УДК 004.85+004.056.53
DOI: 10.31673/2409-7292.2024.040003

Гайдур Г. І., Гахов С. О., Гамза Д. Є.

МОДЕЛЬ ВИЯВЛЕННЯ ШКІДЛИВОЇ АКТИВНОСТІ В ІНФОРМАЦІЙНІЙ СИСТЕМІ ОРГАНІЗАЦІЇ НА ОСНОВІ ГІБРИДНОЇ КЛАСИФІКАЦІЇ

Стаття присвячена дослідженню методів виявлення шкідливих процесів в інформаційній системі організацій з використанням методів машинного навчання. Кількість атак, які згідно з статистикою, збільшуються з кожним роком, свідчить про те, що методи виявлення вторгнень потребують подальшого розвитку. Застосування машинного навчання може підвищити здатність систем захисту щодо виявлення шкідливих процесів. Для вирішення проблеми виявлення шкідливих процесів в роботі запропоновано модель гібридної класифікації виявлення вторгнень в інформаційній системі організації. Запропонована модель побудована на використанні декількох методів машинного навчання. До складу цих методів було включено метод опорних векторів, дерева рішень та *k*-найближчих сусідів. Опис основних етапів застосування визначених методів запропонованої моделі надає узагальнене розуміння особливостях роботи. Було використано набори даних CSE-CIC-IDS2018 Канадського інституту кібербезпеки. Для обробки набору даних запропоновано основні етапи, які необхідні для коректної роботи запропонованої моделі. Очікується, що такий підхід дозволить скоротити час роботи моделі та покращити показники якості та точності виявлення шкідливої активності в інформаційній системі організації. В основу гібридної класифікації покладено метод асамблевого навчання - стекінг, який передбачає навчання кількох моделей однаковому наборі даних, а потім використання ще одного класифікатора, який навчається на вихідних даних цих моделей та визначає, як їх комбінувати для отримання кінцевого прогнозу. Отже, запропонована модель виявлення шкідливої активності дозволить отримувати результати прогнозів, які будуть порівнюватися з результатами моделі на основі кожного окремого обраного алгоритму машинного навчання.

Ключові слова: кібербезпека, кібератака, шкідлива активність, ботнет, машинне навчання, класифікація.

Вступ та постановка проблеми

Сьогодні спостерігається стрімкий розвиток інформаційних технологій, який супроводжується не менш стрімким зростанням викликів з якими зіштовхується галузь кібербезпеки. За інформацією щорічних звітів компанії CheckPoint [1] лише в 2022 загальна кількість кібератак зросла на 38% у порівнянні з 2023 роком. Згідно статистики за 2022 рік [1, 2] – ми бачимо, що корпоративні мережі підприємств за типом кібератак переважно піддавалися атакам саме ботнетів (31% від усіх кібератак). Ботнети становлять серйозну загрозу, тому що мають величезний розмір та потенціал для нанесення шкоди інформаційній системі організації. Це пов’язано з тим, що на сьогоднішній день ботнет складається не тільки з комп’ютерів та смартфонів, але й з великої кількості IoT пристроїв, які легко можуть бути скомпрометовані через низьку захищеність.

Ботнет є мережею пристроїв заражених шкідливими бот-програмами, які керуються централізовано та дозволяють віддалено виконувати різноманітні шкідливі дії такі як скоординовані DDoS атаки, розсилка спаму, криптомайнінг, використання обчислювальних потужностей для перебору ключів шифрування тощо. Пристроїв у такій мережі може бути тисячі, сотні тисяч чи навіть мільйони. Зважаючи на зростання номенклатури пристроїв, які можуть бути заражені шкідливою бот-програмою, майбутні ризики, з якими доведеться зустрітися є достатньо великі. Як приклад, особливо розповсюдженою активністю є застосування ботнетів для DdoS-атак, які згідно з звітом компанії Netscout за перше півріччя 2023 року здійснили 7.9 млн. DDoS атак, що на 31% склало більше ніж у 2022 році [1]. Отже, зростання загрози з боку ботнетів, які є шкідливим процесом, потребує покращення методів їх виявлення, які є однією з першочергових задач для фахівців кібербезпеки. Тому тема дослідження є актуальною.

Мета статті. Аналіз робіт, щодо виявлення шкідливої активності в інформаційних системах організацій в основному зосереджено на виявленні ботнетів, які застосовують один обраний метод класифікації або кластеризації. Тому дана робота буде зосереджена на створенні моделі виявлення шкідливих процесів функціонування ботнетів з метою

підвищення якості та точності методів виявлення ботнетів, в основі якої буде застосовано гібридну модель машинного навчання.

Виклад основного матеріалу

1. Огляд існуючих методів виявлення ботнетів

Основою для виявлення процесів функціонування ботнету можуть бути методи:

Аналіз мережевого трафіку. Один із основних методів виявлення ботнетів базується на аналізі мережевого трафіку. Це включає в себе моніторинг різних параметрів, таких як об'єм передачі даних, типові маршрути, шаблони споживання ресурсів і частота взаємодії між вузлами мережі. Аномалії в цих параметрах можуть свідчити про невідомі або підозрілі активності, які можуть бути пов’язані з ботнетом [2, 3].

Виявлення аномалій в системних ресурсах. Ботнети намагаються захопити якнайбільше ресурсів для виконання своїх завдань. Моніторинг системних ресурсів, таких як CPU, пам’ять і мережа, дозволяє виявити надмірне споживання ресурсів. Наприклад, щильний розклад використання CPU або перевантажене розподілення мережевого трафіку може бути індикатором наявності ботнета.

Аналіз поведінки. Сучасні системи виявлення аномальної поведінки використовують технології машинного навчання для аналізу поведінки системи. Моделі, навчені на базі реальних даних, можуть виявляти аномальні патерни, які можуть бути характерними для ботнетів. Наприклад, раптові зміни в робочих годинах або спроби непов’язаного доступу до ресурсів можутьвказувати на потенційно шкідливу активність.

Використання honeypots. Honeypots є системами, які призначені для імітації роботи реальних ресурсів чи служб для залучення зловмисників. Розміщення honeypots у мережі може допомогти виявити спроби вторгнення та привернути увагу ботнетів. Цей метод дозволяє не лише виявити, але і дослідити методи дії ботнетів, що допомагає вдосконалити системи кіберзахисту.

Моніторинг DNS запитів. Ботнети зазвичай встановлюють з’єднання зі своїм централізованим управлінням через DNS. Моніторинг та аналіз DNS запитів може виявити нетипові підключення та комунікації. Підозрілі доменні імена чи нетипова частота запитань можуть слугувати індикаторами наявності ботнета.

2. Методи виявлення ботнетів на основі гібридної класифікації вторгнень

Для гібридної класифікації вторгнень пропонується використовувати методику ансамблевого навчання [4]. Ансамблеве навчання є методом машинного навчання, що полягає в комбінації кількох моделей для отримання кращих прогнозних результатів. Основна ідея полягає в тому, що ансамбль (або група) моделей, в цілому, дає більш точні і надійні прогнози, ніж окрема модель. Це досягається за рахунок того, що різні моделі можуть ефективно вивчати різні аспекти даних, і, комбінуючи їх, можливо отримати більш повне і точне прогнозування щодо аномальної поведінки. Ансамблеві методи, такі як беггінг, бустінг та стекінг, використовуються для підвищення стабільності та точності прогнозування [5].

Ансамблеве навчання може складатися з трьох основних видів:

Бегінг (Bagging): цей метод передбачає навчання кількох моделей на різних піднаборах вихідного наборуданих. Після чого, результати цих моделей комбінуються (зазвичай через голосування або усереднення) для отримання кінцевого прогнозу. Прикладом бегінгу є випадковий ліс (Random Forest).

Бустінг (Boosting): цей метод також передбачає навчання кількох моделей, але в цьому випадку, кожна наступна модель намагається виправити помилки попередньої. Це досягається шляхом надання більшої ваги тим прикладам, які були класифіковані неправильно попередніми моделями. Прикладом бустінгу є AdaBoost або Gradient Boosting.

Стекінг (Stacking): цей метод передбачає навчання кількох моделей на тих самих даних, а потім використання ще одного класифікатора (мета моделі), який навчається на вихідних даних цих моделей та визначає, як їх комбінувати для отримання кінцевого прогнозу. Цей

© Гайдур, Г. І., Гахов, С. О., & Гамза, Д. Є. (2024). Модель виявлення шкідливої активності в інформаційній системі організації на основі гібридної класифікації. Сучасний захист інформації, 4(60), 30–38. https://doi.org/10.31673/2409-7292.2024.040003.

підхід може призводити до підвищення якості передбачень, особливо в тих випадках, коли різні моделі демонструють гарні результати для різних частин обраних параметрів [6].

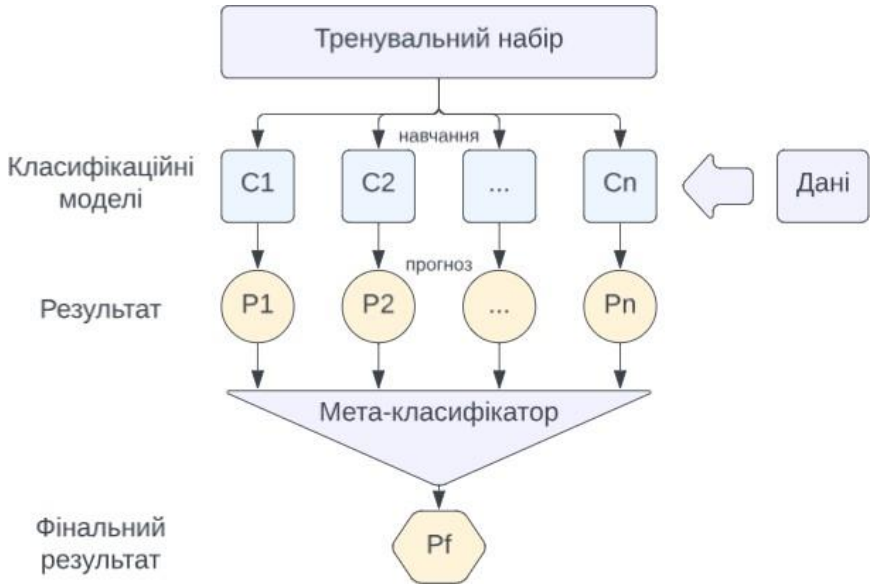

Рис. 1. Запропонована схема методу асамблевого навчання

У цій роботі пропонується застосувати саме стекінг, тому що він має низку переваг:

1. Стекінг дозволяє комбінувати різні моделі, які можуть добре працювати на різних частинах даних або за різними аспектами проблеми. Це дозволяє створювати більш загальні моделі, які можуть краще узагальнюватися до нових, невідомих даних.
2. Стекінг може допомогти зменшити ризик перенавчання, оскільки він використовує прогнози базових моделей як ознаки для мета-моделі, а не просто використовує їх прогнози як кінцевий результат.
3. Щоб досягти максимального результату, стекінг може використовувати різні типи моделей, які мають різні сильні та слабкі сторони. Це дозволяє максимізувати потенційні переваги кожної моделі.
4. Якщо одна модель має тенденцію робити помилки на певних типах даних або для певних класів, інші моделі можуть компенсувати ці слабкості, що призводить до загальної кращої продуктивності.
5. Стекінг дозволяє створювати складніші моделі, включаючи різні комбінації базових моделей та методів об’єднання їх прогнозів. Це дає більшу гнучкість в розробці моделі і може допомогти в досягненні кращих результатів.

Отже, використання стекінгу може допомогти покращити якість та узагальненість моделі шляхом комбінування переваг різних моделей та зменшення їхніх недоліків.

Методика стекінгу складається з таких основних кроків:

1. Ділення навчального набору на піднабори. Починаючи з навчального набору даних, дані розділяються на два чи більше піднабори. Зазвичай використовуються два рівні: один для навчання базових моделей і другий для навчання метамоделі.
2. Навчання базових моделей. Кожна базова модель навчається на одному з піднаборів. Ці моделі можуть бути різнорідними, наприклад, деревами рішень, методами опорних векторів, нейронними мережами тощо.
3. Створення прогнозів базових моделей. Коли базові моделі навчені, вони використовуються для прогнозування на другому (залишковому) піднаборі та на тестовому наборі даних.

4. Створення метаданих. Прогнози, отримані в результаті дії базових моделей на другому піднаборі, стають входом для метамоделі. Ці прогнози стають метаданими або новими функціями для метамоделі.

5. Навчання метамоделі. Метамодель навчається на метаданих, що були отримані з базових моделей. Метамодель вивчає, як поєднувати прогнози базових моделей для отримання кращого фінального результату.

6. Створення фінальних передбачень. За допомогою метамоделі робляться фінальні передбачення на тестовому наборі даних.

Мета-класифікатор у гібридного методу машинного навчання складається з моделі, яка навчається на виходах інших моделей. Така модель називається метамоделлю, яка аналізує результати роботи базових моделей (моделей першого рівня) і визначає, як їх комбінувати для отримання кінцевого прогнозу. Це дозволяє виходити за рамки можливостей окремих моделей та підвищує точність передбачень.

3. Вибір набору даних

Набір даних CSE-CIC-IDS2018 був створений Комп'ютерним відділом науково-дослідних лабораторій безпеки систем Канадського університету, щоб допомогти у виявленні вторгнень у мережу. Набір даних включає в себе характеристики трафіку, який був симульований в п'ятиденний період [7]. Цей трафік включає в себе нормальний трафік, а також шкідливий трафік, симульований за допомогою різних сценаріїв атак. Дані були зібрані за допомогою 9-ти машин, що працюють під управлінням Windows і двох машин з ОС Linux. Кожен з цих комп'ютерів був налаштований на виконання різних задач, включаючи перегляд веб-сторінок, надсилання електронних листів, переклад даних, стрімінг відео та ін.

Щоб зібрати цей набір даних, було використано різні методики виявлення вторгнень, зокрема Snort для виявлення атак на основі сигнатур, Suricata для виявлення зловмисних поведінок на основі правил, та Bro для виявлення вторгнень на основі аналізу мережевого трафіку. Набір даних CSE-CIC-IDS2018 включає в себе близько 16 мільйонів записів, що представляють 80 різних типів атак, включаючи DDoS, DoS, проникнення, сканування портів, атаки на веб-застосунки та ін. Він також включає в себе широкий спектр мережевих характеристик, включаючи джерело та призначення IP-адрес, порти, протоколи, кількість пакетів, об'єм даних та інше.

Таблиця 1

Загальний розподіл типу трафіку у CSE-CIC-IDS2018

Мітка	**Кількість**	**%**
Benign	2,856,035	63.111
Bot	286,191	6.324
Brute Force	513	0.011
DDoS	1,289,544	28.497
Infiltration	93,063	2.056
SQL injection	53	0.001
Total	4,525,399	100

Набір даних CSE-CIC-IDS2018 є оптимальним вибором для даного дослідження з кількох причин. По-перше, він містить реальний мережевий трафік, який було зібрано протягом п'яти днів. Це забезпечує велику кількість даних для аналізу і дозволяє протестувати запропоновану модель на різних сценаріях атак. По-друге, цей набір даних містить дані нормального, так і шкідливого трафіку, що дозволяє навчити запропоновану модель розпізнавати різні види атак. По-третє, набір даних було зібрано з використанням різних методів виявлення вторгнень, що дозволяє нам використовувати ці методи як основу для нашої

© Гайдур, Г. І., Гахов, С. О., & Гамза, Д. Є. (2024). Модель виявлення шкідливої активності в інформаційній системі організації на основі гібридної класифікації. Сучасний захист інформації, 4(60), 30–38. https://doi.org/10.31673/2409-7292.2024.040003.

моделі. По-четверте, набір даних включає велику кількість мережевих характеристик, які можуть бути використані для створення детальних моделей машинного навчання.

Отже, з усіх цих причин, набір даних CSE-CIC-IDS2018 ми вважаємо є найкращим вибором для запропонованого дослідження.

4. Підготовка набору даних

Початковий набір даних містить величезну кількість інформації, яка містить як нормальний так і шкідливий трафік. Поряд з цим можуть міститися некоректні, відсутні чи повторювані значення. У початковому вигляді дані не готові для користування у моделях машинного навчання, оскільки можуть негативно вплинути на продуктивність. Для початку необхідно підготувати таку інформацію шляхом очищення, трансформації та зменшення цього набору даних(табл.2).

Таблиця 2

Опис вмісту набору даних

Директорія	Вміст
Original Network Traffic and Log data	Оригінальні pcap файли запису мережевого трафіку та та логи запису подій (event logs)
Processed Traffic Data for ML Algorithms	Набір csv файлів у якому вибрано та представлено понад 80 характеристик мережевого трафіку

Більшість ботнет трафіку міститься у Processed Traffic Data for ML Algorithms/Botnet-Friday-02-03- 2018_TrafficForML_CICFlowMeter.csv. Саме цей день запису даних містить необхідну нам інформацію, а саме записи ботнет трафіку від ботнету Apec (табл. 2). Але для створення надійної моделі увесь датасет після підготовки буде злитий у один файл. Таке злиття дозволить досягти наступних результатів, яке полягає у наступному:

навчаючись на різноманітному наборі даних датасету, модель навчиться виявляти ботнет-трафік на фоні різних умов та нормальних патернів трафіку, що допоможе в кращому узагальненні на нові дані;

об’єднання всіх днів краще імітує реальні умови, де нормальний та шкідливий трафік змішуються з часом, що допомагає моделі краще диференціювати між нормальним та шкідливим трафіком на практиці.

Очищення датасету

Датасет містить багато надлишкової інформації, яка не несе корисного навантаження і може бути видалена. Очищення даних полягає у видаленні від пустих або помилкових значень, видаленні колонок з однаковими або типовими значеннями, видаленні колонок за параметрами, що нерелевантні для даного дослідження.

Для дослідження очищення даних полягає у наступному:

видалення нерелевантних колонок метаданих: “Fwd Header Len”, “Dst Port”, “Timestamp”;

видалення колонок в яких значення виключно 0: “Bwd PSH Flags”, “Fwd URG Flags”, “CWE Flag Count”, “Fwd Byts/b Avg”, “Fwd Pkts/b Avg”, “Fwd Blk Rate Avg”, “Bwd Pkts/b Avg”, and “Bwd Blk Rate Avg”;

видалення значеннь “NaN” і заміна їх на 0 видалення “+inf”, “-inf”;

видалення від’ємних значень todo: put a table with the data entries before and after cleaning with % of valid data.

Нормалізація даних

Після очищення датасету, необхідно провести нормалізацію даних. Нормалізація - це процес масштабування числових даних, щоб всі значення знаходилися в межах одного діапазону. Це важливий крок у підготовці даних для машинного навчання, оскільки алгоритми машинного навчання часто працюють краще з нормалізованими даними. В цій роботі для нормалізації даних використано метод один з найрозповсюдженіших методів - мінімально-

максимальне масштабування (min-max scaling) яке масштабує функцію до фіксованого діапазону [0, 1] шляхом віднімання мінімального значення функції ($x_{\min}$) з поточного значення (x), а потім розділення результату на діапазон, як це показано у формулі (1)

$$X_{new} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (1)$$

Балансування даних. Багато алгоритмів машинного навчання працюють краще, коли набір даних рівномірно збалансований. В нашому випадку, це означає, що кількість прикладів нормального трафіку і ботнет трафіку повинна бути приблизно однаковою. Якщо один клас даних представлений у вибірці значно частіше, ніж інший, це може призвести до перенавчання моделі.

Зменшення розміру датасету. Зменшення розміру датасету включає застосування технік, які дозволяють зменшити кількість параметрів у датасеті без значної втрати інформації. Це може бути досягнуто за допомогою методів, таких як аналіз основних компонент (PCA), автоматичне кодування або відбір параметрів. Ці методи дозволяють зменшити обсяг даних, але при цьому зберегти важливу інформацію для навчання моделей машинного навчання.

Розділення даних на тренувальні та тестові набори. Одним з критично важливих кроків у процесі підготовки датасету для машинного навчання є розділення даних на тренувальний та тестовий набори. Тренувальний набір даних використовується для тренування моделі, тоді як тестовий набір даних використовується для перевірки якості та ефективності моделі.

Існують різні пдходи для розподілення датасету: 80/20, 70/30, 50/50 та ін. У даній роботі набір даних розбиваєтья у пропорції 80/20, тобто 80% буде використано для навчання і 20% для тестування.

5. Зниження розмірності

Вибір ознак. Найбільш підходящими методами вибору характеристик для попередньої обробки набору даних CSE-CIC-IDS2018 для виявлення ботнетів, ймовірно, будуть комбінація фільтруючих методів, методів обгортки та вбудованих методів. Фільтруючі методи, такі як хі-квадратний тест, ANOVA та коефіцієнт кореляції, можуть використовуватися для ідентифікації та видалення нерелевантних характеристик на основі їх статистики.

Методи обгортки, такі як рекурсивне виключення характеристик, послідовний вибір характеристик та генетичні алгоритми, використовують модель машинного навчання для оцінки комбінації характеристик і зберігають найкращі з них. Вбудовані методи, як-от LASSO, Elastic Net та Ridge Regression, включають вибір характеристик як частину процесу навчання моделі. Оскільки набір даних CSE-CIC-IDS2018 є високовимірним, важливо зменшити його розмірність. Вищезазначені методи можуть допомогти визначити найбільш релевантні характеристики для завдання виявлення ботнетів. Однак вибір методів повинен керуватися конкретними вимогами до завдання машинного навчання та характером даних.

6. Вибір алгоритмів класифікації

KNN (K-найближчі сусіди). Алгоритм *k*-найближчих сусідів (KNN) є алгоритмом машинного навчання, який використовується для класифікації та регресії. Він відноситься до типу “лінивого навчання”, оскільки не будує експліцитну модель під час навчання, а замість цього зберігає всі навчальні приклади і використовує їх безпосередньо під час прогнозування [8]. Основна ідея KNN полягає в тому, що виходи нового примера прогнозуються на основі виходов його *k* найближчих сусідів в просторі вхідних параметрів. “Близькість” зазвичай вимірюється за допомогою таких метрик, як евклідова відстань або манхеттенська відстань. При використанні KNN для класифікації, вихід нового прикладу є класом, який є найбільш поширеним серед *k* найближчих сусідів. При використанні KNN для регресії, вихід нового прикладу є середнім виходом *k* найближчих сусідів. Одним з ключових параметрів в KNN є число сусідів *k*. Число *k* впливає на міру гладкості прогнозування: менше значення *k* веде до більш гнучкого, але менш стабільного прогнозування, тоді як більше

© Гайдур, Г. І., Гахов, С. О., & Гамза, Д. Є. (2024). Модель виявлення шкідливої активності в інформаційній системі організації на основі гібридної класифікації. Сучасний захист інформації, 4(60), 30–38. https://doi.org/10.31673/2409-7292.2024.040003.

значення k веде до більш стабільного, але менш гнучкого прогнозування. Хоча KNN може бути дуже ефективним для певних задач, він також може бути досить повільним при великих обсягах даних, оскільки потребує обчислення відстаней між кожним новим прикладом і всіма навчальними прикладами.

Алгоритм роботи KNN складається з наступних етапів:

Етап ініціалізації. На етапі ініціалізації KNN просто зберігає весь навчальний набір даних.

Етап обчислення. На етапі обчислення відстаней, за потреби в прогнозі для нової точки даних KNN обчислює відстань між новою точкою даних і кожною існуючою точкою даних у навчальному наборі. Відстань може бути обчислена різними методами, наприклад, Евклідова відстань, яка обраховується за (2):

$$d(x,y) = \sqrt{\sum_{i-1}^{n}(x_i - y_i)^2}, \tag{2}$$

де x_i та y_i - це вектори ознак нової та існуючої точок даних відповідно, а n - кількість ознак.

Етап вибору K найближчих сусідів. Після обчислення відстаней KNN вибирає K найближчих сусідів до нової точки даних на основі обчислених відстаней. K - це гіперпараметр, зазвичай вибирається як непарне число, щоб уникнути нерішучих голосувань.

Етап більшості голосів (класифікація) або середнє значення (регресія). У випадку класифікації KNN використовується метод більшості голосів. Тобто, клас, який найчастіше зустрічається серед K найближчих сусідів, призначається новій точці даних. У випадку регресії, KNN використовує середнє значення цільових значень K найближчих сусідів як прогноз для нової точки даних.

Метод опорних векторів. Метод опорних векторів (SVM) - це потужний та гнучкий алгоритм машинного навчання, призначений для розрізнення і регресії. Основна ідея SVM полягає в пошуку гіперплощини, яка найкраще розділяє дані на класи [9]. SVM використовує так званий “ядерний трюк/метод”, що дозволяє ефективно виконувати обчислення в високовимірному просторі, навіть коли дані вихідного простору не є лінійно роздільними. Ядро може бути будь-якою функцією, яка відповідає умовам ядра. Найчастіше використовуються лінійне, поліноміальне, радіально- базисне функціональне (RBF) та сигмоїдні ядра [10].

Опорні вектори – це дані, які знаходяться найближче до гіперплощини. Ці точки визначають гіперплощину, і вони є "опорними векторами". Відстань від цих опорних векторів до гіперплощини називається “маржою”. SVM намагається максимізувати цю маржу, щоб створити якомога більше простору між класами. SVM має декілька ключових параметрів, включаючи параметр *C*, який контролює відхилення (тобто дозволяє або запобігає помилкам класифікації на тренувальних даних) та параметри ядра. Наприклад, у випадку RBF-ядра два ключових параметри – це *gamma* (який контролює ширину гауссової функції, що використовується як ядро) та *C*. Однією з переваг SVM є те, що він може забезпечити дуже хорошу точність класифікації, особливо на невеликих або середніх наборах даних і з високою розмірністю вхідних параметрів. Однак вони можуть бути відносно повільними для навчання, особливо на великих наборах даних, і їх може бути складно налаштувати через потребу вибору відповідних параметрів ядра та *C*. Алгоритм роботи SVM складається з наступних етапів:

Етап пошуку опорних векторів та гіперплощини. Основна мета SVM - знайти оптимальну гіперплощину, яка найкращим чином розділяє класи даних. Гіперплощина задається рівнянням (3):

$$w \times x + b = 0 \tag{3}$$

де w - вектор ваги, x - вектор ознак, а b - зсув (bias).

© Гайдур, Г. І., Гахов, С. О., & Гамза, Д. Є. (2024). Модель виявлення шкідливої активності в інформаційній системі організації на основі гібридної класифікації. Сучасний захист інформації, 4(60), 30–38. https://doi.org/10.31673/2409-7292.2024.040003.

Етап максимізації маржі. SVM шукає оптимальну гіперплощину, яка максимізує маржу між опорними векторами різних класів. Маржа обчислюється як відстань між гіперплощиною і найближчими до неї точками кожного класу. Мета - знайти вектор ваг w та зсув b, що максимізують маржу.

Етап визначення функції втрат і оптимізація. SVM використовує функцію втрат для знаходження оптимальної гіперплощини. Функція втрат (зазвичай опорні вектори мають нульову втрату) визначається як (4):

$$L(w,b) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\varepsilon_i \quad (4)$$

де ε_i – змінна ослаблення, а C – параметр регуляризації, який контролює баланс між маржою та помилками класифікації.

Етап оптимізації через розв'язок задачі квадратичного програмування (QP): Задача оптимізації SVM може бути сформульована як задача квадратичного програмування і вирішується числовими методами оптимізації для знаходження оптимальних параметрів w і b.

Етап застосування ядерних методів. SVM може використовувати ядерні методи для проекції даних у вищерозмірний простір, де вони можуть бути краще розділені. Функція відстані в просторі ознак може бути виражена через ядерну функцію (5):

$$K(x_i, x_j) = \varphi(x_i)\cdot\varphi(x_j), \quad (5)$$

де $K(x_i, x_j)$ - ядерна функція, а $\varphi(x)$ - функція відображення вищерозмірного простору.

Випадковий ліс (Random Forest). Випадковий ліс – це алгоритм машинного навчання, який використовується для задач класифікації, регресії та інших. Він включає в себе створення великої кількості дерев рішень під час навчання та потім використання режиму середнього прогнозу цих дерев для прогнозування [11].

Алгоритм роботи Random Forest складається з наступних етапів:

Етап побудови дерев рішень. Кожне дерево T_i у лісі будується на випадковій підмножині даних D_i, яка вибирається за допомогою бутстрепу, тобто з повтореннями. Для кожного вузла дерева випадковим чином вибирається підмножина ознак для розгалуження.

Етап побудови дерев рішень. Для кожного вузла дерева j, вибирається найкращий розбиття на підмножини $S_{j,m}$ та $S_{j,l}$ на основі певного критерію (наприклад, ентропії або невизначеності Джині). Критерій, що використовується для розбиття, зазвичай максимізує вибіркову інформацію G або мінімізує помилкову кількість класифікації.

Етап голосування більшості. Після побудови всіх дерев, для класифікації нового вхідного вектору x, всі дерева голосують за клас, який є найчастіше представлений серед їх прогнозів. Формально, прогноз $\hat{y}$ для вхідного вектору x обчислюється як (5):

$$\hat{y} = mode\left(T_1(x), T_2(x), \ldots, T_n(x)\right), \quad (5)$$

де $T_i(x)$ – прогноз i-го дерева для вектора x, а $mode$ – функція, що повертає найбільш часто елемент який зустрічається.

Однією з переваг випадкового лісу є його гнучкість. Він може використовуватись для вирішення різних задач, а саме класифікацію, регресію та інше. Крім того, він може обробляти дані з великою кількістю параметрів та категоріальні параметри.

Результат дослідження. Отже, для проведення експериментальних досліджень пропонується наступна модель оцінки ефективності виявлення шкідливої активності в

інформаційній системі організації, яка для виконання поставленої задачі буде реалізована на основі гібридної класифікації (Рис.2).

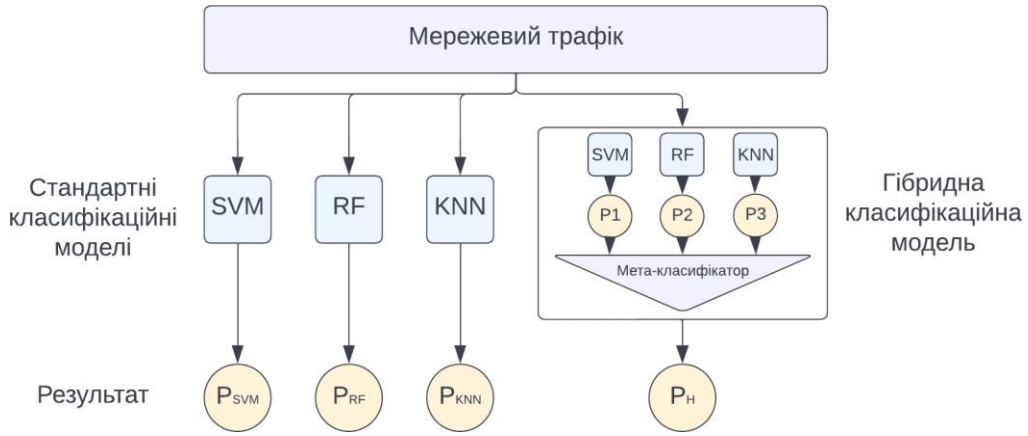

Рис. 2. Модель гібридної класифікації виявлення шкідливої поведінки

Висновок

Головною відмінністю даної моделі є те, що для оцінки ефективності роботи гібридної класифікаційної результат її прогнозів буде порівнюватися з результатами моделі на основі кожного окремого взятого алгоритму з наведених вище (Рис 2). Очікуваний результат дослідження повинен підвищити якість та точність виявлення шкідливих процесів в інформаційній системі організації.

Перелік посилань:

1. Check Point Research Reports. Check Point Blog. https://blog.checkpoint.com/2023/01/05/38-increase-in-2022-global-cyberattacks.
2. Гайдур, Г. І., Гахов, С. О., Дмітрієв, В. Є., & Бондаренко, Н. В. (2021). Виявлення аномалій трафіку в інформаційних системах організацій з використанням методів Machine Learning на основі алгоритмів прогнозування категорійних полів. 1., (4), 41–53. DOI: 10.31673/2412-4338.2021.044153
3. Гайдур, Г. І., Гахов, С. О., & Марченко, В. В. (2022). Method for constructing a dynamic model of a logical object of the information system and determining the law of its functioning. Radioelectronic and Computer Systems, 0(1), 129–140. doi: 10.32620/reks.2022.1.10.
4. Mohammed, A., & Kora, R. (2023). A comprehensive review on ensemble deep learning: Opportunities and challenges. Journal of King Saud University - Computer and Information Sciences, 35(2), 757–774. doi: 10.1016/j.jksuci.2023.01.014
5. Ganaie, M. A., Hu, M., Malik, A. K., Tanveer, M., & Suganthan, P. N. (2022). Ensemble deep learning: A review. Eng. Appl. Artif. Intell., 115, 105151. doi: 10.1016/j.engappai.2022.105151
6. Wizards, D. S. (2023). Guide to Simple Ensemble Learning Techniques - Data Science Wizards - Medium. Medium. Retrieved from https://medium.com/@datasciencewizards/guide-to-simple-ensemble-learning-techniques-2ac4e2504912.
7. IDS 2018 | Datasets | Research | Canadian Institute for Cybersecurity | UNB. (2023, December 21). Retrieved from https://www.unb.ca/cic/datasets/ids-2018.html .
8. Zhang, Y., Cao, G., Wang, B., & Li, X. (2019). A novel ensemble method for k-nearest neighbor. Pattern Recognit., 85, 13–25. doi: 10.1016/j.patcog.2018.08.003.
9. Cervantes, J., Garcia-Lamont, F., Rodríguez-Mazahua, L., & Lopez, A. (2020). A comprehensive survey on support vector machine classification: Applications, challenges and trends. Neurocomputing, 408, 189–215. doi: 10.1016/j.neucom.2019.10.118.
10. Roy, A., & Chakraborty, S. (2023). Support vector machine in structural reliability analysis: A review. Reliab. Eng. Syst. Saf., 233, 109126. doi: 10.1016/j.ress.2023.109126.
11. Breiman, L. (2001). Random Forests. Machine Learning, 45(1), 5–32. doi: 10.1023/A:1010933404324.

Надійшла 26.11.2024