класифікації об'єктів. Він припускає, що всі ознаки незалежні між собою, що робить його особливо ефективним для текстових даних та категоріальних ознак. Порівнюючи з деревом прийняття рішень наївний баєсівський класифікатор працює на основі ймовірності входження даних до класу, тоді як Decision Tree розділяє дані на основі порогових значень ознак. NB може бути менш точним у складних задачах, де ознаки сильно залежать одна від одної.

Алгоритм k-Nearest Neighbors (kNN) – це метод, що класифікує об'єкти на основі їхніх найближчих сусідів. Він шукає k найближчих точок даних до нового прикладу і визначає клас на основі більшості його сусідів. kNN не вимагає попереднього навчання, він просто зберігає набір даних. В порівнянні з Decision Tree, kNN може бути менш ефективним у великих наборах даних з багатьма ознаками через високу обчислювальну складність. З недоліків можна виділити чутливість до шуму та викидів у даних, а також низьку швидкість визначення класу для великої кількості даних.

Також в роботі [4] використовувалися звичайні штучні нейронні мережі (ANN), які можна сказати моделюють людський мозок, використовуючи шари нейронів, які передають сигнали один одному. Вони складаються з вхідних, прихованих і вихідних шарів, де кожен шар містить нейрони з вагами, які змінюються під час навчання. Однак ANN в порівнянні з деревами рішень вимагає більшої кількості даних для навчання і налагодження параметрів.

Gradient Boosting – це метод, який використовує ансамбль слабких моделей, щоб послідовно покращувати прогноз шляхом оптимізації градієнтного спуску. Кожна наступна модель намагається зменшити помилку попередньої моделі. GB покращує результати шляхом комбінації декількох слабких моделей, тоді як Decision Tree працює незалежно від інших моделей. Але з недоліків можна виділити високу чутливість до високо-змінюваних даних, що може призводити до перенавчання. Також алгоритм вимагає налаштування багатьох гіперпараметрів для досягнення оптимальної продуктивності.

Обраний для дослідження в цій роботі алгоритм Decision Tree. Дерево рішень розділяє дані на основі набору правил, що максимізують інформаційний приріст або зменшують ентропію на кожному кроці. Вузли дерева представляють ознаки, а гілки – умови розбиття. Він дозволяє легко інтерпретувати результати, а також добре працює з категоріальними та числовими даними, однак може бути схильним до перенавчання при занадто глибокому дереві, а також не завжди добре підходить для задач з нерегулярною структурою даних. Проте останнє не грає ролі в даній задачі класифікації атак.

Необхідно більш детально розглянути датасет, який використовувався під час дослідження.

Опис та характеристики використаного датасету

У дослідженні було використано датасет CICIDS2017 [6], який є великим набором даних, який використовується для досліджень в області кібербезпеки, зокрема для виявлення аномалій та атак в мережевому трафіку. Цей набір даних створено Canadian Institute for Cybersecurity і містить різноманітні типи мережевих атак, що робить його дуже корисним для навчання та тестування моделей машинного навчання.

CICIDS2017 був створений для того, щоб надати дослідникам зразки мережевого трафіку, які відображають реальні сценарії атаки та нормальної роботи мережі. Цей набір даних служить важливим ресурсом для розробки і тестування методів виявлення кіберзагроз, включаючи виявлення аномалій, інвазивного аналізу і реагування на інциденти.

Як зазначають розробники датасету, для створення CICIDS2017 використовувалось реальне корпоративне мережеве середовище, що включало сервери, маршрутизатори, комутатори та кінцеві пристрої. Дані були зібрані протягом семи днів, включаючи нормальний трафік, що складається з повсякденних дій користувачів, таких як перегляд веб-сторінок та доступ до баз даних, а також атакуючий трафік, що включав DoS, DDoS, brute force, SQL ін'єкції та сканування портів. Для генерації атак використовувались інструменти, такі як hping3 для DoS та DDoS атак і Metasploit для здійснення експлойтів. Всі мережеві пакети були захоплені за допомогою інструментів Wireshark або Tcpdump, а також велись журнали активності. Дані були ретельно анотовані, що включає мітки для нормального трафіку та різних типів атак, і поділені на навчальні та тестові набори. Це середовище забезпечує достатню кількість прикладів для обох видів трафіку і створює репрезентативний набір даних для моделювання реальних загроз. Тому цей датасет є цілком репрезентативним та відображає реальні дані.

В датасеті наявно багато колонок (які будемо називати характеристиками), що відображають різні аспекти мережевого трафіку. Наприклад, в датасеті наявна колонка dst_port, яка представляє порт призначення, flow_duration, що вимірює тривалість потоку в мілісекундах, та tot_fwd_pkts, яка показує загальну кількість пакетів, що були відправлені вперед у певному потоці. Колонка totlen_fwd_pkts відображає загальну довжину цих пакетів у байтах, а flow_byts_s і flow_pkts_s показують середню кількість байтів і пакетів за секунду відповідно. Всі ці колонки разом дозволяють детально аналізувати поведінку мережевого трафіку, що є важливим для виявлення аномалій та атак.

Процес підготовки даних до тренування

Загалом в датасеті CICIDS2017 міститься багато даних з різними атаками сгрупованими по днях, але в рамках дослідження нас цікавить лише DOS/DDOS, тому було обрано файл «Wednesday-workingHours», який містить 692 тисячі записів атак даного типу. В ньому 79 унікальних характеристик (колонок), за якими можна досліджувати дані та будувати класифікатор.

На рис. 1 наведено повний алгоритм роботи з даними, який необхідний для підготовки датасету до тренування.

Рис. 1. Послідовність роботи з даними для підготовки до тренування

Опишемо більш детально послідовність роботи з даними згідно з рис. 1. Спершу дані потрібно зчитати з файлів. Ці дані були у вигляді файлу «Wednesday-workingHours.pcap_ISCX.csv» в форматі «csv», який можна зчитувати за допомогою бібліотек мови Python.

Далі, наступний етап – розмітка даних на класи. У кожному файлі представлено певний клас трафіку (атака або нормальний), тому додається колонка "label" для маркування: 0 означає відсутність атаки, а 1 – її наявність. Це потрібно тому що алгоритми машинного навчання в основному працюють з числовими даними.

Третій етап передбачає видалення невалідних або некорисних даних. Це включає рядки з пропущеними чи неправильними значеннями, наприклад текст у числових колонках. Некорисні дані, такі як IP-адреси, MAC-адреси, порти, протоколи та мітки часу, також видаляються. Крім того, видаляються дані з надто великими значеннями, що можуть спричиняти помилки, наприклад, числа потоку байтів і пакетів за секунду (flow_byts_s, flow_pkts_s).

Четвертий крок – розподіл даних на тренувальну та тестову частини. Модель навчається на тренувальній частині та тестується на даних, які вона раніше не бачила, щоб оцінити її точність. Дані поділяються на X (ознаки) та Y (мітки класу, атака чи нормальний трафік). В даному випадку розподіл був 80/20, де 20 % датасету пішло на тестувальну вибірку. Таким чином для тренування було 553356 даних, а для проведення тестів – 138339 екземплярів.

П’ятий крок – масштабування даних. Оскільки деякі колонки мають великі значення, а інші – малі, необхідно масштабувати дані, щоб модель не надавала перевагу більшим значенням. Для цього використовується MinMaxScaler, який нормалізує дані. Формула, за якої відбувається масштабування даних:

$$\frac{X-Xmin}{Xmax-Xmin} \tag{1}$$

де X – поточний елемент колонки для масштабування; $Xmin$ – мінімальне значення колонки; $Xmax$ – максимальне значення колонки.

Формула (1) застосовується до кожного елемента кожної колонки у наборі даних. Таким чином, можна отримати нормовані значення, навіть якщо одна колонка оперує числами у діапазоні від 1000 до 10000 та інша від 1 до 10, то в результаті нормування даних, будуть дві колонки, які оперують числами від 0 до 1. Такі числа дуже зручні для тренування, та дають гарні результати, тому що модель дивиться на колонки як на рівноправні за важливістю.

Далі розглянемо показники, за якими можна оцінювати якість моделей, а також опишемо процес тренування класифікатору та наведено порівняння з класифікаторами з інших робіт.

Показники оцінки якості моделей

Наведемо опис характеристик для оцінки якості моделей. За цими характеристиками буде також відбуватися порівняння побудованої в цій роботі моделі з класифікаторами з інших робіт.

Confusion Matrix: таблиця, яка показує кількість правильних і неправильних передбачень, розділених на категорії. Вона містить чотири значення: True Positives (TP) – кількість правильно ідентифікованих атак; False Positives (FP) – кількість нормального трафіку, помилково ідентифікованого як атака; True Negatives (TN) – кількість правильно ідентифікованого нормального трафіку; False Negatives (FN) – кількість атак, які залишилися непоміченими.

Accuracy: частка правильних передбачень серед усіх передбачень. Розраховується за формулою

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{2}$$

Precision: частка правильних передбачень атак серед усіх передбачених атак. Розраховується за формулою

$$Precision = \frac{TP}{TP+FP} \tag{3}$$

Recall (Sensitivity): частка правильно ідентифікованих атак серед усіх реальних атак. Розраховується за формулою

$$Recall = \frac{TP}{TP+FN} \tag{4}$$

F1 Score: гармонійне середнє між Precision та Recall, що враховує як помилки пропуску атак, так і помилки ідентифікації нормального трафіку як атаки. Розраховується формулою

$$F1 = 2 * \frac{Precision*Recall}{Precision+Recall} \tag{5}$$

Далі наведемо опис підходу до тренування моделі дерева прийняття рішень, а також візуально показано як можна інтерпретувати результати моделі.

Тренування класифікатора на основі дерева прийняття рішень

Після збору та обробки даних настає етап тренування моделі, який потребує певного часу та налаштування гіперпараметрів. Наприклад, для дерева прийняття рішень використовувалися такі параметри: random_state для забезпечення відтворюваності результатів, max_depth для обмеження глибини дерева, min_samples_split для визначення мінімальної

кількості вибірок для розділення вузла та max_features для вказівки кількості ознак для вибору найкращого рішення.

Мова програмування Python має популярну бібліотеку sklearn, в якої використовується багато методів і класів для машинного навчання, включаючи клас DecisionTreeClassifier для дерева прийняття рішень. Використовуючи цей клас, можна передати дані для тренування, налаштувати гіперпараметри та запустити процес тренування за допомогою методу fit().

Тренування відбувається досить швидко, бо самі дані не є дуже великими та алгоритм досить швидкий. Для тренування моделі дерева прийняття рішень обрані наступні гіперпараметри: random_state=1, щоб забезпечити відтворюваність результатів і однаковий розподіл випадковості під час перемішування даних; max_depth=150, що встановлює максимальну глибину дерева, обмежуючи його, щоб уникнути перенавчання; та min_samples_split=15, що визначає мінімальну кількість вибірок, необхідних для розділення вузла дерева, забезпечуючи баланс між складністю моделі та її здатністю узагальнювати нові дані.

Результати моделі показують високу ефективність у виявленні DoS/DDoS атак. Зведена таблиця конфузійної матриці демонструє, що модель правильно класифікувала 87,749 випадків, як нормальний трафік (True Negatives), і 50,552 випадків, як атаки (True Positives). Лише 27 нормальних запитів були помилково ідентифіковані як атаки (False Positives), і лише 11 атак залишилися непоміченими (False Negatives). Загальна точність моделі становить 99.97 %, що свідчить про її високу здатність правильно класифікувати трафік. Показник Precision дорівнює 99.95 %, що означає, що майже всі випадки, класифіковані як атаки, дійсно були атаками. Показник Recall складає 99.98 %, що свідчить про здатність моделі виявляти майже всі реальні атаки. Значення F1 score, яке враховує як Precision, так і Recall, становить 99.96 %, підкреслюючи загальну ефективність моделі.

Аналіз та порівняння результатів моделі зі схожими моделями з інших робіт

Проведемо порівняння та інтерпретуємо результати порівняння зі схожими роботами, котрі будували подібний класифікатор DOS/DDOS трафіку. В табл. 1 можна побачити порівняння результатів тренування моделей з досліджень [3 – 5] з результатами тренування моделі на основі дерева прийняття рішень в цій роботі.

Таблиця 1

Порівняння якості створеної та моделей з інших робіт

Дослідження	Датасет/характеристики	Алгоритм	Метрики			
			Accuracy	Precision	Recall	F1 міра
[3]	CSICIDS2017 / 10	RF	0.8680	0.9963	0.8629	не зазначено
		NB	0.7999	0.8603	0.9006	не зазначено
		SVM	0.7988	0.8436	0.9244	не зазначено
[4]	Власний / 6-10	KNN (8)	0.9830	0.9772	0.9773	0.9770
		NB (10)	0.9487	0.9329	0.9205	0.9201
		SVM (10)	0.9215	0.9023	0.9020	0.9021
		ANN (6)	0.9144	0.8811	0.8772	0.8789
[5]	CICIDS2018 / 26	RF	0.9913	0.9843	0.9992	0.9913
		KNN	0.9886	0.9801	0.9982	0.9885
		SVM	0.9689	0.9583	0.9812	0.9685
		XGBoost	0.9894	0.9806	0.9994	0.9894
		DecisionTree	0.9895	0.9847	0.9947	0.9875
Ця робота	CICIDS2017 / 75	DecisionTree	0.9997	0.9994	0.9997	0.9996

Результати роботи [3] не мають значення F1 міри, проте це не зовсім важливо через доволі низьку точність в побудові класифікатору, що робить подальше порівняння не потрібним. Низьку точність з роботи [1] теоретично могло спричинити невдале обрання характеристик з датасету. Проте, ця робота ставила більше собі за мету показати необхідність обирати правильні характеристики, тож вони могли навмисно взяти таку малу кількість.

В табл. 1 наведено відомості з роботи [4], які були взяті з порівняльної таблиці для методу обирання ознак, який автори прозвали «Wrapper» (вони описали його як підхід до відбору

ознак використовує алгоритм навчання для оцінки корисності підмножини ознак). Робота показала вже більш менш непогані результати, де алгоритм KNN показав найкращі результати в усіх метриках. Точність (Accuracy) складає 98.30 %, що свідчить про високу здатність алгоритму правильно класифікувати вхідні дані. Precision та Recall також мають високі значення (97.72 % та 97.73 % відповідно), що вказує на високу точність та повноту класифікації. F1-міра, яка є гармонійним середнім між Precision та Recall, дорівнює 97.70 %, що підкреслює баланс між цими двома метриками.

Дослідження [5] використовує схожий датасет з дослідженням в цій роботі, проте вони вирішили обрати менше характеристик з датасету, але більше ніж в інших роботах [4, 5]. Серед використаних алгоритмів найкращі результати показав алгоритм Random Forest (RF) з Accuracy 0.9913, Precision 0.9843, Recall 0.9992, та F1 міра 0.9913. Також високі показники продемонстрував алгоритм K-Nearest Neighbors (KNN) з Accuracy 0.9886, Precision 0.9801, Recall 0.9982, та F1 міра 0.9885. Інші алгоритми, такі як SVM, ANN, XGBoost і DecisionTree, також мали значні результати, але поступалися RF і KNN у точності та інших метриках.

Проте, робота [5] також показала, що хоча й алгоритм RF набрав більше точність, алгоритм дерев прийняття рішень показував найкращий приріст від зміни характеристик датасету. Цього не відображено в фінальній табл. 1, але як вже було сказано в минулих розділах – алгоритм дерев прийняття рішень дав найкращий приріст у точності з 0.9554 (для датасету без обирання певних характеристик) до 0.9895 – тобто, приріст у точності після обрання кращих характеристик з датасету склав 0.0341, де алгоритм RF посів друге місце з приростом всього лише 0.0278. Також в цій роботі був явний недолік. Автори зазначили, що навчали всі алгоритми з параметрами за замовчуванням, але для більшої якості моделі необхідно постійно обирати параметри, підбираючи найкращі, які дадуть більшу точність. У власній реалізації дерев прийняття рішень так і відбулося, що дало кращі результати.

Найкращий результат отримала власна реалізація з алгоритмом DecisionTree. Точність (accuracy), яка вимірює частку правильних передбачень серед усіх передбачень досягла значення 0.9997, Precision (точність позитивних передбачень) показує, скільки з передбачених атак дійсно були атаками – стало 0.9994, Recall (чутливість), який відображає здатність моделі виявляти всі реальні атаки став 0.9997, а F1 міра є гармонійним середнім між Precision і Recall, що дає збалансовану оцінку якості моделі набула значення 0.9996. У порівнянні з іншими роботами, жоден алгоритм не досяг такого рівня точності, точності позитивних передбачень, чутливості та гармонійного середнього, як DecisionTree у власній роботі. Такі результати можна пояснити вибором іншого датасету, відбором релевантних характеристик з датасету, а також більш вдалим алгоритмом для цієї задачі, яким є дерево прийняття рішень. Також головного «конкурента» з роботи [3] вдалося перевершити завдяки підбиранню правильних параметрів до алгоритму під час навчання. Отже, можна сказати, що поставлена гіпотеза підтвердилася.

Висновки

У дослідженні вивчено ефективність методу дерев прийняття рішень для виявлення атак типу DOS/DDOS. Було проаналізовано роботи [3 – 5], які дали розуміння, що саме дерева прийняття рішень мають кращий потенціал серед інших методів, якщо обирати правильні характеристики з датасету, а також налаштовувати параметри тренування моделі. Обрані в цій роботі параметри для тренування були наступні: random_state=1, щоб забезпечити відтворюваність результатів і однаковий розподіл випадковості під час перемішування даних; max_depth=150, що встановлює максимальну глибину дерева, обмежуючи його, щоб уникнути перенавчання; та min_samples_split=15, що визначає мінімальну кількість вибірок, необхідних для розділення вузла дерева, забезпечуючи баланс між складністю моделі та її здатністю узагальнювати нові дані.

Результати моделі показують високу ефективність у виявленні атак типу DoS/DDoS. Зведена таблиця конфузійної матриці відображає, що модель правильно класифікувала

87,749 випадків як нормальний трафік (True Negatives) і 50,552 випадки як атаки (True Positives). Лише 27 нормальних запитів були помилково ідентифіковані як атаки (False Positives), і лише 11 атак залишилися непоміченими (False Negatives). Загальна точність моделі становить 99.97 %, що підкреслює її високу здатність правильно класифікувати трафік. Precision досягає 99.95 %, що свідчить про те, що майже всі випадки, визнані моделлю як атаки, є дійсно атаками. Recall складає 99.98 %, що підтверджує здатність моделі виявляти практично всі реальні атаки. Значення F1 score, яке узагальнює як Precision, так і Recall, становить 99.96 %, що підкреслює загальну ефективність моделі.

Також побудована модель має приріст по точності 0.0102 (близько 1 %) з 0.9895 до 0.9997 серед досліджуваних дерев прийняття рішень, а також приріст точності 0.0084 порівнюючи з найкращою реалізацією серед аналізованих – приріст з 0.9913 (точність кращого алгоритму KNN з роботи [5]) до 0.9997 (точність побудованої в рамках роботи моделі).

Порівняльний аналіз з результатами інших досліджень підтвердив переваги використання дерев прийняття рішень у виявленні атак DOS/DDOS. Звичайно, за умови вибору правильних параметрів під час навчання (бо від цього залежить наприклад крок навчання моделі, або кількість ітерацій по датасету, та багато іншого, що потрібно підбирати), вибору відповідного датасету (він повинен якогомога краще характеризувати типи атак, щоб під час навчання модель мала більше узагальчующих даних, які описують ту чи іншу атаку), а також його правильної попередньої обробки (видалення з даних викидів, обробка числових колонок та інше). Проведена робота показала, що вибір іншого датасету та відбір релевантних характеристик значно вплинули на досягнення найкращих результатів. Додатково, підбір оптимальних параметрів моделі під час навчання підсилив ефективність нашої системи порівняно з конкурентами.

Використання даного методу в системах безпеки мереж може значно покращити їхню здатність до виявлення та захисту від кібератак.

Список літератури:

1. Сєвєрінов О.В., Шевцов В.О., Сокол-Кутиловська А.С. Аналіз сучасних методів атак на електронні ресурси органів управління // Системи озброєння і військова техніка. 2017. №1. С. 65–68.
2. Сєвєрінов О.В., Хрєнов А.Г., Поляков А.О. Аналіз сучасних методів атак на автоматизовані системи управління військами та інформаційні мережі // Системи обробки інформації. 2015. №9. С.101–104.
3. Amer A. Abdulrahman, Mahmood K. Ibrahem, Evaluation of Ddos Attacks Detection in a CICIDS2017 Dataset Based on Classification Algorithms [Електронний ресурс]. Режим доступу: https://www.academia.edu/71363307/Evaluation_of_Ddos_Attacks_Detection_in_a_CICIDS2017_Dataset_Based_on_Classification_Algorithms
4. Polat H., Polat O., Cetin A. Detecting DDoS Attacks in Software-Defined Networks Through Feature Selection Methods and Machine Learning Models // Sustainability. 2020. 12(3). P.1035. https://doi.org/10.3390/su12031035
5. Liu Z., Wang Y., Feng F., Liu Y., Li Z, Shan Y. A DDoS Detection Method Based on Feature Engineering and Machine Learning in Software-Defined Networks // Sensors. 2023. 23(13). P.6176. https://doi.org/10.3390/s23136176
6. IDS 2017 | Datasets | Research | Canadian Institute for Cybersecurity | UNB [Електронний ресурс]. Режим доступу: https://www.unb.ca/cic/datasets/ids-2017.html.

Надійшла до редколегії 05.06.2024

Відомості про авторів:

Кавецький Максим Сергійович – Харківський національний університет радіоелектроніки, магістр кафедри безпеки інформаційних технологій факультет комп'ютерної інженерії та управління; Україна; e-mail: maksym.kavetskyi@nure.ua; ORCID: https://orcid.org/0009-0008-7419-1029

Сєвєрінов Олександр Васильович – канд. техн. наук, доцент, Харківський національний університет радіоелектроніки, професор кафедри безпеки інформаційних технологій, факультет комп'ютерної інженерії та управління; Україна; e-mail: oleksandr.sievierinov@nure.ua; ORCID: https://orcid.org/0000-0002-6327-6405

Гвоздьов Роман Юрійович – Харківський національний університет радіоелектроніки, аспірант кафедри безпеки інформаційних технологій; Україна; e–mail: roman.hvozdov@nure.ua; ORCID https://orcid.org/0000-0002-5408-943X

Смірнов Антон Олександрович – канд. техн. наук, Харківський національний університет радіоелектроніки, доцент кафедри безпеки інформаційних технологій, факультет комп'ютерної інженерії та управління; Україна; e-mail: anton.smirnov@nure.ua, ORCID: https://orcid.org/0000-0003-4121-3902