ALGERIAN DEMOCRATIC AND POPULAR

REPUBLIC MINISTRY OF HIGHE REDUCATION AND SCIENTIFIC RESEARCH

UNIVERSITY KASDI MERBAH OUARGLA

FACULTY OF NEW INFORMATION AND COMMUNICATION TECHNOLOGIES

DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

PROFESSIONAL MASTER THESIS

Domain: Mathematics and Computer Science

Field: Computer science

Speciality: Network Administration and Security

Presented by:

Talbi Maria Hibat Errahmane

Hadji Soumia

THEME

Semi-supervised learning-based IDS using active learning

Publicly discussed on: 24/06/2024

Before the Jury:

Mr.Benbezziane	**President**	**UKM Ouargla**
Mr. Khaldi Yacine	**Supervisor**	**UKM Ouargla**
Mrs. Hanane Azzaoui	**Examiner**	**UKM Ouargla**

Academic year:2023/2024

Dedication

This work is dedicated to:

To my parents, who have supported me through the whole study process.

To my sister and my brothers who stood by me until the last minute.

Also, I would like to dedicate this thesis to my friends, for their support and encouragement during my work.

Soumia

Dedication

To those who taught me the meaning of life and illuminated my path to success,

To my beloved mother and father, this achievement is a testament to your unwavering support.

To my beloved mother, the source of my inspiration and strength, who has spared no effort in ensuring my comfort and education.

To my dear father, my primary pillar of support, who has generously offered his valuable guidance and advice.

To all my family members and friends who stood by me at every step of this journey.

To everyone who contributed to achieving this accomplishment, whether with a kind word or a sincere prayer.

I dedicate this modest work to you all as a token of my deep gratitude and love

Maria Hibat Errahmane

Acknowledgments

Praise be to Allah SWT.

We express our infinite gratitude for the patience and perseverance He has granted us to successfully complete this scientific journey and conclude this modest work.

We wish to express our heartfelt thanks to our parents for their unwavering support and countless sacrifices, without which we would not be here today.

We are deeply grateful to our supervisor **Dr.Khaldi Yacine**,for agreeing to supervise us and for providing invaluable advice and expertise.

Additionally, we extend our gratitude to everyone who has contributed to our educational journey in various ways.

To those who encouraged, supported, and inspired us on our path to success, your contributions are deeply appreciated and will always be remembered.

Soumia, Maria Hibat Errahmane

Abstract

Due to the increasing number of attacks, IDS systems are becoming more essential to protect networks and devices from malicious attacks.

The current study aims to enhance Intrusion Detection System performance by utilizing the least amount of labeled instances.To achieve the objective of this study, a semi-supervised learning approach was used that combines machine learning and active learning method, using the CICIDS 2017 dataset. Our experimental evaluation shows the effectiveness of incorporating active learning into semi-supervised learning for IDS, as the proposed approach significantly outperforms traditional machine learning methods in terms of accuracy, precision, detection rate, and false positive rate.

Keywords: IDS, active learning, cybersecurity, CICIDS2017, Machine learning

Résumé

Avec l'augmentation des attaques, les systèmes IDS deviennent de plus en plus importants pour protéger les réseaux et les appareils contre les attaques malveillantes.

L'étude actuelle vise à améliorer les performances du système de détection d'intrusion en utilisant le moins d'instances étiquetées. Pour atteindre l'objectif de cette étude, une approche d'apprentissage semi-supervisé a été utilisée qui combine l'apprentissage automatique et la méthode d'apprentissage actif, en utilisant l'ensemble de données CICIDS 2017. Notre évaluation expérimentale montre l'efficacité de l'intégration de l'apprentissage actif dans l'apprentissage semi-supervisé pour l'IDS, car l'approche proposée surpasse considérablement les méthodes traditionnelles d'apprentissage automatique en termes d'exactitude, de précision, de taux de détection et de taux de faux positifs.

Mots-clés: IDS, apprentissage actif, cybersécurité, CICIDS2017, Apprentissage automatique

الملخص

بسبب الزيادة المستمرة في الهجمات، أصبحت أنظمة الكشف عن التسلل (IDS) ضرورية لحماية الشبكات والأجهزة من التهديدات الخبيثة. تهدف هذه الدراسة إلى تحسين أداء نظام الكشف عن الاختراق باستخدام أقل عدد ممكن من العينات المصنفة.

لتحقيق هذا الهدف، تم اعتماد نهج تعلم شبه خاضع للإشراف يجمع بين التعلم الآلي والتعلم النشط، وذلك باستخدام مجموعة بيانات CICIDS 2017. أظهرت نتائج التقييم التجريبي أن دمج التعلم النشط في التعلم شبه الخاضع للإشراف يعزز بشكل كبير فعالية نظام الكشف عن الاختراق، متفوقًا على الأساليب التقليدية للتعلم الآلي من حيث الدقة، الضبط، معدل الإكتشاف ومعدل الانذارات الايجابية الكاذبة.

الكلمات المفتاحية : أنظمة الكشف عن اتسلل ، التعلم النشط، الأمن السيبراني، CICIDS2017، التعلم الآلي

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

List of acronyms

AI	Artificial Intelligence
ANN	Artificial Neural Network
CEAL	Cost-Effective Active Learning mode
DoS	Denial-of-Service attack
DDoS	Distributed Denial-of-Service attack
DR	Detection Rate
FAR	False Alarm Rate
FN	False Negative
FP	False Positive
FPR	False Positive Rate
GUI	Graphical User Interface
HTML	HyperText Markup Language
HIDS	Host-Based Intrusion Detection System
IDS	Intrusion Detection System
IoT	Internet of Things
kNN	k-Nearest Neighbors
ML	Machine Learning
NIDS	Network Intrusion Detection System
NLP	Natural Language Processing
NMT	Neural Machine Translation
RAM	Random Access Memory
ReLU	Rectified Linear Unit
SQLi	Structured Query Language Injection
SVM	SVM - Support Vector Machine
TN	True Negative
TP	True Positive
TPR	True Positive Rate
XSS	Cross-Site Scripting

CHAPTER

1

GENERAL INTRODUCTION

1.1 Background and Motivation

Nowadays, the Internet is widely used, as the number of people using the Internet in 2024 reached about 5.35 billion[1], so it indicates that there are many computers, phones, systems and networks connected to the outside world, which means securing devices and networks and their confidentiality from intrusions has become a necessity [2].

For this reason security techniques such as Intrusion Detection Systems (IDS) have to be introduced.

IDS is a type of detection system that plays an important role in protection by monitoring software and hardware traffic in the network and detecting malicious activities that try to modify, delete or steal data [3].

The performance of the actual IDS relies on the preset rules defined by experts and usually machine learning methods (generally, supervised learning) are applied to detect attacks [4,5]. To obtain high accuracy, supervised learning methods need the acquisition of a large amount of labeled data, which is very expensive and difficult, as opposed to semi-supervised learning, which only requires a small amount of labeled data to obtain a good result or better than fully labeled training data [6].

For that , an active learning algorithm (a type of semi-supervised learning closer to traditional supervised learning) is used to define if the activities are malicious or not, and this is done by labeling data incrementally during the training phase [7].

1.2 Problem Statement

Intrusion detection systems (IDSs) play an important role in cyber security by alerting administrators to suspicious activity across networks and devices [3]. To realise the full power of this technology however, administrators must first overcome a variety of challenges and one of the biggest challenges is the High False Alarm Rates which means that an IDS can label a normal activity as suspicious and vice versa [8]. Also, traditional IDS approaches rely on a large amount of labeled datasets to train the machine learning model, which can be difficult as real-world attack data are uncommon and scarce [7].

As a result of these challenges, a new semi-supervised learning approach is used in IDS which is active learning, which only requires a small amount of labeled data to obtain a good result [7].

1.3 Objectives of the Study

This study seeks to make significant enhancements in model performance by utilizing both semi-supervised and active learning in combination. One objective of the research is to utilize both labeled and unlabeled data together in training the model, strengthening the effectiveness of semi-supervised learning and enhancing model accuracy.

The second objective involves incorporating an active learning approach into the context of semi-supervised oversight. The main objective is to significantly enhance model performance metrics like precision and recall, in comparison to utilizing solely labeled data or all unlabeled data while avoiding active learning involvement. The study's goal is to save time and money by efficiently utilizing unlabeled data, leading to increased efficiency and cost reduction in data annotation.

1.4 Scope and Limitations

We plan to develop a program utilizing traditional artificial neural networks (ANNs) for anomaly detection, followed by a comparison with ANNs enhanced with active learning techniques.

Our focus will be on analyzing the prevalent types of intrusion detection systems, including Denial of Service (DoS) attacks, Distributed Denial of Service (DDoS) attacks, Cross-site Scripting (XSS) attacks, SQL injection (SQLi) attacks, Brute force attacks, leaks, port scans, and Robot attacks.

This choice is motivated by the challenges associated with obtaining a substantial amount of data for less common attack types, coupled with the rarity of these attacks.

1.5 Structure of the Thesis

The remainder of this thesis is structured as follows:

- Chapter 2: It provides a comprehensive overview of intrusion detection systems (IDS), including its types, methodologies, and common challenges associated with this field and also talks about active learning technique, how it works and examples of its applications.

Also this chapter identifies the current research gaps that it aims to address, namely the integration of semi-supervised learning with active learning in IDS.

• Chapter 3: It outlines the research design, data collection, and preparation process. It explains the semi-supervised learning approach, including algorithm selection and configuration, and describes the integration of active learning with semi-supervised learning.

It also discusses evaluation metrics for evaluating IDS performance.

• Chapter 4: This chapter presents experimental setup details and performance evaluation results, including comparisons with baseline models.

This chapter then analyzes how semi-supervised learning and active learning can be combined in IDS, discusses their implications, acknowledges their limitations, and suggests future research directions.

CHAPTER

2

LITERATURE REVIEW

2.1 Introduction

For the purpose of getting more details about both IDS and active learning, in this chapter we are going to discuss the IDS, its types and methodologies, and the most common challenges. We also talk about active learning techniques its applications and phases.

And since we will use a semi-supervised technique in our work, we will present some works implemented using this approach for detecting malicious activities.

2.2 Intrusion Detection Systems

2.2.1 Definition of IDS

Detection of intrusions (IDS) involves monitoring the activities happening on a device or network and analyzing them in order to identify suspicious behavior and policy violations, which include attempts to compromise the confidentiality, integrity, availability, or security of a computer or network [9].

As its name implies, an intrusion detection system detects and alerts network administrators to take corrective action when an intrusion occurs (high detection rate and a low false alarm rate)within your IT infrastructure. It is possible to implement this technology in either a hardware or software manner [10].

2.2.2 Benefits Of IDS

The IDS has several advantages, including:

- With the IDS, administrators can keep an eye on routers, firewalls, and servers, to gain a holistic view of a network's security [11].
- Companies benefit from it as an additional layer of protection [11].
- By analyzing attacks, identifying their patterns, and implementing effective controls, it helps administrators manage and secure their networks [12].

2.2.3 The types of IDS

As shown in the figure below, there are three types of IDS:

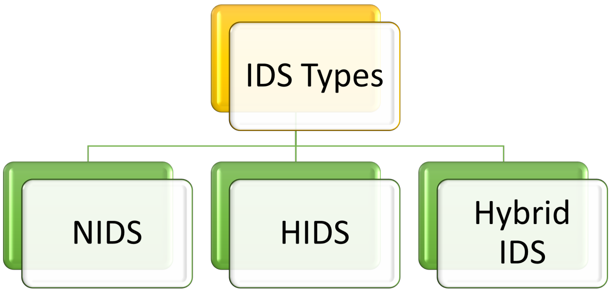


Figure 2.1: The types of IDS

a) NIDS

Network Intrusion Detection Systems (NIDS) are solutions that screen a network at several strategic points [12].In order to use a NIDS, you must install it in your network infrastructure [12]. When the NIDS is installed, it will analyse all packets that arrive and detect any suspicious activity that may pose a serious threat, such as denial-of-service attacks and botnet attacks [12].

The NIDS will notify directors as soon as an attack or abnormal behavior is detected [12].

a.1) Advantages of NIDS

- Networks can be monitored effectively with a few properly placed NDISs [13].
- NIDS are harder to detect by intruders and attackers because they are invisible [13].
- NIDS detects events in real-time which enables fast response [14].

a.2) Disadvantages of NIDS

- It may have difficulty identifying signs of intrusion the more traffic it analyzes [14].
- It is difficult for NIDS to determine if an attack has been successful[13,15]. As a result administrators must manually investigate each attacked host after NIDS detects an attack to determine whether it has indeed been accessed [16].

b) HIDS

This type is placed on independent devices such as servers or workstations [10]. HIDS monitors packets coming from and leaving the device by using log files and comparing them to previously collected information, including the owner and the size of a file, the last modification date etc., then alerts the administrator if suspicious or malicious activity occurs [12,16].

b.1) Advantages of HIDS

- Local user activities can be accurately monitored using HIDS, because it can be installed on computers and servers [13].
- The device affected can be easily pinpointed using HIDS [13].

b.2) Disadvantages of HIDS

- HIDS utilizes a lot of CPU resources [15].
- It is difficult to deploy and manage, especially when there are a lot of hosts to protect[9].

c) The Hybrid IDS

It combines NIDS and HIDS, meaning both devices and networks are monitored [13].

This solution combines the strengths of both previous types, which elevates the competence of intrusion detection [13].

2.2.4 The methodologies of IDS

As shown in the figure below, there are two key methodologies for IDS:

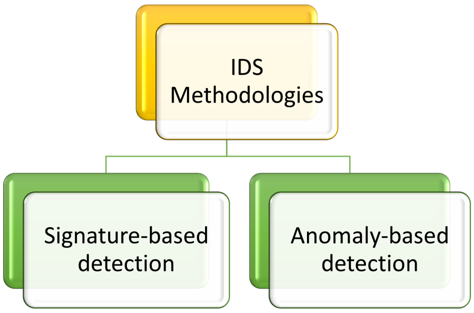


Figure 2.2: The methodologies of IDS

a) Signature-based detection

In this technique, known as misuse detection [13], patterns of known malicious activity are stored in a data set, and new attackers are detected by comparing them with the stored patterns [16].

Although detecting known attacks with this technology is effective, it does not work against novel or unknown attacks where signatures do not yet exist such as "zero-day" attacks [13,15].

Signature-based detection also faces challenges in building flawless signatures to cover all known attacks [15,17]. This method will be less effective if there are any mistakes in defining these signatures (increase false alarm rate).

b) Anomaly-based detection

There are numerous researchers who use an anomaly-based intrusion detection system because it is capable of detecting unknown and zero-day attacks even though machine learning systems don't know about these attacks during the training phase, as well as the ability to customize the normal network behavior profile, such as protocols and packet sizes etc [17].

Due to this customization, an attacker cannot clearly discern what activities (behaviors) he can carry out without being detected, and that is why it is also called a behavior-based intrusion detection system [15,17]. The main disadvantage of this method is the higher probability of false positives [17].

2.2.5 The challenges of IDS

As the figure below shows, the major challenges faced by the intrusion detection system are the following:

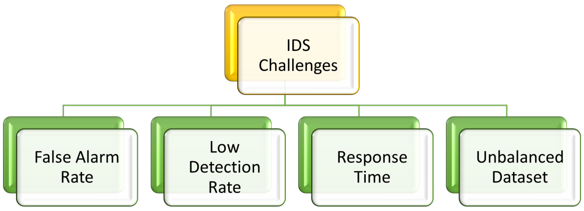


Figure 2.3: The key challenges of IDS

a) False Alarm Rate

When the IDS detects suspicious activity, an alert is raised, but the activity may actually be normal (False Positive). A high false alarm rate will waste both time and resources on non-existent attacks [8].

b) Low Detection Rate

It is the opposite of false positive rate, where the IDS fails to detect malicious attacks (False Negatives), leaving the network vulnerable to attack [8].

c) Response Time

The IDS should be able to detect threats and raise alerts quickly. However, IDS sometimes take a long time to respond to threats, giving attackers enough time to access a network or specific devices and cause damage [8].

d) Unbalanced Dataset

The IDS learns to detect threats by analyzing data that includes both normal and abnormal activity. If this dataset is unbalanced (usually the number of examples of normal traffic is greater than malicious traffic), the IDS may be effective at detecting normal traffic, but less effective at detecting malicious threats [8].

2.3 Semi-supervised Learning in IDS

Semi-supervised learning has emerged as a focal point in the realm of intrusion detection systems (IDS) primarily because traditional supervised learning methods require a vast number of labeled examples, which poses significant challenges[6]. One pivotal advantage of this approach lies in its adept utilization of unlabeled data, presenting an extra informational reservoir without the imperative of complete labeling. This not only streamlines the data preprocessing phase but also enriches the learning process. Moreover, the integration of semi-supervised learning enhances the performance of detection systems by discerning suspicious patterns and behaviors with heightened accuracy and efficacy, ultimately contributing to more robust and reliable security measures[6]. Numerous researchers have proposed a variety of semi-supervised learning approaches that incorporate unlabeled data into the learning process.

[18] demonstrated in their study that semi-supervised classification methods exhibit superior performance compared to supervised classifiers. Furthermore, they demonstrated that semi-supervised clustering methods can significantly enhance the performance of purely unsupervised clustering methods. The accuracy of the semi-supervised method reached 53.44 %, in comparison to 26.31% for supervised learning.

[19] proposed a semi-supervised technique using a modified Mahanalobis distance based on PCA (M-PCA) for anomaly detection in network traffic. They extended the K-means clustering algorithm to group similar data points and build a normal traffic profile, aiming to reduce noise in anomalies and enhance the quality of the training dataset. The results of this method were 91% accuracy with 10.6% false alarms on a pre-partitioned NSL-KDD dataset.

[20] developed an ensemble learning approach to anomaly detection. This intrusion detection system preprocesses network data using Weka, then builds an ensemble model combining Naive Bayes, Neural Networks, and J48 decision trees to identify normal traffic and specific attack types. This system has used various individual classification methods and its ensemble model on the KDD99 and NSL-KDD data sets to verify the performance of the model. The results of the proposed model demonstrate high accuracy with a very low number of false alarms. Notwithstanding these favorable outcomes, the group learning approach is more intricate and time-consuming.

[21] developed an intrusion detection system that integrates unified learning with semi-supervised learning in order to enhance privacy preservation, improve training and inference efficiency, and achieve greater accuracy. The system uses the unified learning methodology to detect anomalies on Internet of Things (IoT) devices. The methodology comprises an iterative training process for an unsupervised model, utilising local data on the devices. This is followed by aggregation and fine-tuning with a limited quantity of labelled data on a central server. They use the UNSW-NB15 dataset to train their model. This methodology circumvents the sharing of sensitive data while facilitating cooperative learning. This model demonstrated remarkable performance when compared to traditional machine learning models trained on labeled data.

Integrating semi-supervised learning into intrusion detection systems (IDS) has proven to be a valuable strategy that provides advantages over traditional supervised methods through the efficient use of unlabeled data. This approach not only simplifies data preprocessing, but also improves learning outcomes and allows for more accurate and efficient detection of suspicious patterns and behaviors.

2.4 Active Learning

2.4.1 Definition of active learning

It is a semi-supervised approach [7]. Unlike supervised learning, active learning only provides an initial subset of labeled data from a larger unlabeled dataset and selects the data to learn from to achieve higher accuracy [7]. This technology allows programs to actively query authoritative sources (programmers or labeled datasets) to learn the correct prediction for a given problem [7].

The goal of this technique is to speed up the learning process, especially if we don't have a large labeled dataset to practice traditional supervised learning methods [22].

2.4.2 The principles of active learning

As shown in Figure 2.4, active learning has four key steps:

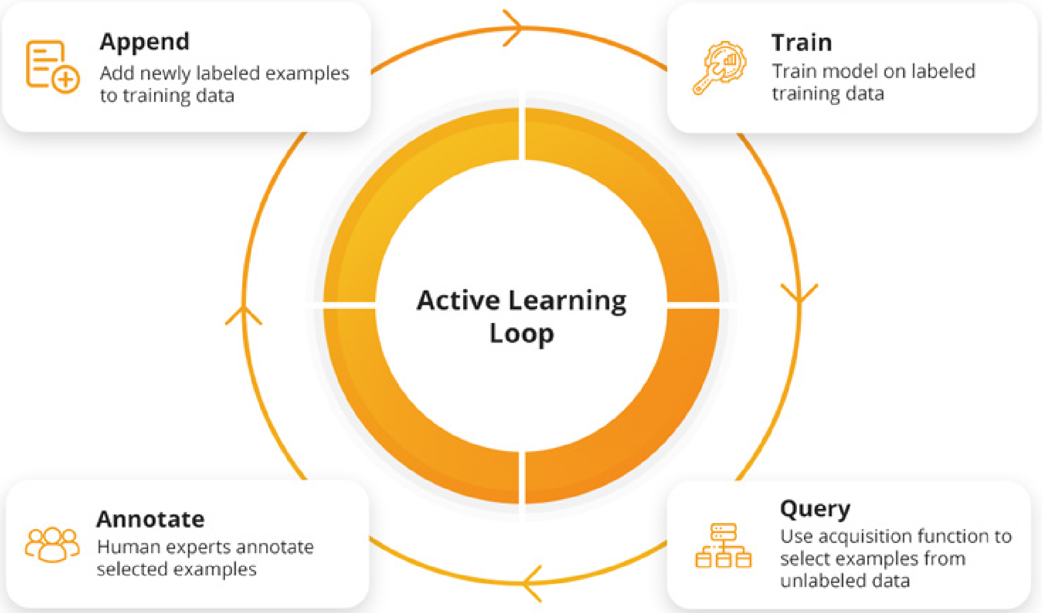

Append
Add newly labeled examples to training data
Train
Train model on labeled training data
Active Learning Loop
Annotate
Human experts annotate selected examples
Query
Use acquisition function to select examples from unlabeled data

Figure 2.4: The steps of the active learning process [23]

a) Query

The model uses a learning (acquisition) function to select a subset or a sample of unlabeled data that it believes will be the most beneficial for further labeling [22,23].Typically,the model focuses on samples that he is unsure of [22,23].

b) Annotate

In this phase, a human expert assigns labels to the selected samples [22,23].

c) Append

Once the human expert has finished labeling the samples, the newly labeled samples are added to the training data set so that the model can be trained on it [22,23].

d) Train

After that, the model is re-trained on the new training set [22,23].This procedure is repeated iteratively and every time the model is more effective, as it is exposed to more labeled data [22,23].

2.4.3 The applications of Active learning

Recently, active learning has been employed in several domains. It could be useful in many modern machine learning problems that require a high degree of accuracy ,using as few labeled instances as possible such as:

a) Natural Language Processing(NLP)

One popular area of active learning is natural language processing (NLP). This is because many applications of NLP require large amounts of labeled data, and the cost of labeling this data is high [24]. Using active learning, the amount of labeled data needed can be reduced significantly, and the number of experts needed to do so can be reduced as well.

For example, [25] employed active learning to address the NMT(the Neural Machine Translation) problem, which is based on deep learning models to translate text from one language to another.

b) Image Classification

It is the process of determining to which category or categories an image belongs[24].

For example, the CEAL (Cost-Effective Active Learning model) employs Active Learning in the image classification problem; this model is different from typical approaches that only consider the most informative and significant samples. It instead proposes to automatically select and annotate unlabeled samples [24].

c) Object Detection

It is a Computer Vision technique where the object is to detect objects in an image. This differs from image classification because in image classification, the entire image is categorized into a single category [24].

On the other hand, object detection can classify objects in a single image into different categories [24].

2.5 Gap Analysis

Intrusion detection systems (IDS) are essential components in the field of cybersecurity, continuously safeguarding networks from unauthorized access and malicious activities. However, the biggest challenge they face is the occurrence of false positives. False positives are alerts that are triggered without an actual attack, or situations where harmless events are misidentified as malicious, resulting in the inefficient use of resources and energy. Conversely, false negatives occur when a system is unable to detect an attack and continues to operate normally, allowing the attack to proceed undetected. This could result in the theft of sensitive data or disruption of system functionality. The utilization of outdated datasets may result in the generation of inaccurate or irrelevant results, which could potentially lead to an increase in the number of false alarms.

In this context, our thesis focuses on filling this gap by proposing a novel framework that combines semi-supervised learning with active learning in IDS. This integration aims to improve the accuracy, scalability, and adaptability of anomaly detection systems while minimizing the need for extensive labeled data. By harnessing the power of both semi-supervised and active learning paradigms, our approach seeks to enhance threat detection capabilities, reduce false positives, and optimize resource utilization in cybersecurity operations.

2.6 Conclusion

As we see in this chapter, active learning is a vast field, especially in the cybersecurity domain such as specifically IDS. However, it is still an open problem. We also talked about the IDS technique, and took a look at some existing works that use semi-supervised learning to know the power of this approach.

CHAPTER

3

ACTIVE-LEARNING BASED INTRUSION DETECTION SYSTEM

3.1 Introduction

Timely detection of attacks is an important thing. For that, in recent years many techniques such as machine learning have been proposed for the detection of malicious traffic.

In this chapter, we will discuss the used techniques in our research that aim at effective classification of traffic. We will also explain in detail our proposed model using active learning. Finally, we will introduce the machine learning approach and explain its types.

3.2 Machine Learning and its Approaches

Machine learning (ML) is a subfield of artificial intelligence (AI) that provides computers with the ability to learn from data. It contains many learning algorithms, which can be classified into different types as shown in Figure 3.1:

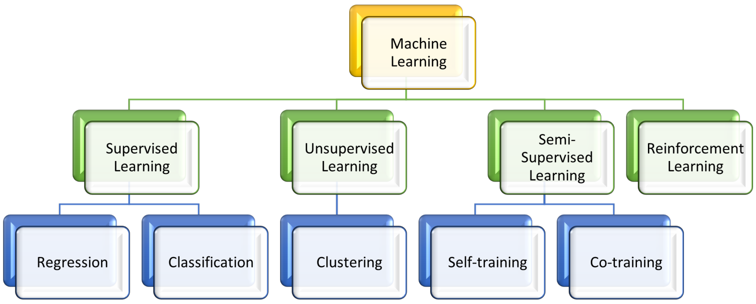


Figure 3.1: Machine Learning Approaches

3.2.1 Supervised Learning

It is the most common and simplest type. The principle of this type is very simple, as if the machine were a student who needed information and supervision from a teacher (training data), so, in order to predict new results based on the training data, the machine creates a model by finding the relationship between the inputs or features, X, and the outputs, Y. After that, the machine predicts new results based on the model [26]. There are many supervised learning algorithms such as artificial neural networks(ANN), K nearest neighbors(kNN) and support vector machines(SVM) [26].

As shown in the Figure 3.1 supervised learning includes two tasks: regression and classification.

a) Regression

The concept of regression is used when we are trying to predict continuous values, such as the prices of cars [27].

b) Classification

Its major purpose is to predict the classification of labels (classes), such as the classification of important or spam emails [27].

3.2.2 Unsupervised Learning

Unsupervised learning relies on unlabeled training data, and the machine tries to learn independently without supervision [28]. Its main advantage is that it can be used to find patterns in data that humans may not easily identify [28]. There are many unsupervised learning algorithms, including K-means and isolated forests [28]. There are several types of unsupervised learning . The most well-known is clustering, as shown in the Figure 3.1.

a) Clustering

It is an important algorithm in unsupervised learning, where data is divided into groups containing similar elements [27]. When you post your photo on a social network site that collects photos of the same person for an organization, the site doesn't know which photos you are putting up and how many people are in them, but it will separate all the faces in the images into groups containing similar faces [27].

3.2.3 Semi-Supervised Learning

It bridges supervised learning and unsupervised learning by using both labeled and unlabeled data to train the model, eliminating the limitations of the previous two types. Supervised learning requires a lot of time to collect learning data, and the main disadvantage of unsupervised learning is that it provides less accurate results [29]. Semi-supervised learning methods are useful in cases where obtaining a sufficient amount of labeled data is difficult or expensive, but large amounts of unlabeled data are easy to acquire [29]. There are a variety of techniques used in semi-supervised learning. The most popular are self-training and co-training, according to Figure 3.1.

a) Self-training

In self-training, a model is trained on labeled data, then used to label unlabeled data [6]. The newly labeled data is augmented with the labeled data, and the process is repeated until convergence is achieved [6]. This technique is suitable for a variety of tasks, including image classification and natural language processing [6].

b) Co-training

In this technique, two models are trained on two different views of the data, and then the unlabeled data is labeled using the two models [6]. In particular, this method is useful when there are multiple representations of the data, such as text and images [6].

3.2.4 Reinforcement Learning

Reinforcement learning is a method of learning how to behave optimally in an environment so as to gain maximum rewards [30]. It is one of the techniques developers use to train their machine learning models. The importance of this method lies in the fact that it allows an agent (feature in a video game or a robot) to learn how to navigate the environment in which it was created [30]. Using a feedback system, usually involving rewards and punishments, the agent optimizes its behavior over time [30]. Gaming, resource management, and robotics are some of the fields in which it is used [30].

3.3 Research Design

3.3.1 Semi-supervised learning technique

A semi-supervised learning method can improve accuracy by taking advantage of unlabeled data, which adds valuable information about key patterns in the data [31]. Also, semi-supervised learning can be a great way to reduce both the time and cost of machine learning projects because labeling data is very expensive and time-consuming [31].

In many cases, it is not practical to label all the data required for the problem. Thus, semi-supervised learning reduces the time and cost of collecting data, allowing for faster model training, since not all data must be labeled [31].

3.3.2 Active learning technique

The active learning domain is characterized by several important applications that demonstrate its utility in real-life, such as natural language processing, image classification, and many more [24]. The use of this technology has many significant benefits. Selecting the most informative samples for labeling reduces labeling costs and improves the efficiency and performance of machine learning models [22]. Through active learning, the model can learn faster than traditional models by focusing on the most relevant samples [22].

3.3.3 ANN algorithm

The artificial neural network is one of the common techniques in machine learning. Its structure and function are similar to those of the human brain, which contains interconnected units called neurons that receive inputs and produce electrical signals that are transmitted between neurons [32].

As shown in Figure 3.2, ANN typically has three key layers: an input layer, an output layer, and hidden layers [32]. Each layer can have a number of neurons [32].

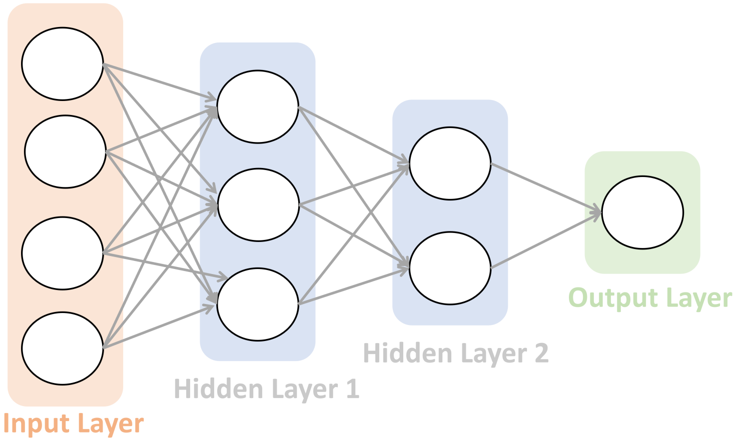


Figure 3.2: ANN layers

The input layer receives external data that needs to analyze or learn [32]. This data then passes through one or several hidden layers, converting the input into data valuable to the output layer [32]. Lastly, the output layer provides output in the form of an artificial neural network's response to the input data provided [32]. ANN is a powerful algorithm that can be used in several domains such as Image Recognition, Speech Recognition and Financial Forecasting [33].

3.3.4 The proposed scheme

In this part, we will explain our approach to improving the performance of IDS.

In this thesis, we leverage the strengths of each precident technique and use it, we propose a semi-supervised active learning model based on the ANN (Artificial Neural Network) algorithm, which could be helpful to improve the performance of IDS on the basis of accuracy, detection rate, precision, and false alarm rate.

As illustrated in Figure 3.3, our proposed approach has two primary phases: a supervised training phase, and a semi-supervised active learning phase.

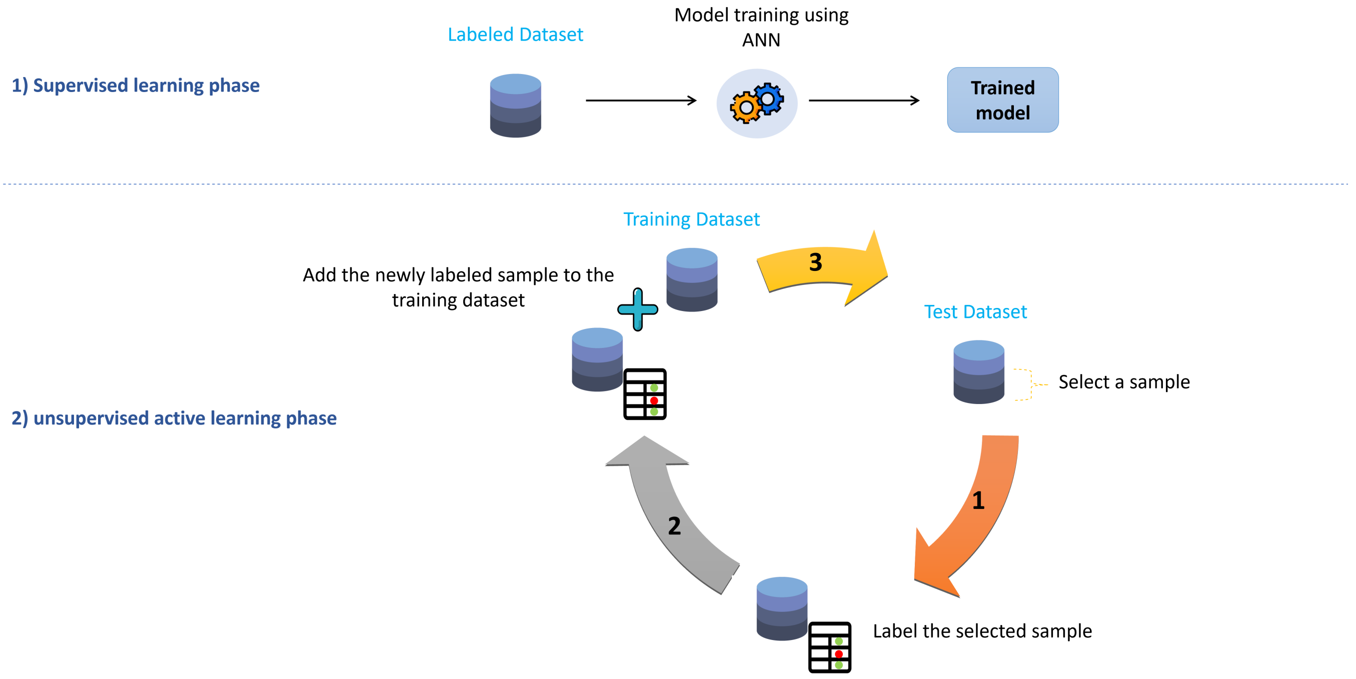

1) Supervised learning phase
Labeled Dataset
Model training using ANN
Trained model
2) unsupervised active learning phase
Training Dataset
3
Test Dataset
Select a sample
1
Label the selected sample
2
Add the newly labeled sample to the training dataset

Figure 3.3: The schema of our proposed method

In the first phase, we trained the classification model using a labeled IDS dataset, through the ANN algorithm i.e. there is a label for each type of traffic, and the IDS model learns from those examples to be able to classify each one.

After that, we performed semi-supervised active learning using the test data during the test phase. The first step was to choose a sample of N points from the test dataset, then to label or annotate this sample, then to add it to the training set, and finally to retrain the model on the updated training set. This process is repeated incrementally.

3.4 Active Learning Strategy

Active learning and semi-supervised learning are two distinct techniques that, when combined, can significantly enhance the performance of machine learning models. Active learning enables the model to strategically select the most informative unlabeled data for labeling [22], while semi-supervised learning leverages both labeled and unlabeled data for training [6].

The integration of active learning with semi-supervised learning starts with a small labeled dataset used for initial model training. As the model interacts with a large unlabeled dataset, it employs specific criteria to select the most informative instances[22]. These criteria often involve choosing instances where the model is most uncertain about categorization, thereby focusing on challenging cases that can enhance the model's ability to distinguish between categories[22]. The selected unlabeled instances are then sent to a human classifier or expert for labeling[22]. The newly labeled cases are incorporated into the training data, and the model is retrained using the expanded dataset[22]. This iterative process continues until the pool of unlabeled data is fully utilized, resulting in a model that benefits from both labeled and unlabeled data to improve its performance[22].

This combined approach empowers models to learn effectively even with limited labeled data, reducing labeling costs and achieving superior accuracy compared to using labeled data alone[22].

3.5 Evaluation Metrics

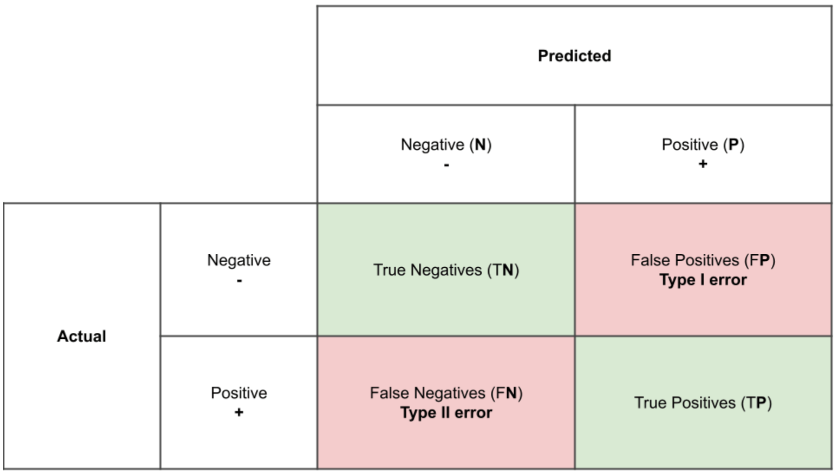
The effectiveness of IDS is determined by their capacity to classify correctly normal and suspicious traffic. Some of the most known metrics that we use in our work are the following:

3.5.1 Confusion Matrix

In this matrix, correct and incorrect guesses are grouped in an NxN matrix.

Confusion matrices have two axes, one for predicted labels and one for actual labels (Classes are represented by N) [34].

As shown in figure 3.4, in binary classifications, such as normal or malicious activity, the confusion matrix has a 2x2 structure:



		Predicted	
		Negative (**N**) **-**	Positive (**P**) **+**
Actual	Negative **-**	True Negatives (T**N**)	False Positives (F**P**) **Type I error**
	Positive **+**	False Negatives (F**N**) **Type II error**	True Positives (T**P**)

Figure 3.4: Confusion Matrix [35]

- True positive (TP): Abnormal ativity are correctly predected as an abnormal.
- False positive (FP): Normal ativity are wrongly predected as abnormal.
- True negative (TN): Normal ativity are correctly predected as normal.
- False negative (FN): Abnormal ativity are wrongly predected as normal.

3.5.2 Accuracy

It represents the percentage of correct predictions made by a classification model [35]. Mathematically, accuracy is calculated as follows:

$$Accuracy = \frac{TN+TP}{TN+FP+TP+FN}$$ [35]

3.5.3 Precision

It identifies the percentage of correct predictions among the total predicted samples [35], as shown below:

$$Precision = \frac{TP}{TP+FP} \text{ [35]}$$

3.5.4 Detection Rate

Detection Rate, also called true positive rate (TPR). It represents the number of correct positive samples divided by the number of all samples that should have been positive [36].It is calculated as follows:

$$DetectionRate = \frac{TP}{TP+FN} \text{ [36]}$$

3.5.5 False Alarm Rate

False Alarm Rate, also called true positive rate (FPR). It is the percentage of negative samples that are considered positive, in comparison to all other negative data [36]. It is identified as follows:

$$FalseAlarmRate = \frac{FP}{FP+TN} \text{ [36]}$$

3.6 Conclusion

In this chapter, we introduced our main method that we work on. Furthermore, we explained why we decided to use each technique and finally we talked about the used evaluation metrics to evaluate our model. In the next chapter, we will present the used dataset and show all the used tools, how we implement our work and the results that we get.

CHAPTER

4

EXPERIMENTAL RESULTS AND DISCUSSION

4.1 Introduction

After presenting the theoretical aspect of our work and machine learning approach in the previous chapters, this chapter will focus on the experimental results we obtained, including the tools used, the dataset, and discuss the achieved results.

4.2 Experimental Setup

4.2.1 Material Equipment

To implement our project, we used a Lenovo T440s ThinkPad laptop, and their features are listed below:

- Processor: Intel (R) Core (TM) i5-7200U CPU @ 2.5GHz 2.71GHz
- Memory (RAM, random memory in phones and computers): 8,00 GB
- Operating System: Windows 10 Pro(64-bit).

4.2.2 Software Tools

a) Python programming language

Python was created in the 1980s by Guido van Rossum, during his research at the National Research Institute for Mathematics and Computer Science in the Netherlands [37]. The first version of Python came out in 1991, following the ABC programming language [38].

Python is a free and open-source, object-oriented, high-level, and easy programming language in terms of reading and usage and is closer to human language [37]. Also it is compatible with multiple operating systems such as Windows, Mac, Linux, and others [39].

Figure 4.1: Python logo [40]

b) Navigator Anaconda

Anaconda Navigator is a desktop graphical user interface GUI and package management platform for Python that enables easy installation, updating, and removal of Python packages [41]. It supports various operating systems, including Windows, macOS, and Linux [41].

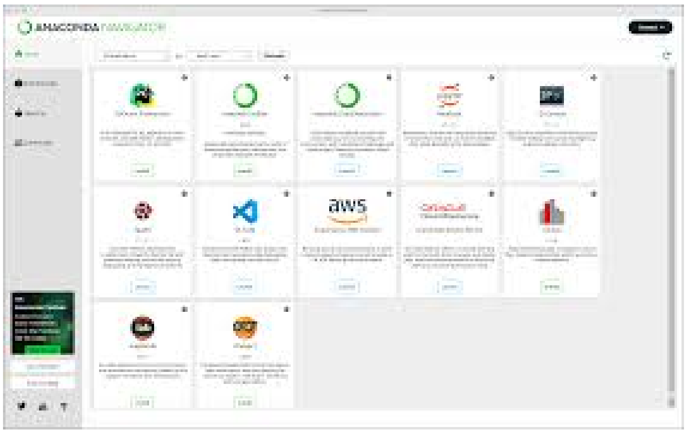


Figure 4.2: Navigator Anaconda interface [42]

c) Jupyter Notebook

Jupyter Notebook is a free, open-source tool from the Jupyter Project. This creation allows you to create and share content documents in which your code is contained, as well as visualize the data [43].

Figure 4.3: Jupyter logo [43]

d) Google Colab

Google Colaboratory, or Colab, is a cloud-based Jupyter Notebook that allows you to write, execute, and share Python code with other members of the team while using your browser [44].

It is also possible to document LaTeX documents, HTML documents, diagrams, and images with Colab [44].

Figure 4.4: Google Colab logo [45]

4.2.3 Library

a)TensorFlow

TensorFlow is an open-source machine learning framework developed by the Google Brain team, it provides a comprehensive ecosystem for building and deploying machine learning models [46]. It is particularly well-suited for tasks involving deep neural networks [46].

Figure 4.5: TensorFlow logo [47]

b) Pandas

Pandas is a powerful and open-source library in Python designed for data manipulation and analysis. It provides a variety of data structures and tools for working with tabular data [48].

Figure 4.6: Pandas logo [49]

c) Matplotlib

Matplotlib is a powerful Python library that lets you create all sorts of visualizations, from basic static charts to interactive and even animated graphics [50]. Matplotlib integrates seamlessly with other scientific Python libraries like NumPy, making data analysis and visualization a streamlined process [50].

Figure 4.7: Matplotlib logo [51]

d) NumPy

NumPy, short for Numerical Python. It is an open-source library used for working with arrays [52]. NumPy can perform operations such as element-wise computations, linear algebra, statistical analysis, and more, with remarkable speed [52].

Figure 4.8: NumPy logo [53]

e) Scikitlearn

Scikit-learn is an essential library for Python, widely used in machine learning projects [54]. It provides a comprehensive suite of machine learning tools, including mathematical, statistical, and general-purpose algorithms, which form the foundation of many machine learning technologies [54].

Figure 4.9: Scikitlearn logo [55]

4.2.4 The Dataset

In our research, we relied on the CICIDS-2017 advanced dataset to train and evaluate our model due to its reliability and it is one of the most recent and widely used datasets in the field of intrusion detection systems [56].CICIDS-2017 is a dataset proposed by the Canadian Institute of Cybersecurity, and it also attracts the attention of many researchers because it covers attacks not included in older datasets [56]. The dataset contains the latest features and threats like infiltration attacks, web attacks, DDoS attacks, heart-bleed attacks, brute force attacks, and botnet attacks. A network of firewalls, modems, switches, routers, as well as various operating systems including Windows, Mac OS X, and Ubuntu was used to collect data for five days for the CICIDS2017 study [56].

4.2.5 Data Preprocessing

Our approach starts with preprocessing the CICIDS2017 dataset, which means cleaning, filtering, and choosing the most relevant features from the data. A number of records in this dataset are redundant, unnecessary, contain null and unknowns and infinite values, so they all need to be addressed.

Figure 4.10 highlights the three main phases which are: Data Initialization, Data Preparation, and Splitting the dataset.

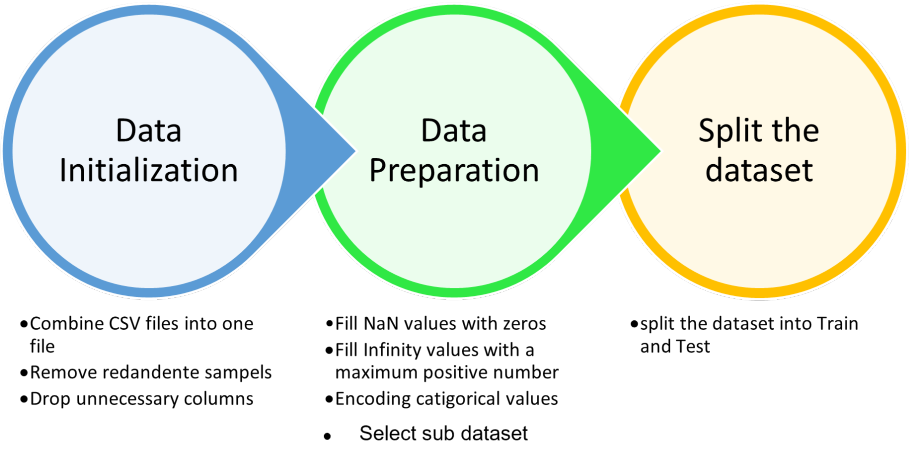

Data Initialization
Data Preparation
Split the dataset
•Combine CSV files into one file
•Remove redandente sampels
•Drop unnecessary columns
•Fill NaN values with zeros
•Fill Infinity values with a maximum positive number
•Encoding catigorical values
• Select sub dataset
•split the dataset into Train and Test

Figure 4.10: Data Preprocessing phases

Data initialization

The dataset is divided into eight files. Processing separate files is a hard task. As a result, we combined those files to form only one file that contained all instances of the previous files.

Then from the total of 85 features, we eliminate redundant values and all features with zero values due to having not enough impact on the classification, also we drop network-specific features such as IP addresses, Flow IDs, and timestamps. , thus leaving only 69 features in the dataset.

Data preparation

In this phase we start by substituting unknown values with zeros and infinities with a maximum positive number much larger than the sum of all features' maximum values.Before the training phase, all variables must be in numerical form. So, we converted categorical variables (Label) into numerical. For binary classification, in the "Label" column, all benign traffic is transformed into "0" and all malicious traffic is transformed into "1". For multi classification each type of traffic in the "Label" column takes a numeric value from zero to nine where label zero represents benign traffic.Considering the vast number of instances and the incompatibility of instance numbers with some types of attack as shown in Figure 4.11, we selected a custom subset that represented about 15% of the whole data set and we randomly reduced instances of Benign, DDoS, DoS, and PortScan in order to balance their distribution.In addition, we substituted unknown values with zeros and infinities with a maximum positive number much larger than the sum of all features' maximum values.

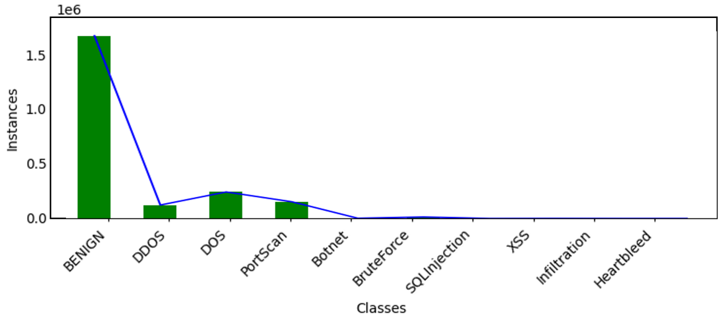


Figure 4.11: Data Distribution

Finally we selected the target feature (Label), placed it in a vector, and dropped it from the original dataset.

Split the dataset

In the splitting phase, we split the dataset into train and test, where the test size is defined as 30% of the total dataset and the rest 70% represents the train data.

Table 4.1 and Table 4.2 show the distribution of each class among the train and test data for both binary and multi-class classification.

Tableau 4.1: Data distribution of the custom CICIDS2017 train and test dataset in binary classification

Class	Train set	Test set
BENIGN	28008	12059
Attacks	53571	34957

Tableau 4.2: Data distribution of the custom CICIDS2017 train and test dataset in multiclass classification

Class	Train set	Test set
BENIGN	28008	12059
DDOS	13941	5871
DoS	13926	5973
PortScan	14032	6089
Botnet	1414	585
Brute Force	9735	4155
SQL injection	14	3
XSS	457	202
Infiltration	29	15
Heartbleed	8	5

4.3 Proposed classification model

4.3.1 Binary classification model

As illustrated in Table 4.3, to build up our IDS model for binary classification, we utilized a three-layer ANN with only one hidden layer of 32 neurons, the input layer consists of 68 neurons, and since our model has two different outputs (Benign or Attack), the output layer neurons are selected as 1. For the input and the hidden layer we used ReLU as an activation function which is a function that returns the same value if x is greater or equal to zero and it returns zero in the other cases [57]. The ReLU equation is shown below.

$$f(x) = \max(0, x)$$ [58]

Also, we used Sigmoid as an activation function for the output layer which represents a logistic function that takes output values between zero and one [58]. It is generally used in the output layer to resolve binary classification problems [58]. The equation below represents a Sigmoid equation.

$$f(x) = \frac{1}{1+e^{-x}}$$ [58]

Tableau 4.3: ANN model-Binary classification

Layer	Neurons number	Activation function
Input	68	Relu
Hidden	32	Relu
Output	1	Sigmoid

The following step is to compile our model and for that we used the hyper-parameters shown in Table 4.4.

Hyperparameters	Values
Optimizer	rmsprop
loss	binary crossentropy
metrics	accuracy

Tableau 4.4: ANN hyperparameters-Binary classification

In the training phase, the model is fitted with training data with batch sizes equal to 20 and 5 epoch where our model achieves its maximum accuracy at it. After that we used active learning technique where the model selects each time a sample of 1000 instances from the test data after that it gives labels for those instances finally the newly labeled data is added to the training set and the model is re-trained on the updated training set. This method is repeated iteratively.

4.3.2 Multiclass classification model

According to Table 4.5, the IDS model for multi-class classification consists of one hidden layer. After many tries with different sizes of neurons, 68 is chosen for the input layer which is equal to the number of features in our dataset, 55 neurons for the hidden layer and since our model has 10 different traffic as outputs we have chosen 10 for the output layer.

Tableau 4.5: ANN model-Multi class classification

Layer	**Neurons number**	**Activation function**
Input	68	Relu
Hidden	55	Relu
Output	10	Softmax

We used ReLU as an activation function for both the input and the hidden layer, and Softmax for the output layer which gives outputs as probabilities where the sum of the results is equal to one [59].If we apply it to a multi-class classification, it gives values between zero and one, so that the target class is the one with the highest probability [59]. Softmax equation is illustrated below.

$$f_i(x) = \frac{exp(x_i)}{\sum_{j=1}^{J} exp(x_j)}$$ [59]

To compile our ANN model we used the hyper-parameters shown in Table 4.6.

Tableau 4.6: ANN hyperparameters-Multi class classification

Hyperparameters	**Values**
Optimizer	rmsprop
loss	categorical crossentropy
metrics	accuracy

Finally for the training phase, our training set is fitted with 2 epochs and a batch size equal to 32. After that we used active learning method, where the model selects each time a sample of 5000 instances from the test data to enhance the performance of our proposed method.

4.4 Performance Evaluation

4.4.1 Performance Evaluation for binary classification

After creating, compiling, and training the model, it comes to the evaluation step, which is an important phase to determine the model's effectiveness and if it is possible to use it in real-world scenarios. As we said in the previous chapter ,we used the Confusion Matrix, Accuracy, Precision, Detection Rate, and False Alarm Rate for assessing our proposed model.

The confusion matrixes of traditional learning and active learning are illustrated in Figure 4.12 and Figure 4.13 respectively. Because benign traffic occupies most of the dataset, the value of TN in both confusion matrixes is the largest compared to the other metrics.

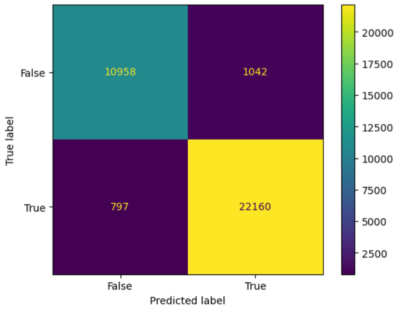


Figure 4.12: Confusion Matrix-Binary Classification for the traditional ANN

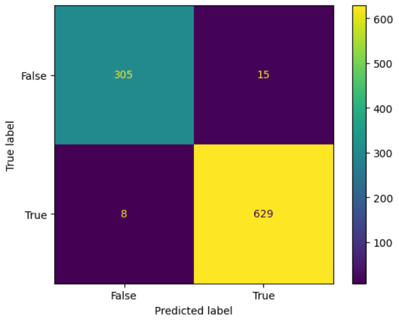


Figure 4.13: Confusion Matrix-Binary Classification for the enhanced ANN using active learning

Because benign traffic occupies most of the dataset, the value of TN in both confusion matrixes is the largest compared to the other metrics.Figure4.14 illustrates the accuracy of both benign and malicious traffic using traditional and active learning. We notice high accuracy for both classes using active learning compared to classical learning.

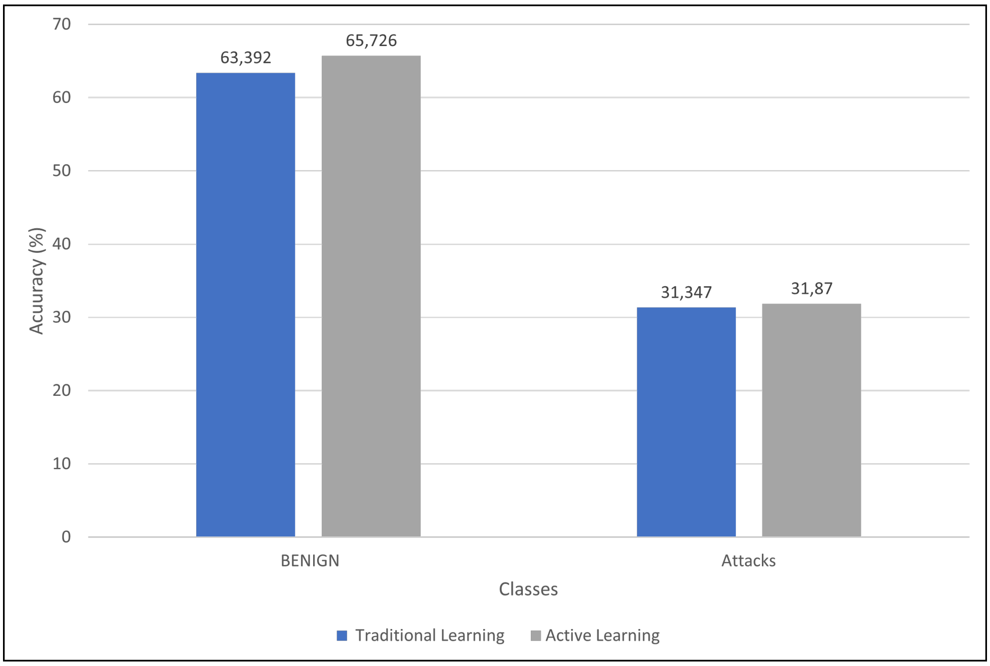


Figure 4.14: Accuracy-Binary Classification

As shown in Figure 4.15 and Figure 4.16, the precision and detection rate are high for the two methods but they are more highest using the active learning technique.

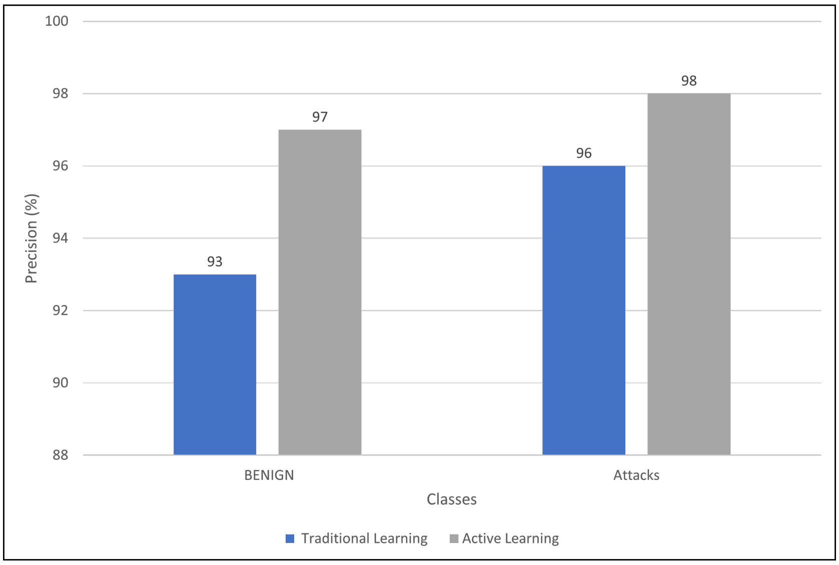


Figure 4.15: Precision Binary Classification

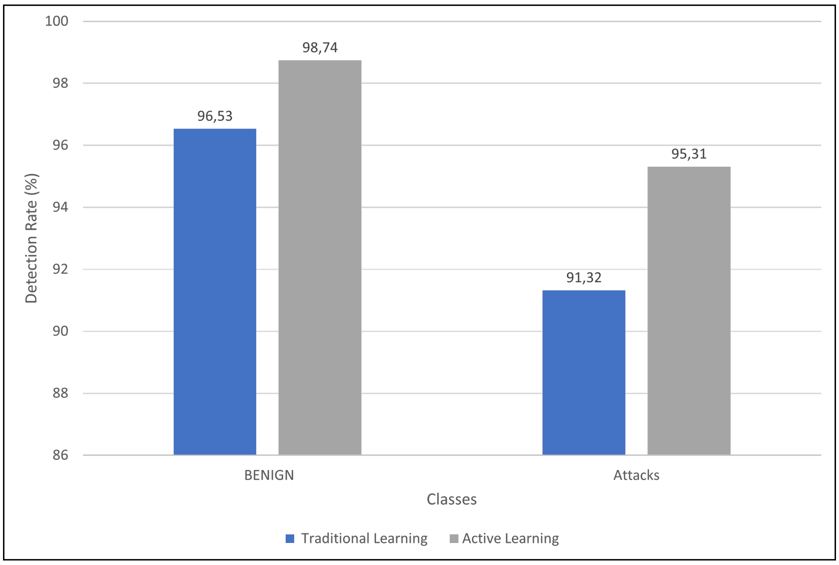


Figure 4.16: Detection Rate-Binary Classification

For the false alarm rate and based on Figure 4.17, we noticed that it is very low for all classes.It represent just 0,09% for benign traffic, 0,05% for the other attacks using traditional learning and 0,04% for benign traffic, 0,02% for the other traffic using active learning, which is considered low compared to the other method.

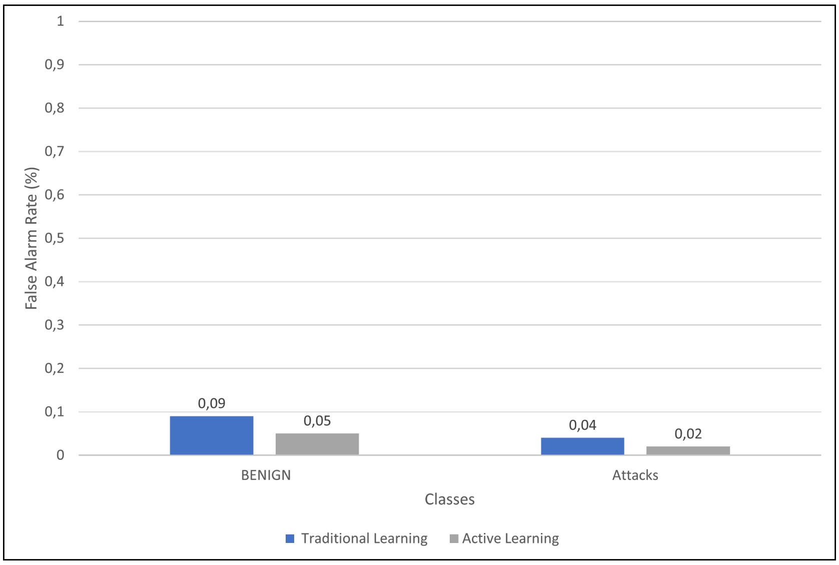


Figure 4.17: False Alarm Rate-Binary Classification

Table 4.7 shows the overall performance results obtained from our experiments using the CICIDS2017 dataset. These results highlight the performance of our approach using active learning compared to traditional learning based on different performance metrics.

Tableau 4.7: Performance Metrics-Binary classification

Performance Metrics	**Values using traditional ANN**	**Values using active learning ANN**
Accuracy	94.74	97.6
Precision	94.5	97.5
Detection Rate	93,93	97.03
False Alarm Rate	0.065	0.035

Figure 4.18 shows a comparison between the performance of traditional learning using ANN and enhanced ANN by active learning in terms of the number of accurate predictions across instances in binary classification.

It can be seen that the number of predictions for both models increases near-linearly in each instance, also the enhanced ANN model performs better than the traditional ANN in the number of correct predictions.

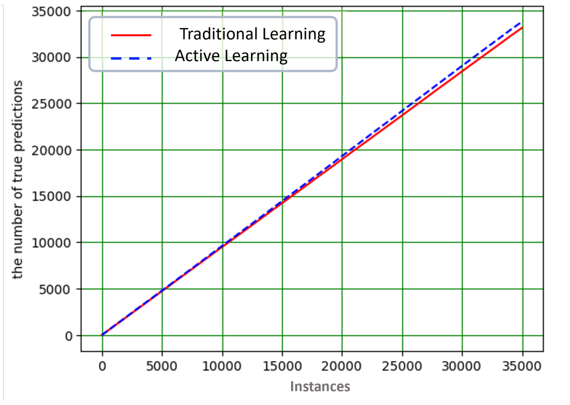


Figure 4.18: Comparison between the traditional ANN and the active learning ANN-Binary Classification

4.4.2 Performance Evaluation for multi-class classification

With the purpose of making predictions based on attack types, We created a multi-class classifier, compiled it, and fitted it. Following that comes the evaluation phase. The confusion matrix of each method is illustrated in Figure 4.19 and Figure 4.20.

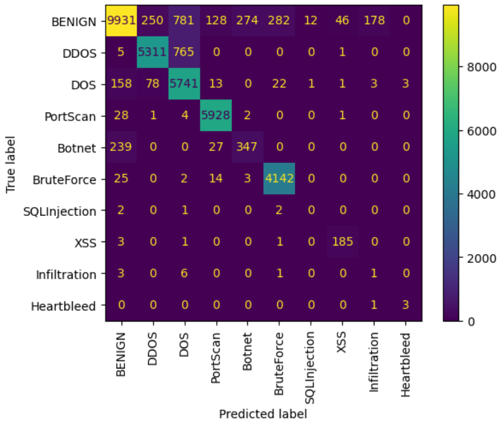


Figure 4.19: Confusion Matrix-Multi -Class Classification for the traditional ANN

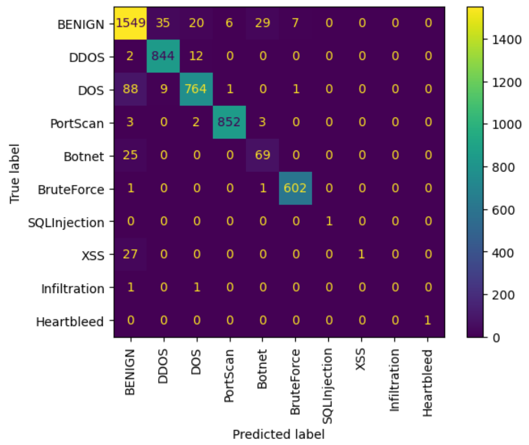


Figure 4.20: Confusion Matrix MultiClass Classification for the enhanced ANN using active learning

We notice that the traditional model was able to classify eight categories relatively correctly, and the remaining one which is SQL injection was not able to classify them properly. The active learning model was also able to classify eight categories relatively correctly including the SQL injection attack, however it was not able to classify the Infiltration attack correctly.

Figure 4.21 illustrates the accuracy for each traffic type using the two methods.

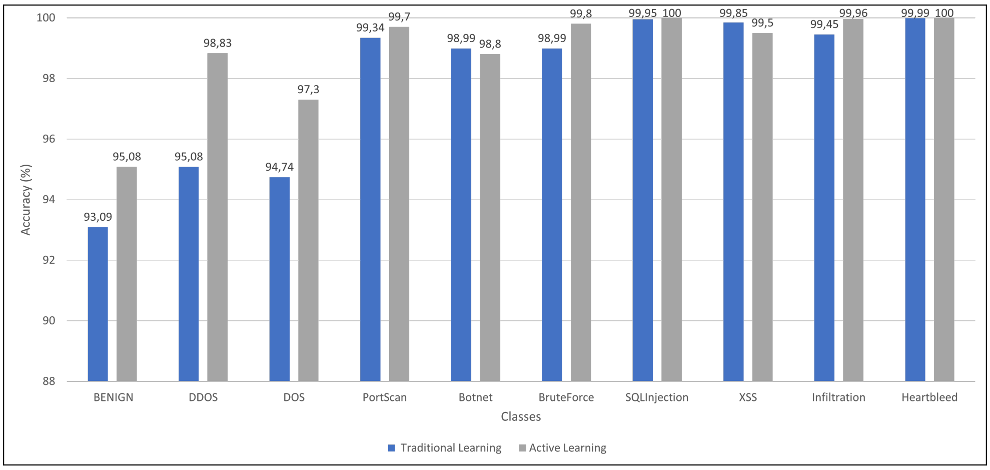


Figure 4.21: Accuracy-Multi -Class Classification

We notice a very high accuracy for all types in both methods where the lowest value is 93.09% for benign traffic using the traditional method and the lowest value is 95.08% for benign traffic using the active learning method.It is also can be seen that the active learning accuracy's are better than the traditional learning accuracy's.

Figure 4.22 illustrates the precision for each traffic type using the two methods.

We notice a high precision for all types using active learning compared to the other method, except for the Infiltration attack the precision is 0% since the active learning model was not able to classify this type correctly.

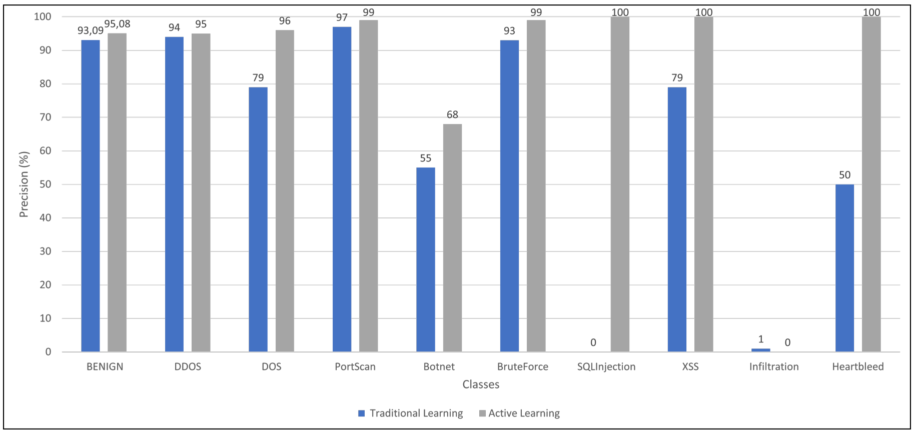


Figure 4.22: Precision-Multi -Class Classification

The detection rate results for active learning and traditional learning are shown in Figure 4.23. We notice an acceptable percentage for the two methods, however the The detection rate for active learning is higher than traditional learning except for the Infiltration type.

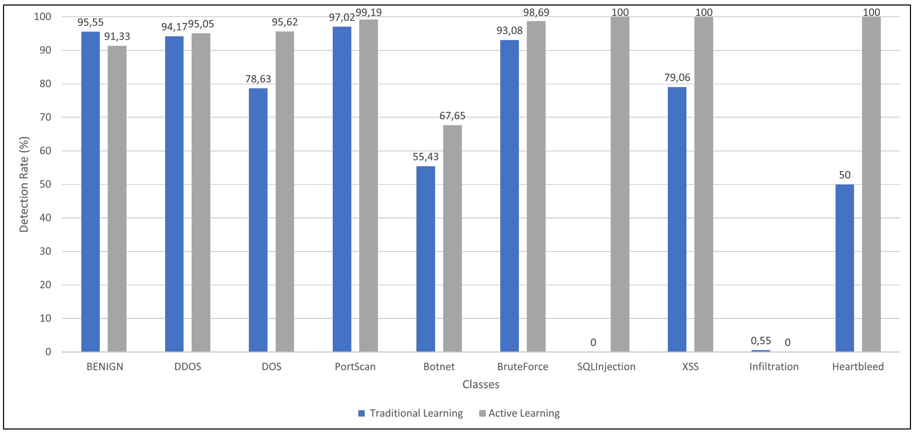


Figure 4.23: Detection Rate-Multi -Class Classification

The results of false alarm rate for active learning and traditional learning are shown in Figure 4.24.

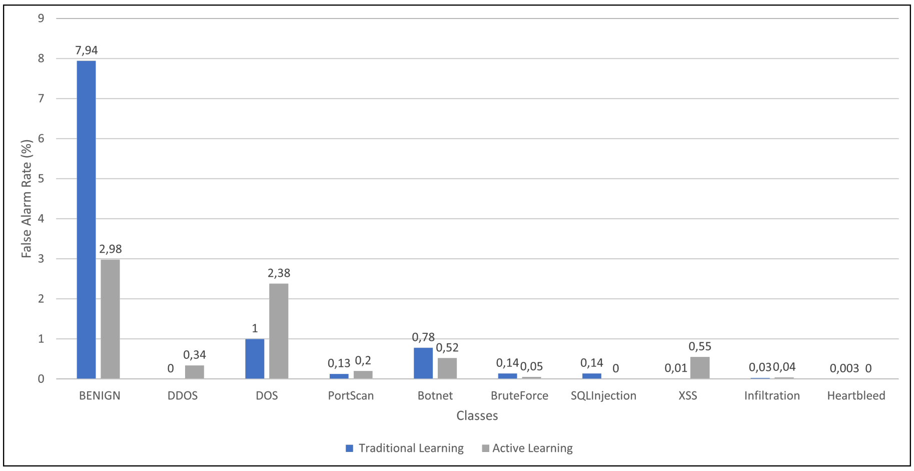


Figure 4.24: False Alarm Rate-Multi -Class Classification

It can be seen that both models have very low false alarm rate, where the higher value for the traditional one is 7.94% for benign and the lower value is 0.003% only for Heartbleed, and for the active learning technique the higher percentage is 2.98% for benign traffic and the lower value is 0% for both SQL Injection and Heartbleed.

The overall performance results for the two methods are summarized in Table 4.8 below:

Tableau 4.8: Performance Metrics- Multi -Class classification

Performance Metrics	**Values using traditional ANN**	**Values using active learning ANN**
Accuracy	90.37	94.47
Precision	64,12	85,21
Detection Rate	64.35	84.75
False Alarm Rate	1.02	0.71

Figure 4.25 compares traditional learning using ANN with enhanced learning through active learning based on the number of accurate predictions across instances in multi-class classification.

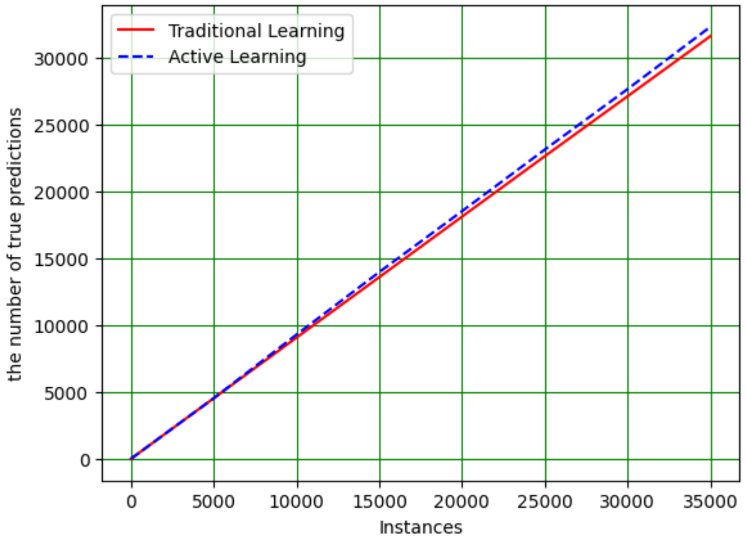


Figure 4.25: Comparison between the traditional ANN and the active learning ANN-Multi -Class Classification

In each instance, the number of predictions increases near-linearly for both models, but it is remarkable that the enhanced ANN model shows better results than the traditional ANN model in terms of correct predictions.

4.5 Discussion

Evaluation of a semi-supervised learning based intrusion detection system using active learning provided many interesting implications as it significantly contributed to higher performance metrics than traditional ANN.This progress means that the system has become more accurate at detecting malicious activities, reducing the rate of false positive alarms. As a result, false alarm rates decrease, which is beneficial to reducing the negative effects of false alarms. This was especially noticeable in attacks of the kind SQL Injection and Heartbleed. This enhancement in performance is due to the utilization of active learning.

Table 4.9 illustrates some recent works that focused on intrusion detection through the implementation of different machine learning techniques and neural networks. Based on Table 4.9, it is clear that our proposed method outperforms all the mentioned works and definitively proves its effectiveness.

Tableau 4.9: A comparison between our model and other models

Publications	**Model**	**Type**	**Accuracy**	**Precision**	**DR**	**FAR**
[60]	MLP	Multi-class	87.630%	-	-	-
[60]	Random Forest	Binary	96,578%	-	96,565%	3,410%
[60]	Naive Bayes	Binary	67,553%	-	42,190%	7,085%
[61]	KNN	Binary	57.76%	-	-	-
[62]	MLP	Multi-class	-	-	77%	-
[62]	KNN	Multi-class	-	-	96%	-
[62]	Quadratic Discriminant Analysis	Multi-class	-	97%	-	-
[63]	MLP	Binary	91%	-	-	-
[64]	CNN	Binary	95%	96%	-	-

4.6 Limitations

An important limitation of our research lies in the uneven distribution of data in the CICIDS2017 dataset. Specifically, the scarcity of certain types of attacks, such as SQL injection and enfiltration, poses significant obstacles for the model's ability to effectively learn and detect these attacks.

The dataset's imbalance means that there are far fewer examples of these rare attack types compared to more common ones, leading to a skewed learning process. Consequently, the model tends to perform well on detecting frequent attacks but struggles with accurately identifying the underrepresented ones.

This imbalance results in lower detection rates for these specific categories of attacks, which in turn negatively impacts the overall performance and reliability of the intrusion detection system. The inability to adequately detect rare but potentially severe threats undermines the system's effectiveness and highlights the need for more balanced and comprehensive datasets, as well as advanced techniques to address data imbalance in training.

4.7 Conclusion

In this chapter, we presented the different tools used, the dataset and all steps we have followed to achieve the best performance for our model. We have also discussed results in terms of accuracy, precision, detection rate and false alarm rate.

CHAPTER

5

GENERAL CONCLUSION

In light of the success achieved in the cybersecurity domain, particularly in the IDS field, using different artificial intelligence techniques, and after reviewing the literature and existing methods in this field, we decided to work on IDS using artificial intelligence methods.

To improve traffic classification compared to existing work based on machine learning, we developed an IDS approach that integrates both semi-supervised learning and active learning.

For that we used the ANN method for both binary and multi-class classification and we used the CICIDS2017 dataset inspired by some previous works, which have worked on it to resolve the same problem.

This thesis was divided into four theoretical chapters and one application chapter, beginning with a general introduction and concluding with a general conclusion.

Firstly, we talk about IDs, active learning, and some previous works.

We explained also all the steps of our methodology as well as the concepts we used in it, and we presented evaluation metrics so that we could compare the model's performance to other existing models.

In the following chapter, we mentioned the used tools, the dataset and all preprocessing steps, our proposed models for both binary and multi-class classification, and finally, we presented all the results we obtained in detail.

It is clear from our results and from the previous works that the proposed method using active learning enhanced the performance of the IDS and that our results were promising, where we achieved accuracy equal to 97% for binary classification and 94% for multiclass classification.

Future work should focus on addressing the limitations and challenges identified in our current research to enhance the effectiveness and robustness of intrusion detection systems (IDS).

- To resolve the data imbalance issue inherent in the CICIDS2017 dataset, our future plans include transitioning to the more comprehensive and updated CICIDS2019 dataset. The CICIDS2019 dataset offers a richer and more diverse set of attack scenarios, including additional types of cyber-attacks and improved data quality that reflects more current and realistic network traffic conditions.
- This innovative approach, Semi-Supervised Learning Based Using Active Learning, can be used in areas that need a large amount of data but are not available or have no names, such as the Internet of Things.

Bibliography

[1] :Kemp, S. (2024, January 31). Internet use in 2024 — DataReportal – Global Digital Insights. DataReportal – Global Digital Insights. https://datareportal.com/reports/digital-2024-deep-dive-the-state-of-internet-adoption

[2] :Görnitz, N., Kloft, M., Rieck, K., & Brefeld, U. (2009). Active learning for network intrusion detection. https://doi.org/10.1145/1654988.1655002

[3] : Kanimozhi, V., & Jacob, T. P. (2021). Artificial Intelligence outflanks all other machine learning classifiers in Network Intrusion Detection System on the realistic cyber dataset CSE-CIC-IDS2018 using cloud computing. ICT Express, 7(3), 366–370. https://doi.org/10.1016/j.icte.2020.12.004

[4] : B, S., & K, M. (2019). Firefly algorithm based feature selection for network intrusion detection. Computers & Security, 81, 148–155. https://doi.org/10.1016/j.cose.2018.11.005

[5]: Intrusion Detection Systems: Supervised learning in intrusion detection system — Saylor Academy. (n.d.). Saylor Academy. https://learn.saylor.org/mod/book/view.php?id=29755&chapterid=5434

[6] :Semi-Supervised Learning, Explained with Examples. (2024, March 29). AltexSoft. https://www.altexsoft.com/blog/semi-supervised-learning/

[7] : Appen. (2023, December 11). ML Techniques: Active Learning vs Weak Supervision. https://www.appen.com/blog/ml-techniques-active-learning-vs-weak-supervision

[8] : Aljanabi, M., Ismail, M. A., & Ali, A. H. (2021). Intrusion Detection Systems, Issues, Challenges, and Needs. the International Journal of Computational Intelligence Systems/International Journal of Computational Intelligence Systems, 14(1), 560. https://doi.org/10.2991/ijcis.d.210105.001

[9] : Boumaza, A., & Naceur, M. D. (2013). Les Systèmes De Détection D'intrusions Coopératifs Avec La Logique De Description [Mémoire de master, Université Daad Dahleb].https://bucket.theses-algerie.com/files/repositories-dz/1516788808449786.pdf

[10] : Maza, S. (2019). Un Système De Vérification Et De Validation De La Sécurité Et L'intégration Évolutive Adaptative De La Protection Dans Les Systèmes D'informations Avancés [Thèse de doctorat, Université Ferhat Abbas - Sétif -1]. https://bucket.theses-algerie.com/files/repositories-dz/1430746674145447.pdf

[11] : Intrusion Detection System (IDS): Types, Techniques, and Applications. (2024, June 6). https://www.knowledgehut.com/blog/security/intrusion-detection-system

[12] :GeeksforGeeks. (2024, May 23). Intrusion Detection System (IDS). GeeksforGeeks. https://www.geeksforgeeks.org/intrusion-detection-system-ids/

[13] :DAOUADI, Z., & ABBAS, A. (2016). DETECTION D'INTRUSION DANS LES RESEAUX VANETS [MEMOIRE DE MASTER, Université de Larbi Tébessi]. https://bucket.theses-algerie.com/files/repositories-dz/2771625447504138.pdf

[14] : Jacky. (2024, May 29). What is Network Intrusion Detection System (NIDS)? - Sapphire.net. wordpress-331244-3913986.cloudwaysapps.com. https://www.sapphire.net/insights/nids/

[15] : Alshamy, R., & Ghurab, M. (2020). A Review of Big Data in Network Intrusion Detection System: Challenges, Approaches, Datasets, and Tools. ResearchGate. https://www.researchgate.net/publication343452562_A_Review_of_Big_Data_in_Network_Intrusion_Detection_System_Challenges_Approaches_Datasets_and_Tools

[16] : Ould Bechiry, A. (2021). Minimizing The Rate Of False Positives In Intrusion Detection Systems By Considering The Context Changes [Master's Thesis, University of Blida 1]. https://bucket.theses-algerie.com/files/repositories-dz/3182537155006481.pdf

[17]: Rastegari, S. (2015). Intelligent network intrusion detection using an evolutionary computation approach. https://ro.ecu.edu.au/cgi/viewcontent.cgi?article=2761&context=theses

[18]: Chen, C., Gong, N. Y., & Tian, N. Y. (2008). Semi-supervised learning methods for network intrusion detection. Conference Proceedings/Conference Proceedings – IEEE International Conference on Systems, Man, and Cybernetics. https://doi.org/10.1109/icsmc.2008.4811688

[19]: Duong, N. H., & Hai, H. D. (2015). A semi-supervised model for network traffic anomaly detection. https://doi.org/10.1109/icact.2015.7224759

[20]: Varal, A. S., & Wagh, S. K. (2018). Misuse and Anomaly Intrusion Detection System using Ensemble Learning Model. https://doi.org/10.1109/icrieece44171.2018.9009147

[21]: Aouedi, O., Piamrat, K., Muller, G., & Singh, K. (2022). FLUIDS: Federated Learning with semi-supervised approach for Intrusion Detection System. https://doi.org/10.1109/ccnc49033.2022.9700632

[22]: Hvilshøj, F. (2024, April 30). Active Learning in Machine Learning: Guide & Strategies [2024]. https://encord.com/blog/active-learning-machine-learning-guide/

[23]: Active Learning? – Definition, Strategies, Algorithms, Models. (2024, April 29). clickworker.com. https://www.clickworker.com/ai-glossary/active-learning/

[24]: Kundu, R. (2024, April 10). Active Learning in Machine Learning [Guide & Examples]. V7. https://www.v7labs.com/blog/active-learning-guide

[25]: Platanios, E. A., Stretcu, O., Neubig, G., Poczos, B., & Mitchell, T. (2019). Competence-based Curriculum Learning for Neural Machine Translation. https://doi.org/10.18653/v1/n19-1119

[26]: Theobald, O. (2017). Machine Learning for Absolute Beginners: A Plain English Introduction. Afrique du Sud: Scatterplot Press.

[27]: Müller, A. C., & Guido, S. (2016). Introduction to Machine Learning with Python: A Guide for Data Scientists. O'Reilly Media.

[28]: Géron, A. (2019). Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. O'Reilly.

[29]: Reddy, Y. C. a. P., Viswanath, P., & Reddy, B. E. (2018). Semi-supervised learning: a brief review. International Journal of Engineering & Technology, 7(1.8), 81. https://doi.org/10.14419/ijet.v7i1.8.9977

[30]: Hashemi-Pour, C., & Carew, J. M. (2023, August 16). reinforcement learning. Enterprise AI. https://www.techtarget.com/searchenterpriseai/definition/reinforcement-learning

[31]: Panarin, R. (2023, July 13). What Semi-Supervised Learning Is and How Does It Work. Custom Software Development Company. https://maddevs.io/blog/what-is-semi-supervised

[32]: GeeksforGeeks. (2023, June 2). Artificial Neural Networks and its Applications. GeeksforGeeks. https://www.geeksforgeeks.org/artificial-neural-networks-and-its-applicat

[33]: S Akash, & S Akash. (2023, February 14). Top 10 Applications of Artificial Neural Networks in 2023. Analytics Insight. https://www.analyticsinsight.net/artificial-intelligence/top-10-applications-of-artificial-neural-networks-in-2023

[34]: GeeksforGeeks. (2024, May 13). Confusion Matrix in machine learning. GeeksforGeeks. https://www.geeksforgeeks.org/confusion-matrix-machine-learning/

[35]: Learn And Code Confusion Matrix With Python. (n.d.). NBSHARE. https://www.nbshare.io/notebook/626706996/Learn-And-Code-Confusion-Matrix-With-Python/

[36]: What is False Positive Rate. (2024, February 15). Iguazio. https://www.iguazio.com/glossary/false-positive-rate/

[37]: GeeksforGeeks. (2024a, March 26). What is Python? it s Uses and Applications. GeeksforGeeks. https://www.geeksforgeeks.org/what-is-python/

[38]: GeeksforGeeks. (2024c, June 3). History of Python. GeeksforGeeks. https://www.geeksforgeeks.org/history-of-python/

[39]: What is Python? — Teradata. (2023, October 12). https://www.teradata.com/glossary/what-is-python

[40]: ZDNet, E. (2024, April 25). Python, une définition en un clic. ZDNET. https://www.zdnet.fr/lexique-it/python-une-definition-39928695.htm

[41]: Anaconda Navigator Plateforme de gestion de packages pour Python. (2023, May 23). Diskod. https://diskod.ma/produit/anaconda-navigator/

[42]: Anaconda Navigator — Anaconda documentation. (n.d.). https://docs.anaconda.com/free/navigator/

[43]: The Jupyter Notebook — Jupyter Notebook 7.2.0 documentation. (n.d.). https://jupyter-notebook.readthedocs.io/en/stable/notebook.html

[44]: Google Colab. (n.d.). https://research.google.com/colaboratory/faq.html?hl=fr

[45]: Mike. (2024, June 5). Google Colab. BeginCodingNow.com. https://begincodingnow.com/google-colab/

[46]:TensorFlow: Serdar Yegulalp. What is tensorflow? the machine learning library explained, [On-line; accessed May 18, 2023]. https://www.infoworld.com/article/3278008/what-is-tensorflow-the-machine-learning-library-explained.html.

[47]: Kadimisetty, A. (2018, July 9). TensorFlow — A hands-on approach - Towards Data Science. Medium. https://towardsdatascience.com/tensorflow-a-hands-on-approach-8614372f

[48]: Pandas for data management and data analysis. (2023, August 17). Svitla Systems, Inc. Retrieved May 8, 2024, from https://svitla.com/blog/pandas-for-data-management-and-data

[49]: Hutson, G. (2020, October 5). Python Pandas Pro – Session Three – Setting and Operations. Hutsons-hacks. https://hutsons-hacks.info/python-pandas-pro-session-three-settin

[50]: Matplotlib :Margaret, R. (2019, July 2). Matplotlib. techopedia. Retrieved may 8, 2024, https://www.techopedia.com/definition/33861/matplotlib

[51]: File:Matplotlib icon.svg - Wikimedia Commons. (2015, March 11). https://commons.wikimedia.org/wiki/File:Matplotlib_icon.svg

[52]: NumPy :NumPy Introduction. (n.d.). NumPy. Retrieved may 8, 2024, from https://www.w3schools.com/python/numpy/numpy_intro.asp

[53]: AhmedAgiza. (2022, December 28). Getting Started with NumPy: A Beginner's Guide - AhmedAgiza - Medium. Medium. https://medium.com/@AIisDUMB/getting-started-with-nu

[54]: Scikit-learn :Margaret ,R. (2019, July 2). Scikit-learn. techopedia. Retrieved may 8, 2024. https://www.techopedia.com/definition/33860/scikit-learn

[55]: Ivory. (2024, January 30). Installing Scikit Learn Using pip: A Beginner's Guide — Python Central. Python Central. https://www.pythoncentral.io/installing-scikit-learn-using-

[56]: IDS 2017 — Datasets — Research — Canadian Institute for Cybersecurity — UNB. (n.d.-b). https://www.unb.ca/cic/datasets/ids-2017.html

[57]: ReLU Activation Function — Dremio. (2023, September 21). Dremio. https://www.dremio.com/wiki/relu-activation-function/

[58]: Sharma, S. (2022, November 20). Activation Functions in Neural Networks - Towards Data Science. Medium. https://towardsdatascience.com/activation-functions-neural-network

[59]: Saxena, S. (2024, June 14). Introduction to Softmax Activation Function for Neural Network. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/04/introduction-to-softmax-for-neural-network/

[60]: Ahmim, A., Ferrag, M. A., Maglaras, L., Derdour, M., & Janicke, H. (2020). A Detailed Analysis of Using Supervised Machine Learning for Intrusion Detection. In Springer proceedings in business and economics (pp. 629–639). https://doi.org/10.1007/978-3-030-36126-6_70

[61]: Aksu, D., Üstebay, S., Aydin, M. A., & Atmaca, T. (2018). Intrusion Detection with Comparative Analysis of Supervised Learning Techniques and Fisher Score Feature Selection Algorithm. In Communications in computer and information science (pp. 141–149). https://doi.org/10.1007/978-3-030-00840-6_16

[62]: Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A. (2018). Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. https://doi.org/10.5220/0006639801080116

[63]: Ustebay, S., Turgut, Z., & Aydin, M. A. (2018). Intrusion Detection System with Recursive Feature Elimination by Using Random Forest and Deep Learning Classifier. https://doi.org/10.1109/ibigdelft.2018.8625318

[64]: Getman, A. I., Goryunov, M. N., Matskevich, A. G., Rybolovlev, D. A., & Nikolskaya, A. G. (2023). Deep Learning Applications for Intrusion Detection in Network Traffic. Trudy Instituta Sistemnogo Programmirovaniâ RAN/Trudy Instituta Sistemnogo Programmirovaniâ, 35(4), 65–92. https://doi.org/10.15514/ispras-2023-35(4)-3