МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ТЕХНОЛОГІЙ ТА ДИЗАЙНУ

ФАКУЛЬТЕТ МЕХАТРОНІКИ ТА КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
(повна назва факультету/інституту)

КАФЕДРА КОМП'ЮТЕРНИХ НАУК
(повна назва випускової кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на тему:

Застосування імовірнісного класифікатора Байєса для виявлення несанкціонованих доступів до комп'ютерних мереж

Рівень вищої освіти другий (магістерський)
(перший (бакалаврський) / другий (магістерський))

Спеціальність 122 Комп'ютерні науки
(код і найменування спеціальності)

Спеціалізація (за наявності) _____
(код і найменування спеціальності)

Освітня програма Комп'ютерні науки
(назва освітньої програми)

Виконав(-ла): студент(-ка) МгІТ-2-22

Путієнко В.Р.
(прізвище та ініціали)

Науковий керівник д.ф.-м.н.,проф.
Краснитський С.М.
(науковий ступінь, вчене звання, прізвище та ініціали)

Рецензент к.т.н. доц. Яхно В.М.
(науковий ступінь, вчене звання, прізвище та ініціали)

Київ 2023

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ТЕХНОЛОГІЙ ТА ДИЗАЙНУ

Факультет / Інститут _____Факультет мехатроніки та комп'ютерних технологій_____
(повна назва факультету/інституту)

Кафедра_____Кафедра комп'ютерних наук_____
(повна назва кафедри)

Рівень вищої освіти _____другий (магістерський)_____
(перший (бакалаврський) / другий (магістерський))

Спеціальність _____122 Комп'ютерні науки_____
(код і назва спеціальності)

Спеціалізація _____
(код і назва спеціальності)

Освітня програма _____Комп'ютерні науки_____
(назва освітньої програми)

ЗАТВЕРДЖУЮ
Завідувач кафедри ____КН_____
(абревіатура кафедри)
_____ Володимир ЩЕРБАНЬ
(підпис) (Власне ім'я та ПРІЗВИЩЕ)
«_____» _____ 2023 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

_____Путієнко Віталію Ростиславовичу_____
(прізвище, ім'я, по батькові студента)

1. Тема кваліфікаційної роботи ___Застосування імовірнісного класифікатора Байєса для виявлення несанкціонованих доступів до комп'ютерних мереж_____

Науковий керівник роботи_____Краснитський Сергій Михайлович, д.ф.-м.н.,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
професор,_____
_____

затверджені наказом КНУТД від «12» вересня 2023 року № 210-уч

2. Вихідні дані до кваліфікаційної роботи__Розробки кафедри комп'ютерних комп'ютерних наук, рекомендована література, додатки._____
_____

3. Зміст кваліфікаційної роботи (перелік питань, які потрібно опрацювати)
____Вступ, Розділ 1. Аналіз предметної області; Розділ 2. Алгоритмічне_____ забезпечення; Розділ 3. Програмне забезпечення; Додатки._____
_____
_____
_____

Дата видачі завдання _____01.08.2023____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапу кваліфікаційної роботи	Орієнтовний термін виконання	Примітка про виконання
1	Вступ	12.08.2023	Підпис керівника і студента
2	Розділ 1. Аналіз предметної області	14.09.2023	Підпис керівника, студента та консультанта
3	Розділ 2. Алгоритмічне забезпечення	18.09.2023	Підпис керівника, студента та консультанта
4	Розділ 3. Програмне забезпечення	17.10.2023	Підпис керівника, студента та консультанта
5	Висновки	19.10.2023	Підпис керівника і студента
6	Оформлення (чистовий варіант)	25.10.2023	Підпис керівника і студента
7	Подача кваліфікаційної роботи науковому керівнику для відгуку (за 14 днів до захисту)	12.11.2023	Підпис керівника і студента
8	Подача кваліфікаційної роботи для рецензування (за 12 днів до захисту)	14.11.2023	Підпис керівника, студента консультанта, рецензента
9	Перевірка кваліфікаційної роботи на наявність ознак плагіату (за 10 днів до захисту)	16.11.2023	Підпис керівника і студента на підставі довідки фахівця відділу моніторингу якості підготовки фахівців та аналітичної роботи
10	Подання кваліфікаційної роботи на завідувачу кафедри (за 7 днів до захисту)	19.11.2023	Підпис завідувача кафедри

З завданням ознайомлений:

Студент(-ка) _____ Віталій ПУТІЄНКО
(підпис) (Власне ім'я та ПРІЗВИЩЕ)

Науковий керівник _____ Сергій КРАСНИТСЬКИЙ
(підпис) (Власне ім'я та ПРІЗВИЩЕ)

АНОТАЦІЯ

Путієнко Віталій. Застосування імовірнісного класифікатора Байєса для виявлення несанкціонованих доступів до комп’ютерних мереж.

Дипломна магістерська робота за спеціальністю 122 - «Комп’ютерні науки».– Київський національний університет технологій та дизайну, Київ, 2023 рік.

Дипломна робота присвячена дослідженню та розробці системи виявлення несанкціонованих доступів до комп'ютерної мережі з використанням методів машинного навчання.

У роботі проаналізовано існуючі підходи до виявлення вторгнень, розглянуто можливості застосування наївного байєсівського класифікатора для класифікації мережевого трафіку. Розроблено програмне забезпечення системи на основі мови Python з використанням бібліотек Pandas, NumPy, Scikit-Learn. Для навчання та тестування моделі використовувався набір даних NSL-KDD. Побудовано класифікатор на основі наївного байєсівського підходу з попереднім обиранням ознак.

Показана можливість застосування обраного підходу для виявлення несанкціонованих доступів на основі аналізу мережевого трафіку

ANNOTATION

Putiienko Vitalii. Application of the Bayesian probability classifier for detecting unauthorized access to computer networks.

Master's thesis in specialty 122 - "Computer Science". - Kyiv National University of Technology and Design, Kyiv, 2023.

The thesis is devoted to the research and development of a system for detecting unauthorized access to a computer network using machine learning methods.

The paper analyzes existing approaches to intrusion detection, considers the possibilities of using the Naive Bayes classifier to classify network traffic. The software of the system was developed using the Python language with the Pandas, NumPy, Scikit-Learn libraries. The NSL-KDD dataset was used for training and testing the model. A classifier based on the Naive Bayes approach with pre-selection of features has been built.

The possibility of applying the chosen approach for detecting unauthorized accesses based on the analysis of network traffic is shown.

ЗМІСТ

ВСТУП

Зі збільшенням залежності від комп'ютерних мереж для різних аспектів нашого життя забезпечення безпеки та цілісності цих мереж стало критичною проблемою. Кібератаки та мережеві вторгнення становлять серйозну загрозу для організацій, урядів і окремих осіб, що призводить до витоку даних, фінансових втрат і збоїв у роботі послуг. Системи виявлення вторгнень (IDS) відіграють вирішальну роль у виявленні та пом'якшенні таких загроз, надаючи додатковий рівень захисту для комп'ютерних мереж.

Системи виявлення вторгнень спрямовані на виявлення та реагування на несанкціоновані дії в мережі. Традиційно IDS покладалися на методи на основі правил або сигнатур, де попередньо визначені правила або сигнатури використовуються для ідентифікації відомих моделей атак. Незважаючи на те, що ці підходи ефективні проти відомих атак, вони часто не можуть виявити нові та розвиваються методи атак, які не відповідають існуючим сигнатурам. Це обмеження вимагає розробки більш досконалих і адаптивних методів, які можуть ефективно виявляти аномалії та класифікувати мережевий трафік як звичайний або шкідливий.

Алгоритми машинного навчання привернули значну увагу в області виявлення вторгнень завдяки своїй здатності автоматично вивчати шаблони та виявляти аномалії в мережевому трафіку. Ці алгоритми можуть аналізувати величезні обсяги даних і виявляти шаблони, які можуть свідчити про вторгнення або ненормальну поведінку. Використовуючи машинне навчання, системи виявлення вторгнень можуть адаптуватися до нових моделей атак і підвищити точність виявлення.

Одним із популярних алгоритмів машинного навчання, який використовується для завдань класифікації, є наївний класифікатор Байєса.

Наївні класифікатори Байєса — це імовірнісні моделі, які базуються на теоремі Байєса, припускаючи, що ознаки є умовно незалежними з урахуванням мітки класу. Незважаючи на свою простоту та припущення, наївні класифікатори Байєса продемонстрували багатообіцяючі результати в різних областях, включаючи категоризацію тексту, фільтрацію спаму та медичну діагностику. Їх ефективність, масштабованість і здатність обробляти масиви даних великого розміру роблять їх привабливим вибором для виявлення вторгнень у комп'ютерні мережі.

Ефективність систем виявлення вторгнень значною мірою залежить від їх здатності точно й ефективно ідентифікувати зловмисну діяльність у мережевому трафіку. Крім того, на продуктивність систем виявлення вторгнень впливають такі фактори, як різноманітність і складність мережевих протоколів, обсяг мережевого трафіку та швидкість, з якою відбуваються атаки. Це вимагає дослідження алгоритмів машинного навчання, які можуть впоратися з проблемами, властивими мережевим даним, і забезпечити своєчасне й точне виявлення вторгнень.

Метою цього дослідження є вивчення використання наївного класифікатору Байєса як підходу машинного навчання для виявлення вторгнень у комп'ютерні мережі. Цей класифікатор відомий своєю простотою та масштабованістю, а дослідження показали хороші результати в різних завданнях класифікації.

Це дослідження має на меті зробити внесок в вдосконалення методів виявлення вторгнень і підвищити безпеку комп'ютерних мереж. Відомості, отримані в результаті цього дослідження, можуть допомогти в розробці більш надійних і адаптивних систем виявлення вторгнень, здатних ефективно і точно ідентифікувати та пом'якшувати ризики мережевих вторгнень.

Розділ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Введення в машинне навчання

Машинне навчання - це галузь штучного інтелекту, цілью якої є навчити комп'ютери виконувати складні завдання без прямого програмування. В основі машинного навчання лежать статистичні методи, які дозволяють алгоритмам вдосконалюватися на основі даних, а не жорстких правил.

З точки зору науки, машинне навчання - це методологія, що включає моделювання, налаштування параметрів, підготовку даних та оптимізацію для досягнення конкретної мети. Метою машинного навчання як дослідницького підходу є знаходження оптимальних рішень та прогнозів на основі наявних даних. Машинне навчання базується на індуктивному підході, де алгоритми "вчаться" на конкретних прикладах пар "вхідні дані - вихід", а потім застосовують набуті знання до нових даних. Тобто машинне навчання полягає у тому, щоб навчити комп'ютерну систему аналізувати великі обсяги реальних даних, будувати моделі закономірностей, тестувати їх та використовувати для подальших прогнозів[3].

Існують чотири основні підходи в машинному навчанні: кероване навчання, некероване навчання, частково кероване навчання та навчання з підкріпленням [32].

Кероване навчання, відоме також як навчання з вчителем, визначається використанням позначених наборів даних для навчання алгоритмів класифікації даних чи прогнозування результатів. Під час подачі вхідних даних модель коригує свої вагові коефіцієнти, допоки не буде адекватно підлаштована. Це відбувається як частина процесу крос-валідації для уникнення перенавчання чи недонавчання. Кероване навчання допомагає організаціям вирішувати

різноманітні реальні задачі великого масштабу, наприклад, класифікувати спам в окрему папку від вашої пошти. Деякі методи, що використовуються при навчанні з вчителем, включають нейронні мережі, наївний байєсів класифікатор, лінійну регресію, логістичну регресію, Random forest та метод опорних векторів.

Некероване навчання використовує алгоритми машинного навчання для аналізу та кластеризації непозначених наборів даних. Ці алгоритми виявляють приховані закономірності або групування даних без необхідності втручання людини. Здатність цього методу виявляти подібності та відмінності в інформації робить його ідеальним для дослідницького аналізу даних, стратегій крос-продажів, сегментації клієнтів та розпізнавання зображень і патернів. Звичайні підходи для цього - метод головних компонент та сингулярне розкладання. Інші алгоритми некерованого навчання включають нейронні мережі, кластеризацію k-середніх та ймовірнісні методи кластеризації.

Частково кероване навчання є компромісом між керованим та некерованим навчанням. Під час навчання воно використовує менший позначений набір даних для керування класифікацією та виділення ознак з більшого непозначеного набору даних. Частково кероване навчання може вирішити проблему недостатньої кількості позначених даних для алгоритму керованого навчання. Воно також корисне, якщо позначити достатню кількість даних занадто проблематично.

Навчання з підкріпленням - це модель машинного навчання, схожа на навчання з вчителем, але алгоритм не навчається на зразках даних. Ця модель вчиться методом спроб і помилок у процесі роботи. Послідовність успішних результатів закріплюється, щоб розробити найкращу рекомендацію чи стратегію для конкретної задачі.

Машинне навчання широко застосовується в різних сферах, включаючи комп'ютерний зір, обробку природної мови, рекомендаційні системи, біоінформатику тощо. У цій роботі розглядається застосування імовірнісного класифікатора Байєса, одного з популярних алгоритмів навчання з вчителем, для виявлення несанкціонованих доступів до комп'ютерних мереж.

1.2 Огляд методів виявлення несанкціонованого доступу

Під процесом виявлення атак розуміється оцінка подій та інформаційних потоків інформаційної системи (ІС) шляхом аналізу журналів реєстрації операційних систем і додатків, а також мережевого трафіку.

Поточні підходи до виявлення вторгнень можна широко класифікувати на дві категорії: виявлення аномалій та виявлення зловживань(сигнатурний підхід).

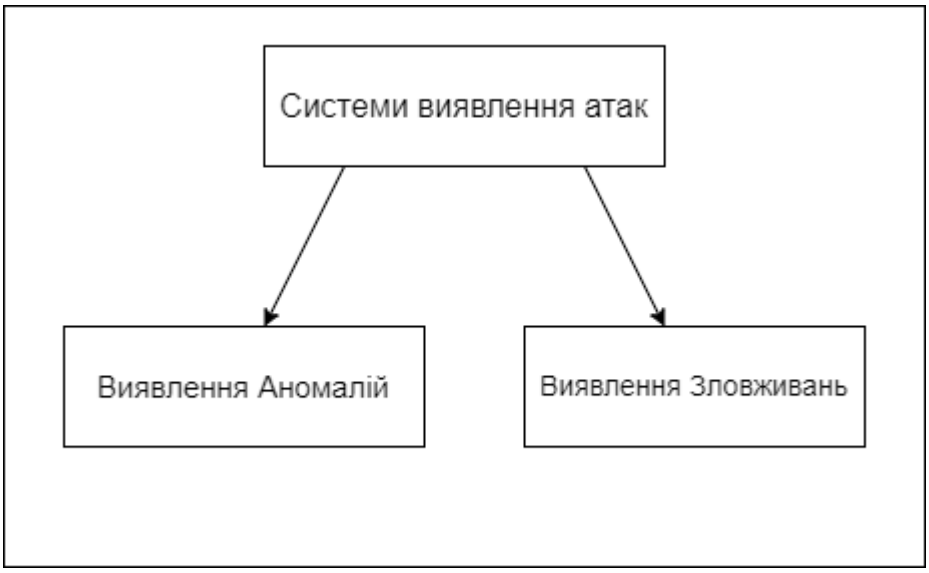

Рис. 1.1

Виявлення аномалій ґрунтується на припущенні, що вторгнення часто проявляється як аномалія. Зазвичай тут використовують показники, що свідчать про вторгнення, та виявляють статистично великі відхилення на цих показниках. Прикладами можуть бути надзвичайно велика кількість мережевих

з'єднань протягом певного інтервалу часу, надзвичайно висока активність процесора або використання пристроїв, які зазвичай не використовуються. Цей підхід був широко досліджений та реалізований в багатьох системах. Він намагається кількісно оцінити припустиму поведінку та виявляти аномальну поведінку як вторгнення.

Інша техніка виявлення вторгнень, виявлення на основі сигнатур, намагається закодувати знання про атаки у чітко визначені шаблони і відстежує випадки виникнення цих шаблонів. Наприклад, зловживання вразливостей fingerd і sendmail, які були використані в атакі Інтернет-черв'яка, відносяться до цієї категорії. Ця техніка конкретно представляє знання про неприпустиму поведінку та намагається виявити її випадки[25].

Ключова особливість виявлення зловживань полягає у тому, що воно оперує заздалегідь вивченим і добре відомим зразком небезпечних дій, які можуть стати частиною вторгнення. Цей набір знань про відомі атаки називається базою сигнатур. Зазвичай ця база оновлюється регулярно для врахування нових типів атак або вразливостей, що виявляються з часом[11].

Коли система виявлення вторгнень працює в режимі виявлення зловживань, вона постійно аналізує активність системи та мережі, шукаючи відповідність між цією активністю та відомими сигнатурами атак. Якщо вона виявляє шаблон атаки в потоці даних, спрацьовує тривога, що попереджає про можливе вторгнення або високий ризик безпекового порушення.

Проте, виявлення на основі зловживань має свої недоліки. Воно просте і точне, але не може виявити нові типи атак чи нові варіанти відомих атак. Отже, IDS не здатні їх розпізнати. Іншими недоліками є велика кількість хибних спрацювань. Хибно позитивне спрацювання буває, коли нормальну активність помилково класифікують як шкідливу. А хибно негативне - коли шкідливу як

нормальну. Виявлення зловживань потребує частого оновлення сигнатур для якісного виявлення. А методи на основі аномалії мають високий рівень хибних спрацювань.

Переваги підходу на основі сигнатур полягають у високій точності виявлення відомих атак, так як система працює на основі чітко визначених правил та шаблонів. Крім того, відсутність аномалій у зареєстрованій активності може зменшити кількість помилкових спрацювань тривог.

Підсумовуючи, сигнатурні методи мають низький рівень хибних спрацьовувань, але високий рівень пропуску атак. Аномальні методи навпаки схильні до хибних спрацьовувань, але менш імовірно пропустять реальну атаку.

1.3 Поняття IDS

Система виявлення вторгнень, відома також як СВВ (Intrusion Detection System, IDS), є системою, яка автоматизує процес аналізу подій в інформаційно-комунікаційній системі з метою забезпечення її безпеки. В сучасному світі ці системи вважається невід'ємною складовою інфраструктури безпеки, що має вирішальне значення[1].

Систему виявлення вторгнень можна описати на дуже макроскопічному рівні як детектор, що обробляє інформацію, яка надходить із системи, що захищається (рис. 1.2). Він також може запускати операції для запуску процесу аудиту, наприклад, запит номерів версій програм може бути використаний для отримання інформації про версії програмного забезпечення, які використовуються в системі. Це може бути корисно для виявлення застарілих версій програм, для яких відомі вразливості, або для ідентифікації потенційних проблем у забезпеченні безпеки. Цей детектор використовує три види інформації: довгострокову інформацію, пов'язану з технікою, що використовується для виявлення вторгнень (база знань про атаки, наприклад),

конфігураційну інформацію про поточний стан системи та опис інформації аудиту подій, які відбуваються в системі[13].

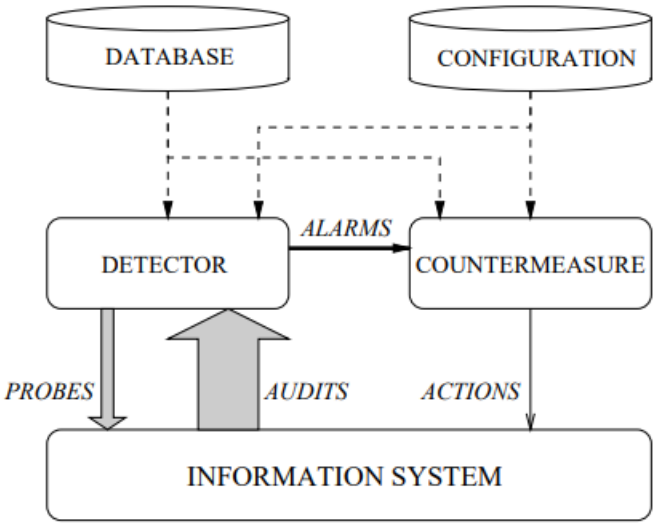

Рис. 1.2 – Архітектура IDS

Щоб подолати обмеження систем на основі сигнатур, було введено IDS на основі аномалій. Ці системи встановлюють базову лінію нормальної поведінки мережі та виявляють відхилення від цієї базової лінії як потенційні вторгнення. IDS на основі аномалій використовують методи статистики, машинного навчання або інтелектуального аналізу даних для аналізу моделей мережевого трафіку та виявлення аномалій. Цей підхід дозволяє виявляти невідомі та нові атаки, але також може призвести до хибних спрацьовувань через природну складність визначення того, що є «нормальною» поведінкою.

Аномальна активність може виявлятися через різноманітні ознаки, такі як істотне збільшення кількості з'єднань за короткий період, високе використання ресурсів центрального процесора та мережі, або активність на периферійних пристроях, які зазвичай залишаються невикористаними. Імовірно, будь-яке відхилення від типового профілю поведінки суб'єкта можна вважати

аномальним. Аномальна поведінка не завжди має на меті зловмисницькі дії або атаку[26].

Операція сучасних систем виявлення вторгнень з метою забезпечення інформаційної безпеки інформаційних систем полягає в розв'язанні наступних основних завдань:

- Аналіз активності в захищеній ІС з метою виявлення ознак, що можуть свідчити про спробу атаки або незвичайної активності в ІС.
- Виявлення ворожих дій або незвичайної активності та ідентифікація типу атаки або аномалії, якщо такі з'явилися.
- Прийняття рішення щодо блокування атаки або аномалії, яке може бути автоматичним або здійснюватися за допомогою уповноважених осіб.
- Внесення змін до ІС з метою унеможливлення подібних атак на майбутнє.
- Заблокування атак або аномалій після їхнього виявлення.
- Повідомлення уповноважених суб'єктів ІС про інцидент, якщо таке повідомлення є необхідним[2].

Гібридні IDS поєднують сильні сторони підходів на основі сигнатур і аномалій. Використовуючи як попередньо визначені сигнатури, так і методи виявлення аномалій, гібридні IDS прагнуть підвищити точність виявлення та зменшити помилкові спрацьовування. Ці системи пропонують більш комплексний та адаптивний підхід до виявлення вторгнень, поєднуючи переваги як методів на основі сигнатур, так і на основі аномалій.

1.4 Класифікація систем

Сфера систем виявлення вторгнень стикається з декількома проблемами, включаючи зростаючу складність атак, високий обсяг і швидкість мережевого трафіку, необхідність виявлення в реальному часі та появу зашифрованого або обфускованого мережевого трафіку. Дослідники та практики вирішують ці

проблеми, досліджуючи передові методи, такі як машинне навчання, інтелектуальний аналіз даних, штучний інтелект і глибоке навчання, щоб підвищити точність і ефективність виявлення вторгнень.

Крім того, інтеграція аналізу загроз, алгоритмів виявлення аномалій і методів аналізу поведінки покращила можливості IDS. Ця інтеграція дозволяє IDS співвідносити та контекстуалізувати різні події, виявляти складні моделі атак і надавати точніші сповіщення та відповіді.

Класифікація системи виявлення вторгнень - це поділ на категорії системи виявлення атак (вторгнень), які призначені для виявлення фактів несанкціонованого доступу в комп'ютерну систему або мережу. Існують чотири основні категорії IDS:

1. Система виявлення вторгнень у мережі (NIDS)
2. Система виявлення вторгнень на хост-рівні (HIDS)
3. Віртуальна система виявлення вторгнень (VMIDS)
4. Система виявлення несанкціонованого фізичного доступу (PIDS)

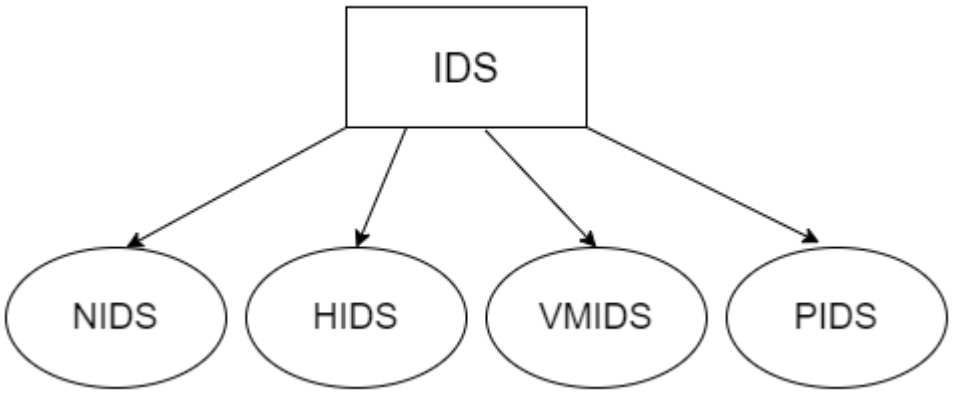

Рис. 1.3

NIDS (Network Intrusion Detection System) - це система виявлення вторгнень у мережі. Вона використовується для виявлення та контролю небажаних або шкідливих активностей, що відбуваються в комп'ютерних мережах.

Ця система працює, аналізуючи мережевий трафік та шукаючи ознаки, що можуть вказувати на атаку або ненормальну поведінку. Система аналізує патерни мережевого трафіку на вміст підозрілих активностей, таких як спроби несанкціонованого доступу. Коли NIDS виявляє підозрілу активність, система сповіщає адміністратора мережі або виконує певні автоматичні заходи для забезпечення безпеки[31]. NIDS відстежують мережевий трафік у стратегічних точках комп’ютерної мережі, таких як маршрутизатори чи комутатори.

NIDS добре підходять для виявлення атак на рівні мережі, таких як сканування портів, атаки на відмову в обслуговуванні (DoS) або зараження шкідливим програмним забезпеченням у мережі. NIDS може бути реалізований як апаратний пристрій або програмне забезпечення, яке встановлюється на сервер або спеціальну мережеву приставку. Вона може працювати в режимі реального часу, постійно моніторячи мережевий трафік, або аналізувати записані дані для подальшого виявлення вторгнень[10].

HIDS (Host-based Intrusion Detection System) - це система виявлення вторгнень на хост-рівні. Вона встановлюється безпосередньо на окремому комп'ютері або сервері і працює для виявлення аномалій та небажаної активності на самому хості. HIDS аналізує певну поведінку хоста (на рівні кінцевих точок), включаючи те, які програми використовуються, до яких файлів здійснюється доступ і яка інформація зберігається в журналах ядра. NIDS може ідентифікувати зловмисника до того, як він встигне здійснити зловживання, тоді як HIDS діє як другий рівень захисту і вживає заходів на рівні кінцевих точок, якщо система зламана [30].

Основна мета HIDS полягає в ранньому виявленні атак та вторгнень на окремий хост. Вона може використовувати різноманітні методи для виявлення вторгнень, такі як аналіз сигнатур (порівняння з попередньо відомими шаблонами атак), виявлення аномалій (виявлення незвичайної активності) або комбінацію обох. Після виявлення підозрілої активності HIDS може сповістити адміністратора системи або вжити заходів для запобігання атаки, наприклад, блокування підключень, зупинка підозрілих процесів або виконання інших заходів безпеки.

VMIDS (virtual machine-based intrusion detection system) - це віртуальна система виявлення вторгнень. Вона використовується для виявлення атак або небажаної активності в віртуальних середовищах, таких як віртуальні машини або хмарні інфраструктури.

Система виявлення вторгнень на основі віртуальних машин (VMIDS) схожа на одну або комбінацію будь-яких з двох вищезазначених IDS, але впроваджується віддалено через віртуальну машину. Це найновіший з чотирьох типів IDS, який все ще вдосконалюється. Більшість постачальників IT-послуг використовують VMIDS. VMIDS менш інтрузивні, ніж традиційні системи IDS, оскільки їх можна впровадити без фізичного візиту постачальника. Вони потенційно мають краще покриття, ніж будь-яка з трьох інших IDS, але можуть викликати деякі проблеми, якщо інтернет-з'єднання переривається[29].

VMIDS забезпечує аналіз трафіку, активності та подій в віртуальному середовищі з метою виявлення потенційних загроз безпеці. Вона спостерігає за діяльністю віртуальних машин, мережевими з'єднаннями, змінами в конфігурації та іншими аспектами віртуального середовища.

Основна мета VMIDS полягає в ідентифікації аномальної або підозрілої активності в віртуальних інфраструктурах. Вона використовує методи аналізу

сигнатур або виявлення аномалій для розпізнавання зловмисницьких дій, вторгнень або порушень безпеки. Після виявлення підозрілої активності VMIDS може сповістити адміністратора системи, інтегруватись з системою керування віртуальними середовищами або вжити заходів для негайного реагування на загрозу. Це можуть бути блокування мережевих з'єднань, ізоляція або припинення роботи віртуальних машин, зміна конфігурації або виконання інших заходів безпеки.

VMIDS є важливим елементом безпеки віртуальних середовищ, оскільки вони дозволяють виявляти та реагувати на вторгнення в межах віртуальної інфраструктури. Вони допомагають забезпечити безпеку віртуальних машин, даних та додатків, що розміщені в таких середовищах.

PIDS (Physical Intrusion Detection System) - це система, яка використовується для виявлення несанкціонованого фізичного доступу до обмежених зон та складається з різних датчиків, таких як датчики руху, магнітні контакти, вібраційні, акустичні, датчики нахилу, які реагують на рух, вібрацію, звук, магнітні поля та інші фізичні явища. При спрацюванні цих датчиків система видає сигнал тривоги та може активувати інші системи безпеки.

Системи фізичного виявлення вторгнень призначені для виявлення можливих загроз фізичної безпеки в області обмеженого доступу. Ці системи включають в себе різні компоненти, такі як відеоспостереження, пристрої обмеження доступу до дверей, датчики руху та датчики розбиття скла. Зазвичай вони можуть діяти самостійно або взаємодіяти з іншими рішеннями для попередження фізичних вторгнень, наприклад, з системами блокування вхідних дверей[17].

Датчики можуть бути активними і пасивними, система має один або декілька рівнів виявлення. Для передачі даних використовуються дротові та бездротові

інтерфейси до центрального контролера, який оповіщає оператора. Дані також можуть передаватися в хмарні системи моніторингу. Система інтегрується з іншими системами безпеки та використовує спеціальні чутливі кабелі. Вона застосовується для захисту периметру різних об'єктів та запобігання несанкціонованому доступу.

Переваги та недоліки різних IDS:

Тип IDS	**Переваги**	**Недоліки**
NIDS (мережеві)	- Моніторинг всієї мережі - Виявлення зовнішніх атак - Розгортання на стратегічних точках	- Потребують сенсорів на ключових точках - Можуть пропустити внутрішні атаки
HIDS (хостові)	- Моніторинг окремого хоста - Виявлення внутрішніх атак - Аналіз детальної активності	- Потребують встановлення на кожен хост - Обмежена видимість мережі
VMIDS (віртуальні)	- Захист віртуальної інфраструктури - Масштабованість	- Відсутність фізичного доступу - Складність налаштування
PIDS (фізичні)	- Висока надійність - Раннє виявлення - Простота встановлення	- Обмежена зона охоплення - Локальне

		спрацьовування - Вразливість до саботажу

1.5 Сучасні дослідження в області машинного навчання для виявлення несанкціонованого доступу

Існують різні методи та системи, які допомагають виявити несанкціонований доступ. Вони базуються на аналізі порядку використання інформаційних ресурсів, що захищають, та здатні здійснювати певні самостійні дії з виявлення, ідентифікації і усунення причин, що їх викликали.

Багато дослідників працюють над створенням систем виявлення вторгнень на основі машинного навчання. Поширеним підходом є використання нейронних мереж для класифікації трафіку як нормального або аномального. При цьому мережі навчаються на великих наборах даних, щоб надалі розпізнавати відхилення від норми.

Інші методи включають в себе машинне навчання на основі графів для виявлення зв'язків між пристроями, кластеризацію для групування схожих зразків даних, а також алгоритми посилення для навчання систем на малих наборах даних. Крім аналізу мережевого трафіку, дослідження також ведуться в області аналізу журналів подій та поведінкового аналізу користувачів для виявлення ознак вторгнення.

У 2012 році Аріф Джамал Малік та інші дослідники представили систему виявлення вторгнень, яка базується на використанні методу Random Forest та оптимізаційного алгоритму PSO. Для вибору відповідних ознак для класифікації вторгнень використовується бінарний PSO. Алгоритм Random Forest використовується як класифікатор. Їхній метод складається з двох основних

кроків: вибір ознак і подальша класифікація. Цей метод був реалізований авторами за допомогою середовища MATLAB [7].

Метод дерева рішень у завданні класифікації або прогнозування полягає у поділі вихідних даних на групи досягнення однорідності (або майже однорідності) їх підмножин. Загальні правила, які випливають з такого розділення, дозволяють прогнозувати цільову змінну, використовуючи оцінку вхідних ознак для нових даних (предикторів).

Іньїго Лопес-Ріобо Ботана, Карлос Ейрас-Франко та Ампаро Альонсо-Бетансос у своїй статті[8] розглянули новий підхід до пояснення алгоритму виявлення аномалій (ADMNC - Anomaly Detection in Mixed Numerical and Categorical Input Spaces) на основі дерев регресії. Робота пропонує покращений алгоритм, який використовує формулювання моделі ADMNC для надання попереднього пояснення на основі дерева класифікації та регресії. Пояснення представляється у вигляді сегментації вхідних даних на однорідні групи, які можна описати декількома змінними, що надає нову інформацію для обґрунтування рішень. Результати експериментів наведено на реальних великих наборах даних, зосереджуючись на області виявлення вторгнень у мережу.

У праці "Identify Features and Parameters to Devise an Accurate Intrusion Detection System Using Artificial Neural Network " автори показують, що точність та ефективність IDS можна підвищити шляхом отримання хороших навчальних параметрів та вибору правильних ознак для проектування будь-якої штучної нейронної мережі [24].

Також було запропоновано динамічну модель "Інтелектуальна система виявлення вторгнень", засновану на специфічному підході для виявлення вторгнень. Використовуються техніки нейронних мереж та нечіткої логіки з моніторингом мережі, що використовують прості методи Data Mining для

обробки мережевих даних. Система поєднує виявлення аномалій та зловживань. Нечіткі правила типу "якщо-то" дають можливість абстрактно описати атаки безпеки. [21].

Висновки до розділу 1

Існують різні підходи до виявлення вторгнень, включаючи методи на основі сигнатур, виявлення аномалій та їх комбінації. Кожен підхід має свої переваги і недоліки. Для підвищення ефективності активно застосовуються методи машинного навчання та інтелектуального аналізу даних, що дозволяє краще виявляти невідомі загрози. Розрізняють різні типи систем виявлення вторгнень: мережеві, хостові, віртуальні та фізичні, кожна зі своїми особливостями. Для аналізу даних та класифікації трафіку застосовуються різноманітні моделі машинного навчання, такі як нейронні мережі, дерева рішень, кластеризація тощо. З розвитком технологій з'являються нові види кібератак, тому потрібні подальші дослідження для вдосконалення методів виявлення та запобігання вторгнень.

РОЗДІЛ 2 АЛГОРИТМІЧНЕ ЗАБЕЗПЕЧЕННЯ

2.1 Опис наївного байєсівського класифікатора

Одним з підходів до виявлення несанкціонованого доступу є використання методів машинного навчання, зокрема імовірнісних класифікаторів. Вони дозволяють аналізувати великі обсяги даних про активність в мережі та виявляти аномальну поведінку, що може вказувати на спробу атаки.

Класифікація є фундаментальною проблемою в машинному навчанні та інтелектуальному аналізі даних. Метою в класифікації є побудова класифікатора на основі набору навчальних прикладів з мітками класів. Зазвичай, приклад E представлено у вигляді кортежу значень атрибутів $(x_1, x_2, ..., x_n)$, де x_i - значення атрибуту X_i. Нехай C позначає змінну класифікації, а c - її значення. Припускається, що є лише два класи: + (позитивний клас) або – (негативний клас)[33].

Класифікатор - це функція, що призначає мітку класу прикладу. З ймовірнісної точки зору, відповідно до правила Байєса, ймовірність прикладу $E = (x_1, x_2, ..., x_n)$, що належить класу c, дорівнює

$$p(c \mid E) = \frac{p(E \mid c)p(c)}{p(E)}$$

E класифікується як клас $C = +$ тоді і лише тоді, коли

$$f_b(E) = \frac{p(C = + \mid E)}{p(C = - \mid E)} \geq 1$$

де $f_b(E)$ є байєсівським класифікатором.

Припустимо, що всі атрибути незалежні за умови заданого значення змінної класу; тобто,

$$p(E \mid c) = p(x_1, x_2, \dots, x_n \mid c) = \prod_{i=1}^{n} p(x_i|c)$$

Тоді отримуємо класифікатор:

$$f_{nb}(E) = \frac{p(C = +)}{p(C = -)} \prod_{i=1}^{n} \frac{p(x_i|C = +)}{p(x_i|C = -)}$$

Функція $f_{nb}(E)$ називається наївним байєсівським класифікатором, або просто наївний Байєс. Рисунок 2.1 показує приклад наївного Байєса. В наївному Байєсі кожен вузол атрибуту не має батьківських вузлів, окрім вузла класу[33].

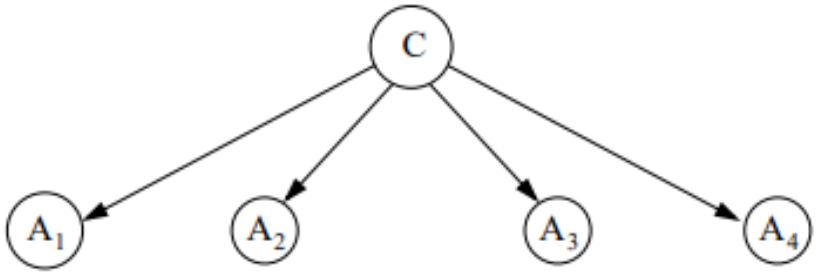

Рис. 2.1

Іншими словами багато класифікаторів можна розглядати як обчислення набору дискримінантних функцій прикладу, по одній для кожного класу, і віднесення прикладу до того класу, функція якого є максимальною. Якщо E - приклад, а $f_i(E)$ - дискримінантна функція, що відповідає i-му класу, обраний клас C_k - це клас, для якого

$$f_k(E) > f_i(E) \; \forall \; i \neq k$$

Припустимо, що приклад є вектором атрибутів, як це зазвичай буває у програмах класифікації. Нехай v_{jk} - це значення атрибута A_j у прикладі, $P(X)$ позначає ймовірність X, а $P(Y \mid X)$ позначає умовну ймовірність Y для X. Тоді одним з можливих наборів дискримінантних функцій є[14]:

$$f_i(E) = P(C_i) \prod_{j=1}^{a} P(A_j = v_{jk} | C_i)$$

Класифікатор, отриманий за допомогою цього набору дискримінантних функцій та оцінки відповідних ймовірностей з навчальної вибірки є наївним байєсівським класифікатором.

Прості байєсівські класифікатори набули популярності останнім часом і, як виявилося, демонструють досить високу ефективність. Ці ймовірнісні підходи базуються на сильних припущеннях щодо генерації даних та формулюють ймовірнісну модель, яка відображає ці припущення. Далі вони використовують колекцію позначених прикладів для оцінки параметрів генеративної моделі. Класифікація нових прикладів здійснюється за допомогою теореми Байєса, обираючи клас, який найбільш ймовірно згенерував приклад.

Наївний байєсівський класифікатор є найпростішою з цих моделей, оскільки він припускає, що всі атрибути прикладів незалежні одне від одного при умові класу. Це так зване "наївне припущення Байєса". Хоча це припущення очевидно неправдиве для більшості завдань реального світу, наївний байєсівський класифікатор часто добре справляється з класифікацією. Цей парадокс пояснюється тим, що оцінка класифікації є лише функцією знаку (у випадку бінарних випадків) оцінки функції; апроксимація функції може бути низькою, але точність класифікації залишається високою. Завдяки припущенню про незалежність, параметри для кожного атрибута можна вивчати окремо, що

суттєво спрощує процес навчання, особливо коли кількість атрибутів велика[19].

Наївний байєсівський класифікатор був використаний для класифікації текстів, аналізу тональності, фільтрації спаму та інших задач, де потрібно визначити ймовірність приналежності спостереження до одного з класів. Він був вибраний через свою простоту, швидкість та ефективність.

Імовірнісний класифікатор Байєса застосовується у різних областях з метою класифікації об'єктів на кілька категорій або класів. Основними цілями використання наївного байєсівського класифікатора є:

- Класифікація тексту - наприклад, віднесення новин, електронних листів чи документів до певних категорій. Наївний класифікатор Байєса добре працює з текстовими даними.
- Виявлення спаму - класифікація електронних листів як спам чи не спам.
- Аналіз тональності тексту - визначення емоційного забарвлення тексту, наприклад позитивний чи негативний відгук.
- Діагностика захворювань на основі симптомів.
- Розпізнавання образів - класифікація зображень за вмістом.
- Виявлення кібератак - в контексті кібербезпеки, наївні байєсівські класифікатори можуть застосовуватися для виявлення аномальних дій та вразливостей в комп'ютерних мережах.
- Рекомендаційні системи - передбачення рейтингу користувача для продукту на основі його попередніх даних.

Плюси та мінуси імовірнісного класифікатора Байєса:

Переваги	**Недоліки**

Простота побудови моделі	Сильна залежність від припущення про незалежність ознак
Швидкість навчання	Може давати слабкі результати, якщо припущення про незалежність порушено
Ефективність класифікації нових прикладів	Чутливість до шумових даних
Масштабованість - працює з великими даними	
Не вимагає багато даних для навчання	
Інтерпретованість моделі	
Можливість додавання нових ознак без перенавчання	

2.2 Типи імовірнісних класифікаторів Байєса

Існують чотири основні типи наївних байєсівських класифікаторів: гаусівський, біноміальний, мультиноміальний та доповнювальний.

Наївний байєсівський класифікатор Гаусса (Gaussian Naive Bayes) - це один з методів наївного байєсовського класифікатора, який використовує нормальний (гаусівський) розподіл для моделювання ймовірностей класів з неперервними ознаками[20].

Ймовірність ознак вважається Гаусовою:

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right)$$

Параметри σ_y та μ_y оцінюються з використанням методу максимальної імовірності.

Основна ідея цього методу полягає у тому, що вважається, що значення кожної ознаки в кожному класі розподілені незалежно за гаусівським законом. Тобто, для кожного класу вимірювання ознаки розподілені за нормальним розподілом, характеризованим середнім значенням і стандартним відхиленням.

Наївний байєсовський класифікатор Гаусса є ефективним методом, особливо в ситуаціях, коли дані мають неперервний характер. Він широко використовується в багатьох областях, включаючи розпізнавання образів, фінансовий аналіз, біомедичні дослідження та інші.

Цей метод підходить для виявлення несанкціонованих доступів в мережі, якщо ознаки мають неперервний характер. Наприклад, можна використовувати цей метод, якщо ознаки включають числові значення, такі як пропускна здатність мережі, швидкість передачі даних, часові інтервали тощо. За допомогою гаусівського розподілу можна моделювати розподіл цих ознак і використовувати ймовірності для класифікації об'єктів, виявлення аномалій або несанкціонованого доступу.

Біноміальний наївний байєсовський класифікатор (Bernoulli Naive Bayes) - це метод наївного байєсовського класифікатора, який використовує біноміальний розподіл для моделювання ймовірностей класів з бінарними ознаками. У біноміальному наївному байєсовському класифікаторі вважається, що кожна ознака є бінарною з двома можливими значеннями, такими як "присутність" або "відсутність".

Цей метод реалізує наївні алгоритми навчання та класифікації Байєса для даних, розподілених відповідно до багатовимірних розподілів Бернуллі; тобто, ознак може бути декілька, але кожна з них вважається двійковою (Бернуллі,

булевою) змінною. Тому цей клас вимагає, щоб вибірки були представлені у вигляді двійкових векторів ознак; якщо на вхід подаються будь-які інші дані, то клас може бінаризувати вхідні дані (залежно від параметра бінаризації)

Правило прийняття рішення для наївного Байєса Бернуллі базується на

$$P(x_i \mid y) = P(x_i = 1 \mid y)x_i + (1 - P(x_i = 1 \mid y))(1 - x_i)$$

При класифікації тестового документа модель Бернуллі використовує бінарну інформацію про наявність термінів, ігноруючи кількість входжень, в той час як мультиноміальна модель враховує кількісні входження. В результаті, модель Бернуллі часто припускається багатьох помилок при класифікації великих документів. Наприклад, вона може віднести всю книгу до класу "Китай" через єдине входження терміну "Китай"[18].

Мультиноміальний наївний байєсовський класифікатор (Multinomial Naive Bayes) реалізує наївний байєсівський алгоритм для мультиноміально розподілених даних і є одним з двох класичних варіантів наївного Байєса, що використовуються в класифікації текстів. Розподіл параметризується векторами $\theta_y = (\theta_{y1}, \ldots, \theta_{yn})$для кожного класу y, де n - кількість ознак (у класифікації текстів - розмір словника), а θ_{yi} - ймовірність $P(x_i \mid y)$появи ознаки і в зразку, що належить до класу y [20].

Параметри оцінюється за допомогою згладженої версії максимальної правдоподібності, тобто підрахунку відносної частоти:

$$\hat{\theta}_{yi} = \frac{N_{yi} + \alpha}{N_y + \alpha n}$$

де $N_{yi} = \sum_{x \in T} x_i$ - кількість разів, коли ознака і з'являється у вибірці класу y в навчальній множині T, а $N_y = \sum_{i=1}^{n} N_{yi}$ - загальна кількість всіх ознак для класу.

Мультиноміальний наївний байєсовський класифікатор часто використовується як базовий метод в класифікації текстів, оскільки він швидкий і простий у реалізації. Більше того, за умови відповідної попередньої обробки, він конкурентний із більш просунутими методами, включаючи машини опорних векторів. Однак, цей класифікатор у стандартному вигляді не є повністю байєсівським. Принаймні, не в тому сенсі, що апостеріорний розподіл параметрів оцінюється за навчальними документами, а потім використовується для прогнозного висновку для нового документа[27].

Доповнювальний наївний Байєсівский класифікатор (Complement Naive Bayes) особливо підходить для роботи з незбалансованими наборами даних. В доповнювальному наївному байєсівському класифікаторі, замість обчислення ймовірності належності елемента певному класу, алгоритм обчислює ймовірність належності елемента всім класам.

Для кожного класу обчислюємо ймовірність того, що даний екземпляр не належить до нього. Після обчислення для всіх класів, ми перевіряємо всі обчислені значення і вибираємо найменше. Найменше значення (найнижча ймовірність) вибирається, тому що це найнижча ймовірність того, що він не належить до цього конкретного класу. Це означає, що він має найвищу ймовірність насправді належати до цього класу. Отже, цей клас і вибирається. Зокрема, класифікатор використовує статистику з доповнення кожного класу для обчислення ваг моделі. Процедура розрахунку ваг виглядає наступним чином:

$$\hat{\theta}_{ci} = \frac{\alpha_i + \sum_{j:y_j \neq c} d_{ij}}{\alpha + \sum_{j:y_j \neq c} \sum_k d_{kj}}$$

$$w_{ci} = \log \hat{\theta}_{ci}$$

$$w_{ci} = \frac{w_{ci}}{\sum_j |w_{cj}|}$$

де суми беруться по всіх документах j, що не належать до класу x, d_{ij} - або кількість входжень, або tf-idf значення терміну i в документі j, α_i - гіперпараметр згладжування, подібний до того, що використовується в Мультиноміальному наївному байєсівському класифікатору(MNB), $\alpha = \sum_i \alpha_i$. Друга нормалізація вирішує проблему домінування довших документів в оцінках параметрів MNB. Правило класифікації має вигляд:

$$\hat{c} = \arg \min_c \sum_i t_i w_{ci}$$

Тобто, документ віднесено до класу, який має найгіршу відповідність доповнень[22].

Вибір гаусовського методу для виявлення вторгнень на основі датасету трафіку можна обґрунтувати декількома аспектами. По-перше, гаусівський метод підходить для даних з безперервними характеристиками, які часто присутні у датасетах трафіку, такими як пропускна здатність, розміри пакетів, тривалість пакетів тощо. Використання гаусового розподілу дозволяє апроксимувати ці характеристики та виявляти аномалії, що можуть вказувати на вторгнення. По-друге, гаусівський метод може моделювати нормальну поведінку в мережі та виявляти відхилення від цього профілю. Шляхом побудови базового профілю нормального трафіку на основі статистики він може виявляти незвичайну або аномальну активність, що може свідчити про вторгнення. По-третє, гаусівський метод гнучкий і дозволяє враховувати

взаємозв’язки між різними характеристиками датасету трафіку. Це особливо важливо, оскільки взаємозв’язки між характеристиками можуть впливати на виявлення аномалій. Гаусівський метод може аналізувати ці взаємозв’язки і допомагати точніше виявляти аномалії. Також, гаусівський метод є широко вивченим і добре підтримуваним, що спрощує його використання та впровадження. Існує багато готових реалізацій цього методу, що полегшує його застосування та дослідження в контексті виявлення несанкціонованих доступів в мережі.

2.3 Обирання ознак

Побудова ефективних моделей машинного навчання вимагає ретельного аналізу та підготовки даних, що використовуються для навчання. Одним з ключових етапів цього процесу є обирання ознак, або feature selection. У багатьох задачах машинного навчання вихідний набір даних може містити велику кількість ознак, деякі з яких є надлишковими або неінформативними для моделі. Використання таких ознак не тільки уповільнює навчання, але і може призвести до перенавчання та зниження якості моделі на нових даних. Відбір ознак дозволяє вирішити цю проблему шляхом зменшення розмірності даних та вибору найбільш значущих предикторів для конкретної задачі моделювання.

Обирання ознак - це процес, який обирає підмножину ознак з початкового набору ознак таким чином, що простір ознак оптимально зменшився згідно з певним критерієм. Існує три основні класи алгоритмів обирання ознак: фільтровий метод(Filter methods), обгортковий метод(wrapper methods) та вбудований метод(embedded methods) (рис. 2.2).

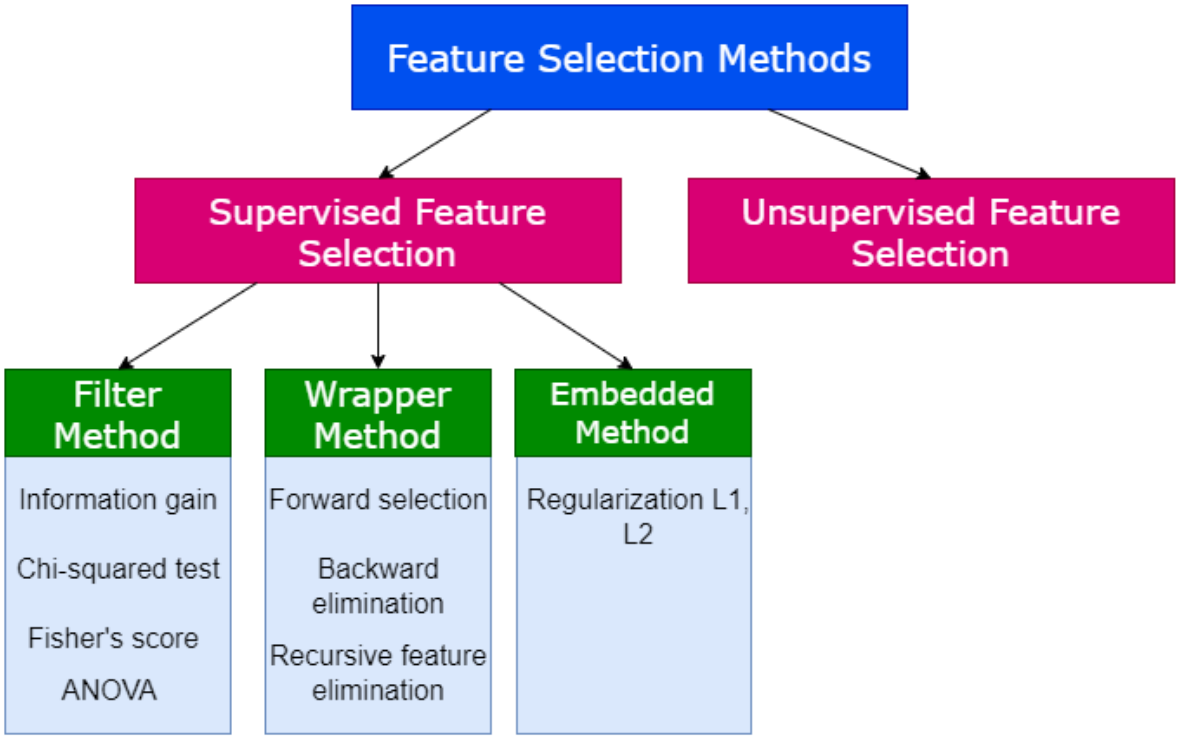

Рис. 2.2

Алгоритм відбору ознак можна розглядати як комбінацію методу пошуку для пропонування нових підмножин ознак разом із мірою оцінки, яка оцінює різні підмножини ознак. Ідеально, метод відбору ознак буде перебирати всі можливі підмножини комбінацій ознак, які можна отримати з даного набору даних, і знаходити комбінацію ознак, яка забезпечує найкращу продуктивність моделі машинного навчання. На практиці це зазвичай не є можливим через обчислювальні витрати. Крім того, різні підмножини ознак можуть забезпечувати оптимальну продуктивність для різних алгоритмів машинного навчання. Це означає, що існує не лише одна оптимальна підмножина ознак, а потенційно багато оптимальних підмножин залежно від алгоритму машинного навчання, який буде використаний. Тому протягом років було розроблено

багато різних методів відбору ознак, щоб охопити якомога більше застережень та обмежень.

Фільтровий метод оцінює релевантність ознак на основі внутрішніх характеристик даних. Ця категорія є етапом попередньої обробки, який не залежить від алгоритмів машинного навчання.. Процес фільтрації складається з двох ключових етапів. Для створення моделі класифікації спочатку відбувається індивідуальне ранжування ознак на основі певної критеріальної міри, таких як кореляція Пірсона та ентропія [5]. По-друге, вона відбирає ознаки з найкращим рейтингом, використовуючи порогове значення. Решта ознак вважаються непотрібними та неінформативними. Після цього класифікатор отримує на вхід відібрану підмножину ознак[9].

Рис. 2.3 Фільтровий метод обирання ознак

Деякі методи, що використовуються в фільтрації [15]:

- Інформаційний приріст - це кількість інформації, яку надає ознака для ідентифікації цільового значення, та вимірює зменшення ентропії. Інформаційний приріст кожної ознаки обчислюється з урахуванням цільових значень для відбору ознак.

- Критерій Хі-квадрат - це статистичний тест, який застосовується до груп категоріальних ознак для оцінки ймовірності кореляції або асоціації між ними за допомогою їхнього частотного розподілу.
- Критерій Фішера відбирає кожну ознаку незалежно згідно з її балами за критерієм Фішера, що призводить до субоптимального набору ознак. Чим вищий критерій Фішера, тим краще обрана ознака.
- Коефіцієнт кореляції Пірсона є мірою кількісної оцінки зв'язку між двома безперервними змінними та напрямком зв'язку зі значеннями від -1 до 1.
- Аналіз дисперсії (ANOVA) - це інструмент аналізу, який використовується в статистиці та ділить спостережувану сукупну змінність всередині набору даних на дві частини: систематичні фактори та випадкові фактори. Систематичні фактори мають статистичний вплив на даний набір даних, в той час як випадкові фактори - ні. Аналітики використовують тест ANOVA, щоб визначити вплив незалежних змінних на залежну змінну в дослідженні регресіїх[6].

У методі обгорткового обирання ознак, виходячи із певного алгоритму машинного навчання, використовується підхід, який адаптується до конкретних наборів даних. Цей метод оперує за допомогою "жадібного" аналізу, оцінюючи всі можливі комбінації ознак на основі обраного критерію оцінки. Сам критерій оцінки визначається відповідно до призначення завдання; для задач регресії, наприклад, він може використовувати показники якості, такі як p-значення, R-квадрат, скоригований R-квадрат, тоді як для задач класифікації - точність, влучність, повнота, f1-міра і т. д. На завершення, метод вибирає оптимальну комбінацію ознак, яка забезпечує найкращі результати для конкретного алгоритму машинного навчання.

На практиці, обгорткові методи часто використовують такі підходи:

- Внесення ознак (Forward selection)
- Усунення ознак (Backward elimination)
- Рекурсивне виключення ознак

Рис. 2.3 Обгортковий метод обирання ознак

Метод внесення ознак є ітераційним підходом, де ми починаємо з порожнього набору змінних і продовжуємо додавати змінну, яка найкраще покращує нашу модель, після кожної ітерації. Критерієм зупинки є те, що додавання нової змінної не покращує продуктивність моделі.

При методі усунення ознак ми починаємо з повної моделі (включаючи всі незалежні змінні), та після кожної ітерації видаляємо найменш значущу ознаку. Критерієм зупинки є відсутність покращення продуктивності моделі після видалення ознаки[4].

Також в обгортковому методі застосовується рекурсивне виключення ознак (RFE) - цей метод "жадібної" оптимізації відбирає ознаки, рекурсивно розглядаючи все менший і менший набір ознак. Іноді метод використовують, щоб пояснити деяку кількість "найважливіших" ознак, що впливають на результати; а іноді для зменшення дуже великої кількості змінних (приблизно 200-400), і залишають тільки ті, які роблять хоч якийсь внесок у модель, а всі інші виключаються. Оцінювач навчається на початковому наборі ознак, а їх важливість визначається за допомогою атрибуту feature_importance_attribute.

Потім найменш важливі ознаки видаляються з поточного набору ознак, поки не залишиться необхідна кількість ознак.

Крім сортування ознак, RFE може показати, чи є ці ознаки важливими навіть для заданої кількості ознак. Оскільки вибір конкретної кількості ознак може бути неоптимальним, RFE допомагає визначити, чи дана кількість ознак варта уваги, або оптимальна кількість ознак може бути меншою або більшою, ніж задана.

Переваги обгорткових підходів полягають у взаємодії між процесом відбору підмножини ознак і вибором моделі, а також у можливості враховувати взаємозв'язки між ознаками. Однак загальним недоліком цих методів є підвищений ризик перенавчання порівняно з методами фільтрів, і вони вимагають значних обчислювальних ресурсів, особливо в разі побудови класифікатора з високою обчислювальною складністю.

У вбудованих методах пошук оптимальної підмножини ознак вбудовано в конструкцію класифікатора, і його можна розглядати як пошук в об'єднаному просторі підмножин ознак і гіпотез. Так само, як і обгорткові підходи, вбудовані підходи, таким чином, є специфічними для певного алгоритму навчання. Перевага вбудованих методів полягає в тому, що вони передбачають взаємодію з моделлю класифікації, але водночас є набагато менш обчислювально інтенсивними, ніж обгорткові методи[23].

Рис. 2.4 Вбудований метод обирання ознак

Основним методом із цієї категорії є регуляризація. Регуляризація додає штрафний член до різних параметрів моделі машинного навчання, щоб уникнути надмірного пристосування моделі. Цей штрафний член додається до коефіцієнтів; отже, він зменшує деякі коефіцієнти до нуля. Ці ознаки з нульовими коефіцієнтами можуть бути видалені з набору даних. Типи методів регуляризації - це L1 регуляризація (регуляризація Лассо) або еластичні мережі (L1 і L2 регуляризація). Таким чином, регуляризація - це своєрідний штраф за зайву складність моделі, який дає змогу захистити себе від перетренування в разі наявності непотрібних ознак[16].

Також застосовується методи на основі дерев - такі методи, як Random Forest і Gradient Boosting, також надають нам значення важливості ознак як спосіб вибору ознак. Значення важливості ознак вказує нам, які ознаки мають більший вплив на цільову ознаку.

Висновки до розділу 2

У розділі 2 було розглянуто алгоритмічне забезпечення для виявлення несанкціонованого доступу за допомогою методів машинного навчання.

Зокрема, був описаний наївний байєсівський класифікатор, який базується на застосуванні теореми Байєса для класифікації даних. Розглянуто різні види

наївних байєсівських класифікаторів, такі як гаусівський, біноміальний, мультиноміальний та доповнювальний. Показано, що саме гаусівський класифікатор найкраще підходить для аналізу даних мережевого трафіку, оскільки він дозволяє ефективно моделювати безперервні характеристики трафіку, такі як часові інтервали, обсяг пакетів тощо.

Також розглянуто важливий етап підготовки даних - обирання ознак. Описано основні методи: фільтровий метод, обгортковий метод та вбудовані методи. Показано, що для конкретної задачі виявлення вторгнень найкраще підходить саме фільтровий метод відбору ознак. Він дозволить відібрати найбільш значущі характеристики трафіку на основі статистичних критеріїв.

В розділі запропоновано використати гаусівський наївний байєсівський класифікатор з попереднім застосуванням фільтрового методу відбору ознак для ефективного виявлення несанкціонованого доступу на основі аналізу даних мережевого трафіку. Це дозволить побудувати точну модель для класифікації нормальної та аномальної поведінки в мережі.

Розділ 3. ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ

3.1 Засоби та методи

Для реалізації практичної частини дипломної роботи було використано наступне програмне забезпечення:

Мова програмування Python широко використовується для задач аналізу мережевого трафіку і класифікації мережевих подій з кількох причин. По-перше, Python - це високорівнева мова програмування з простим синтаксисом, що робить її доступною для широкого кола програмістів та дослідників. Велика спільнота розробників і наявність багатьох бібліотек, таких як scikit-learn і pandas, сприяють швидкому розробленню та ефективному аналізу даних мережевого трафіку. По-друге, Python має потужні бібліотеки для обробки даних, візуалізації та статистичного аналізу, що робить його ідеальним інструментом для підготовки та обробки даних перед класифікацією. Зручність роботи з даними у формі датафреймів дозволяє швидко і ефективно виконувати операції з даними, необхідні для підготовки вхідних даних для моделей класифікації.

Python має багатий вибір бібліотек для машинного навчання та інтелектуального аналізу даних, які дозволяють легко реалізувати і навчати моделі класифікації, включаючи наївний байєсівський класифікатор, як в даному випадку. Ця гнучкість і розширюваність робить Python ідеальним інструментом для розв'язання завдань аналізу мережевого трафіку та виявлення відмінностей між різними класами мережевої активності, такими як атаки та нормальний трафік.

Anaconda - це платформа для наукового обчислення та управління пакетами, яка дозволяє легко створювати і управляти віртуальними

середовищами Python та встановлювати необхідні бібліотеки. Вона забезпечує зручний спосіб управління залежностями та середовищами.

Scikit-learn є однією з ключових бібліотек для машинного навчання в Python. Вона надає набір класичних алгоритмів машинного навчання, включаючи наївний класифікатор Байєса, який був використаний у цьому дослідженні.

Для вибору ознак було застосовано метод SelectKBest з бібліотеки scikit-learn. SelectKBest реалізує процедуру фільтрації, дозволяючи обрати k найбільш значущих ознак на основі заданого статистичного критерію. У якості критерію було обрано ANOVA F-значення між ознакою та ціллю. між ознакою та цільовою змінною.

Бібліотека Pandas використовується для роботи з даними у форматі таблиць. Вона дозволяє завантажувати, обробляти та аналізувати дані з легкістю, що робить її незамінним інструментом для підготовки даних перед використанням моделей машинного навчання.

Numpy - це бібліотека для чисельних обчислень в Python. Вона надає підтримку для роботи з масивами та матрицями, що робить обчислення швидкими та ефективними.

Бібліотека Matplotlib використовуються для візуалізації даних та результатів дослідження. Matplotlib.pyplot дозволяє налаштовувати графіки докладніше.

3.2 Вибір та збір даних

Для навчання та тестування моделі класифікатора використовувався відомий набір даних NSL-KDD. Це удосконалена версія оригінального набору KDD Cup 99, який містить мережевий трафік з модельованими вторгненнями та атаками. NSL-KDD виправляє деякі істотні недоліки попереднього набору даних, зокрема дублювання записів у навчальній та тестовій вибірках, що призводило до завищеної оцінки якості алгоритмів машинного навчання.

Для зручності моделювання в NSL-KDD всі типи атак було об'єднано в один узагальнений клас "anomaly"[12]. Таким чином, задача зводиться до бінарної класифікації трафіку на нормальний і аномальний. Це спрощує побудову та оцінку моделей виявлення вторгнень[28].

Навчальна вибірка KDDTrain+ містить 125973 рядки даних, що описують мережеві з'єднання з 41 ознакою кожне. Ознаки характеризують базові характеристики з'єднання та виділені за допомогою експертних знань параметри, що дозволяють ідентифікувати вторгнення. Серед ознак - протокол, тривалість з'єднання, кількість даних переданих по TCP та UDP, кількість невдалих спроб логіну та інші.

Кожне з'єднання позначене одним з 23 класів - normal для нормального трафіку або одним з 22 типів атаки, таких як buffer overflow, port sweep, DoS (denial of service), які були об`єднанні в один клас. Це дозволяє будувати та тестувати моделі класифікації вторгнень.

Для побудови оптимальної моделі класифікатора навчальну вибірку KDDTrain+ було розбито на дві частини співвідношенням 80/20. Більша частина (100778 прикладів) використовувалася для навчання моделі, менша (25195 приклади) - для внутрішньої валідації та підбору оптимальних гіперпараметрів.

Такий розподіл даних на навчальну, та тестові вибірки є оптимальним підходом, що дозволяє побудувати якісну модель машинного навчання та

об'єктивно оцінити її здатність класифікувати раніше невідомі дані. Використання різних за складністю тестових наборів дає уявлення про роботу моделі в умовах, наближених до реальних.

Набір даних NSL-KDD є загальноприйнятим бенчмарком для тестування алгоритмів виявлення вторгнень. Незважаючи на деякі обмеження, пов'язані з віком цих даних, він дозволяє ефективно оцінювати якість та порівнювати різні підходи. Широке використання NSL-KDD в науковій літературі робить результати, отримані на цьому наборі, більш зрозумілими та порівнюваними. Тому для вирішення поставленого завдання - побудови та оцінки наївного класифікатора Байєса, набір даних NSL-KDD є оптимальним вибором, що поєднує репрезентативність, зручність та можливість порівняння з іншими методами машинного навчання для виявлення мережевих атак.

Датасет NSL-KDD є удосконаленою версією відомого набору даних KDD Cup 99, який широко застосовується в задачах виявлення вторгнень в комп'ютерні мережі. Він був розроблений в 2009 році групою дослідників з Університету Нью-Брансвіка для подолання деяких істотних недоліків оригінального KDD Cup 99.

Однією з головних проблем KDD Cup 99 була наявність великої кількості дублікатів записів в навчальній та тестовій вибірках. Це призводило до зміщення результатів, оскільки алгоритми машинного навчання отримували завищені показники точності на таких даних. Для вирішення цієї проблеми, в NSL-KDD були видалені всі дублікати, що дозволяє об'єктивно оцінювати ефективність різних методів. Ще однією важливою проблемою KDD Cup 99 був сильний дисбаланс класів - кількість прикладів нормального трафіку на порядки перевищувала кількість атак. Для NSL-KDD класи були збалансовані шляхом проріджування даних.

Набір даних NSL-KDD є одним з найбільш популярних наборів даних для навчання та тестування систем виявлення вторгнень. Набір даних широко використовується в наукових дослідженнях і в промисловості.

До переваг набору даних NSL-KDD відносяться:

- Набір даних є великим і різноманітним.
- Набір даних містить дані про різні типи атак.
- Набір даних широко використовується в наукових дослідженнях і в промисловості.

До недоліків набору даних NSL-KDD відносяться:

- Набір даних був створений у 2009 році, і він може не відображати сучасні загрози.
- Набір даних містить дані про атаки, які були сфальсифіковані.

Набір даних NSL-KDD використовується для навчання та тестування систем виявлення вторгнень. Системи виявлення вторгнень використовують набір даних для навчання алгоритмів, які можуть виявляти атаки. Набір даних NSL-KDD також використовується для наукових досліджень. Дослідники використовують набір даних для розробки нових алгоритмів виявлення вторгнень і для оцінки ефективності існуючих алгоритмів.

3.3 Архітектура та загальний опис системи

Система виявлення вторгнень базується на використанні наївного байєсівського класифікатора. Дані для навчання та тестування моделі беруться з відкритого набору даних NSL-KDD. Цей набір містить інформацію про мережевий трафік та відповідні мітки класів вторгнень або нормальної активності.

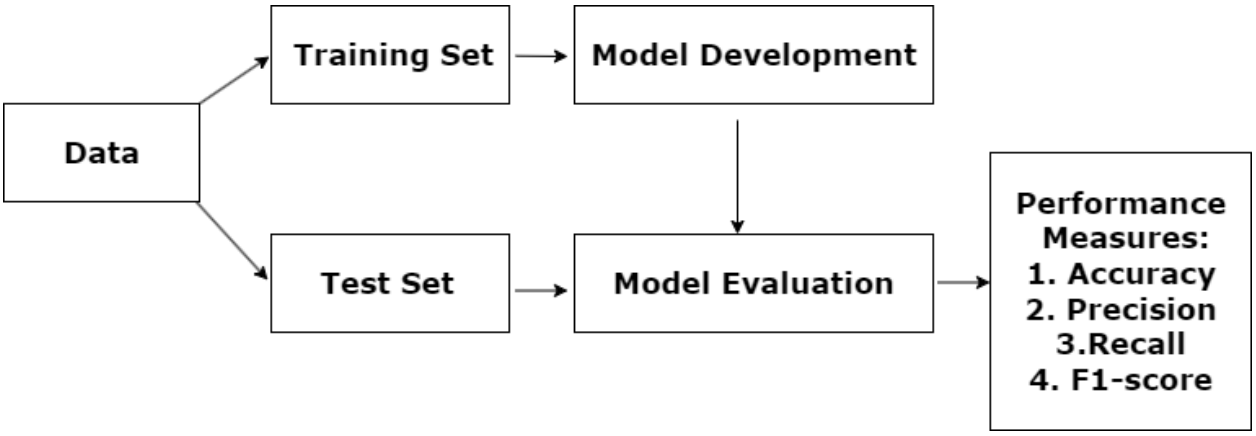

Рис. 3.1. Архітектура системи

Перед поділом даних на навчальну та тестову вибірки, виконується їх попередня обробка. Також застосовується метод відбору найбільш інформативних ознак для побудови оптимальної моделі класифікатора. Далі дані розбиваються на навчальну та тестову частини. На навчальній вибірці будується модель наївного байєсівського класифікатора за допомогою бібліотеки scikit-learn.

Після навчання модель тестується та оцінюється її точність. Також будується матриця невідповідностей для аналізу результатів. Розроблена система дозволяє класифікувати мережевий трафік за допомогою алгоритму машинного навчання та виявляти потенційні кібератаки.

3.4 Реалізація програмного коду

```python

def preprocess_dataset(dataset):

    protocol_type = {'tcp': 1, 'udp': 2, 'icmp': 3}
    flag = {'OTH': 1, 'REJ': 2, 'RSTO': 3, 'RSTOS0': 4, 'RSTR': 5, 'S0': 6, 'S1': 7, 'S2': 8, 'S3': 9, 'SF': 10, 'SH': 11}
    service = {'aol': 1, 'auth': 2, 'bgp': 3, 'courier': 4, 'csnet_ns': 5, 'ctf': 6, 'daytime': 7, 'discard': 8, 'domain': 9,
               'domain_u': 10, 'echo': 11, 'eco_i': 12, 'ecr_i': 13, 'efs': 14, 'exec': 15, 'finger': 16, 'ftp': 17,
               'ftp_data': 18, 'gopher': 19, 'harvest': 20, 'hostnames': 21, 'http': 22, 'http_2784': 23, 'http_443': 24,
               'http_8001': 25, 'imap4': 26, 'IRC': 27, 'iso_tsap': 28, 'klogin': 29, 'kshell': 30, 'ldap': 31, 'link': 32,
               'login': 33, 'mtp': 34, 'name': 35, 'netbios_dgm': 36, 'netbios_ns': 37, 'netbios_ssn': 38, 'netstat': 39,
               'nnsp': 40, 'nntp': 41, 'ntp_u': 42, 'other': 43, 'pm_dump': 44, 'pop_2': 45, 'pop_3': 46, 'printer': 47,
               'private': 48, 'red_i': 49, 'remote_job': 50, 'rje': 51, 'shell': 52, 'smtp': 53, 'sql_net': 54, 'ssh': 55,
               'sunrpc': 56, 'supdup': 57, 'systat': 58, 'telnet': 59, 'tftp_u': 60, 'tim_i': 61, 'time': 62, 'urh_i': 63,
               'urp_i': 64, 'uucp': 65, 'uucp_path': 66, 'vmnet': 67, 'whois': 68, 'X11': 69, 'Z39_50': 70}

    dataset.protocol_type = [protocol_type.get(item, 0) for item in dataset.protocol_type]
    dataset.flag = [flag.get(item, 0) for item in dataset.flag]
    dataset.service = [service.get(item, 0) for item in dataset.service]

    return dataset

```

Рис. 3.2

Функція preprocess_dataset (рис 3.2) призначена для попередньої обробки датасету перед побудовою та навчанням моделі машинного навчання. Вона виконує перетворення деяких категоріальних ознак з текстового представлення у числове, що є необхідним кроком для подальшого навчання моделі.

Зокрема, функція створює відображення для трьох ознак: protocol_type, flag та service. Ці ознаки є категоріальними, тобто містять обмежену кількість можливих текстових значень, таких як назви мережевих протоколів, прапорці TCP або назви мережевих сервісів. Щоб модель могла працювати з такими даними, їх необхідно представити у числовому вигляді.

Саме для цього і створюються відображення у вигляді словників Python. Ключами служать текстові значення ознак, а значеннями - числові коди цих категорій. Наприклад, протоколу TCP відповідає код 1, UDP - 2, ICMP - 3. Подібним чином кожному прапорцю TCP та кожному мережевому сервісу зі списку присвоюється унікальне числове значення.

Далі за допомогою спискових вкладених виразів ці відображення застосовуються до відповідних стовпців датасету для заміни текстових значень

на числові коди. Якщо для якогось значення відображення не існує, то ставиться код 0.

Таким чином виконується числова трансформація даних, необхідна для застосування методів машинного навчання. Перетворений датасет повертається як результат роботи функції для подальшого використання в процесі побудови та навчання моделі класифікації чи виявлення аномалій.

```
29
30  dataset = pd.read_csv("C:\\Users\\putie\\Desktop\\magistr\\projects\\nsl\\KDDTrain+.csv")
31
32  dataset = preprocess_dataset(dataset)
33
34  X = dataset.iloc[:, 0:41].values
35  y = dataset.iloc[:, 41].values
36
37  X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
38
```

Рис. 3.3

На рис. 3.3 код виконує завантаження датасету, його попередню обробку та розбиття на навчальну і тестову вибірки для подальшого навчання та перевірки моделі машинного навчання.

Спочатку за допомогою бібліотеки Pandas завантажується датасет у CSV форматі з файлу по заданому шляху. Це основний набір даних для наступної роботи. Після завантаження датасет проходить попередню обробку за допомогою функції preprocess_dataset, яка була описана вище. Далі з датасету виділяються матриці ознак X та цільових змінних y. Матриця X містить усі характеристики об'єктів датасету, а вектор y - відповідні їм значення цільової змінної, яку потрібно буде передбачити моделлю.

Наступним кроком є розбиття вихідного датасету на навчальну та тестову вибірки за допомогою функції train_test_split. Це стандартна процедура в машинному навчанні. Навчальна вибірка використовується для налаштування

параметрів і навчання моделі, а тестова - для незалежної оцінки її якості. Розмір тестової вибірки встановлюється як 20% від загального обсягу даних.

```
46
47  classifier = GaussianNB()
48
49  selector = SelectKBest(score_func=f_classif, k=20)
50  selector.fit(X_train, y_train)
51  X_train_selected = selector.transform(X_train)
52  X_test_selected = selector.transform(X_test)
53
54
55  clf = classifier.fit(X_train_selected, y_train)
56  pred = clf.predict(X_test_selected)
57
```

Рис. 3.4

Частина коду на рис 3.4 створює та навчає модель класифікації на основі наївного байєсівського класифікатора з попереднім відбором найбільш значущих ознак. Спочатку створюється об'єкт GaussianNB - реалізація наївного байєсівського класифікатора з гаусовими розподілами ймовірностей ознак.

Наївний байєсівський класифікатор використовує теорему Байєса для визначення того, чи є доступ легітимним чи несанкціонованим. Теорема Байєса стверджує, що ймовірність того, що подія *y* відбудеться при умові, що відбулася подія *X*, розглядається як ймовірність того, що подія *X* відбудеться при умові, що відбулася подія *y*, помножена на ймовірність взагалі настання події *y* і поділена на ймовірність взагалі настання події *X*.

У контексті виявлення несанкціонованого доступу, подія *y* є несанкціонованим доступом, а подія *X* є набором ознак, які були виявлені:

$$P(y \mid x_1, \dots, x_n) = \frac{P(y)P(x_1, \dots, x_n \mid y)}{P(x_1, \dots, x_n)}$$

Імовірність того, що подія X відбудеться, визначена на основі статистичних даних про нормальний і аномальний трафік.

Набір ознак, які класифікатор використовує для виявлення несанкціонованого доступу, включають:

- Тривалість з'єднання
- Кількість невдалих спроб входу в систему
- Тип запиту
- Дані, які запитуються
- Чи було здійснено багато з'єднань з тієї ж IP-адреси і т. ін.

Наївний байєсівський класифікатор ґрунтується на припущенні, що ознаки незалежні одна від одної. Це означає, що ймовірність того, що зразок належить до певного класу, можна обчислити як добуток ймовірностей кожної ознаки:

$$P(y \mid x_1, \dots, x_n) \propto P(y) \prod_{i=1}^{n} P(x_i \mid y)$$

$$\Downarrow$$

$$\hat{y} = \arg\max_y P(y) \prod_{i=1}^{n} P(x_i \mid y)$$

На практиці, припущення про незалежність ознак часто не виконується. Однак, наївний байєсівський класифікатор часто все одно забезпечує хорошу якість класифікації, навіть якщо це припущення не виконується.

Отже, X_train та X_test - це набори ознак (вхідні дані), а y_train - це набір відповідей (класів) для відповідних прикладів у тренувальному наборі.

Навчання наївного байєсівського класифікатора полягає в оцінці параметрів розподілів ймовірностей ознак для кожного класу.

У задачах виявлення несанкціонованого доступу, класом "аномалія" може бути будь-яка подія, яка не відповідає нормальному поведінці користувача або системи. Наприклад, несанкціонований доступ може включати в себе спробу входу в систему з невідомої IP-адреси, спробу запуску програми з невідомого джерела або спробу доступу до заборонених файлів або каталогів.

Далі для відбору найбільш інформативних ознак використовується метод SelectKBest. Вказується, що потрібно відібрати 20 найкращих ознак з навчальної вибірки за цією мірою. Об'єкт SelectKBest використовує функцію f_classif, яка оцінює статистичну значимість кожної ознаки для задачі класифікації. Статистична значимість ознаки визначається як ймовірність того, що вона випадково має таку ж або більшу значимість, ніж фактична. Об'єкт SelectKBest повертає набір ознак, який містить найбільш значущі ознаки.

Після відбору ознак, виконується трансформація навчальної і тестової вибірок - залишаються лише обрані 20 стовпців даних. Далі класифікатор навчається на навчальній вибірці з відібраними ознаками. На завершення здійснюється предикт - прогнозування міток класів для тестової вибірки, також з використанням лише відібраних ознак.

```
57
58  cm = confusion_matrix(y_test, pred)
59  ac = 100 * accuracy_score(y_test, pred)
60  pr = 100 * precision_score(y_test, pred, average='weighted')
61  rc = 100 * recall_score(y_test, pred, average='weighted')
62  f1 = 100 * f1_score(y_test, pred, average='weighted')
63
64  print("Accuracy:", ac)
65  print("Precision:", pr)
66  print("Recall:", rc)
67  print("F-measure:", f1)
68  print("Confusion Matrix:", cm)
69
70  tp_indices = np.where((y_test == 'anomaly') & (pred == 'anomaly'))[0]
71
72  print("True positives(considered to be anomalies):", tp_indices)
73
```

Рис. 3.5

На рис. 3.5 зображено блок коду, який розраховує та виводить основні метрики якості та результати роботи навченого класифікатора на тестовій вибірці. Спочатку обчислюється матриця невідповідностей confusion_matrix між реальними мітками класів тестової вибірки y_test та прогнозованими класифікатором pred. Вона показує розподіл кількості правильних та неправильних предиктів по кожному з класів.

Далі розраховуються ключові метрики:

- accuracy_score - точність (відсоток правильних предиктів)
- precision_score - влучність (відсоток істинно позитивних серед усіх позитивних прогнозів)
- recall_score - повнота (відсоток виявлених істинно позитивних прикладів)
- f1_score - середнє гармонійне точності та повноти

Ці метрики дають уявлення про ефективність та якість роботи класифікатора на тестових даних.

Також виділяються індекси прикладів, що були правильно класифіковані як "аномалії". Це допомагає проаналізувати, наскільки добре модель знаходить саме цілий клас аномалій, що часто є найважливішою задачею.

```
74
75  class_labels = ['Normal', 'Anomaly']
76
77  plt.imshow(cm, interpolation='nearest', cmap=plt.cm.Blues)
78  plt.title('Confusion Matrix')
79  plt.colorbar()
80
81  tick_marks = np.arange(len(class_labels))
82  plt.xticks(tick_marks, class_labels)
83  plt.yticks(tick_marks, class_labels)
84
85  thresh = cm.max() / 2.0
86  for i in range(len(class_labels)):
87      for j in range(len(class_labels)):
88          plt.text(j, i, format(cm[i, j], 'd'),
89                   horizontalalignment="center",
90                   color="white" if cm[i, j] > thresh else "black")
91
92  plt.ylabel('True label')
93  plt.xlabel('Predicted label')
94
95  plt.show()
96
```

Рис. 3.6

Частина коду на рис. 3.6 будує матрицю невідповідностей для візуального представлення результатів роботи класифікатора. Спочатку визначаються мітки класів (нормальна поведінка та аномалія). Далі за допомогою matplotlib відображається сама матриця невідповідностей cm, де кожен елемент показує кількість прикладів, що належать до класу i та були класифіковані як клас j. Встановлюються підписи для осей матриці у вигляді назв класів. Щоб матриця була більш наочною, комірки забарвлюються залежно від значення у них.

В результаті отримується наочне графічне представлення розподілу прикладів між реальними та прогнозованими класами. Це дозволяє швидко оцінити якість класифікатора та виявити типові помилки на конкретних класах.

3.5 Оцінювання та аналіз результатів

Для оцінки якості побудованої моделі наївного класифікатора Байєса використовувалися наступні метрики:

Точність (accuracy) - відсоток правильно класифікованих прикладів загалом, як позитивних, так і негативних.

$$Accuracy = \frac{True\ Negatives + True\ Positives}{True\ Negatives + False\ Positives + True\ Positives + False\ Negatives}$$

Влучність (Precision) - відношення кількості правильно класифікованих позитивних прикладів до загальної кількості випадків, класифікованих як позитивні.

$$Precision = \frac{True\ Positives}{True\ Positives + \ False\ Positives}$$

Повнота (recall) - відношення кількості правильно ідентифікованих позитивних прикладів до загальної кількості позитивних прикладів у вибірці. Висока повнота означає, що модель знаходить більшість фактичних вторгнень.

$$Recall = \frac{True\ Positives}{True\ Positives + \ False\ Negatives}$$

F1-score - усереднення влучності та повноти, що дає загальне уявлення про якість моделі.

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

Всі ці метрики оцінюються від 0 до 1: чим ближче значення критерію до 1, тим вища продуктивність, а коли воно наближається до 0, продуктивність системи зменшується. Ці метрики були помножені на 100 для більш зручного представлення, де максимальне значення вже буде 100, а мінімальне - 0.

Додатково використовувалася матриця невідповідностей (confusion matrix), що дозволяє детальніше проаналізувати результати роботи моделі. Матриця невідповідностей показує, які типи помилок робить класифікатор при предикуванні класів.

В нашому випадку матриця має два рядки та два стовпці, оскільки є два можливих класи - normal (нормальний трафік) та anomaly (можливе вторгнення). По вертикалі відображаються фактичні значення класів з тестової вибірки, а по горизонталі - предиковані класифікатором значення. Наприклад, False Positives - це приклади нормального трафіку, неправильно ідентифіковані як атаки.

Елементи матриці невідповідностей відповідають наступним значенням:

- TN (True Negatives) - кількість випадків, коли нормальний трафік було правильно ідентифіковано (фактичний і предикт - normal).
- TP (True Positives) - кількість випадків, коли вторгнення було правильно ідентифіковане (фактичний класс - anomaly, предикт - anomaly).
- FP (False Positives) - кількість випадків, коли модель хибно визначила нормальний трафік як атаку (фактичний класс - normal, предикт - anomaly).
- FN (False Negatives) - кількість пропущених вторгнень, коли атака була класифікована як нормальний трафік (фактичний класс - anomaly, предикт - normal).

Аналіз цих метрик дозволяє краще зрозуміти сильні та слабкі сторони моделі. Наприклад, висока кількість FN означає, що модель часто пропускає реальні вторгнення.

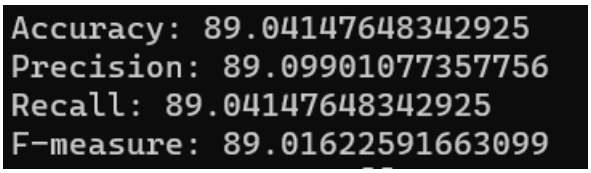

Рис. 3.7 - Результати

Точність (accuracy) склала 89%, що означає загалом хороший відсоток правильної класифікації як нормального трафіку, так і вторгнень.

Влучність (precision) становила 89%, тобто більшість випадків, визначених моделлю як атаки, дійсно були атаками. Це свідчить про низький рівень хибних спрацювань.

Повнота (recall) дорівнювала 89%, тобто модель знаходила більшість реальних вторгнень у тестовій вибірці. Водночас 11% атак все ж було пропущено.

F-міра, що усереднює влучність і повноту, склала 89%, що також вказує на достатньо високу загальну ефективність моделі.

Розроблений класифікатор продемонстрував задовільну якість на тестовій вибірці, досягнувши близько 89% за основними метриками. Застосування цього методу машинного навчання дозволяє ефективно аналізувати великі обсяги даних та вирішувати складні завдання класифікації.

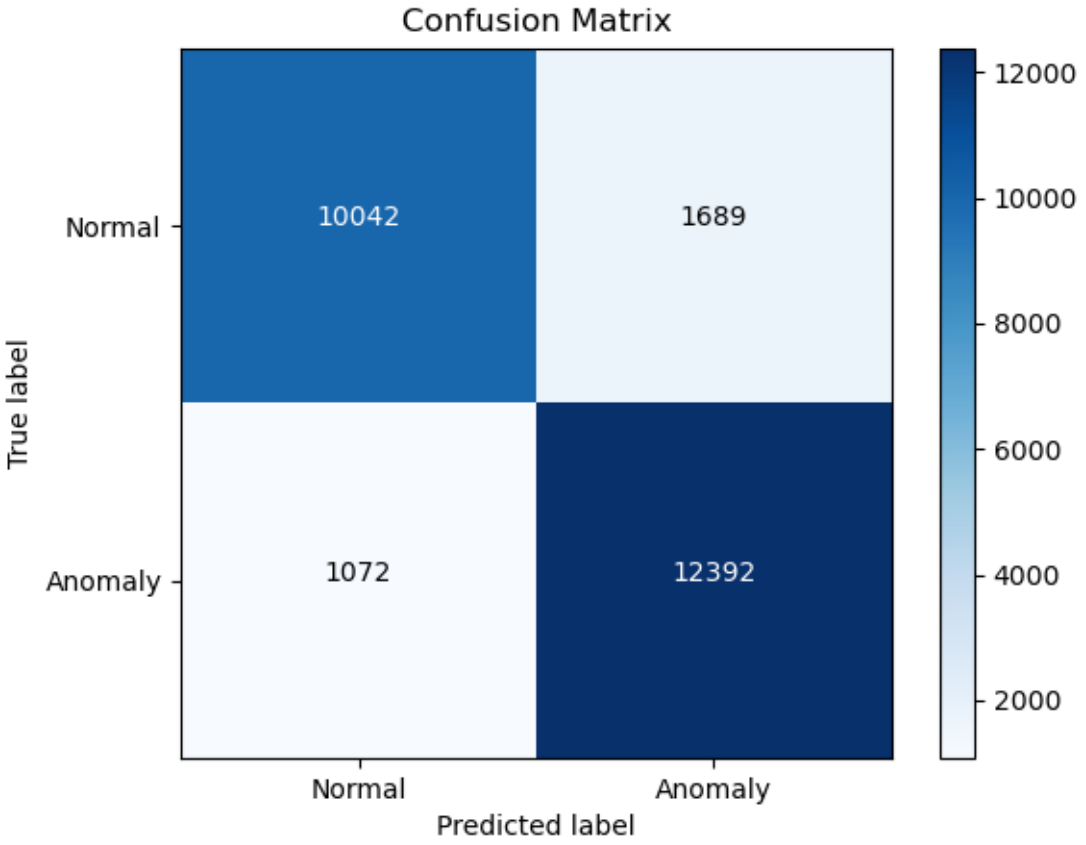

Рис. 3.8 - Матриця невідповідностей

Аналіз матриці невідповідностей показав, що основним типом помилок були False Positives - хибні спрацювання моделі. Водночас кількість False Negatives (пропущені вторгнення) була помірною.

Для перевірки результатів, ми виводимо на екран тестовий датасет (рис 3.9), в якому вказані номери рядків та вказано, чи це аномалія чи нормальний трафік. Після класифікації ми виводимо рядки (рис 3.10), які класифікатор визнав правильно як аномалії, тобто true positives, і порівнюємо це з реальним станом для наочної правильності роботи класифікатора. Таким чином можна переконатися, що класифікатор дійсно зміг розпізнати аномальні записи у тестових даних.

```
y_test (labels):
         Label
0       normal
1       normal
2      anomaly
3      anomaly
4       normal
...        ...
25190   normal
25191   normal
25192   normal
25193  anomaly
25194   normal

[25195 rows x 1 columns]
```

Рис. 3.9

```
True positives(considered to be anomalies): [    2     3     6 ... 25186 25188 25193]
```

Рис. 3.10

Висновки до розділу 3

У даному розділі було описано програмне забезпечення та методи, використані для реалізації практичної частини дипломної роботи. Зокрема, для розробки системи виявлення мережевих вторгнень було обрано мову програмування Python та низку спеціалізованих бібліотек для аналізу даних та машинного навчання, таких як Pandas, NumPy, Scikit-Learn.

Для навчання та тестування моделі класифікатора було використано відомий відкритий набір даних NSL-KDD, що містить інформацію про мережевий трафік з модельованими кібератаками різних типів. Цей набір пройшов попередню обробку - перетворення категоріальних ознак у числові, дискретизацію, розбиття на навчальну та тестову вибірки.

На навчальній вибірці було побудовано модель на основі наївного байєсівського класифікатора з використанням бібліотеки Scikit-Learn. Також

застосовувався метод відбору найбільш інформативних ознак. Після навчання модель тестувалася на незалежній тестовій вибірці та оцінювалися метрики якості роботи класифікатора.

Вибране програмне забезпечення та методи дозволили ефективно вирішити поставлене завдання - розробити та оцінити систему виявлення несанкціонованих доступів до комп’ютерних мереж основі імовірнісного класифікатора Байєса.

ВИСНОВКИ

У роботі було досліджено можливість застосування імовірнісного класифікатора Байєса для виявлення вторгнень в комп'ютерні мережі на основі аналізу мережевого трафіку.

Проведено детальний огляд існуючих методів та систем виявлення вторгнень, проаналізовано їх сильні та слабкі сторони. Розглянуто основні типи систем виявлення вторгнень: мережеві, хостові, віртуальні та фізичні. Обговорено переваги та недоліки підходів на основі сигнатур, виявлення аномалій та їх поєднання. Також було обґрунтовано доцільність застосування наївного байєсівського класифікатора для даної задачі. Детально описано різні типи наївних байєсівських класифікаторів: гаусівський, біноміальний, мультиноміальний та доповнювальний. Проаналізовано методи відбору ознак, зокрема фільтровий метод, обгортковий метод та вбудований метод.

Розроблено програмне забезпечення системи виявлення вторгнень з використанням мови Python та спеціалізованих бібліотек Pandas, NumPy, Scikit-Learn. Реалізовано функції попередньої обробки даних, підготовки навчальної та тестової вибірок. Побудовано оптимізовану модель гаусівського наївного байєсівського класифікатора з попереднім відбором найбільш інформативних ознак. Модель детально протестовано на відомому репрезентативному наборі даних NSL-KDD.

Експериментальне дослідження продемонструвало задовільну якість розробленої моделі, що досягла близько 89% точності на тестових даних. Проаналізовано отримані результати за допомогою різних метрик та матриці невідповідностей.

Проведене дослідження підтвердило ефективність застосування оптимізованого наївного байєсівського класифікатора для виявлення несанкціонованих доступів на основі аналізу мережевого трафіку.

Запропонований підхід до виявлення вторгнень є перспективним напрямком застосування методів машинного навчання в сфері кібербезпеки. Отримані результати можуть слугувати основою для подальших досліджень в цій галузі.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

1. ІС - Інформаційна система

2. ADMNC (*Anomaly Detection in Mixed Numerical and Categorical Input Spaces*) - Виявлення аномалій у змішаних числових і категоріальних вхідних просторах

3. ANOVA (*ANalysis Of VAriance*) - Дисперсійний аналіз

4. HIDS (*Host-based Intrusion Detection System*) - Система виявлення вторгнень на хост-рівні

5. IDS (*Intrusion Detection System*) – Система виявлення вторгнень

6. NIDS (*Network Intrusion Detection System*) - Система виявлення вторгнень у мережі

7. PIDS (*Physical Intrusion Detection System*) Система виявлення несанкціонованого фізичного доступу

8. PSO (*Particle swarm optimization*) - Метод рою часток

9. RFE (*Recursive Feature Elimination*) - Рекурсивне виключення ознак

10. VMIDS (*Virtual machine-based intrusion detection system*) Віртуальна система виявлення вторгнень

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Грайворонський М. В., Новіков О. М.. Безпека інформаційно-комунікаційних систем — 2009. — 608 с.

2. Зоріна Т.І. Системи виявлення і запобігання атак в комп'ютерних мережах / Т.І. Зоріна // Вісник східноукраїнського національного університету імені Володимира Даля. – 2013. – № 15 (204) ч.1. – С. 48 – 54.

3. Маляр М. М. Машинне навчання у процесі прийняття рішень / М. М. Маляр, Н. М. Маляр-Газда, М. М. Шаркаді Комбінаторні конфігурації та їхні застосування: Матеріали XXII Міжнародного науково-практичного семінару імені А. Я. Петренюка (Запоріжжя - Кропивницький, 15-16 травня 2020 року) / за ред. Г. П. Донця – Кропивницький : ПП «Ексклюзив-Систем», 2020.– С. 84-86.

4. A comprehensive guide to Feature Selection using Wrapper methods in Python [Електронний ресурс] – Режим доступу до ресурсу: https://www.analyticsvidhya.com/blog/2020/10/a-comprehensive-guide-to-feature-selection-using-wrapper-methods-in-python/

5. Akhiat, Y., Chahhou, M., & Zinedine, A. Feature selection based on graph representation. In 2018 IEEE 5th International Congress on InformationScience and Technology (CiSt) (pp. 232-237).IEEE. (2018, October).

6. Analysis of Variance (ANOVA) Explanation, Formula, and Applications [Електронний ресурс] – Режим доступу до ресурсу: https://www.investopedia.com/terms/a/anova.asp

7. Arif Jamal Malik, Waseem Shahzad and Farrukh Aslam Khan, Network Intrusion Detection Using Hybrid Binary PSO and Random Forests Algorithm, Security and Communication Networks, (2012).

8. Botana, I. L.-R., Eiras-Franco, C., & Alonso-Betanzos, A. (2020). Regression Tree Based Explanation for Anomaly Detection Algorithm. Proceedings, 54(1),7. https://doi.org/10.3390/proceedings2020054007

9. Bouchlaghem Y., Akhiat Y., and Amjad S., "Feature Selection: A Review and Comparative Study," E3S Web Conf., vol. 351, pp. 1–6, 2022, doi: 10.1051/e3sconf/202235101046.

10. Casas, P., Mazel, J., & Owezarski, P. (2012). Unsupervised network intrusion detection systems: Detecting the unknown without knowledge. Computer Communications, 35(7), 772-783.

11. Daş R., Karabade A., and Tuna G., "Common network attack types and defense mechanisms," 2015 23rd Signal Process. Commun. Appl. Conf. SIU 2015 - Proc., pp. 2658–2661, 2015, doi: 10.1109/SIU.2015.7130435.

12. Dataset NSL-KDD[Електронний ресурс] – Режим доступу до ресурсу: https://github.com/codenpython/NSL-KDD-dataset

13. Debar H. An Introduction to Intrusion-Detection Systems. – 2009. [Електронний ресурс] – Режим доступу: https://www.researchgate.net/publication/228589845_An_Introduction_to_Intrusion-Detection_Systems

14. Domingos P. and Pazzani M.. On the optimality of the simple Bayesian classifier under zero-one loss. Machine Learning, 29:103–130, 1997.

15. Feature Selection Techniques in Machine Learning [Електронний ресурс] – Режим доступу до ресурсу: https://www.geeksforgeeks.org/feature-selection-techniques-in-machine-learning/

16. Feature Selection Techniques in Machine Learning [Електронний ресурс] – Режим доступу до ресурсу: https://www.javatpoint.com/feature-selection-techniques-in-machine-learning

17. Guide to Physical Intrusion Detection Systems [Електронний ресурс] – Режим доступу до ресурсу: https://info.verkada.com/perimeter-security/physical%20intrusion%20detection%20systems-pids/

18. Manning C.D., Raghavan P. and Schütze H. (2008). Introduction to Information Retrieval. Cambridge University Press, pp. 234-265.

19. McCallum, A. and Nigam K. «A Comparison of Event Models for Naive Bayes Text Classification». In AAAI/ICML-98 Workshop on Learning for Text Categorization, pp. 41-48. Technical Report WS-98-05. AAAI Press. 1998.

20. Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

21. Norbik Bashah, Idris Bharanidharan Shanmugam, and Abdul Manan Ahmed," Hybrid Intelligent Intrusion Detection System" World Academy of Science, Engineering and Technology, 2005

22. Rennie, J. D., Shih, L., Teevan, J., & Karger, D. R. (2003). Tackling the poor assumptions of naive bayes text classifiers. In ICML (Vol. 3, pp. 616-623)

23. Saeys Y, Inza I, Larranaga P. A review of feature selection techniques in bioinformatics. bioinformatics. 2007;23(19):2507–17.

24. Saman M. Abdulla, Najla B. Al-Dabagh, Omar Zakaria, Identify Features and Parameters to Devise an Accurate Intrusion Detection System Using Artificial Neural Network, World Academy of Science, Engineering and Technology 2010.

25. Sandeep Kumar and Eugene Spafford. A pattern matching model for misuse intrusion detection. In Proceedings of the 17th National Computer Security Conference, pages 11–21, October 1994.

26. Scarfone Karen. Guide to Intrusion Detection and Prevention Systems (IDPS) — 2007. [Електронний ресурс] – Режим доступу: NIST SP 800-94, Guide to Intrusion Detection and Prevention Systems (IDPS)

27. Shuo X. , Li Y. , and Wang Z. , "Bayesian multinomial Naïve Bayes classifier to text classification," in Park J., Chen SC., Raymond Choo KK. (eds) Advanced Multimedia and Ubiquitous Engineering. Singapore: Springer. 2017.

28. Tavallaee M., Bagheri E., Lu W., and Ghorbani A., “A Detailed Analysis of the KDD CUP 99 Data Set,” Submitted to Second IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA), 2009.

29. The four types of IDS and how they can protect your business [Електронний ресурс] – Режим доступу до ресурсу: https://www.outsourceitcorp.com/the-four-types-of-ids-and-how-they-can-protect-your-business/

30. What Is a Host Intrusion Detection System (HIDS) and How It Works [Електронний ресурс] – Режим доступу до ресурсу: https://heimdalsecurity.com/blog/host-intrusion-detection-system-hids/

31. What is a network intrusion detection system (NIDS)?. [Електронний ресурс] – Режим доступу: https://datadome.co/learning-center/what-is-a-network-intrusion-detection-system/

32. What is machine learning? [Електронний ресурс] – Режим доступу до ресурсу: https://www.ibm.com/topics/machine-learning

33. Zhang H. (2004). The optimality of Naive Bayes. Proc. FLAIRS