

Міністерство освіти і науки України
Державний університет «Житомирська політехніка»
Факультет інформаційно-комп'ютерних технологій
Кафедра комп'ютерної інженерії та кібербезпеки

ПОЯСНЮВАЛЬНА ЗАПИСКА

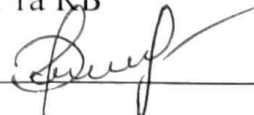
до випускної кваліфікаційної роботи магістра
на тему:

«Дослідження детектора аномалій на основі нейронної
мережі Azure для рішення завдань кібербезпеки»

Виконав студент 2-го курсу, групи КБм-21-1
спеціальності 125 «Кібербезпека»

 Олексюк Б. Ю.

Керівник, старший викладач кафедри
КІ та КБ

 Байлюк Є. М.

Рецензент, доцент кафедри КІ та КБ, к.т.н.

 Бродський Ю.Б.

Житомир 2022

ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА КІБЕРБЕЗПЕКИ
СПЕЦІАЛЬНІСТЬ 125 «КІБЕРБЕЗПЕКА»

ЗАТВЕРДЖУЮ

Завідувач кафедри
комп'ютерної інженерії
та кібербезпеки

 Єфіменко А. А.

«25» 10 2022 р.

ЗАВДАННЯ

на випускню кваліфікаційну роботу

Студента Олексюк Богдана Юрійовича

Тема роботи: Дослідження детектора аномалій на основі служб нейронної мережі Azure для рішення завдань кібербезпеки

затверджена Наказом по університету від «25» жовтня 2022 р. № 501/С

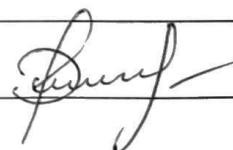
Термін здачі студентом закінченої роботи – «19» грудня 2022 р.

Вихідні дані для роботи:

Консультанти з випускної кваліфікаційної роботи із зазначенням розділів, що їх стосуються

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Байлюк Є. М.	2.09.22	2.09.22
2	Байлюк Є. М.	6.10.22	6.10.22
3	Байлюк Є. М.	4.11.22	4.11.22

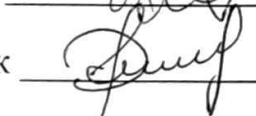
Керівник



Календарний план

№ з/п	Назва етапів випускної кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Постановка завдання. Пошук, огляд та аналіз літературних джерел.	2 листопада 2022 – 5 листопада 2022	Виконано
2	Пректування структури системи	6 листопада 2022 – 12 листопада 2022	Виконано
3	Розробка конвеєра та налаштування конструктура	13 листопада 2022 – 26 листопада 2022	Виконано
4	Тестування	27 листопада 2022 – 4 грудня 2022	Виконано
5	Захист	26 грудня 2022	Виконано

Студент  Олексюк Б. Ю.

Керівник  Байлюк Є. М.

РЕФЕРАТ

Кваліфікаційна робота складається з проекту детектора аномалій на основі служб машинного навчання Azure та пояснювальної записки до нього. Пояснювальна записка до роботи містить 64 сторінок, 29 ілюстрацій та 2 таблиць.

Метою дослідження є виявлення переваг та недоліків нейронних мереж за рахунок розробки та застосування детектора на основі служб машинного навчання Azure. Об'єктом дослідження є процес пошуку аномалій у мережевому трафіку на основі нейронної мережі Azure з використанням KDD Cup. Предметом дослідження є алгоритм класифікації нейронні мережі машинного навчання студії Microsoft Azure для вирішення завдання виявлення аномалій.

В роботі визначено основні завдання на основні завдання на побудову та дослідження детектора, який попередить та захистить від зловмисних вторгнень в системи. Також в роботі проаналізовані метод нейронних мереж та його типи, обґрунтовано актуальність даного детектора аномалій та описана сфера його використання, обґрунтований вибір виробника обладнання – Microsoft. Розроблені, сформовані та налагоджені загальний проект детектора. Наведено дослідження переваг та недоліків технологій машинного навчання Azure, алгоритм побудови детектора аномалій, дослідження та порівняння алгоритму різних тестових базах даних, рекомендації щодо застосування технологій нейронної мережі в майбутньому. Також наводяться результати ефективності навчання моделі детектора аномалій.

КЛЮЧОВІ СЛОВА: ЗАХИСТ, КЛАСИФІКАЦІЯ, МАШИННЕ НАВЧАННЯ, AZURE, ДЕТЕКТОР АНОМАЛІЙ, НЕЙРОННІ МЕРЕЖІ.

					КБМ.КР.М – 125 – 22 – ПЗ			
Зм.	Арк.	№ докум.	Підпис	Дата	Дослідження детектора аномалій на основі нейронної мережі Azure для рішення завдань кібербезпеки	Літ.	Арк.	Аркушів
Розроб.		Олексюк Б. Ю.		19.12.22				
Керівник		Байлюк Є. М.		19.12.22			4	
Рецензент.		Бродський Ю. Б.		19.12.22		Житомирська політехніка, група КБм-21-1		
Зав. каф.		Єфіменко А. А.		19.12.22				
					Пояснювальна записка			

ABSTRACT

The qualification work consists of an anomaly detector project based on Azure machine learning services and an explanatory note for it. The explanatory note to the work contains 64 pages, 29 illustrations and 2 tables.

The purpose of the research is to identify the advantages and disadvantages of neural networks due to the development and application of a detector based on Azure machine learning services. The object of research is the process of finding anomalies in network traffic based on the Azure neural network using KDD Cup. The subject of the study is the neural network classification algorithm of the Microsoft Azure studio for solving the problem of anomaly detection.

The work defines the main tasks for the construction and research of a detector that will warn and protect against malicious intrusions into systems. The work also analyzes the method of neural networks and its types, substantiates the relevance of this anomaly detector and describes the scope of its use, substantiates the choice of the equipment manufacturer - Microsoft. The general design of the detector was developed, formed and adjusted. A study of the advantages and disadvantages of Azure machine learning technologies, an algorithm for building an anomaly detector, research and comparison of the algorithm in various test databases, recommendations for the use of neural network technologies in the future are given. The results of training efficiency of the anomaly detector model are also given.

KEYWORD: SECURITY, NETWORK CLASSIFICATION, MACHINE LEARNING, AZURE, ANOMALY DETECTOR, NEURON NETWORKS.

ЗМІСТ

РЕФЕРАТ	4
ABSTRACT	5
ВСТУП.....	8
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ОБҐРУНТУВАННЯ АКТУАЛЬНОСТІ ТЕМИ РОБОТИ.....	10
1.1. ОБҐРУНТУВАННЯ АКТУАЛЬНОСТІ ТЕМИ РОБОТИ	10
1.2. АНАЛІЗ МЕТОДІВ МАШИННОГО НАВЧАННЯ.....	13
1.3. АНАЛІЗ НЕЙРОННОЇ МЕРЕЖІ ЯК МЕТОДУ МАШИННОГО НАВЧАННЯ	17
1.4. ТИПИ НЕЙРОННИХ МЕРЕЖ	21
1.5. АНАЛІЗ ІСНУЮЧИХ МЕТОДИК ВИЯВЛЕННЯ АНОМАЛІЙ І ВТОРГНЕНЬ З ВИКОРИСТАННЯМ МЕТОДІВ МАШИННОГО НАВЧАННЯ.....	27
1.6. АНАЛІЗ АНАЛІТИЧНИХ МОЖЛИВОСТЕЙ ХМАРНОГО СЕРЕДОВИЩА AZURE З ВИКОРИСТАННЯМ МАШИННОГО НАВЧАННЯ	29
1.7. Постановка завдання	31
РОЗДІЛ 2. РОЗРОБКА ХМАРНОГО РІШЕННЯ ЗАДАЧІ КЛАСИФІКАЦІЇ НА ОСНОВІ КОНСТРУКТОРА МАШИННОГО НАВЧАННЯ AZURE.....	32
2.1. Розробка архітектури рішення з використанням студії машинного навчання AZURE.....	32
2.2. Аналіз вхідних даних та їх попередня обробка	37
2.3. Розробка моделі для класифікації на основі нейронної мережі	44
РОЗДІЛ 3. ПЕРЕВІРКА ПРАЦЕЗДАТНОСТІ ТА ЕФЕКТИВНОСТІ РІШЕННЯ ЗАДАЧІ КЛАСИФІКАЦІЇ	47
3.1. Показники якості класифікації при виявленні аномалій.....	47
3.2. Дослідження та порівняння алгоритму нейронних мереж студії машинного навчання AZURE	51
3.3 Рекомендації по застосуванню результатів дослідження	56
ВИСНОВКИ.....	59
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	61

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

KDD – Knowledge Discovery in Databases

MH – машинне навчання

IDS – Intrusion Detection System

DOS – Denial-of-service

R2L – remote to local

U2R – user to root

DL – Deep learning

IDS – Intrusion Detection System

DM – Data Mining

ANN – Artificial Neural Network

SNN – Spiking neural network

MSE – Mean Squared Error

CNN – Convolutional neural network

RNN – Recurrent neural network

SVM – Support vector machine

RBAC – Role Based Access Control

PR-curve – Precision-Recall Curve

ВСТУП

Актуальність теми. Поява технологій, починаючи від смартфонів і закінчуючи великомасштабними системами зв'язку, призвела до надзвичайно цифрового взаємопов'язаного суспільства та величезного використання Інтернету.

За оцінками, на сьогоднішній день у світі існує понад 5 мільярдів розумних пристроїв і 3 мільярди користувачів Інтернету. Це кіберз'єднання широко використовується в різноманітних програмах, таких як онлайн-банкінг і покупки, електронна пошта, документи або обмін важливою інформацією, відеочат та ігри тощо. Отже, багато даних у терабайтах на секунду створюються, обробляються, обмінюються та зберігаються різними програмами, а також Інтернетом речей (IoT). Насправді вважається, що 90% даних у світі сьогодні було згенеровано лише за останні чотири роки.

Зростання використання Інтернету та пов'язаних з ним послуг також збільшує кількість випадків кібератак щороку.

Протягом останніх десятиліть було запроваджено кілька заходів протидії кібератакам, які зазвичай називають системами виявлення вторгнень (IDS). В останні роки методи обчислювального інтелекту, включаючи машинне навчання (ML), глибоке навчання (DL) і інтелектуальний аналіз даних (DM), були використані для забезпечення кібербезпеки.

Хорошу базу для створення рішень штучного інтелекту надають сучасні хмарні сервіси, наприклад, такі як Microsoft Azure. Тому тема роботи дослідження детектора аномалій на основі нейронної мережі Azure для рішення завдань кібербезпеки є **актуальною**.

Об'єктом дослідження є процес пошуку аномалій у мережевому трафіку на основі служб машинного навчання Azure з використанням бази даних KDD Cup.

Предметом дослідження алгоритм класифікації конструктора нейронна мережа студії Microsoft Azure для вирішення завдання виявлення аномалій.

Метою дослідження є дослідження методу класифікації студії машинного навчання Microsoft Azure нейронна мережа з використанням KDD Cup для виявлення аномалій та вторгнень.

Перед нами стоять декілька завдань:

- Ознайомлення з технологіями служб машинного навчання Azure;
- Визначення алгоритму побудови детектора аномалій;
- Дослідження переваг та недоліків технологій машинного навчання Azure;
- Вивчення та налаштування технологій служб машинного навчання Azure;
- Розроблення технологія виявлення аномалій та вторгнень в KDD Cup з використанням методів машинного навчання Azure.

Наукова новизна дослідження полягає в використанні можливостей більш сучасної студії машинного навчання середовища Microsoft Azure для розробки моделі виявлення аномалій і вторгнень з використанням бази даних KDD Cup.

Методи досліджень. В даній роботі були використані можливості моделювання хмарного середовища Microsoft Azure, зокрема «нейронна мережа Azure», метод нейронної мережі машинного навчання, набір даних KDD Cup Data set та стандартний набір даних для аудиту KDD.

Апробація результатів:

1. Олексюк Б.Ю., Єфіменко А.А., Вакалюк Т.А. СУЧАСНІ FRAMEWORKS МАШИННОГО НАВЧАННЯ// Тези доповідей V Всеукраїнської науково-технічної конференції «Комп'ютерні технології: інновації, проблеми, рішення», м. Житомир, 01 – 02 грудня 2022 р. – Житомир: Житомирська політехніка, 2022. С. 5.

2. Олексюк Б.Ю., Єфіменко А.А., Вакалюк Т.А. МЕТОДИ ВИЯВЛЕННЯ ВТОРГНЕНЬ З ВИКОРИСТАННЯМ МАШИННОГО НАВЧАННЯ// Тези доповідей V Всеукраїнської науково-технічної конференції «Комп'ютерні технології: інновації, проблеми, рішення», м. Житомир, 01 – 02 грудня 2022 р. – Житомир: Житомирська політехніка, 2022. С. 8.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ОБҐРУНТУВАННЯ АКТУАЛЬНОСТІ ТЕМИ РОБОТИ

1.1. Обґрунтування актуальності теми роботи

На сьогоднішній день, Інтернет став частиною нашого повсякденного життя, світ бізнесу пов'язаний з Інтернетом. Щодня багато людей підключаються до Інтернету, щоб скористатися перевагами сучасної бізнес-концепції, відомої як електронний бізнес. Тому покращення зв'язку стало дуже актуальним у сучасному електронному бізнесі.

В Інтернеті існує два бізнес-етапи. По-перше, Інтернет надає чудову бізнес-можливість для залучення клієнтів, водночас створюючи великі ризики для компанії. В Інтернеті є нешкідливі та шкідливі користувачі. Приблизно в той же час компанія робить свою систему даних відкритою для нешкідливих користувачів Інтернету. Зловмисники або хакери часто можуть отримати доступ до внутрішньої інфраструктури компанії з багатьох причин. Є помилки програмного забезпечення, збій адміністративної безпеки та, можливо, залишення системи в конфігурації за замовчуванням.

Зловмисники використовують цілий ряд методів, включаючи злом паролів, однорангову атаку, атаку з перехопленням, підслуховування, атаку на прикладному рівні тощо, щоб усунути вищезазначені вразливості пристроїв і зламати чутливі системи. Таким чином, приватні ресурси організації з Інтернету та користувачів усередині організації потребували певної форми захисту.

Система виявлення вторгнень (IDS) є важливим інструментом, який використовується в кібербезпеці для моніторингу та визначення атак вторгнень. Існує три типи IDS; IDS мережі, IDS хоста та IDS програми.

Мережевий IDS відстежує мережевий пакет, щоб виявити атаку вторгнення. У той час як хост IDS контролює один хост (сервер або

комп'ютер). Нарешті, IDS програми відстежує кілька відомих програм із високим ризиком.

Щоб визначити, чи відбулася атака вторгнення, IDS використовує кілька підходів. По-перше, це підхід на основі сигнатур, коли відома сигнатура атаки вторгнення зберігається в базі даних IDS для відповідності поточним даним системи. Коли IDS знайде відповідність, вона розпізнає це як вторгнення. Такий підхід забезпечує швидке і точне виявлення. Однак недоліком цього є періодичне оновлення бази даних підписів. Крім того, система може бути скомпрометована до того, як буде оновлено найновішу атаку вторгнення.

Другий підхід заснований на аномаліях або поведінці, коли IDS визначає атаку, коли система працює поза нормою. Цей підхід може виявити як відомі, так і невідомі атаки. Однак недоліком цього підходу є низька точність із високим рівнем помилкових тривог.

Нарешті, гібридний підхід використовує підходи як на основі сигнатур, так і на основі аномалій. Цей підхід використовує підхід на основі сигнатур для виявлення відомих атак і підхід на основі аномалій для виявлення невідомих атак. Поєднання обох підходів може забезпечити більш ефективне виявлення, але може збільшити витрати на обчислення[1].

В останні роки доступність великих наборів даних у поєднанні з удосконаленням алгоритмів і експоненціальним зростанням обчислювальної потужності призвели до неперевершеного сплеску інтересу до теми машинного навчання. Нині алгоритми машинного навчання успішно застосовуються для завдань класифікації, регресії, кластеризації або зменшення розмірності великих наборів вхідних даних великої розмірності.

Насправді машинне навчання довело свої надлюдські здібності в багатьох сферах (самокеровані автомобілі, класифікація зображень, тощо). У результаті величезні частини нашого повсякденного життя, наприклад розпізнавання зображень і мовлення, веб-пошук, виявлення шахрайства, фільтрація електронної пошти/спаму, кредитні оцінки і багато іншого, працюють на основі алгоритмів машинного навчання.

Машинне навчання (ML) використовує підхід статистичного моделювання, щоб дізнатися шаблон минулих даних, а потім прогнозує найімовірніший результат, використовуючи нові дані. Таким чином, алгоритм ML був застосований до IDS з використанням підходу на основі аномалій. Як було сказано вище, проблема полягає в тому, щоб побудувати модель, яка може забезпечити високу точність з низьким рівнем помилкових тривог.

Системи виявлення вторгнень можна використовувати проти таких поширених атак:

1. Відмова в обслуговуванні (DoS): це атака, під час якої зловмисник спрямовує потоки запитів на трафік до системи, щоб зробити обчислювальний ресурс або ресурс пам'яті занадто зайнятими або занадто заповненими для обробки законних запитів, і в процесі відмовляє законним користувачам доступ до машини.

2. Зондування (зондування): перевірка мережі комп'ютерів для збору інформації, яка буде використана для компрометації її засобів контролю безпеки.

3. User to Root Attack (U2R): клас експлойтів, у яких зловмисник починає з доступу до звичайного облікового запису користувача в системі (отриманого шляхом перехоплення паролів, атаки за словником або соціальної інженерії) і може використовувати деякі вразливості для отримання кореневого доступу до системи.

4. Віддалена локальна атака (R2L): виникає, коли зловмисник, який має можливість надсилати пакети на машину через мережу, але не має облікового запису на цій машині, використовує певну вразливість, щоб отримати локальний доступ як користувач цієї машини.

5. Впровадження мови структурованих запитів (SQL): відбувається, коли зловмисник вставляє зловмисний код на сервер, який використовує SQL, і змушує сервер розкривати інформацію, яку він зазвичай не робив би.

1.2. Аналіз методів машинного навчання

Неймовірна корисність ШНМ була доведена у значній кількості доменів. Від їх раннього дизайну до всіх нових винаходів і доповнень до підходу, знайшли свій шлях до таких програм, як аналіз даних, біометрія, обробка природної мови, розпізнавання фігури, розпізнавання обличчя і, зрозуміло, виявлення вторгнення та безліч інших.

Нейронні мережі зараз є широко розповсюдженим інструментом, видатним у програмах, де необхідний аналіз даних.

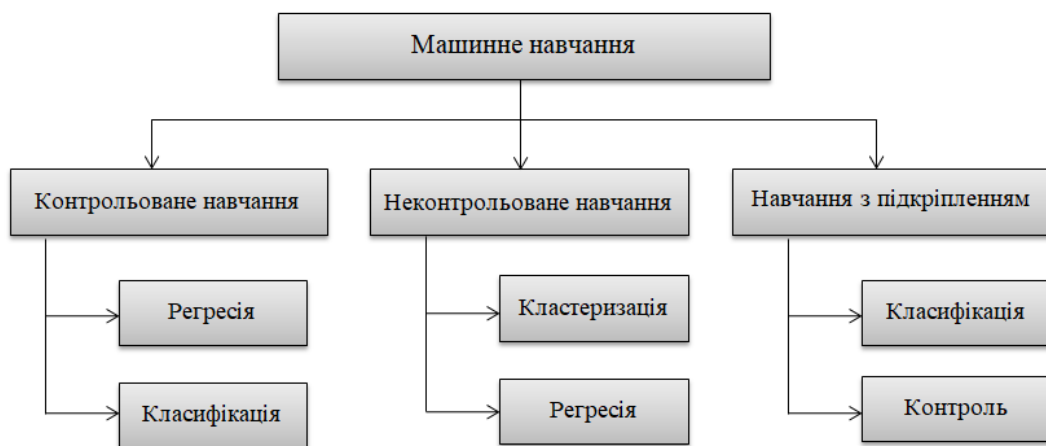


Рисунок 1.1 – Огляд типів та завдань машинного навчання

У найзагальнішому випадку розрізняють два типу машинного навчання: навчання по прецедентах, або індуктивне навчання, і дедуктивне навчання. Оскільки останнє прийнято відносити до області експертних систем, то терміни «машинне навчання» і «навчання по прецедентах» можна вважати синонімами.

Цей метод навчання зараз, як прийнято говорити, в тренді, а ось експертні системи переживають кризу. Бази знань, що лежать в їх основі, важко узгоджувати з реляційною моделлю даних, тому промислові СУБД неможливо ефективно використовувати для наповнення баз знань експертних систем.

Навчання по прецедентах, в свою чергу, поділяють на три основних типи: контрольоване навчання, або навчання з учителем (supervised learning), неконтрольоване навчання (unsupervised learning), або навчання без учителя, і навчання з підкріпленням (reinforcement learning).

Крім названих, розробляються і інші методи навчання: активне, багатозадачне, різноманітне, трансферне і т.д. Особливо успішно розвивається в останні роки «глибоке навчання», при використанні якого можуть успішно поєднуватися алгоритми навчання з вчителем і без вчителя.

1) Контрольоване навчання

Цей метод навчання застосовується у випадках, коли є великі обсяги даних, припустимо — тисячі фотографій домашніх тварин з маркерами (мітками, ярликами): це кішка, а це собака. Необхідно створити алгоритм, за допомогою якого машина могла б по фотографії, яку «не бачила» раніше, визначити, хто на ній зображений: кішка або собака. У ролі «вчителя» в даному випадку виступає людина, яка заздалегідь проставила маркери. Машина сама вибирає ознаки, за якими вона відрізняє кішок від собак. Тому в подальшому знайдений нею алгоритм може бути швидко переналаштований на рішення іншої задачі, наприклад, на розпізнавання курей і качок. Машина знову-таки сама виконає складну і копітку роботу по виділенню ознак, за якими буде розрізняти цих птахів. А нейромережа, яку навчили розпізнавати кішок, можна швидко навчити обробляти результати комп'ютерної томографії.

2) Неконтрольоване навчання

Хоча маркованих, розмічених даних накопичилося вже досить багато, даних без маркерів (міток) все ж набагато більше. Це зображення без підписів, аудіозаписи без коментарів, тексти без анотацій. Завдання машини при неконтрольованому навчанні — знайти зв'язку між окремими даними, виявити закономірності, підібрати шаблони, упорядкувати дані або описати їх структуру, виконати класифікацію даних. Неконтрольоване навчання використовується, наприклад, в рекомендаційних системах, коли в інтернет-магазині на основі аналізу попередніх покупок покупцеві пропонуються товари, які можуть зацікавити його з більшою ймовірністю, ніж інші. Або коли на після перегляду якогось відеокліпу на порталі YouTube відвідувачеві пропонують десятки посилань на ролики, чимось схожі на переглянутий. Або коли Google у

відповідь на один і той же запит ранжує посилання в результатах пошуку для одного користувача інакше, ніж для іншого, оскільки враховує історію пошуків.

3) Навчання з підкріпленням

Таке навчання є окремим випадком контрольованого навчання, але вчителем в даному випадку є «середовище». Машина (її в цій ситуації часто називають «агент») не має попередньої інформації про середовище, але має можливість здійснювати в ній будь-які дії. Середовище реагує на ці дії і тим самим надає агенту дані, які дозволяють йому реагувати на них і вчитися. Фактично агент і середовище утворюють систему зі зворотним зв'язком.

Навчання з підкріпленням використовується для вирішення більш складних завдань, ніж навчання з учителем і без вчителя. Воно використовується, наприклад, в системах навігації для роботів, які навчаються уникати зіткнень з перешкодами шляхом набуття досвіду, отримуючи зворотний зв'язок при кожному зіткненні. Навчання з підкріпленням використовується також в логістиці, при складанні графіків і плануванні завдань, при навчанні машини логічним іграм (покер, нарди, го і ін.).

4) Нейронні мережі і глибоке навчання

Для машинного навчання використовують різні технології та алгоритми. Зокрема, можуть застосовуватися дискримінантний аналіз, байєсовські класифікатори та багато інших математичних методів. Але в кінці XX століття все більше уваги почали приділяти штучним нейронним мережам (ANN). Черговий вибух інтересу до них почався в 1986 році, після істотного розвитку т.зв. «Методу зворотного поширення помилки», який з успіхом застосували при навчанні нейронної мережі.

ANN є системою з'єднаних і взаємодіючих між собою штучних нейронів, виконаних на основі порівняно простих процесорів. Кожен процесор ANN періодично отримує сигнали від одних процесорів (або від сенсорів, або від інших джерел сигналів) і періодично посилає сигнали іншим процесорам.

Всі разом ці прості процесори, з'єднані в мережу, здатні вирішувати досить складні завдання.

Найчастіше нейрони розташовуються в мережі за рівнями (їх ще називають шарами). Нейрони першого рівня — це, як правило, вхідні. Вони отримують дані ззовні (наприклад, від сенсорів системи розпізнавання осіб) і після їх обробки передають імпульси через синапси нейронів на наступному рівні. Нейрони на другому рівні (його називають прихованим, оскільки він безпосередньо не пов'язаний ні з входом, ні з виходом ANN) обробляють отримані імпульси і передають їх нейронам на вихідному рівні. Оскільки мова йде про імітацію нейронів, то кожен процесор вхідного рівня пов'язаний з декількома процесорами прихованого рівня, кожен з яких, в свою чергу, пов'язаний з декількома процесорами рівня вихідного. Така архітектура найпростішої ANN, яка здатна до навчання і може знаходити прості взаємозв'язку в даних.

Глибоке (глибинне) навчання може бути застосоване лише по відношенню до більш складних ANN, що містить кілька прихованих рівнів. При цьому рівні нейронів можуть чергуватися з шарами, які виконують складні логічні перетворення. Кожен наступний рівень мережі шукає взаємозв'язки в попередньому. Така ANN здатна знаходити не тільки прості взаємозв'язки, а й взаємозв'язки між взаємозв'язками. Саме завдяки переходу на нейромережу з глибинним навчанням компанії Google вдалося різко підвищити якість роботи свого популярного продукту «Перекладач». Зокрема, якість перекладу між англійською та французькою мовами підвищився відразу на 7 балів, тобто більш ніж на 20%. Попередня система, яка виконувала фразовий статистичний машинний переклад, домоглася подібного поліпшення за весь час свого існування (з 2006 року).

1.3. Аналіз нейронної мережі як методу машинного навчання

Нейронні мережі, також відомі як штучні нейронні мережі (ANN) або імітовані нейронні мережі (SNN), є підмножиною машинного навчання та є основою алгоритмів глибокого навчання. Їх назва та структура навіяні людським мозком, імітуючи спосіб, яким біологічні нейрони передають сигнали один одному.

Штучні нейронні мережі (ШНМ) складаються з вузлових шарів, що містять вхідний рівень, один або більше прихованих шарів і вихідний рівень. Кожен вузол, або штучний нейрон, з'єднується з іншим і має відповідну вагу та поріг. Якщо вихід будь-якого окремого вузла перевищує вказане порогове значення, цей вузол активується, надсилаючи дані на наступний рівень мережі. В іншому випадку дані не передаються на наступний рівень мережі.

Нейронні мережі покладаються на навчальні дані, щоб навчатися та підвищувати свою точність з часом. Однак, коли ці алгоритми навчання точно налаштовані на точність, вони стають потужними інструментами в інформатиці та штучному інтелекті, що дозволяє класифікувати та кластеризувати дані з високою швидкістю.

Завдання з розпізнавання мовлення або розпізнавання зображень можуть тривати хвилини чи години порівняно з ручною ідентифікацією експертів-людей. Однією з найвідоміших нейронних мереж є пошуковий алгоритм Google.

Неймовірна корисність ШНМ була доведена у значній кількості доменів. Від їх раннього дизайну до всіх нових винаходів і доповнень до підходу, знайшли свій шлях до таких програм, як аналіз даних, біометрія, обробка природної мови, розпізнавання фігури, розпізнавання обличчя і, зрозуміло, виявлення вторгнення та безліч інших[2].

Нейронні мережі зараз є широко розповсюдженим інструментом, видатним у програмах, де необхідний аналіз даних.

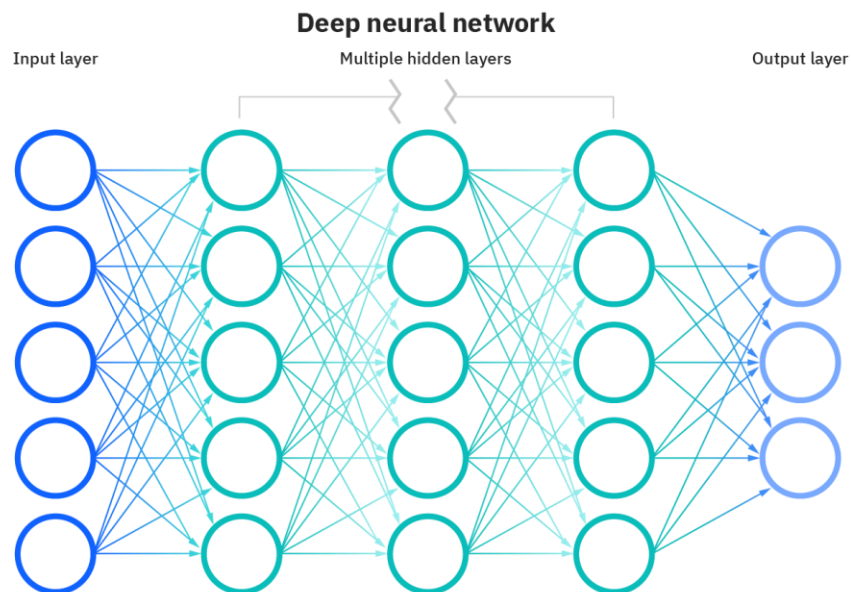


Рисунок 1.2 – Графічне представлення роботи нейронної мережі

Про кожен окремий вузол потрібно думати як про власну модель лінійної регресії, що складається з вхідних даних, вагових коефіцієнтів, зміщення (або порогу) і вихідних даних.

Формула виглядатиме приблизно так:

$$\sum w_i x_i + \text{зміщення} = w_1 x_1 + w_2 x_2 + w_3 x_3 + \text{зміщення} \quad (1.1)$$

$$\begin{aligned} \text{Вихідні дані} = f(x) = 1, \text{ якщо } \sum w_1 x_1 + b \geq 0; 0, \\ \text{якщо } \sum w_1 x_1 + b < 0 \end{aligned} \quad (1.2)$$

Після визначення вхідного рівня призначаються ваги. Ці вагові коефіцієнти допомагають визначити важливість будь-якої заданої змінної, причому більші з них мають більший внесок у результат порівняно з іншими вхідними параметрами. Потім усі вхідні дані множаться на відповідні ваги, а потім підсумовуються.

Після цього вихідні дані проходять через функцію активації, яка визначає вихідні дані. Якщо цей вихід перевищує заданий поріг, він «спрацьовує» (або активує) вузол, передаючи дані на наступний рівень у мережі. Це призводить до того, що вихід одного вузла стає входом наступного вузла. Цей процес передачі даних з одного рівня на наступний визначає цю нейронну мережу як мережу прямого зв'язку.

Розберемо, як може виглядати один окремих вузол за допомогою двійкових значень. Застосовуємо цю концепцію до більш відчутного прикладу, наприклад, чи варто вам займатися серфінгом (Так: 1, Ні: 0). Рішення йти чи не йти є нашим прогнозованим результатом, або y_{hat} . Припустимо, що на прийняття рішення впливають три фактори:

1. Чи хороші хвилі? (Так: 1, Ні: 0)
2. Склад пустий? (Так: 1, Ні: 0)
3. Чи була нещодавня атака акули? (Так: 0, Ні: 1)

Тоді давайте припустимо наступне, надаючи нам такі вхідні дані:

- $X_1 = 1$, оскільки хвилі накачують;
- $X_2 = 0$, оскільки натовп вийшов;
- $X_3 = 1$, оскільки недавнього нападу акули не було.

Тепер нам потрібно призначити деякі ваги, щоб визначити важливість. Більші ваги означають, що певні змінні мають більшу важливість для рішення чи результату.

- $W_1 = 5$, оскільки великі набряки виникають не часто;
- $W_2 = 2$, оскільки ви звикли до натовпу;
- $W_3 = 4$, оскільки ви боїтеся акул.

Нарешті, ми також припустимо порогове значення 3, що означає значення зсуву -3 . З усіма різними вхідними даними ми можемо почати додавати значення у формулу, щоб отримати бажаний результат.

$$Y_{hat} = (1 * 5) + (0 * 2) + (1 * 4) - 3 = 6 \quad (1.3)$$

Якщо ми використовуємо функцію активації, ми можемо визначити, що вихід цього вузла буде 1, оскільки 6 більше за 0. У цьому випадку ви б пішли займатись серфінгом; але якщо ми налаштуємо вагові коефіцієнти або поріг, ми зможемо досягти інших результатів від моделі.

Коли ми спостерігаємо за одним рішенням, як у наведеному вище прикладі, ми бачимо, як нейронна мережа може приймати дедалі складніші рішення залежно від результатів попередніх рішень або рівнів.

У наведеному вище прикладі ми використали персептрони, щоб проілюструвати деякі з математики, що тут грає, але нейронні мережі використовують сигмоподібні нейрони, які відрізняються значеннями від 0 до 1. Оскільки нейронні мережі поведуться подібно до дерев рішень, каскадні дані з одного вузла з іншого боку, наявність значень x від 0 до 1 зменшить вплив будь-якої зміни окремої змінної на вихід будь-якого даного вузла, а згодом і на вихід нейронної мережі.

Коли ми починаємо думати про більш практичні випадки використання нейронних мереж, як-от розпізнавання або класифікація зображень, ми використовуватимемо контрольоване навчання або мічені набори даних для навчання алгоритму. Під час навчання моделі ми захочемо оцінити її точність за допомогою функції витрат (або втрат). Це також зазвичай називають середньоквадратичною помилкою (MSE). У наведеному нижче рівнянні

$\hat{y}$ і представляє індекс вибірки,
 $\hat{y}$ — прогнозований результат,
 y — фактичне значення, i
 m — кількість проб.

$$\text{Cost Function} = \text{MSE} = \frac{1}{2m} \sum_{i=1}^m (\hat{y}^{(i)} - y^{(i)})^2 \quad (1.4)$$

Зрештою, мета полягає в тому, щоб мінімізувати нашу функцію витрат, щоб забезпечити правильність відповідності для будь-якого заданого спостереження. Коли модель коригує свої ваги та зміщення, вона використовує функцію вартості та підкріплення навчання, щоб досягти точки збіжності або локального мінімуму.

Процес, у якому алгоритм коригує свої вагові коефіцієнти, здійснюється за допомогою градієнтного спуску, що дозволяє моделі визначити напрямок для зменшення помилок (або мінімізації функції витрат). З кожним навчальним прикладом параметри моделі налаштовуються для поступового зближення до мінімуму.

Більшість глибоких нейронних мереж є прямими, тобто вони рухаються лише в одному напрямку, від входу до виходу. Однак ви також можете навчити свою модель за допомогою зворотного поширення; тобто рухатися в протилежному напрямку від виходу до входу. Зворотне розповсюдження дозволяє нам обчислити та приписати помилку, пов'язану з кожним нейроном, дозволяючи нам належним чином налаштувати та підігнати параметри моделі.

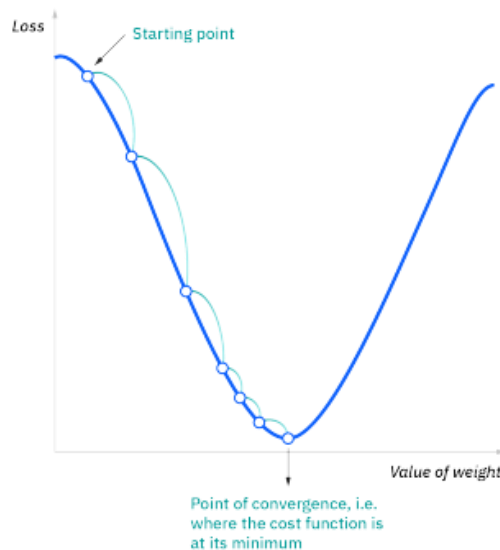


Рисунок 1.3 – Графічне представлення функції вартості нейронної мережі

1.4. Типи нейронних мереж

Нейронні мережі можна класифікувати на різні типи, які використовуються для різних цілей. Вони всі беруть свій початок з найпростішої нейронної системи – перцептрона.

Перцептрон — найстаріша нейронна мережа, створена Франком Розенблатом у 1958 році.

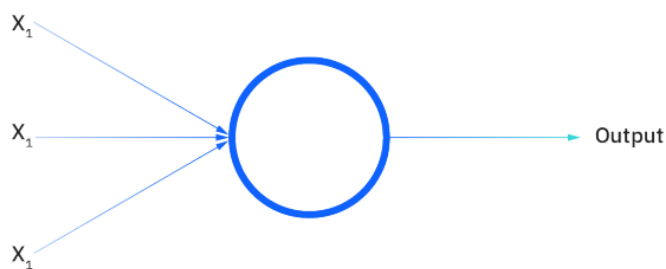


Рисунок 1.4 – Графічне представлення найпершої нейронної мережі

Це не повний перелік типів, але вони найпоширеніші типи нейронних мереж, які можна зустріти для типових випадків використання.

1. Штучні нейронні мережі

Штучні нейронні мережі (ШНМ) складаються з вузлів (перцептронів), які надихаються нейронами мозку. ШНМ має три рівні, а саме вхідний, прихований і вихідний. На основі алгоритму може бути більше одного прихованого шару. Вхідний рівень передає свій вихід до прихованого шару, і, аналогічно, кожен шар передає свій вихід до наступного шару, і, нарешті, вихідний рівень виводить результат[3].

ШНМ були дуже популярні, поки в 1990 році не було винайдено SVM. З розвитком рекурентних, прямого зв'язку та згорткових нейронних мереж ШНМ знову набула популярності в галузі кібербезпеки.

У ШНМ вхідні дані ($x_1, x_2, \dots, x_n$) надаються з вихідною міткою y , де інформація з вхідних даних зважується вектором ваги ($w_1, w_2, \dots, w_n$) під час процесу навчання. Протягом усього процесу навчання ваги регулюються таким чином, щоб мінімізувати похибку навчання, $E = \sum_{i=1}^n |d_i - y_i|$, де похибка є різницею між бажаним результатом (d_i) і фактичним результатом (y_i) нейрона.

Це коригування здійснюється за допомогою градієнтного алгоритму, який називається зворотним поширенням, де процес навчання повторюється туди-сюди, доки модель не отримає помилку, меншу за порогове значення.

Вектор ваги коригується відповідно до наступного рівняння:

$$w_{i,j} \leftarrow w_{i,j} + \Delta w_{i,j} \quad (1.5)$$

де $\Delta w_{i,j} = \eta \delta_j x_{i,j}$, i — вхідний вузол, а j — прихований вузол.

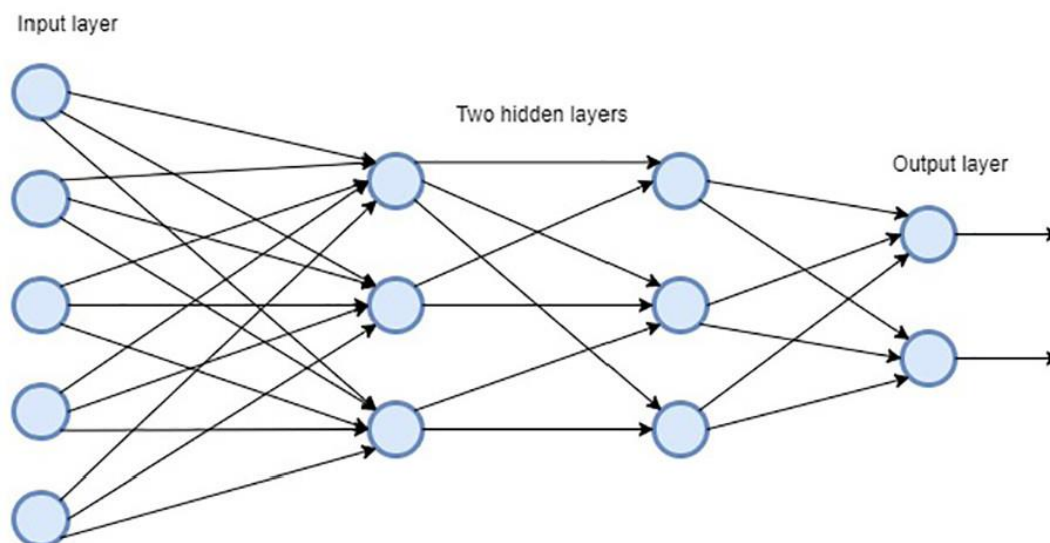


Рисунок 1.5 – Графічне представлення загальної архітектури ШНМ

Приклад архітектури штучної нейронної мережі наведено на рисунку 1.5. Вхідні вузли приймають вхідні дані на вхідному рівні, який обробляється вузлами прихованого шару, а вихідні дані передаються на вихідний рівень.

2. Згорткові нейронні мережі

Глибинне навчання (DL), підклас алгоритмів машинного навчання (ML), в основному використовується для обробки великих навчальних наборів даних за допомогою абстракції та представлення ознак на основі ієрархії. Продуктивність традиційних алгоритмів машинного навчання погіршується, коли набори даних дуже великі, а також через розмірність даних. Щоб вирішити цю проблему, DL використовується за допомогою графічних процесорів (GPU) для обробки великих даних. Серед усіх алгоритмів DL згорткова нейронна мережа (CNN) широко використовується в програмах кібербезпеки.

У CNN є два основних рівні: шар згортки та рівень об'єднання. Рівень згортки згортає вхідні дані за допомогою кількох ядер однакового розміру. Операція згортки отримує ознаки з вхідних даних, надаючи високі значення для заданої позиції, якщо потрібна функція присутня в цьому місці, і навпаки. Як приклад, для даних зображення ядро згортки виконує поелементне

множення кожного ядра значення комірки та відповідне значення пікселів зображення, що перекривається.

Для обчислення точного значення використовується наступна формула (де m — ширина та висота ядра, h — вихід згортки, x — вхід, а w — ядро згортки):

$$h_{ij} = \sum_{k=1}^m \sum_{l=1}^m w_{k,l} x_{i+k-1, j+l-1} \quad (1.6)$$

Рівень об'єднання використовується для зменшення вибірки розмірів об'єктів за допомогою двох типів методів об'єднання (тобто об'єднання максимального об'єднання, об'єднання середнього об'єму). Зокрема, max-pooling вибирає максимальне значення в функціях, обчислених на попередньому рівні, тоді як середнє об'єднання приймає середні значення.

Тобто, в межах ядра механізм об'єднання в конкретній позиції виводить максимальне значення даного вхідного даних, яке підпадає під ядро.

Для математичного представлення:

$$h_{i,j} = \max \{x_{i+k-1, j+l-1} \forall 1 \leq k \leq m, 1 \leq l \leq m\} \quad (1.7)$$

У CNN використовується інший рівень активації, найбільш часто використовуваним з яких є випрямлений лінійний блок, який включає вузли з функцією активації $f(x)=\max(0,x)$. На малюнку 14 показано графічне зображення принципу роботи алгоритму CNN. Серед усіх інших переваг обчислювальна вартість є одним із недоліків CNN.

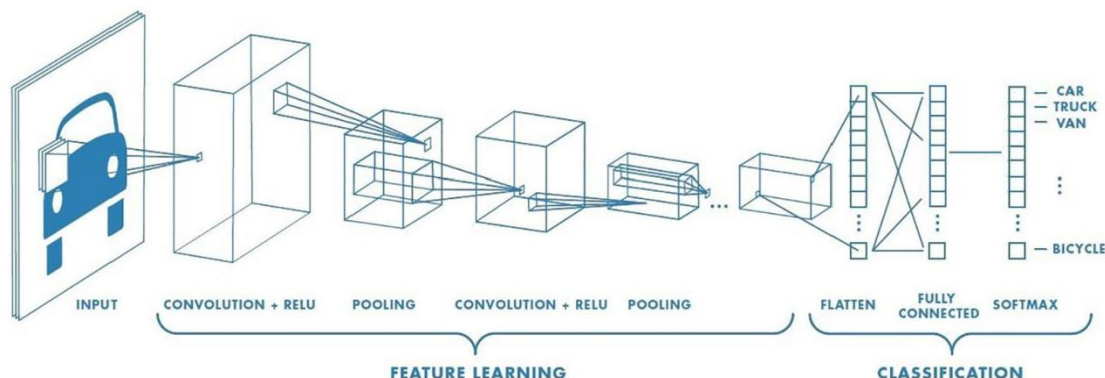


Рисунок 1.6 – Приклад згорткової нейронної мережі для класифікації зображень

Рівень згортки згортає вхідні дані за допомогою кількох ядер однакового розміру, рівень об'єднання використовується для зменшення вибірки розмірів ознак, а повністю підключений рівень зберігає згенероване значення ваги для прогнозування виходу на вихідному рівні(Рисунок 1.6).

Рекурентні(повторювані) нейронні мережі

Програми, де вихідні дані поточного стану залежать від стану кількох попередніх станів (тобто послідовних даних), традиційні алгоритми ML не забезпечують хорошу продуктивність, оскільки в традиційних алгоритмах ML немає взаємозалежності між вхідними та вихідними даними. .

Повторювана нейронна мережа (RNN), ще один алгоритм DL, особливо обробляє ці послідовні дані (наприклад, мовний текст, сенсорні дані) і показує кращу продуктивність, ніж будь-який інший алгоритм.

Подібно до нейронної мережі прямого зв'язку, RNN складається з трьох блоків: блоків введення, блоків прихованих і блоків виводу. Інформаційні потоки в RNN відбуваються лише одним шляхом: від вхідних блоків до прихованих блоків[4].

Основою перевагою рекурентних НМ є динамічність та ітераційність обробки даних, що в перспективі повинно позитивно вплинути на узагальнюючі та обчислювальні можливості. Однак потенційна нестійкість та недостатня дослідженість НМ довільної архітектури із зворотніми зв'язками є серйозними перешкодами для їх широкого застосування. Нестійкість НМ полягає в постійній зміні стану нейронів без виникнення стаціонарного стану мережі. Наслідком нестійкості може бути як колапс процесу навчання, так і невизначеність вихідної інформації в процесі розпізнавання невідомого образу.

Синтез цього одностороннього інформаційного потоку відбувається від попереднього послідовного замаскованого блоку до поточних прихованих блоків синхронізації. Вся інформація в RNN зберігається в прихованих блоках. Принцип роботи RNN показано на рисунку 1.7.

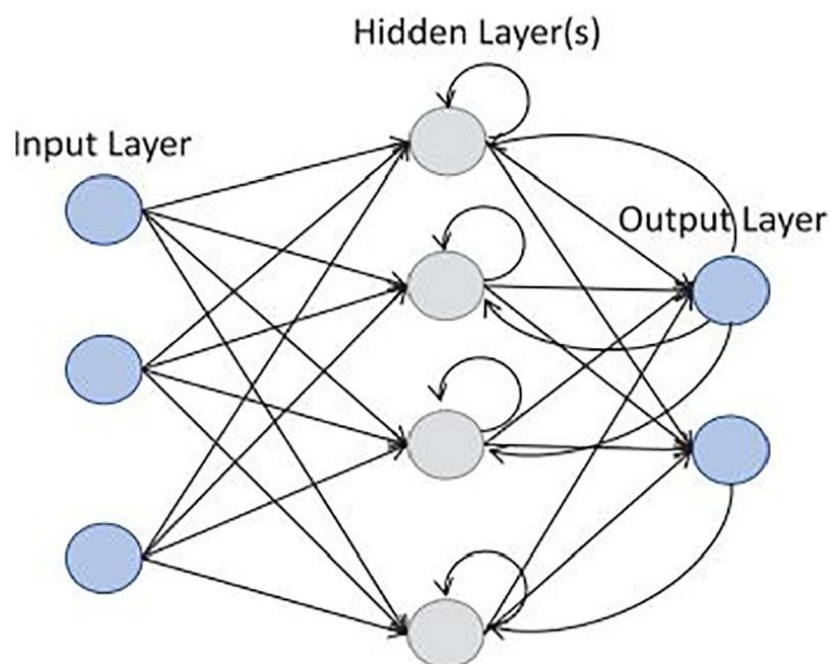


Рисунок 1.7 – Графічне представлення загальної архітектури рекурентних нейронних мереж

RNN обчислює приховану послідовність одиничного вектора $h=(h_1, h_2, \dots, h_T)$, щоб обчислити вихідний одиничний вектор $y=(y_1, y_2, \dots, y_T)$ через $t = 1$ до T ітерації наступного рівняння:

$$h_t = H(W_{xh}x_t + W_{hh}h_t + b_h) \quad (1.8)$$

$$y_t = W_{hy}h_t + b_y \quad (1.9)$$

де W представляє вагову матрицю, b є вектор зміщення, а H представляє рекурентну функцію прихованого шару.

Ці методи навчання глибокої нейронної мережі або їх сукупності та гібридні підходи можна використовувати для інтелектуального вирішення різних проблем кібербезпеки, таких як виявлення вторгнень, ідентифікація шкідливих програм або ботнетів, фішинг, прогнозування кібератак, наприклад, відмова в обслуговуванні (DoS), виявлення шахрайства або кібер-аномалії.

1.5. Аналіз існуючих методик виявлення аномалій і вторгнень з використанням методів машинного навчання

Система виявлення вторгнень (IDS) - це активний процес або пристрій, який аналізує діяльність системи та мережі на предмет несанкціонованої діяльності. Системи виявлення вторгнень (IDS), засновані на сигнатурному підході, зберігають архів відомих порушень і покладаються на зіставлення спостережуваної поведінки для виявлення зловживань. З іншого боку, детектори аномалій визначають патерн нормальної поведінки та сигналізують про будь-які відхилення від норми.

Підхід, що спирається на сигнатури, може бути надзвичайно ефективним проти відомих атак, коли відома «сигнатура зла». Проте триваюча гонка озброєнь породила ряд способів обійти цей підхід. Кіберзлочинці та хакери щодня шукають абсолютно нові та нестандартні способи атак, і тому розумно виявляти функції, що не працюють за межами «сигналу хорошого/нормального».

Аномалії - це шаблон у даних, який не відповідає чітко визначеній нормальній поведінці. Причиною аномалії може бути зловмисна діяльність або якесь вторгнення. Ця аномальна поведінка, виявлена в наборі даних, цікава аналітику, і це є найважливішою характеристикою для виявлення аномалій.

В цьому методі досягається більша моделювальна здатність, ніж у разі використання тривіальних частотних та instance-based методів. Вхідні дані розглядаються як потік дискретних подій, наприклад, системних викликів або ідентифікаторів процесів. Мета – отримати автомат, який моделює вказану послідовність подій. Для багатьох послідовностей характерно, що ймовірність наступного символу, елемента або сигналу залежить від попередніх. Часто вони залежать лише від невеликої кількості попередніх

Існує безліч підходів до розробки алгоритму пошуку аномалій. Але в кожному підході є основні незмінні етапи(рис.1.8).

Етап параметризації - це попередня обробка даних у заздалегідь встановлені формати, які будуть прийняті та відповідатимуть поведінці цільової системи.

Етап навчання - модель побудована на основі нормальної (або ненормальної) поведінки системи. Способи навчання різняться в залежності від типу розглянутого методу виявлення аномалії. Може налаштовуватися вручну або автоматичним методом.

Етап виявлення - якщо модель для системи доступна, її порівнюють із (параметризованим або заздалегідь визначеним) спостережуваним трафіком. Якщо знайдена аномалія, перевіряється чи перевищує (або є меншою, від зазначених моделей відхилень) заздалегідь визначений поріг значень. При виявленні не очікуваного результату спрацює сигнал тривоги.

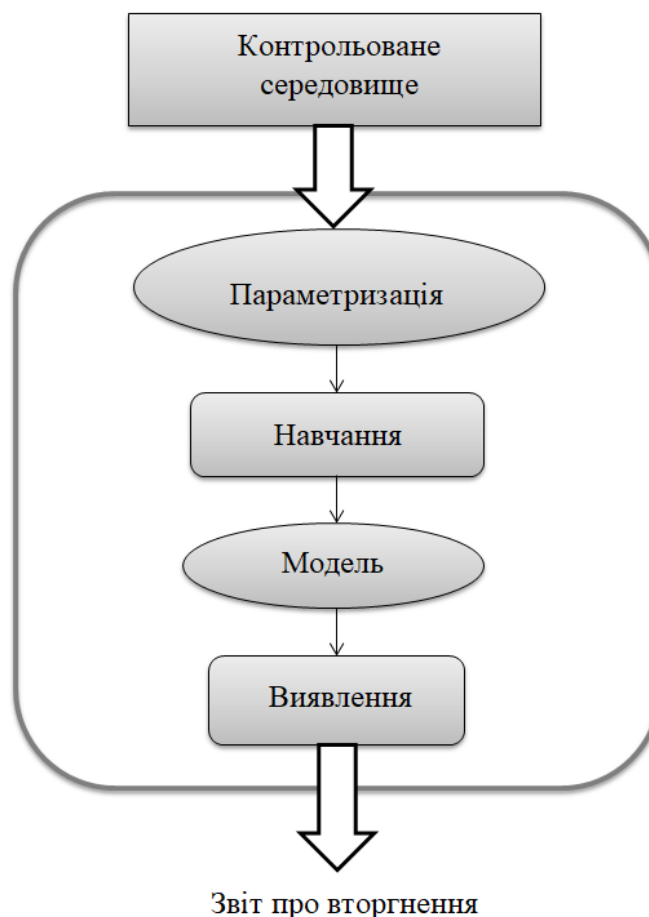


Рисунок 1.8 – Методологія виявлення аномалій

1.6. Аналіз аналітичних можливостей хмарного середовища Azure з використанням машинного навчання

Машинне навчання Azure (Azure ML) — це хмарна служба для створення рішень машинного навчання та керування ними. Його розроблено, щоб допомогти науковцям із обробки даних та інженерам із машинного навчання використовувати наявні навички та фреймворки обробки даних і розробки моделей. Крім того, допомагає масштабувати, розподіляти та розгортати робочі навантаження в хмарі. Azure ML SDK для Python містить класи, які ми можемо використовувати для роботи з Azure ML у передплаті Azure[5].

Azure Machine Learning — це хмарний сервіс для прискорення життєвого циклу проекту машинного навчання та керування ним. Фахівці з машинного навчання, спеціалісти з обробки даних та інженери можуть використовувати його у своїх повсякденних робочих процесах: навчати та розгортати моделі та керувати MLO.

Можна створити модель у машинному навчанні Azure або використати модель, побудовану на платформі з відкритим кодом, як-от Pytorch, TensorFlow або scikit-learn. Інструменти MLOps допомагають контролювати, перенавчати та повторно розгортати моделі.

Машинне навчання Azure призначене для окремих осіб і команд, які впроваджують MLO у своїй організації, щоб запровадити моделі машинного навчання у виробництво в безпечному виробничому середовищі, яке можна перевірити.

Вчені з обробки даних та інженери з машинного навчання знайдуть інструменти для прискорення та автоматизації своїх повсякденних робочих процесів. Розробники програм знайдуть інструменти для інтеграції моделей у програми чи служби. Розробники платформи знайдуть надійний набір інструментів, підкріплений довговічними API Azure Resource Manager, для створення розширених інструментів ML.

Підприємства, які працюють у хмарі Microsoft Azure, знайдуть знайому безпеку та керування доступом на основі ролей (RBAC) для інфраструктури. Ви можете налаштувати проект, щоб заборонити доступ до захищених даних і вибрати операції.

Для створення та підтримки проектів машинного навчання часто потрібна команда з різними навичками. Машинне навчання Azure має інструменти, які допомагають вам:

1. Співпрацювати зі своєю командою за допомогою спільних блокнотів, обчислювальних ресурсів, даних і середовищ.
2. Розробляти моделі справедливості та пояснюваності, відстеження та перевірки, щоб відповідати вимогам щодо походження та аудиту.
3. Швидко та легко розгортати моделі ML у великих масштабах, а також ефективно керуйте ними за допомогою MLOps.
4. Виконуйте робочі навантаження машинного навчання будь-де за допомогою вбудованого керування, безпеки та відповідності.

Машинне навчання Azure інтегрується з хмарною платформою Azure для підвищення безпеки проектів машинного навчання.

Інтеграції безпеки включають:

1. Віртуальні мережі Azure (VNets) із групами безпеки мережі.
2. Azure Key Vault, де ви можете зберігати секрети безпеки, наприклад інформацію про доступ до облікових записів зберігання.
3. Реєстр контейнерів Azure, налаштований за віртуальною мережею.

Фахівці з даних можуть використовувати моделі в машинному навчанні Azure, які вони створили в поширених фреймворках Python, наприклад:

- PyTorch;
- TensorFlow;
- scikit-learn;
- XGBoost;
- LightGBM.

Машинне навчання Azure підтримує оцінку в режимі реального часу та пакетну оцінку (виведення).

Пакетна оцінка або пакетний висновок передбачає виклик кінцевої точки з посиланням на дані. Пакетна кінцева точка виконує завдання асинхронно, щоб паралельно обробляти дані на обчислювальних кластерах і зберігати дані для подальшого аналізу[6].

1.7. Постановка завдання

Проведений аналіз показав що для вирішення проблем кібербезпеки необхідне використання методу машинного навчання, зокрема методу нейронної мережі.

В інформаційній безпеці нейронні мережі - це програмне забезпечення, здатне розпізнавати чи відбуваються в досліджуваному середовищі аномальні події і самостійно вживати необхідних заходів. Штучний інтелект особливо добре справляється з розпізнаванням закономірностей і незвичайної поведінки, тому ШІ бути прекрасним інструментом побудови програмного забезпечення для виявлення аномалій та загроз.

У якості методу виявлення аномалій будуть використовуватись метод класифікації (із застосуванням алгоритму нейронних мереж).

Класифікація - це метод який відносить екземпляр набору даних, що розглядається, до певного класу відповідно до заданої ознаки. Він може бути ефективним як для виявлення зловживань, так і для виявлення аномалій. Класифікацію будемо проводити з використанням конструктора Машинного навчання Azure.

Необхідно побудувати загальну модель з методом класифікації нейронної мережі в Конструкторі конвеєра Машинного навчання Azure, навчити її на вхідних даних, оцінити якість виявлення аномалій і порівняти отриманий результат з відомими результатами методів класифікації.

РОЗДІЛ 2. РОЗРОБКА ХМАРНОГО РІШЕННЯ ЗАДАЧІ КЛАСИФІКАЦІЇ НА ОСНОВІ КОНСТРУКТОРА МАШИННОГО НАВЧАННЯ AZURE

2.1. Розробка архітектури рішення з використанням студії машинного навчання Azure

Azure Machine Learning — це хмарний сервіс для прискорення життєвого циклу проекту машинного навчання та керування ним. Фахівці з машинного навчання, спеціалісти з обробки даних та інженери можуть використовувати його у своїх повсякденних робочих процесах: навчати та розгортати моделі та керувати MLO.

Можна створити модель у машинному навчанні Azure або використати модель, побудовану на платформі з відкритим кодом, як-от Pytorch, TensorFlow або scikit-learn. Інструменти MLOps допомагають контролювати, перенавчати та повторно розгортати моделі. Для роботи буде використовуватися досвід низької кількості коду.

МН Azure має перелік компонентів які взаємодіють між собою в процесі компіляції, розгортання та підтримки моделей машинного навчання(Рис.2.1)

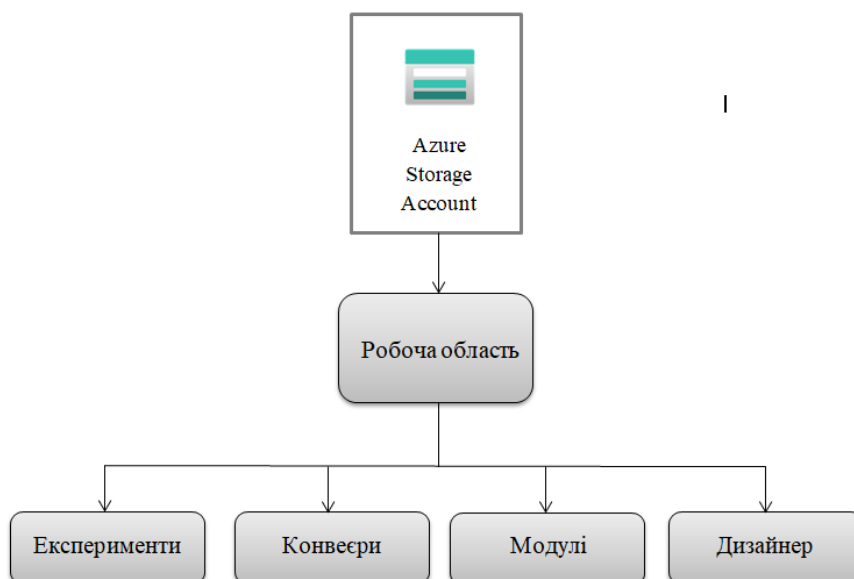


Рисунок 2.1 – Компоненти для роботи Azure Machine Learning

Середовище машинного навчання Azure – це інкапсуляція середовища, у якому виконується ваше завдання машинного навчання. Вони визначають пакети програмного забезпечення, змінні середовища та параметри програмного забезпечення навколо ваших сценаріїв навчання та підрахунку балів. Середовища є керованими та версійними об’єктами у вашій робочій області машинного навчання. Середовища дозволяють відтворювати, перевіряти та переносити робочі процеси машинного навчання на різних обчисленнях.

Конвеєр машинного навчання — це робочий процес для повного завдання машинного навчання. Компоненти є будівельними блоками конвеєра машинного навчання. Коли ви думаєте про компонент, він повинен бути в контексті конвеєра.

Щоб побудувати компоненти, перше, що потрібно, це визначити конвеєр машинного навчання. Для цього потрібно розбити повне завдання машинного навчання на багатоетапний робочий процес. Кожен крок є складовою.

Наприклад, розглядаючи просте завдання машинного навчання використання історичних даних для навчання моделі прогнозування продажів, ми можемо побудувати послідовний робочий процес з обробкою даних, навчанням моделі та етапами оцінки моделі. Для складних завдань ви можете розбити далі. Наприклад, розділемо один крок обробки даних на прийом даних, очищення даних, попередню обробку даних і кроки розробки функцій.

Код для завдання повинен мати можливість виконуватися командою оболонки. Під час розробки ми можемо додати кілька вхідних даних, щоб контролювати, як цей крок буде виконуватися. Наприклад, для етапу навчання можна додати швидкість навчання, кількість епох як вхідні дані для керування навчанням. Ці додаткові входи плюс входи та виходи, необхідні для з’єднання з іншими етапами, є інтерфейсом компонента. Аргумент команди оболонки використовується для передачі входів і виходів у код[7].

Необхідно вказати середовище для виконання команди та код. Середовищем може бути середовище AzureML.

Нарешті, ми можемо упакувати все, включаючи код, cmd, середовище, введення, виходи, метадані разом у компонент. Потім з'єднає ці компоненти разом, щоб створювати конвеєри для робочого процесу машинного навчання. Один компонент можна використовувати в кількох конвеєрах.

Узагальюючи все вище перераховане, можна скласти життєвий цикл проекту машинного навчання Azure(рис.2.2).

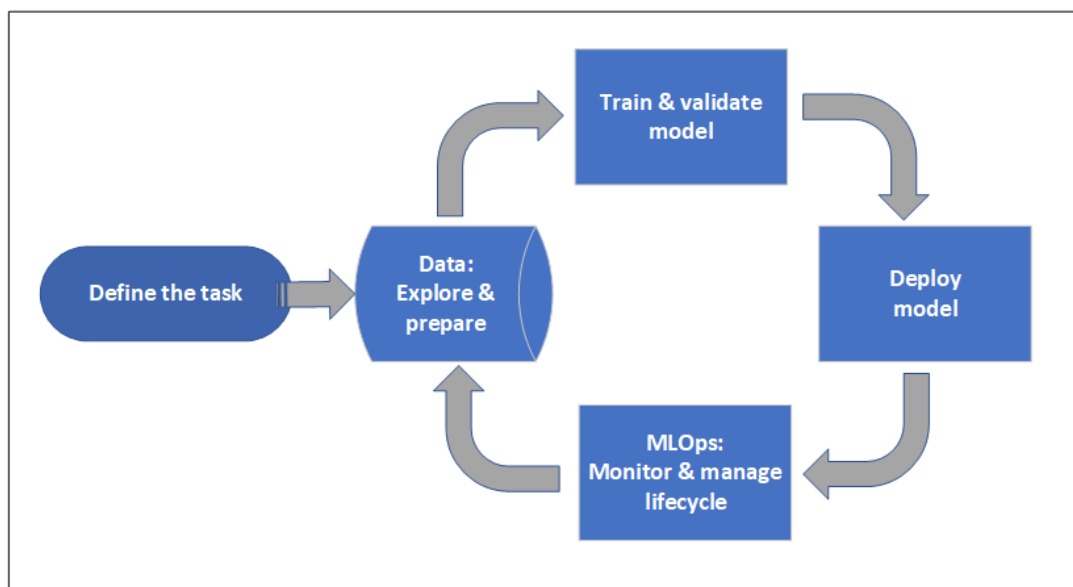


Рисунок 2.2 – Життєвий цикл проекту Azure Machine Learning

Хоча життєвий цикл проекту може відрізнятися залежно від проекту, часто він виглядає як на рисунку.

Робочий простір організовує проект і дає змогу багатьом користувачам працювати над спільною метою. Користувачі в робочій області можуть легко ділитися результатами своїх експериментів в інтерфейсі користувача студії або використовувати активи з версіями для таких завдань, як середовища та посилання на сховище.

Коли проект готовий до введення в дію, роботу користувачів можна автоматизувати в конвеєрі машинного навчання та запускати за розкладом або запитом HTTPS.

Моделі можна розгорнути в керованому рішенні для логічного висновку як для розгортання в режимі реального часу, так і для пакетного розгортання,

абстрагуючись від керування інфраструктурою, яке зазвичай вимагається для розгортання моделей.

Для подальшої роботи потрібно розробити життєвий цикл моделі ML(рис.2.3).

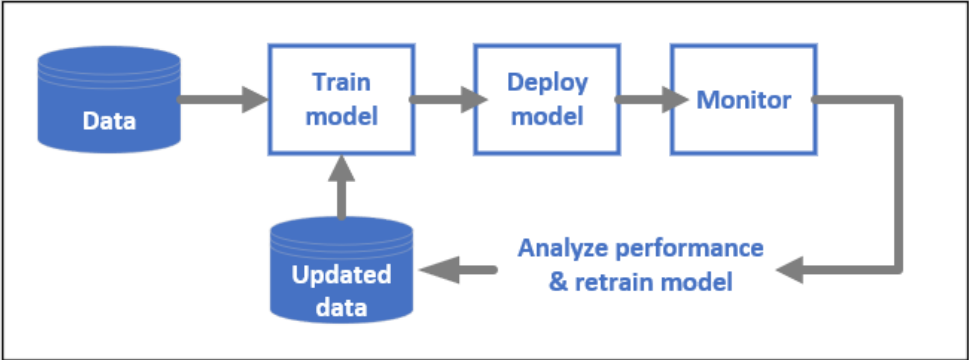


Рисунок 2.3 – Життєвий цикл моделі Machine Learning
Після цих етапів потрібно створити Azure Storage Account(рис.2.4).

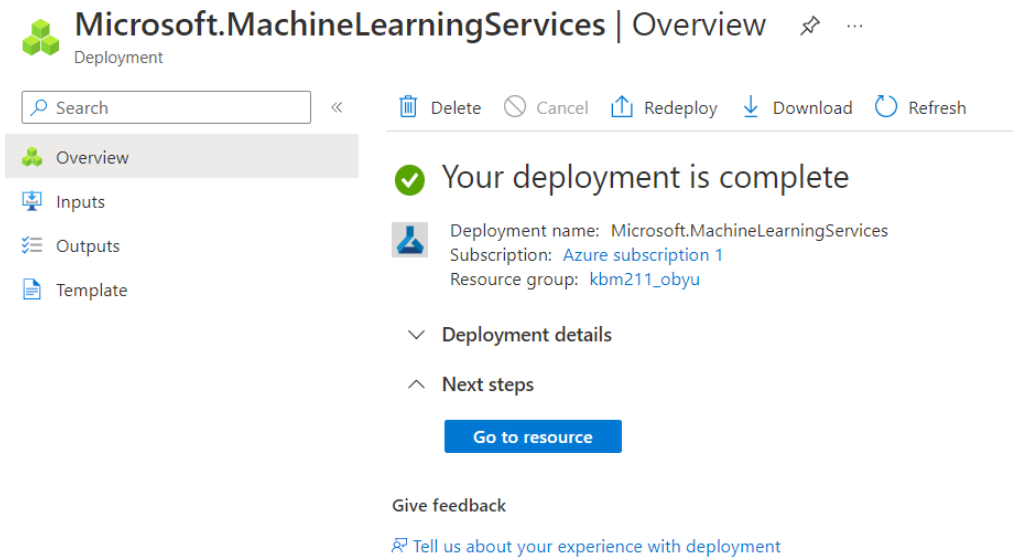


Рисунок 2.4 – Створення Azure Storage Account

Далі для створення робочої області потрібно увійти в портал Azure за допомогою облікових даних підписки Azure та створити ресурс. Після цього, використовуючи сервіс Azure машинне навчання, відкриваємо вікно налаштувань нової робочої області. Як тільки закінчуємо налаштування параметрів робочої області натискаємо Перевірити і створити(рис.2.5).

Azure Machine Learning ...

Create a machine learning workspace

Subscription * ⓘ

Resource group * ⓘ

[Create new](#)

Workspace details

Configure your basic workspace settings like its storage connection, authentication, container, and more. [Learn more](#)

Workspace name * ⓘ

Region * ⓘ

Storage account * ⓘ

[Create new](#)

Key vault * ⓘ

[Create new](#)

Application insights * ⓘ

[Create new](#)

Container registry * ⓘ

[Create new](#)

[Review + create](#) [< Previous](#) [Next : Networking](#)

Рисунок 2.5 - Створення робочої області

Для створення конвеєра потрібно увійти на сайт ml.azure.com та вибрати робочу область, з якою ви хочете працювати. Після цього потрібно назвати конвеєр використовуючи значок “Налаштування” і рядок “Назва проекту”. Також конвеєр працює на цілі обчислення, яка є обчислювальним ресурсом, приєднаним до вашої робочої області. Для її створення потрібно в зоні “Налаштування”, яка відкриється справа від робочої області, вибрати створити ціль обчислення.

Після створення цілі обчислення ви можете використовувати її повторно для подальших запусків. Після створення ресурсу ви можете використовувати його повторно та пропустити цей час очікування для подальших запусків.

Обчислюваний ресурс автоматично масштабується до нульових вузлів, коли він не працює, щоб заощадити витрати. При повторному використанні з затримкою може виникнути приблизно п'ять хвилин очікування, поки він масштабується.

Після цього робоча область конвеєра готова до роботи з конструктором.

2.2. Аналіз вхідних даних та їх попередня обробка

Для навчання моделей використовувати набір даних KDD Cup.

Набір даних KDD Cup 99, отриманий із набору даних оцінки DARPA IDS (Lippmann та ін., 1998), використовувався для змагань KDD Cup 99 (набір даних KDD Cup 99, 2009) — це приблизно 4 гігабайти стиснутих необроблених (двійкових) даних tcpdump 7 тижнів мережевого трафіку, які можна обробити в приблизно 5 мільйонів записів з'єднання, кожна з яких має близько 100 байтів. Два тижні тестових даних містять близько 2 мільйонів записів про підключення, і кожен запис представляє TCP/IP-з'єднання, яке складається з 41 функції, які мають як якісний, так і кількісний характер (Stolfo, Fan, Lee, Prodomidis, & Chan, 2000).

Починаючи з 1999 року KDD99 був найбільш широко використовуваним набором даних для оцінки методів виявлення аномалій. Навчальний набір даних KDD складається приблизно з 4 900 000 одиничних векторів з'єднання, кожен з яких містить 41 функцію та позначається як звичайний або атакуючий з точно одним типом атаки.

Змодельовані атаки належать до однієї з наступних чотирьох категорій:

1. Атака на відмову в обслуговуванні (DoS): це атака, під час якої зловмисник робить певний обчислювальний ресурс або ресурс пам'яті занадто зайнятим або занадто повним для обробки законних запитів, або відмовляє законним користувачам у доступі до машини.

2. User to Root Attack (U2R): це клас експлойтів, у яких зловмисник починає з доступу до звичайного облікового запису користувача в системі (можливо, отриманого шляхом перехоплення паролів, атаки за словником або соціальної інженерії) і може використати деякі вразливості для отримання кореневого доступу до системи

3. Віддалена локальна атака (R2L): виникає, коли зловмисник, який має можливість надсилати пакети на машину через мережу, але не має

облікового запису на цій машині, використовує вразливість, щоб отримати локальний доступ до цієї машини.

4. Зондувальна атака: це спроба зібрати інформацію про мережу комп'ютерів з явною метою обійти засоби її безпеки.

Важливо зазначити, що тестові дані не з того самого розподілу ймовірностей, що й навчальні дані, і включають конкретні типи атак, яких немає в навчальних даних, що робить завдання більш реалістичним. Деякі експерти з вторгнень вважають, що більшість нових атак є варіантами відомих атак, і сигнатури відомих атак можуть бути достатніми для виявлення нових варіантів. Набори даних містять загальну кількість 24 типів тренувальних атак, з додатковими 14 типами лише в тестових даних.

Функції KDD99 можна розділити на три групи:

1) Основні функції: ця категорія інкапсулює всі атрибути, які можна отримати з підключення TCP/IP. Більшість із цих функцій призводять до неявної затримки виявлення.

2) Функції трафіку: ця категорія включає функції, які обчислюються відносно віконного інтервалу та розділені на дві групи:

– Функції «той самий хост»: перевіряйте лише з'єднання за останні 2 секунди, які мають той самий хост призначення, що й поточне з'єднання, і обчислюйте статистику, пов'язану з поведінкою протоколу, сервісом тощо;

– функції «однакова служба»: перевіряйте лише з'єднання за останні 2 секунди, які мають ту саму послугу, що й поточне з'єднання.

Два вищезазначені типи функцій «трафіку» називаються часовими. Однак існує кілька атак повільного зондування, які сканують хости (або порти) з використанням набагато більшого інтервалу часу, ніж 2 секунди, наприклад, кожну хвилину. Як результат, ці атаки не створюють шаблонів вторгнень із часовим вікном у 2 секунди. Щоб вирішити цю проблему, функції «той самий хост» і «та сама служба» обчислюються повторно, але на основі вікна підключення 100 підключень, а не часового вікна 2 секунди. Ці функції називаються функціями трафіку на основі з'єднання.

3) Функції вмісту: на відміну від більшості атак DoS і Probing, атаки R2L і U2R не мають послідовних шаблонів вторгнення. Це пояснюється тим, що атаки DoS і Probing передбачають багато підключень до деяких хостів за дуже короткий проміжок часу; однак атаки R2L і U2R вбудовані в частини даних пакетів і зазвичай охоплюють лише одне з'єднання. Щоб виявити такі види атак, нам потрібні деякі функції, щоб мати можливість шукати підозрілу поведінку в частині даних, наприклад, кількість невдалих спроб входу. Ці функції називаються функціями вмісту[8].

Для кращого розуміння, деякий перелік набору функцій приведений в табл. 2.1

Таблиця 2.1

Основні функції окремих з'єднань TCP

Назва функції	Опис	Тип
duration	тривалість (кількість секунд) з'єднання	безперервна
protocol_type	тип протоколу, напр. tcp, udp тощо	дискретна
service	послуга мережі за призначенням, наприклад, http, telnet тощо.	дискретна
src_bytes	кількість байтів даних від джерела до пункту призначення	безперервна
dst_bytes	кількість байтів даних від пункту призначення до джерела	безперервна
flag	нормальний або стан помилки з'єднання	дискретна
land	1 якщо з'єднання здійснюється з / до одного хосту / порту; 0 інакше	дискретна
wrong_fragment	кількість ``неправильних'' фрагментів	безперервна
urgent	number of urgent packets	безперервна

Для навчання моделі буде використовуватися набір даних – intrusion-detection-KDD[9].

З цього набору даних буде використовуватися 42 стовпця и 125973 рядки.

Для завантаження та попередньої обробки даних використовуємо модулі Конструктора машинного навчання Azure:

1. Import Data - цей модуль використовується для завантаження даних у конвеєр машинного навчання із існуючих хмарних служб даних.

Компонент «Імпорт даних» пропускає реєстрацію вашого набору даних у машинному навчанні Azure та імпортує дані безпосередньо зі сховища даних або URL-адреси HTTP.

Після того, як ви визначите потрібні дані та підключитесь до джерела, Імпорт даних визначає тип даних кожного стовпця на основі значень, які він містить, і завантажує дані у ваш конструктор. Результатом імпорту даних є набір даних, який можна використовувати з будь-яким конструкторським конвеєром(рис.2.6).

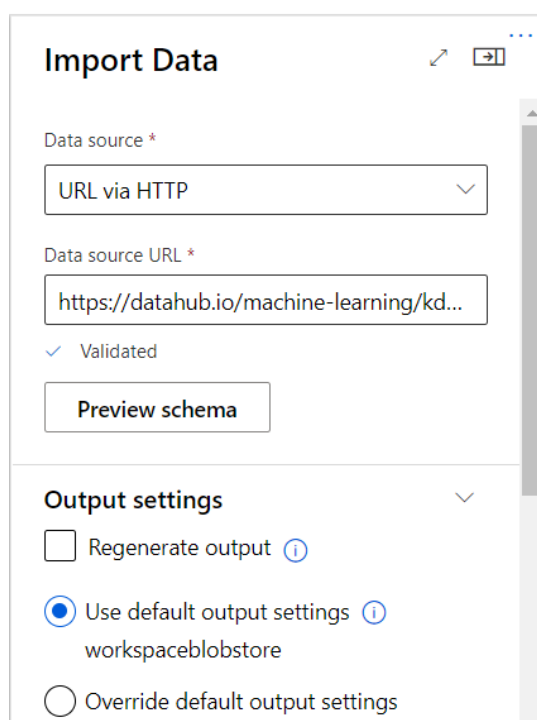


Рисунок 2.6 - Налаштування модуля для роботи з набором даних

Після імпорту даних знадобитися додаткова підготовка для моделювання та аналізу:

Використовуймо «Редагувати метадані», щоб змінити назви стовпців, обробити стовпець як інший тип даних або вказати, що деякі стовпці є мітками чи функціями. Типові зміни метаданих можуть включати:

Edit Metadata використовується в будь-який час, коли потрібно змінити визначення стовпця. Установивши ці метадані, ви можете переконатися, що стовпець ідентифіковано правильно. Налаштування цього параметра не змінює значення даних. Це змінює лише спосіб обробки даних деякими алгоритмами машинного навчання..(рис. 2.7)

Edit Metadata

Column *i ** [Edit column](#)

Column names: class

Data type *i **

Unchanged

Categorical *i **

Categorical

Fields *i **

Unchanged

New column names *i* ...

Output settings >

Run settings >

Рисунок 2.7 - Налаштування модуля для роботи з набором даних

Convert to Indicator Values - цей модуль використовується, щоб «Перетворити у значення показників» у конструкторі машинного навчання Azure, щоб перетворити стовпці, що містять категоріальні значення, у ряд двійкових стовпців-показників.

Компонент повертає два виходи:

- 1) Набір даних результатів: набір даних із стовпцями перетворених значень індикаторів. Невідібрані для очищення стовпці також «пропускаються»[10].

2) Перетворення значень індикатора: перетворення даних, яке використовується для перетворення в значення індикатора, які можна зберегти у вашій робочій області та застосувати до нових даних пізніше(рис. 2.8).

Convert to Indicator Val... [↗](#) [📄](#)

Categorical columns to convert ⓘ * [Edit column](#)

Column names: class

Overwrite categorical columns ⓘ

False ▾

Output settings >

Run settings >

Node information >

Component information >

Рисунок 2.8 - Налаштування модуля для роботи з набором даних

Select Columns in Dataset- цей модуль використовується, щоб вибрати підмножину стовпців для використання в подальших операціях. Цей компонент корисний, коли вам потрібно обмежити стовпці, доступні для подальшої операції, або якщо ви хочете зменшити розмір набору даних, видаливши непотрібні стовпці.

Цей компонент не має параметрів. Ви використовуєте селектор стовпців, щоб вибрати стовпці, які потрібно включити або виключити. (рис. 2.9).

Якщо ви підключили набір даних, який уже заповнений, має з'явитися список доступних стовпців. Якщо стовпці не відображаються, можливо, вам

знадобиться запуснути вищестоящі компоненти, щоб переглянути список стовпців.

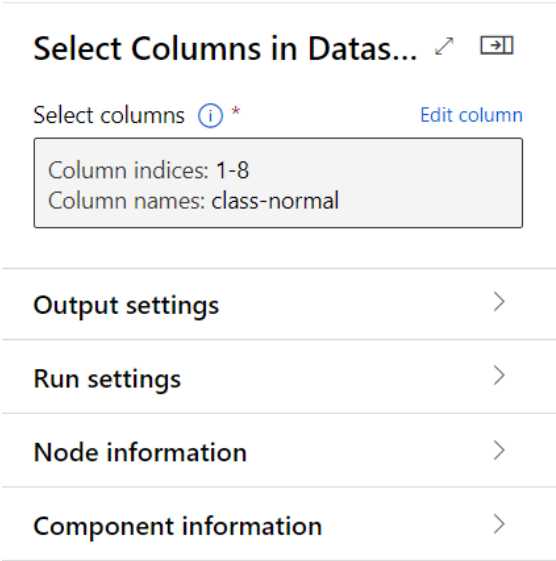


Рисунок 2.9 - Налаштування модуля для роботи з нашим набором даних

В результаті налаштувань в робочому просторі конвеєра ми отримали результат підключення даних(рис. 2.10).

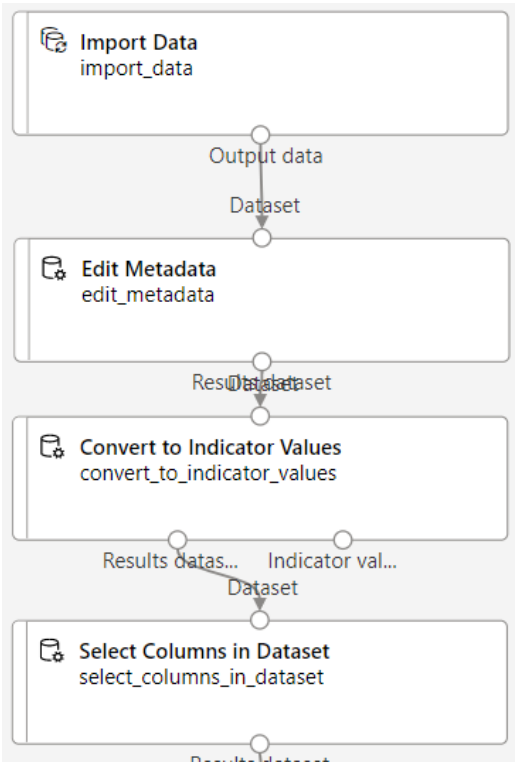


Рисунок 2.10 - Модель завантаження та попередньої обробки даних

Після проведених налаштувань можна перевірити правильність завантаження даних(рис. 2.11)

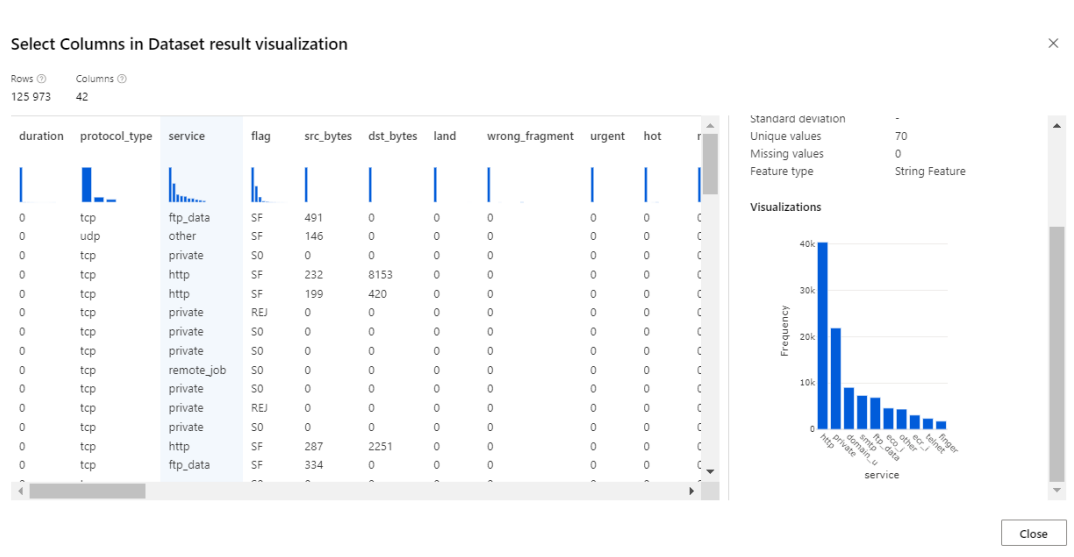


Рисунок 2.11 – Перевірка підвантажених даних

2.3. Розробка моделі для класифікації на основі нейронної мережі

Конструктор Машинного навчання Azure має достатньо модулів двокласової класифікації, які можуть бути використані для визначення аномалій. І серед них є модуль двох-класової нейронної мережі[11].

Нейронна мережа (Two-Class Neural Network module) - класифікація за допомогою нейронних мереж є контрольованим методом навчання, і тому потрібен тегований набір даних, який включає стовпець міток.

Використаємо цей компонент для створення моделі нейронної мережі, яку можна використовувати для прогнозування цілі, яка має лише два значення.

Класифікація за допомогою нейронних мереж — це контрольований метод навчання, тому потрібен тегований набір даних, який містить стовпець міток.

Після того як визначимо модель, навчимо її, надавши тегований набір даних і модель як вхідні дані для моделі навчання. Потім навчену модель можна використовувати для прогнозування значень для нових вхідних даних.

Навчання моделі здійснюється за допомогою модуля «Навчання моделі» (Train Model). Використовуйте цей компонент для навчання моделі класифікації або регресії. Навчання відбувається після того, як ви визначили модель і встановили її параметри, і потребує тегованих даних. Ви також можете використовувати модель навчання, щоб перенавчити існуючу модель новими даними[12].

Далі потрібно винести Score Model для створення прогнозів за допомогою навченої класифікації або регресійної моделі. Після того, як ви створили набір балів за допомогою Score Model:

Щоб створити набір показників, які використовуються для оцінки точності (продуктивності) моделі, ви можете підключити оцінений набір даних до Evaluate Model,

Клацніть компонент правою кнопкою миші та виберіть «Візуалізувати», щоб переглянути зразок результатів.

Оцінка або прогнозоване значення може мати багато різних форматів залежно від моделі та ваших вхідних даних:

Для моделей класифікації Score Model виводить прогнозоване значення для класу, а також імовірність прогнозованого значення.

Для моделей регресії Score Model генерує лише прогнозоване числове значення.

Модуль «Оцінити модель» (*Evaluate Model*) використовується для вимірювання точності навченої моделі.

Якщо для вибору частини вхідного набору даних використовуються такі компоненти, як «Вибір стовпців у наборі даних», переконайтеся, що стовпець «Фактична мітка» (використовується під час навчання), стовпець «Оцінені ймовірності» та стовпець «Оцінені мітки» існують для обчислення таких показників, як AUC, точність для двійкової класифікації / виявлення аномалії. Стовпець фактичних міток, стовпець «Оцінені мітки» існують для обчислення показників для класифікації/регресії за кількома класами. Стовпець «Призначення», стовпці «DistancesToClusterCenter no.X» (X — індекс

центроїда, починаючи від 0, ..., кількість центроїдів-1) існують для обчислення показників для кластеризації.(рис. 2.12).

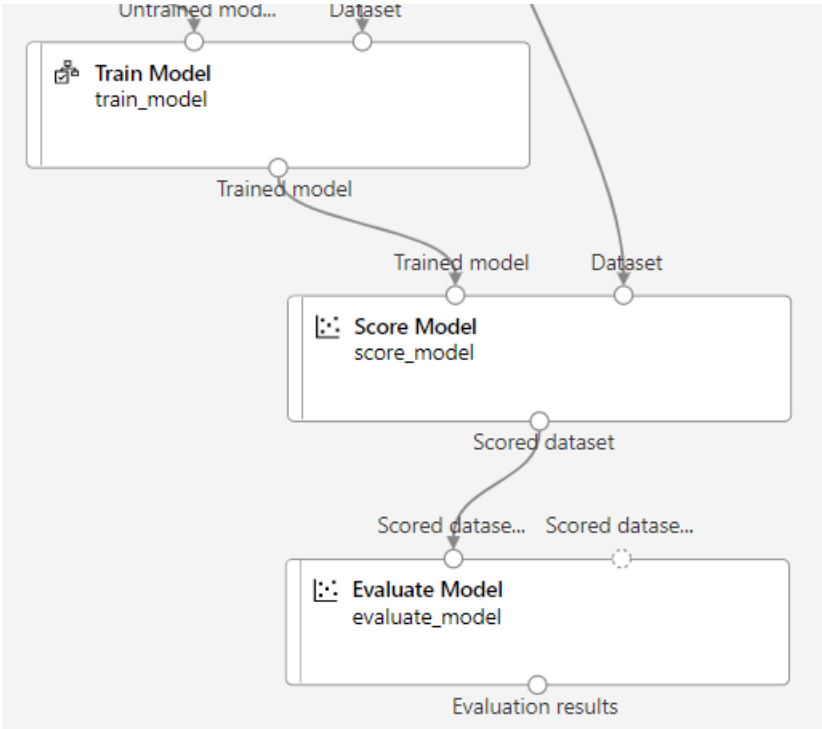


Рисунок 2.12 - Приклад підключення модуля Навчання моделі і Оцінки моделі

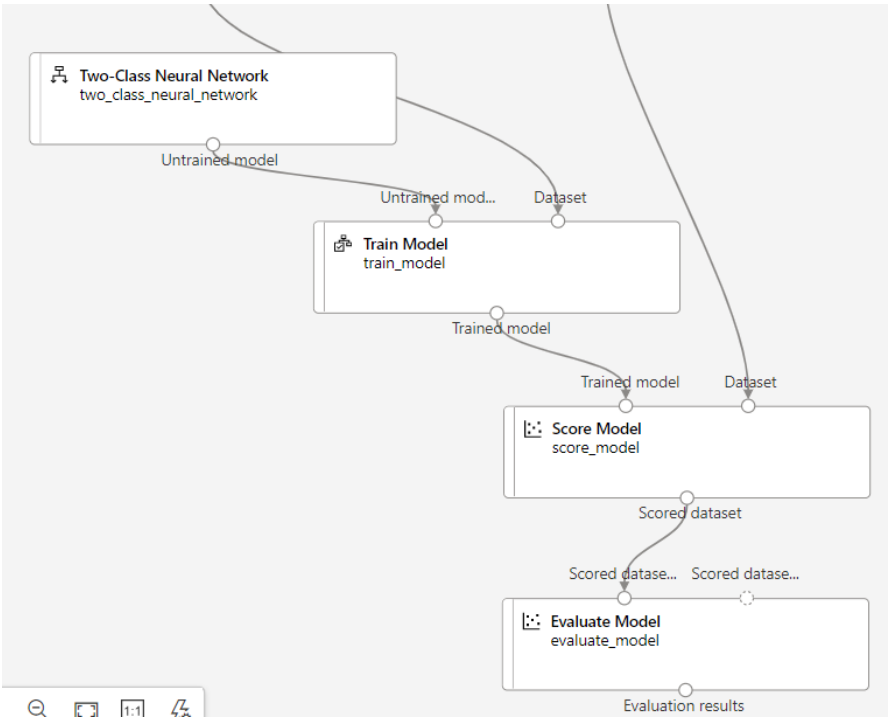


Рисунок 2.13 – Загальний вигляд з’єднання модулів

РОЗДІЛ 3. ПЕРЕВІРКА ПРАЦЕЗДАТНОСТІ ТА ЕФЕКТИВНОСТІ РІШЕННЯ ЗАДАЧІ КЛАСИФІКАЦІЇ

3.1. Показники якості класифікації при виявленні аномалій

Оцінка якості моделі може поділятися на:

- регресійна модель (безперервний вихід);
- модель класифікації (номінальний або двійковий вихід).

Показники, що використовуються в кожній із цих моделей, різні[13].

У задачах класифікації використовуються два типи алгоритмів:

1. Результат класом. Алгоритми, такі як SVM та KNN, створюють вихід класом. Наприклад, у двійковій задачі класифікації виходи будуть або 0, або 1. Однак є алгоритми, які можуть перетворити ці результати класу на ймовірність. Але ці алгоритми погано сприймаються статистичним співтовариством.

2. Результат вірогідністю. Алгоритми, такі як логістична регресія, випадковий ліс, посилення градієнта, дають результати імовірності. Перетворення вихідних даних вірогідності у вихід - це лише питання створення порогової ймовірності.

Проблеми моделі регресії не мають таких невідповідностей у результатах. Вихід завжди має безперервний характер і не потребує подальшої обробки.

В даній роботі використовуються алгоритми двокласової класифікації тому показники якості будуть отримуватись за допомогою моделі класифікації.

Перший спосіб оцінки це матриця плутанини. Матриця плутанини - це матриця $N \times N$, де N - кількість прогнозованих класів. Для розглянутої задачі маємо $N = 2$, а отже, отримуємо матрицю 2×2 . Після проведення класифікації можливе отримання чотирьох видів результатів:

TP (True Positive – істинно позитивний),

TN (True Negative – істинно негативний),

FP (False Positive – псевдо позитивний),

FN (False Negative – псевдо негативний).

Вигляд матриці плутанини

	Прогнозовано НІ	Прогнозовано ТАК
Насправді НІ	істинно негативний(TN)	Псевдо позитивний(FP)
Насправді ТАК	Псевдо негативний(FN)	Істинно позитивний(TP)

Визначень, для матриці плутанини:

- акуратність(Accuracy) - частка загальної кількості правильних прогнозів;

$$\text{Accuracy} = \frac{TP+TN}{Totalinstances} \quad (3.1)$$

- точність(Precision) - показує, як часто насправді передбачено правильно результат;

$$\text{Precision} = \frac{TP}{TotalPredictedYES} \quad (3.2)$$

- відкликання(повнота) або чутливість(Recall or Sensitivity) - частка фактично позитивних випадків, які були правильно визначені;

$$\text{Recall} = \frac{TP}{TotalActuallyYES} \quad (3.3)$$

- специфічність(Specificity) - частка фактично негативних випадків, які були правильно визначені.

$$\text{Specificity} = \frac{TP}{TotalActuallyNO} \quad (3.3)$$

Загалом, ми маємо справу з однією із визначених вище метрик. Наприклад, у фармацевтичній компанії їх більше турбує мінімальний неправильний позитивний діагноз. Отже, їх більше турбуватиме висока специфічність. З іншого боку, модель стирання більше стосуватиметься чутливості. Матриці плутанини зазвичай використовуються лише з моделями виводу класу[14].

F1-оцінка є середнім гармонічним значенням точності та відкликання для класифікаційної задачі.

Обчислюється як середньозважене значення точності та відкликання від 0 до 1, де ідеальне значення F-оцінки дорівнює 1.

$$F_1 = \left(\frac{\text{recall}^{-1} + \text{precision}^{-1}}{2} \right)^{-1} = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}}, \quad (3.4)$$

де β - приймає значення в діапазоні $0 < \beta < 1$, якщо пріоритет віддається точності, і $\beta > 1$, якщо пріоритет віддається відкликанню.

F-оцінка досягає максимуму при повноті і точності, що дорівнює одиниці, і близька до нуля, якщо один з аргументів близький до нуля.

Площа під кривою ROC (AUC - ROC) - одна з найпопулярніших метрик, що використовуються в галузі. AUC забезпечує сукупний показник продуктивності за всіма можливими пороговими значеннями класифікації. Одним із способів інтерпретації AUC є ймовірність того, що модель оцінює випадковий позитивний приклад вище, ніж випадковий негативний приклад.

Для розрахунку кривої буде використовуватися 2 значення з матриці плутанини:

- чутливість;
- специфічність.

Отже, для кожної чутливості ми отримуємо різну специфічність (рис.3.1)

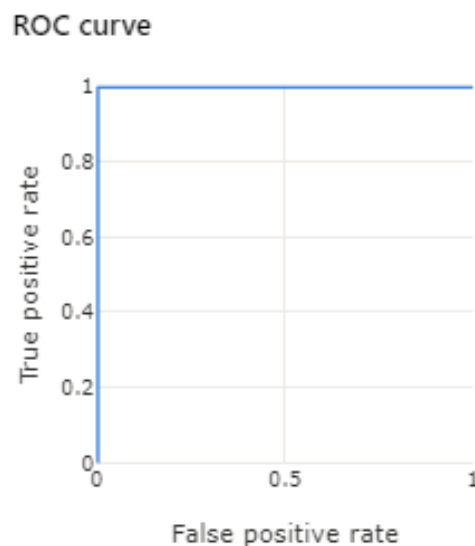


Рисунок 3.1 – Приклад вигляду ROC -кривої

Критерій AUC-ROC стійкий до незбалансованих класів і може бути інтерпретований як ймовірність того, що випадково обраний позитивний об'єкт буде ранжований класифікатором вище (матиме більш високу ймовірність бути позитивним), ніж випадково обраний негативний об'єкт.

Крім ROC-кривої існує PR-крива (від англ. Precision-Recall Curve), що показує відношення точності до повноти.

Крива точність-повнота відображає зв'язок між точністю і повнотою при зміні порога прийняття рішення. Для вирішення деяких задач може знадобитися більш висока повнота, а для деяких - більш висока точність, в залежності від відносної важливості запобігання помилково негативні і хибнопозитивних результатів(рис.3.2).

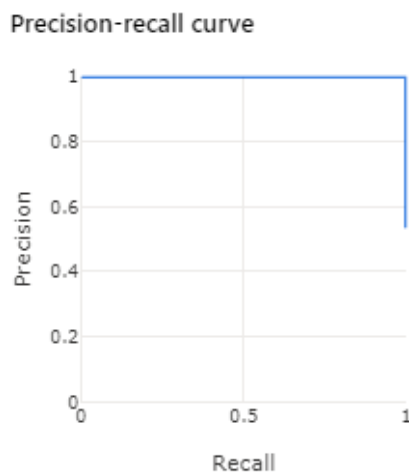


Рисунок 3.2 – Приклад вигляду PR-кривої

Бажано, щоб алгоритм мав як високу точність, так і високу запам'ятовуваність. Однак більшість алгоритмів машинного навчання часто включають компроміс між ними. Хороша крива PR має більшу AUC (площу під кривою). На малюнку вище класифікатор, що відповідає синій лінії, має кращу продуктивність, ніж класифікатор, який відповідає зеленій лінії. Важливо відзначити, що класифікатор, який має вищу AUC на кривій ROC, завжди матиме вищу AUC на кривій PR. із відомими значеннями (основна істина) в ярликах. Необхідно мінімізувати втрати журналу для моделі в цілому.

3.2. Дослідження та порівняння алгоритму нейронних мереж студії машинного навчання Azure

Основна мета цього дослідження полягає в тому, щоб порівняти між собою результати якості алгоритму двокласової класифікації нейронної мережі за допомогою конструктора машинного навчання при вирішенні завдань кібербезпеки[15].

Загальний вигляд конвеєра експерименту в конструкторі машинного навчання Azure показаний в додатку 1.

Щоб оцінити якість створеної моделі потрібно запустити виконання експерименту. Після виконання всіх модулів перевіряємо вихідні дані аналізу модулів.

Це можна зробити двома способами:

1. Зберегти результати в базі даних набору даних для аналізу за допомогою інструментів виходу.
2. Створити виведення даних в конвеєрі машинного навчання Azure.

Щоб перевірити вихідні дані модуля “Аналіз моделей” в конвеєрі, необхідно вказати порт виводу та вибрати елемент “Візуалізація”.

Підключаємо набори даних до обох входів Оцінка моделі, результати будуть містити значення для обох наборів даних або обох моделей.

Після цього виводимо результат оцінки використаного методу на екран конвеєра.

Розглянемо результати отримані для двійкового класифікатора з використанням алгоритму нейронні мережі (Two-Class Neural Network)(рис.3.3 – 3.4).

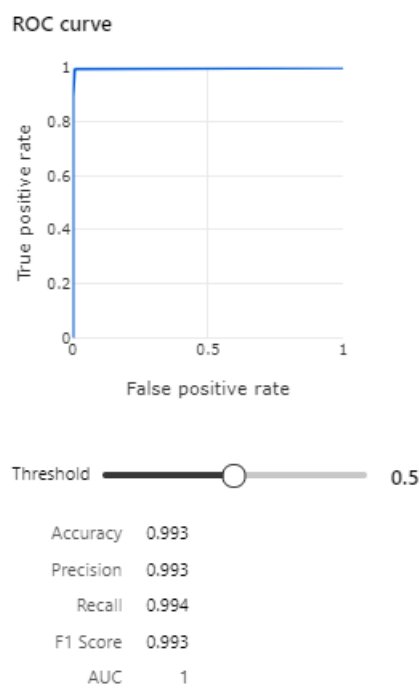


Рисунок 3.3 – Показана крива ROC для моделі двійкового класифікатора з використанням алгоритму нейронні мережі.

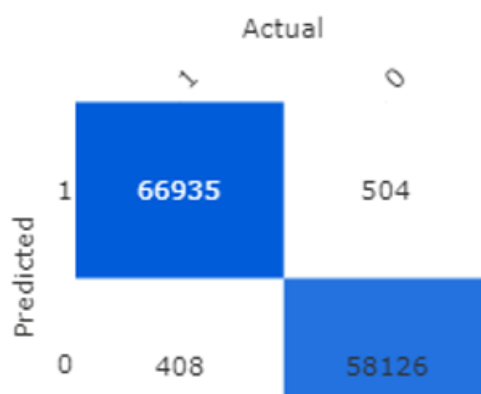


Рисунок 3.4 – Показана матриця плутанини для моделі двійкового класифікатора з використанням алгоритму нейронні мережі.

Точність розглянутої проблеми виявляється 99%. Як видно з двох наведених вище таблиць, позитивне передбачувальне значення є дуже високим, негативне прогнозне значення є досить низьким. Те саме стосується чутливості та специфічності. Це в першу чергу залежить від заданих крайніх значень.

Якщо ми зменшимо наше крайнє значення, дві пари чисел різко зблизяться в значеннях.

Змінюючи крайнє значення можна регулювати і криву ROC. В ідеальній моделі крива йде вгору зліва по верхньому краю, охоплюючи всю область діаграми. Чим більше область під кривою, тим вища продуктивність моделі.

За це відповідає значення AUC, яка приведена разом з іншими метриками.

Якщо значення AUC для моделі має більше значення, ніж значення для моделі бінарної класифікації (0,5), то в цьому випадку модель більш ефективна, ніж вгадування. Чим ближче AUC до 1 тим краще модель розпізнає аномалії(атаки і вторгнення)[16].

Як бачимо за прикладом крайнє значення(Threshold) - 0,5.

Чутливість при цьому крайньому значення становить 99,4%, а специфічність ~ 99,3%. Використовуємо ці значення як координату точки на кривій ROC. Щоб звести цю криву до одного числа, знаходимо площу під цією кривою (AUC – це значення довжини і ширина квадрата).

Якщо AUC це ширина, то площа всього квадрата дорівнює $1 * 1 = 1$. Отже, AUC сам по собі є відношенням під кривою та загальною площею. Для даного випадку ми отримуємо AUC ROC як 100%. Нижче наведено кілька оцінок відповідних значень:

- 90-100 = відмінно (A);
- 80-90 = добре (B);
- 70-80 = справедливо (C);
- 60-70 = бідно (D);
- 50-60 = невдача (F).

Тобто, наш результат поточної моделі підпадає в діапазон відмінно (A). Навчання моделі можна вважати успішним. Навіть дуже.

Також на PR-кривій можна бачити аналогічний результат(рис.3.5)

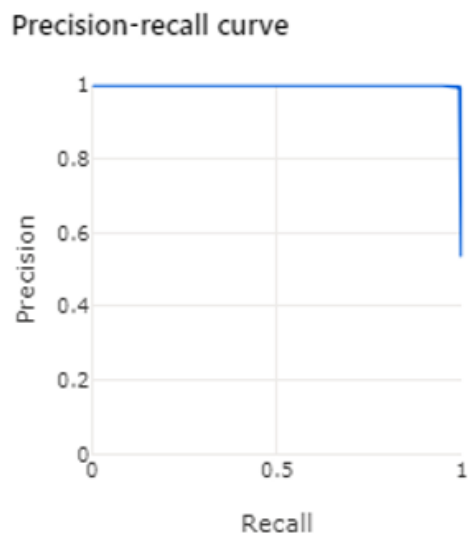


Рисунок 3.5 - Показана крива PR для моделі двійкового класифікатора з використанням алгоритму нейронні мережі.

Для більшої наочності, було взято тестову KDD базу даних новіших років, для перевірки ефективності зібраної моделі машинного навчання(рис.3.6-3.7).

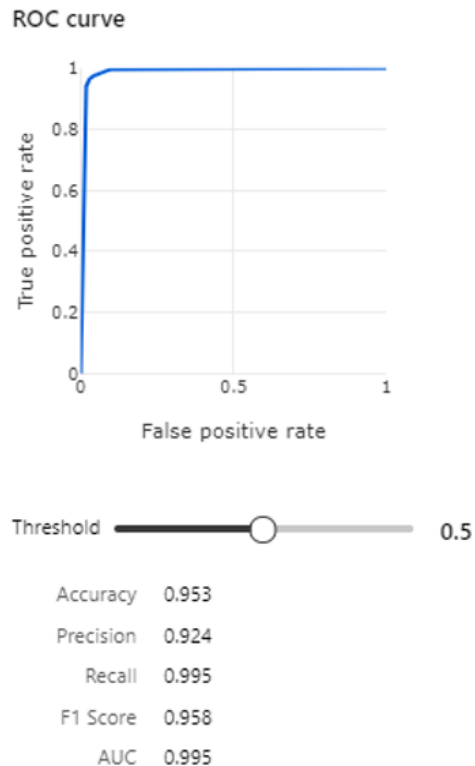


Рисунок 3.6 - Показана крива ROC для моделі двійкового класифікатора з використанням алгоритму нейронні мережі бази даних KDD20

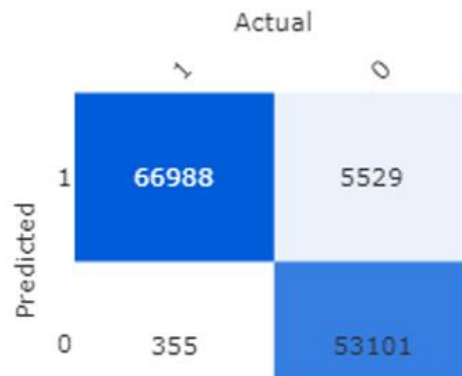


Рисунок 3.7 – Показана матриця плутанини для моделі двійкового класифікатора з використанням алгоритму нейронні мережі бази даних KDD20

Для цього прикладу крайнє значення(Threshold) - 0,5.

Чутливість при цьому крайньому значення становить 99,4%, а специфічність ~ 92,4%. Використовуємо ці значення як координату точки на кривій ROC. Щоб звести цю криву до одного числа, знаходимо площу під цією кривою (AUC – це значення довжини і ширина квадрата)[17].

$$AUC\ ROC = 99,5\%$$

Результат поточної моделі підпадає в діапазон відмінно (А). Навчання моделі можна вважати успішним..

Також на PR-кривій можна бачити аналогічний результат(рис.3.8)

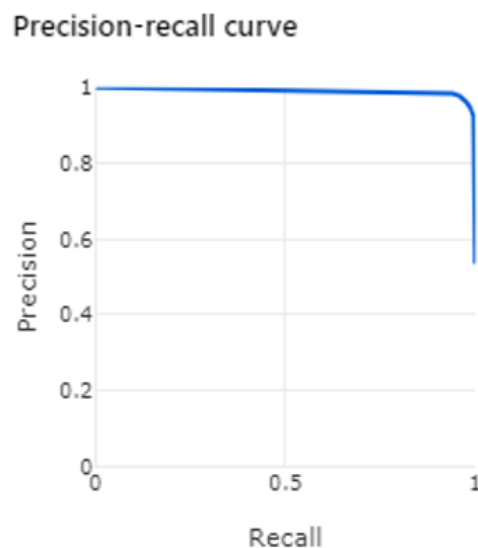


Рисунок 3.8 - Показана крива PR для моделі двійкового класифікатора з використанням алгоритму нейронні мережі бази даних KDD20

Аналіз зведених результатів показує що:

При всіх розглянутих варіантах тестування методу духкласового алгоритму класифікації, а саме нейронної мережі, відсоток успішно знайдених аномалій відповідає позначці відмінно.

Але потрібно зазначити, що при використанні тестових баз даних новішої версії, метрики ефективності зменшувались. З посиланням на це, можна припустити, що в період між виходом версій розроблялись та використовувались нові типи вірусів та інші небезпечні програми.

Хоча натренована система глибинного навчання справилась і з цим.

Потрібно сказати, що метод нейронних мереж показав себе дуже ефективно. Всі метрики показників ефективності були майже ідеальними, особливо показник чутливості.

3.3 Рекомендації по застосуванню результатів дослідження

Вже давно не секрет що машинне навчання, зокрема нейронні мережі масово використовуються в різних сферах діяльності людини, не тільки в кібербезпеці[18].

Машинне навчання і, зокрема, нейронні мережі доцільно використовувати для вирішення бізнес-завдань у випадках, коли:

- накопичено велику кількість різних даних, але програми для їх обробки і систематизації відсутні;
- наявні дані спотворені, не повні або не систематизовані;
- дані настільки різні, що важко виявити зв'язку і закономірності, що існують між ними.

Бізнес-завдання, які можуть вирішуватися засобами машинного навчання і нейронних мереж, але буде важко зробити людині:

- Прогнозування: попиту, обсягу продажів, наповнення складу, завантаження устаткування і інших ресурсів, подальшого розвитку підприємства;

- Виявлення: тенденцій, прихованих взаємозв'язків, аномалій, повторюваних елементів;
- Розпізнавання: фото-, відео-, аудіоконтенту, спроб шахрайства, брехні, внутрішніх загроз, зовнішніх атак на систему безпеки;
- Автоматизація: роботи операторів в онлайн-чатах, телефонних операторів;
- Класифікація: аналіз складу покупців, клієнтів, замовників і сегментація їх за різними параметрами;
- Кластеризація: класифікація за параметрами, які з самого початку не були відомі;
- Розробка: чат-ботів.

Нейромережі вже використовують безліч людей, як для щоденних дрібничок, так і для серйозних рішень. Найбільш помітні.

Найбільша в світі торгова площа Alibaba широко використовує машинне навчання та інші засоби ІІ. В результаті її віртуальні вітрини підлаштовуються під кожного покупця, а система пошуку видає для нього найкращі варіанти. Чат-бот Ali Xiaomi може самостійно впоратися з більшістю звернень клієнтів в техпідтримку. Більш того, розроблена компанією Alibaba нейронна мережа вперше перевершила результати людини при проходженні тестів від Стенфордського університету. У ці тести входять вправи на читання або прослуховування певної інформації, а потім відповіді на перевіірочні питання.

Американська торговельна мережа Target виявила, що за допомогою машинного навчання можна передбачити не лише поведінку покупців, але і зміни в їхньому житті, наприклад, вагітність. Алгоритми Target працюють настільки точно, що за допомогою даних про покупки можуть визначити триместр вагітності жінки, яка їх здійснювала.

Популярний фотохостинг Pinterest використовує машинне навчання, щоб показувати своїм користувачам найцікавіші для них фотографії.

Лукас Бівальд (Lukas Biewald), генеральний директор компанії Figure Eight (колиш. CrowdFlower), яка пропонує безліч проектів машинного навчання, вважає, що воно вже помітно змінює роботу багатьох фірм. І над цим працюють не тільки компанії, які можуть витратити величезні кошти на дослідження і розробки, такі як Google або Microsoft. Насправді, впевнений Лукас Бівальд, кожна компанія зі списку Fortune 500 вже працює набагато ефективніше і заробляє більше грошей завдяки саме машинному навчанню.

Серед компаній з українським корінням слід зазначити стартап Neuromation, який в лютому 2017 року під час ICO залучив \$71,6 млн. інвестицій.

Платформа Neuromation дозволяє створювати штучне навчальне середовище для глибокого навчання нейронних мереж на великій кількості прикладів. Дані для навчання ANN генеруються з використанням обчислювальних потужностей блокчейн-спільноти. Настільки оригінальне рішення компанія прийняла тому, що раніше, в процесі роботи над системами з використанням комп'ютерного зору, зіткнулася з проблемою браку обчислювальних ресурсів. Оренда ресурсів у хмарних сервісів Amazon або Google для стартапу виявилася непідйомною. А через бум майнінгу було практично неможливо купити відеокарти. Так з'явилася ідея брати обчислювальні потужності в оренду у майнерів, яка в підсумку перетворилася на створення нейроплатформи.

Таким чином, перспективним в подальших дослідженнях та розробках детектора аномалій або інших продуктів є використання нейронних мереж та інших методів машинного навчання.

ВИСНОВКИ

В даній кваліфікаційній роботі проведено аналітичну роботу для доведенням актуальності теми роботи, а саме дослідження методу класифікації машинного навчання нейронні мережі при використанні їх в детекторі аномалій для рішення задач кібербезпеки.

Також було проведено розробку детектора аномалій на основі методу нейронних мереж Azure для рішення завдань кібербезпеки.

В результаті дослідження було виявлено, що данна тема має велику актуальність на сьогоднішній день. Дослідження теми нейронних мереж показало що нові підходи та методи не перестають створюватися і не тільки в засобах кібербезпеки, а і в шкідливому програмному забезпеченні.

В роботі було розглянуто нейронні мережі як метод навчання машинного навчання, які використовуються в створенні нових більш ефективніших та зручніших методів.

Також було досліджено типи нейронних мереж, їх різниця між собою, переваги та недоліки.

Було досліджено існуючі методики виявлення аномалій та вторгнень з використанням машинного навчання. Була розглянута система виявлення вторгнень та досліджено природу аномалій. Також було досліджено процес виявлення аномалій та методики виявлення відхилень в досліджуваних даних. При використанні класифікації, кластеризації та методів на основі машинного навчання були виявленні основні етапи процесу виявлення аномалій.

Розроблена технологія виявлення аномалій і вторгнень, яка використовує метод нейронних мереж конструктора машинного навчання Azure.

Оцінка якості навчання і дослідження ефективності класифікації з різними тестовими базами даних проведена за допомогою показників: акуратність, точність (precision), повнота(чутливість), чутливість, специфічність.

Також Була побудована ROC-крива, яка значно упростила процес аналізу продуктивності алгоритмів навчання та методів класифікації машинного навчання

У роботі використані методи візуального математичного моделювання хмарної середовища Microsoft Azure, метод класифікації машинного навчання нейронні мережі, методи програмування.

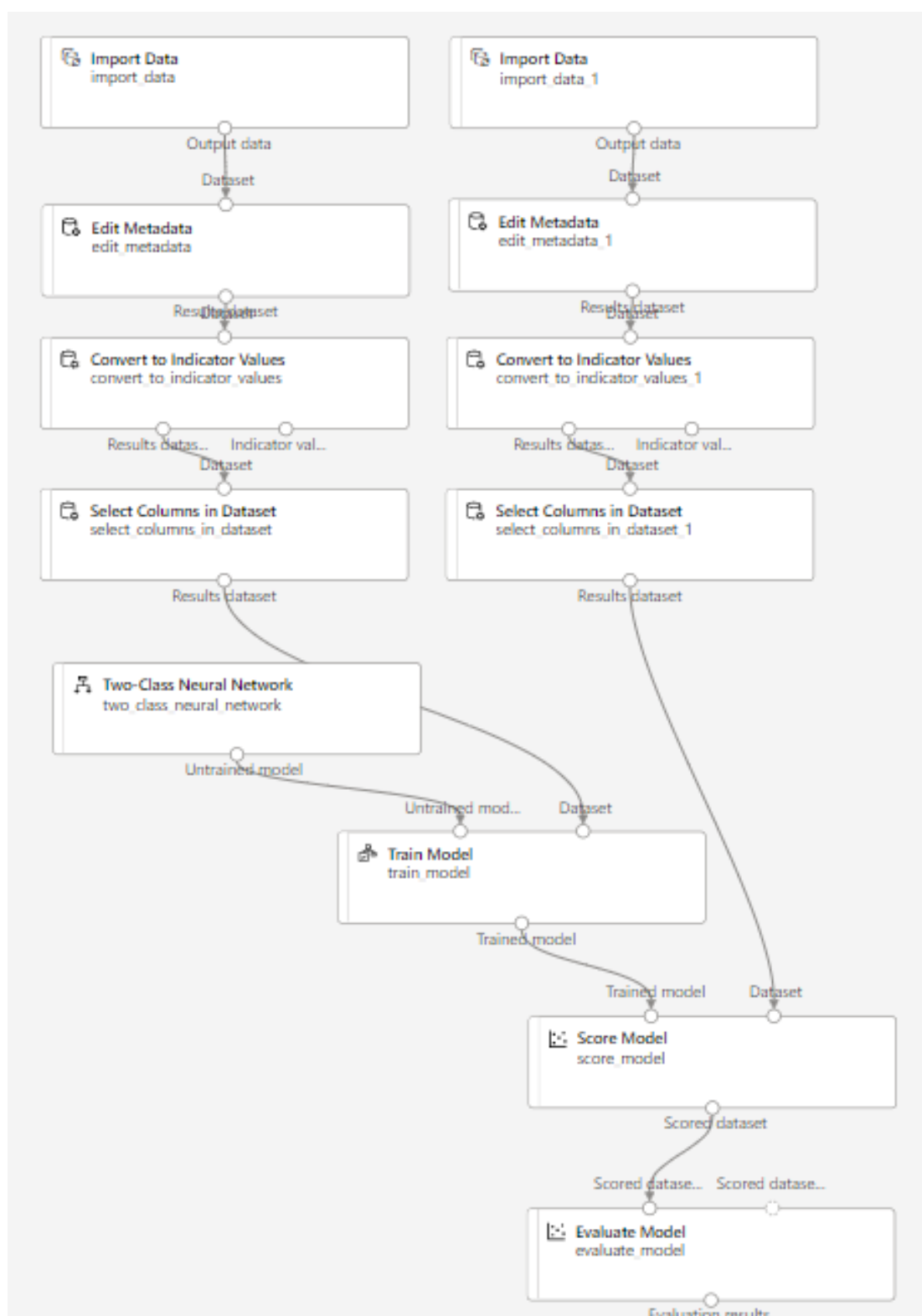
Таким чином, завдання на кваліфікаційну бакалаврську роботу виконано в повному обсязі.

					КБМ.КР.М – 125 – 22 – ПЗ	
						60

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. A.M. Mahfouz, D. Venugopal, S.G. Shiva, Comparative analysis of ml classifiers for network intrusion detection, in: Fourth international congress on information and communication technology, Springer, 2020, pp. 193–207.
2. I.S. Thaseen, B. Poorva, P.S. Ushasree Network intrusion detection using machine learning techniques 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), IEEE (2020), pp. 1-7.
3. Amor N. B., Benferhat S., Elouedi Z., Naive Bayes vs decision trees in intrusion detection systems, In Proceedings of the ACM symposium on Applied computing; 2014; p. 420-424
4. Chitrakar R., . Chuanhe,H., Anomaly detection using Support Vector Machine classification with k-Medoids clustering; In Proceedings of IEEE Third Asian Himalayas International Conference on Internet (AH-ICI); 2012; p. 1-5.
5. A. Drewek-Ossowicka, M. Pietrolaj, J. Rumiński A survey of neural networks usage for intrusion detection systems J. Ambient Intell. Humaniz. Comput., 12 (1) (2021), pp. 497-514
6. Aburomman AA, Ibne Reaz MB (2016) A novel SVM-kNN-PSO ensemble method for intrusion detection system. Appl Soft Comput 38:360–372.
7. Ambusaidi, M., He, X., Tan, Z., Nanda, P., Lu, L. F., and Nagar, U. T., “A novel feature selection approach for intrusion detection data classification”,2014 IEEE 13th International Conference on Trust,Security and Privacy in Computing and Communications, 2014, pp.82-89.
8. Sharma, S. and Gupta, R. K.,“Intrusion Detection System: A Review”,International Journal of Security and Its Applications, vol.9, no.5,2015, pp.69-76.

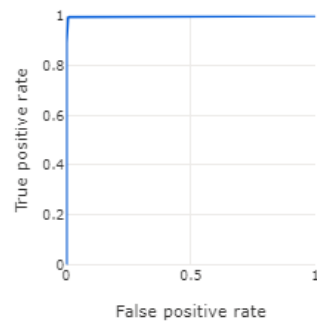
9. Aljawarneh S, Aldwairi M, Yassein MB (2018) Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model. *J Comput Sci* 25:152–160.
10. Top 10 BEST Intrusion Detection Systems (IDS) [2020 Rankings]. Available online: <https://www.softwaretestinghelp.com/intrusion-detection-systems/> (accessed on june 24, 2021)
11. S. Agrawal and J. Agrawal, "Survey on anomaly detection using data mining techniques", *Computer Science*, vol. 60, pp. 708-713, 2015.
12. Du M, Li F, Zheng G, Srikumar V (2017) DeepLog: anomaly detection and diagnosis from system logs through deep learning. In: *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. ACM, pp 1285–1298.
13. Meena, G., Choudhary, R.R., 2017. A review paper on ids classification using kdd 99 and nsl kdd dataset in weka, in: *2017 International Conference on Computer, Communications and Electronics (Comptelix)*, IEEE. pp. 553–558.
14. M.A. Ferrag, L. Maglaras, S. Moschoyiannis, H. Janicke Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study *J. Inf. Secur. Appl.*, 50 (2020), Article 102419
15. Choudhary S., Kesswani N. A survey: Intrusion detection techniques for internet of things *International Journal of Information Security and Privacy(IJISP)*, 13 (2019), pp. 86-105
16. R. Lohiya, A. Thakkar Intrusion detection using deep neural network with antirectifier layer *Appl. Soft Comput. Commun. Netw.*, Springer (2021), pp. 89-105.
17. L. Shang, D.-S. Huang, J.-X. Du, C.-H. Zheng Palmprint recognition using fastica algorithm and radial basis probabilistic neural network *Neurocomputing*, 69 (13–15) (2006), pp. 1782-1786.



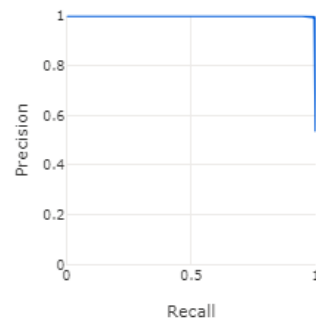
KDD99

● Scored dataset (left port)

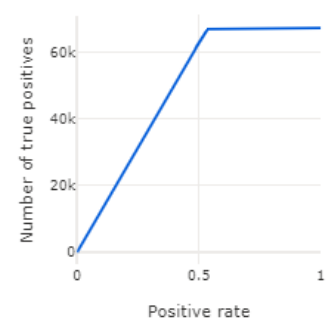
ROC curve



Precision-recall curve

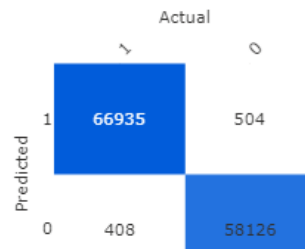


Lift curve



Threshold 0.5

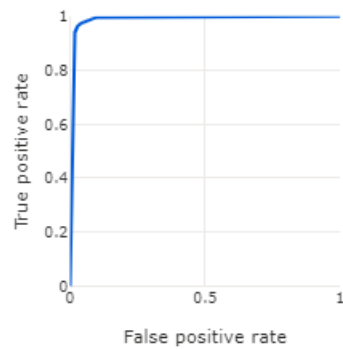
Accuracy 0.993
Precision 0.993
Recall 0.994
F1 Score 0.993
AUC 1



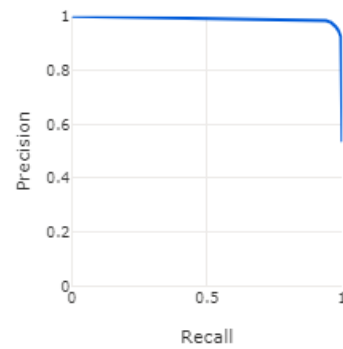
KDD20

● Scored dataset (left port)

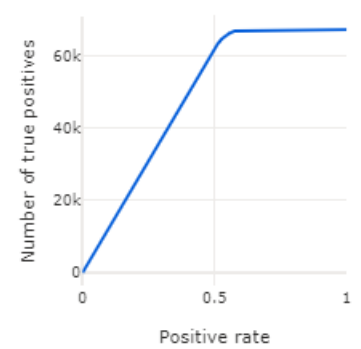
ROC curve



Precision-recall curve



Lift curve



Threshold 0.5

Accuracy 0.953
Precision 0.924
Recall 0.995
F1 Score 0.958
AUC 0.995

