Flow-based Network Intrusion Detection System using Decision Tree over Big Data

BY

Afroza Rahman
ID: 191-15-12465

Tanjina Akter Jame
ID: 191-15-12165

Al Amin
ID: 191-15-12420

This Report Presented in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Computer Science and Engineering

Supervised By

Dr. Md Zahid Hasan
Associate Professor
Department of CSE
Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH

JANUARY 2023

APPROVAL

This Project titled **"Flow-based Network Intrusion Detection System using Decision Tree Over Big Data,"** submitted by Afroza Rahman, ID No: 191-15-12465, Tanjina Akter Jame, ID No: 191-15-12165, and Al Amin, ID No: 191-15-12420 to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfilment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 25 January, 2023.

BOARD OF EXAMINERS

Chairman

Dr. Touhid Bhuiyan
Professor and Head
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner

Dr. Md. Monzur Morshed
Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner

Dewan Mamun Raza
Senior Lecturer
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

External Examiner

25.1.2023

Dr. Ahmed Wasif Reza
Associate Professor
Department of Computer Science and Engineering
East West University

DECLARATION

We hereby declare that this project has been done by us under the supervision of **Md Zahid Hasan, Associate professor, Department of CSE,** Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for the award of any degree or diploma.

Supervised by:

Md Zahid Hasan
Associate Professor
Department of CSE
Daffodil International University

Submitted by:

Afroza

Afroza Rahman
ID: 191-15-12465
Department of CSE
Daffodil International University

Tanjina

Tanjina Akter Jame
ID: 191-15-12165
Department of CSE
Daffodil International University

আল- আমিন

Al Amin
ID: 191-15-12420
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

First we express our heartiest thanks and gratefulness to Almighty Allah for His divine blessing which makes us possible to complete the final year project/internship successfully.

We are really grateful and wish our profound indebtedness to **Md Zahid Hasan**, **Associate professor**, Department of CSE, Daffodil International University, Dhaka, deep knowledge & keen interest of our supervisor in the field of Machine Learning to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts, and correcting them at all stages have made it possible to complete this project.

We would like to express our heartiest gratitude to **Dr. Touhid Bhuiyan**, Head**,** Department of CSE, for his kind help in finishing our project and to other faculty members and the staff of the CSE department of Daffodil International University.

Finally, we must acknowledge with respect the constant support and patients of our parents.

ABSTRACT

In computer networks with constantly increasing traffic volumes, flow-based NIDS is the best option for detecting intrusion attempts. In recent years, different machine learning algorithms have been used to detect intrusions in the network. Some of these algorithms showed outstanding performance but are time-consuming and costly. To overcome these problems, Decision Tree has been proposed. In this research, Decision Tree have been used to identify known and unknown attacks on traffic. It executes decision rules in real-time while creating a tree model. That's why it is time-saving. Random Forest, Support Vector Machine, Naive Bayes, Artificial Neural Network, and Deep Neural Network also have been used to show comparison with the Decision Tree. Obtaining a promising result on the dataset "LUFlow" from Lancaster University, we concluded Decision Tree could be used as an intrusion detection model.

TABLE OF CONTENTS

PAGE NO

Approval i

Declaration ii

Acknowledgement iii

Abstract iv

CHAPTER

Chapter 1: INTRODUCTION 1-3

1.1 Introduction 1

1.2 Problem Statement 2

1.3 Research Objective 3

1.4 Research Questions 3

1.5 Report Layout 3

Chapter 2: LITERATURE REVIEW 4-7

2.1 Related Works 4

2.2 Scope of Problem 6

2.3 Challenges 7

Chapter 3: MATERIAL AND METHODS 8-14

3.1 Background of Flow-based approach 8

3.2 Data Collection and Preprocessing 9

3.3 Decision Tree as Intrusion Detection Model 11

3.4 Machine Learning Algorithms for Comparison 12

3.4.1 Random Forest 12

3.4.2 Support Vector Machine 13

3.4.3 Naive Bayes 13

3.4.4 Artificial Neural Network 13

3.4.5 Deep Neural Network 13

Chapter 4: Experimental Results and Discussion 15-22

4.1 Result and Discussion 15

4.2 Comparative Analysis 21

Chapter 5 Conclusion and Future Work 23-27

5.1 Conclusion 23

5.2 Future Work 23

5.3 Limitations 24

Appendix 25

Reference 26

LIST OF FIGURES

FIGURES	PAGE NO
Figure 3.3: Workflow of the Decision Tree	12
Figure 4.1.1: Accuracy of the models	15
Figure 4.1.2: F1-score of the models on the training and testing dataset	16
Figure 4.1.3: Time consumption for training on LUFlow dataset	17
Figure 4.1.4: Confusion matrix of each model on LUFlow dataset	17

F

LIST OF TABLES

TABLES	PAGE NO
TABLE 3.2: DATASET DESCRIPTION	09
TABLE 4.1.1: PERFORMANCE OF DECISION TREE	18
TABLE 4.1.2: PERFORMANCE OF RANDOM FOREST	18
TABLE 4.1.3: PERFORMANCE OF SUPPORT VECTOR MACHINE	19
TABLE 4.1.4: PERFORMANCE OF NAIVE BAYES	19
TABLE 4.1.5: PERFORMANCE OF ARTIFICIAL NEURAL NETWORK	20
TABLE 4.1.6: PERFORMANCE OF DEEP NEURAL NETWORK	20
TABLE 4.1.7: ACCURACY EVALUATION OF DIFFERENT MODELS	20
TABLE 4.2: COMPARATIVE ANALYSIS	21

CHAPTER 1
INTRODUCTION

1.1 Introduction

According to a new analysis, there has been a record 50% year-over-year rise in hands-on intrusion attempts and significant improvements in threat patterns and adversary tradecraft [1]. Among the most prevalent threats include Network scans, denial-of-service assaults, and brute-force attacks [2]. By generating traffic loads too large for systems to monitor thoroughly, attackers can induce chaos and congestion in network environments, enabling them to conduct undetected cyberattacks. Such malevolent acts put people and groups of organizations, such as government, financial, and health institutions, at risk. A network intrusion detection method has so been suggested.

A network-based intrusion detection system (NIDS) tracks and analyses all network traffic. It is installed at a crucial point on the network to monitor traffic on all network devices. The entire subnet's traffic is examined and compared with a database of known assaults. There are two main approaches for network-based attacks, packet-based and flow-based attacks. The packet-based technique analyzes the data packet information for anomaly identification. Deep packet inspection is used previously, accounting for header and payload data for each packet. Because a large amount of data needs to be processed, deep packet inspection is too expensive for real-time categorization in terms of energy and processing costs. Since flow-based techniques can classify the entirety of the traffic while only analyzing a small portion of the total volume, it provides promising results for real-time traffic classification. Moreover, flow-based intrusion detection does not examine the traffic payload; it simply examines the packet header. Therefore, it is an innovative technique for identifying intrusions on high-speed networks [2].

Because malicious intrusions are becoming more common, we need a way to detect them accurately. In recent years, different types of Machine Learning (ML) based algorithms have been used in Network Intrusion Detection Systems. Such as decision trees, random forests, kneighbors, Support Vector Machines (SVM), and so on. Among these machine learning algorithms, Decision Tree is better.

Most machine learning methods require numerical input and output variables, but Decision trees can simultaneously handle categorical and numerical variables as features. It is efficient and works well with massive data sets. To predict an unlabeled dataset, it is necessary to train on labeled data. Decision trees have several features that make them more adaptable than other classifiers [3]. In this work, the decision tree has been used to categorize input data as benign, outlier, or malicious with other machine learning algorithms.

As for the dataset, we have used Lancaster University's dataset named 'LUFlow' [4] collected from Kaggle. LUFlow is the dataset that is used to train and test different algorithms. This dataset consists of 16 features. The information is gathered using Cisco's Joy product. Numerous measurements pertaining to flow are gathered by this equipment. LUFlow is a flow-based data collection that includes a robust ground truth based on malicious behavior correlation. Through the arrangement of honeypots inside the address space of Lancaster University, it contains telemetry containing newly emerging attack vectors [4].

1.2 Problem Statement

The Intrusion Detection System (IDS) is software that uses various machine learning algorithms to find intruders on a network. IDS protects computer networks from (potential) unauthorized user access and monitors networks or systems for malicious activity. Distinguishing between intrusive and normal network traffic activity is very difficult and time-consuming. Analysts have to go through all of this massive, voluminous data to find the order of intrusions into network connections. Therefore, we need a way to detect network intrusions and reflect current network traffic. So, Network Intrusion Detection System (NIDS) has been proposed. When an attacker attempts to penetrate the network, a NIDS is designed to notify the system administrator.

1.3 Research Objectives

i. To discover illegal access to a computer network by analyzing network data for evidence of malicious behavior.

ii. To protect the network from intrusion is one of the most crucial elements of the system and network administration and security.

iii. To detect violations of corporate security policy and other internal dangers, and to detect and deal with both insider and external attacks.

1.4 Research Questions

i. How does intrusion happen on a network?

ii. On which type of devices can intrusion happen?

iii. Who are the attackers?

iv. Why do they cause intrusion?

v. Can we detect intrusions using different algorithms?

vi. How can we find evidence of harmful activity by searching network data for unauthorized access to a computer network?

vii. How can we identify internal security policy violations and other threats, as well as identify and respond to both insider and outsider attacks?

1.5 Report Layout

i. The introduction to the research, its objectives, and its leading research questions are presented in Chapter 1.

ii. The focus of Chapter 2 is a thorough analysis of the related literature.

iii. The proposed methodology is described briefly in Chapter 3.

iv. The analysis of the results and their relation to previous work are explained in Chapter 4.

v. In Chapter 5, the current research is concluded, along with suggestions for future work.

CHAPTER 2
LITERATURE REVIEW

In this section, different papers have been reviewed based on flow-based intrusion detection systems on different algorithms. Some problems have been discussed with the challenges we faced during the research.

2.1 Related works

Several researchers find Flow-based intrusion detection a hot topic of research to detect intrusions. The identification of malicious flows has been addressed by several flow-based models that employ machine learning and statistical methods. In [5], the authors proposed A two-stage flow-based intrusion detection model. In the beginning stages, hostile flows are distinguished from legitimate network traffic using an improved unsupervised one-class support vector machine. A self-organizing map is used in the second stage to classify harmful flows into various alarm clusters automatically. The technique is evaluated using Sperotto's dataset, showing that the proposed approach obtained promising results.
The study provides a summary of the performance of flow-based and packet-based intrusion detection in high-speed networks [6]. The researchers discovered through their review of the relevant literature that packet-based NIDSs process each packet (payload) received. Although there are few false alarms, it takes a lot of time, making it difficult or even impossible. Conversely, flow-based NIDSs require less processing power overall than payload-based ones, making them the obvious choice for high-speed networks. It still has trouble with false alarm rates that are too high.
In [2], a brand-new algorithm known as the Energy-based Flow Classifier has been proposed (EFC). This classifier based on anomalies applies a statistical model from labeled benign samples using inverse statistics.

The researchers demonstrated that EFC is more flexible to various data distributions than traditional ML-based classifiers and can accurately conduct binary flow classification.

This method obtained a high accuracy of 78% across three independent datasets (CIDDS-001, CICIDS17, and CICDDoS19), which is promising.

The decision tree algorithm was created in [7] based on the C4.5 decision tree technique. Based on several attributes, experimentation is done with the NSL-KDD dataset. The algorithm in this work is made to cope with feature selection and split value. The split value is selected, so the classifier is unbiased towards the most frequent values, and Utilizing information gathering, the most essential attributes are chosen. With the suggested method in the paper, they acquired good accuracy of 75% with even fewer characteristics chosen using information gain instead of training with all the features.

Decision tree-based machine learning algorithms have been described in [8] to identify and categorize intrusions. Depending on the number of features, the testing is carried out using KDDCUP99 data sets. The datasets are processed in three steps per the approach used. Bayesian three modes are examined for various-sized data sets based on the number of attacks. The results of the experiments used in this strategy show that the framework is strong enough.

According to the research in [9], a flow-based intrusion detection system uses ensemble classification machine learning techniques to analyze network flow data. Using the CIDDS-001 flow-based IDS assessment datasets, the ensemble approaches, adaptive boosting, bootstrap aggregation, random forests, and majority voting were examined.. The performance of the combined probabilistic, non-probabilistic, and decision tree classification algorithms is assessed. The experiment's findings show that, with 99% accuracy, the ensemble of decision tree-based classification approaches surpasses the combination of approaches for classifying data that are based on probability and other factors.

In [10], They provide a technique for detecting intrusions as well as a hybrid classification-based approach based on the Decision Tree and K-Nearest Neighbor.

The above experiment uses cross-10-fold validation approaches to test the proposed hybrid classifier using the KDD Cup dataset in conjunction with the decision tree and KNN classifiers.

According to a KDD Dataset experiment, the suggested hybrid classifier achieved 100% accuracy with a 0% false positive rate.

In [11], the research shows a notable enhanced intrusion detection using flow-based network traffic analysis to identify DoS and DDoS attacks. This approach utilizes adjustable threshold settings in the detecting unit based on anomaly detection. Systems can run more efficiently by aggregating packets that are part of the same flow. The results demonstrate the improved performance using DARPA 1999 data collection.

Although much work has been done in flow-based intrusion detection, our technique significantly varies from the previous work. Many works showed outstanding performance on different machine learning algorithms. Among them, the decision tree is one of the standard approaches. A training model is created using a decision tree that may be used to predict the value or class of the target variable by learning fundamental choice rules from historical data. So, we have used the decision tree approach for flow-based network intrusion detection for incredible performance. Since we used a real flow-based dataset to test the proposed framework, the experimental results are pretty accurate.

2.2 Scope of the Problem

Various flow-based classifiers have been presented in recent years using Machine Learning (ML) methods. However, traditional ML-based classifiers have much drawbacks. For example, they need a lot of labeled data for train, which may be challenging. Some studies mention they need high background knowledge of some data to identify the threat. Furthermore, the data availability in Network Intrusion Detection Systems (NIDS) is limited. So, accurate detection criteria can only sometimes be defined. This could lead to lower alert confidence and more false alarms. Due to the fact that some processing is delegated to the probe device, flow-based NIDSs require less processing overall, including during the analysis step.

Therefore, resource consumption is typically low. Some researchers tried to improve previous works by implying a statistical model from labeled benign samples using inverse statistics but have yet to get a satisfactory result. Some studies proposed a two-stage flow-based intrusion detection model.

Still, because they use unsupervised learning that does not require labeled training datasets, their accuracy level was different from what they desired.

2.3 Challenges

The following research issues are those that are specifically focused on this study:

1. **Data Collection:** The data availability in Network Intrusion Detection Systems (NIDS) is limited online. Choosing the correct and labeled data is a hassle.
2. **Data Processing:** Only raw data is available online. So the challenge is to process the raw data, filter it, and encode categorical data into numerical data.
3. **Selecting Machine Learning Approach:** Many researchers employ machine learning techniques in NIDS to accomplish tasks efficiently. Therefore, choosing the best machine learning technique can accurately identify malicious activity.
4. **Accuracy Improvement:** The accuracy of the machine learning model needs to be improved, and choosing the best model is a challenging problem.

CHAPTER 3

Materials and Methods

In this chapter, the background study of the flow-based approach has been discussed with the addition of the description of the dataset. After that, the description of the proposed model is explained with the workflow.

3.1 Background of Flow-based approach

A flow-based network is a series of packets sent from one computer to another, which might be another host, a multicast group, or a broadcast domain [5]. The network flow model may be applied consistently to any protocol, employing any combination of address attributes at the neighboring network and transport levels of the networking stack. Network flow characteristics are designed in such a manner that they are applicable to numerous networking protocol stacks, and traffic flow measurement solutions may be used in multi-protocol contexts [6]. Flow-based data sources are frequently used in applications such as network monitoring, traffic analysis, and security. Flow data or network flow characterizes this strategy. Moreover, it does not deliver any packet payload [7]. Two computer systems are connected by a flow, which is a unidirectional data stream that has the following features:

- Source IP address - The IP address of the source through which traffic is forwarded. This attribute is hidden from the related Autonomous System.
- Source port number - The flow's associated source port number identifies the process that sent the data
- Destination IP address - The server's IP address is related to the flow to which traffic is routed. This attribute is also hidden from the related Autonomous System.
- Destination port number - The flow's destination port number, which identifies the process that would receive the data.
- Protocol number - The flow's protocol number, which is a set of rules for structuring and processing data.

Other than these characteristics number of bytes, number of packets, duration, and system time are also shared.

3.2 Data Collection and Preprocessing

The dataset, LUFlow, is a flow-based dataset that also shares the above features to send packets from one computer to another. This dataset contains a strong ground truth by correlating harmful behavior. Through the arrangement of honeypots within Lancaster University's address space, LUFlow contains telemetry containing newly emergent attack vectors. It is possible to capture and label the labeling mechanism's autonomy continuously and strong ground truth provided by correlation with third-party Cyber Threat Intelligence (CTI) sources. Outliers are flows that could not be classified as malicious yet did not fit within the typical telemetry profile. These are presented to inspire additional investigation to determine the real motivation behind their conduct. Typical traffic from production services, such as ssh and database activity, is likewise recorded and included in this dataset. The description of the attributes of the dataset is given below in Table 3.2:

Table 3.2: Dataset description

Data Attribute	Data Description
src_ip	The IP address of the source through which traffic is forwarded. This attribute is hidden from the related Autonomous System.
src_port	The flow's associated source port number identifies the process that sent the data.
dest_ip	The server's IP address is related to the flow to which traffic is routed. This attribute is also hidden from the related Autonomous System.
dest_port	The flow's destination port number, which identifies the process that would receive the data.
protocol	The flow's protocol number, which is a set of rules for structuring and processing data.

bytes_in	The quantity of data received via that interface (from source to destination).
bytes_out	The quantity of data delivered over that interface (from destination to source).
num_pkts_in	The number of packets received from source to destination
num_pkts_out	The number of packets delivered from destination to source
entropy	Flow’s data field’s entropy, is represented in bits per byte.
total_entropy	The overall entropy of the flow, expressed in bytes, over all of its data fields.
mean_ipt	The flow's incoming time of the transmission of services over a packet-switched IP-based network.
time_start	Flow’s starting time in seconds since the beginning.
time_end	Flow’s finishing time in seconds since the beginning.
duration	Microsecond-level accuracy for the flow duration time.
label	The flow can be either benign, outlier, or malicious.

Every flow feature has been considered for testing with different algorithms. The provided dataset has some 'NA' values that have been filtered. The entire LUFlow dataset is utilized for 30% of the data is used for testing and 70% for training the model.

3.3 Decision Tree as Intrusion Detection Model

Decision trees break complex data into manageable bits, which makes them highly useful for machine learning and data analytics. In these fields, data classification, regression, and prediction analysis are commonly used. We emphasized the decision tree for the network intrusion detection system because experimental findings show that employing the decision tree algorithm would produce high detection rates on several types of network attacks and also boost the system's speed and accuracy. Once the variables have been defined, the decision tree model requires a minimal data-cleaning process. Outliers and cases of missing values have less impact on the results in the decision tree. Decision trees need less data preparation work than other decision procedures. Data collection is a precondition for the analysis. Preprocessing is required to convert the acquired data into the format required by decision tree algorithms. Following the processing of the data, decision trees may be trained using the processed data. Running and evaluating the data is a critical next step in comprehending the resultant model and rule sets. The final stage uses the analysis findings to execute the decision rules in real time. Thus it creates a tree model. Users can further combine decision trees with other algorithms to identify intrusion in more complex situations. Decision trees are also easy to understand since they mimic way people seem to think when making decisions. Because a decision tree displays a tree-like form, its logic is similarly straightforward to comprehend.

Label feature was already classified into three categories in this approach: benign, outlier, and malicious. A single node serves as the decision tree's root, from which it branches out in two or more directions. Each branch offers a variety of potential outcomes, fusing a number of scenarios and unforeseen events to get a final outcome. A workflow of is decision has been illustrated in figure 3.3.

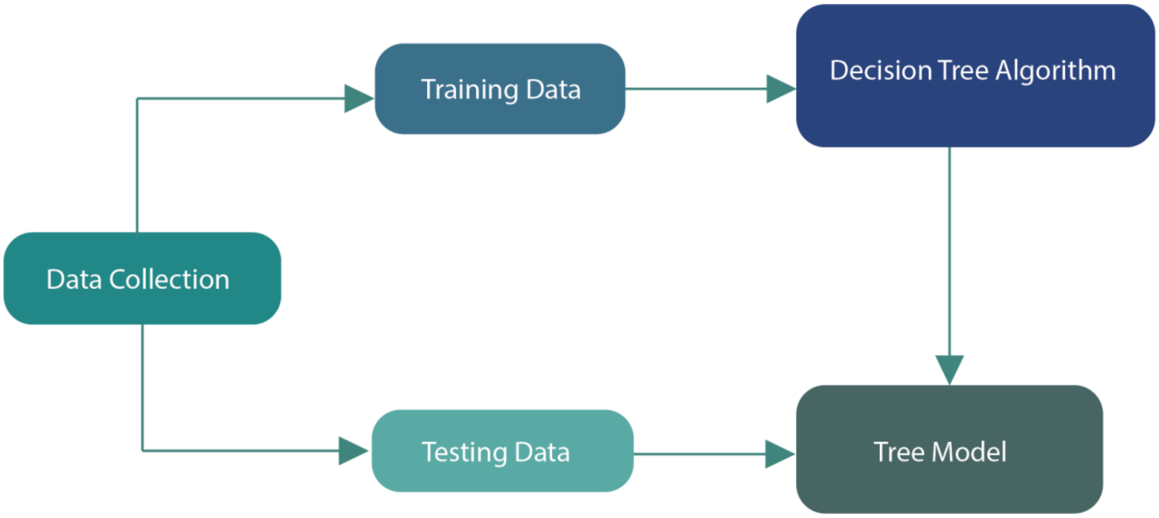

Figure 3.3: Workflow of the Decision Tree

3.4 Machine Learning Algorithms for Comparison

In addition to Decision Tree, five other algorithms have been used in this research. These are Random Forest, Support Vector Machine, Naive Bayes, Artificial Neural Network, and Deep Neural Network. These algorithms have been described below:

3.4.1 Random Forest

The random forest approach, which is used for classification, regression, and other applications, builds a relatively large number of decision trees during the training phase. The classification task's most frequently chosen class is the outcome of the random forest. For regression tasks, a specific tree's average prediction is provided. Random forests fit the situation better because decision trees frequently overfit their trained model. Random forests are often superior to decision trees, although they perform worse than gradient-enhanced trees in terms of accuracy.The effectiveness of them, however, might be impacted by data properties.

3.4.2 Support Vector Machine

One of the most popular supervised learning approaches, support vector machines, is used to handle problems with classification and regression. The SVM algorithm's objective is to create the best decision boundaries or lines for classifying an n-dimensional space. As a result, we will be able to quickly categorize fresh data points in the future. The name of this boundary choice that is ideal is hyperplane. To assist in the creation of hyperplanes, support vector machines select extreme vectors and points. The support vectors used to represent these serious occurrences are the foundation of the SVM methodology.

3.4.3 Naive Bayes

Probability theory is used by a naive Bayes classifier to categorize data. The Bayes theorem is used by naive Bayes classifier systems. The Bayes theorem's most important finding is that the probability of an event can be changed as new information is added. The premise that all characteristics of a data point under consideration are independent of one another is what distinguishes a naive Bayes classifier from other classifiers.

3.4.4 Artificial Neural Network

One or more hidden layers, an output layer, and a node layer are the components of an artificial neural network (ANN). Each node, or artificial neuron, is interconnected with others and comes with a weight and threshold. Any node whose output rises above the specified threshold value is activated and starts sending information to the top layer of the network. In any other case, no data is sent to the following layer of the network.

3.4.5 Deep neural Network

Deep neural networks (DNNs) are ANNs that have a large number of hidden layers between the input and output layers. DNNs and shallow ANNs both have the potential to depict complicated non-linear interactions.

The basic operation of a neural network is to take in a set of inputs, process those inputs using increasingly intricate computations, and then output the findings to deal with practical problems like categorization. We are restricted to feed-forward neural networks.

CHAPTER 4

EXPERIMENTAL RESULTS AND DISCUSSION

The performance of every algorithm has been discussed in this chapter with illustrated images. At last, the accuracy evaluation table has been shown.

4.1 Result and Discussion

All of the ML techniques under evaluation can be used to detect network intrusion detection, according to the analytical outcomes. In this work, Random Forest,Naïve Bayes, Support Vector Machine, Deep Neural Network, and Artificial Neural Network have been used to compare with the Decision Tree. In figure 4.1.1, the accuracy of each model is shown. As can be seen, Decision Tree has outperformed all the other algorithms with an accuracy of 91%. Naïve Bayes showed the lowest accuracy among the other algorithms.

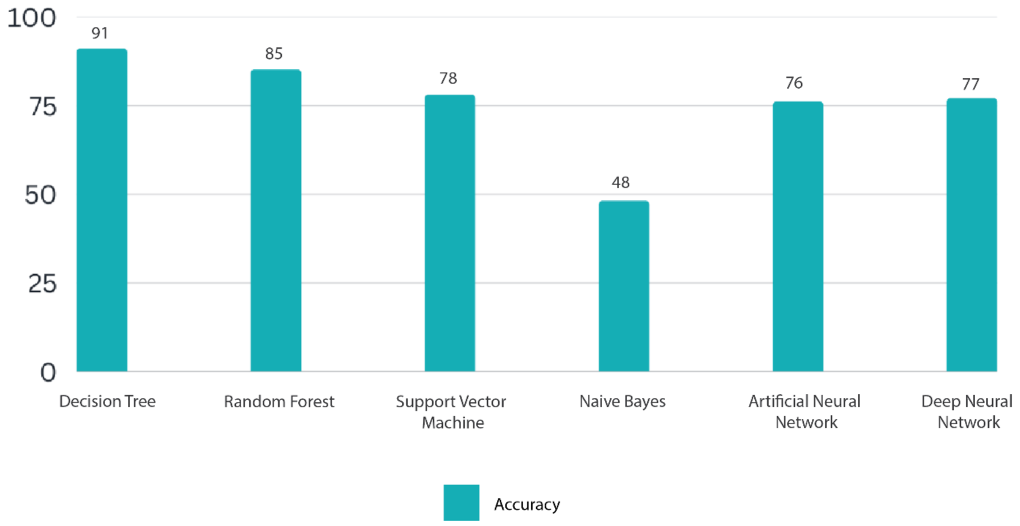

Figure 4.1.1: Accuracy of the models.

According to figure 4.1.2, the F1-score of the models on the training and testing dataset has been shown. In this work, the Decision Tree again showed an outstanding result, while Naïve Bayes Shoed the lowest among the other algorithms. But as shown in Figure 4.1.3, Naïve Bayes has the lowest time consumption for training the LUFlow dataset, while Deep Neural Network took the longest time. Though Decision Tree consumed only 0.16 sec, it could not outperform Naïve Bayes.

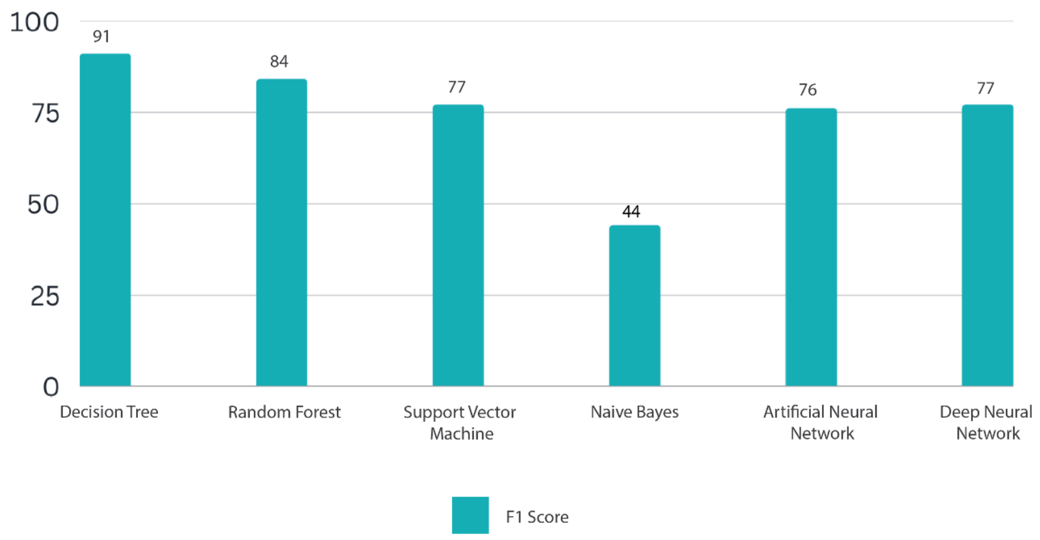

Figure 4.1.2: F1-score of the models on training and testing dataset

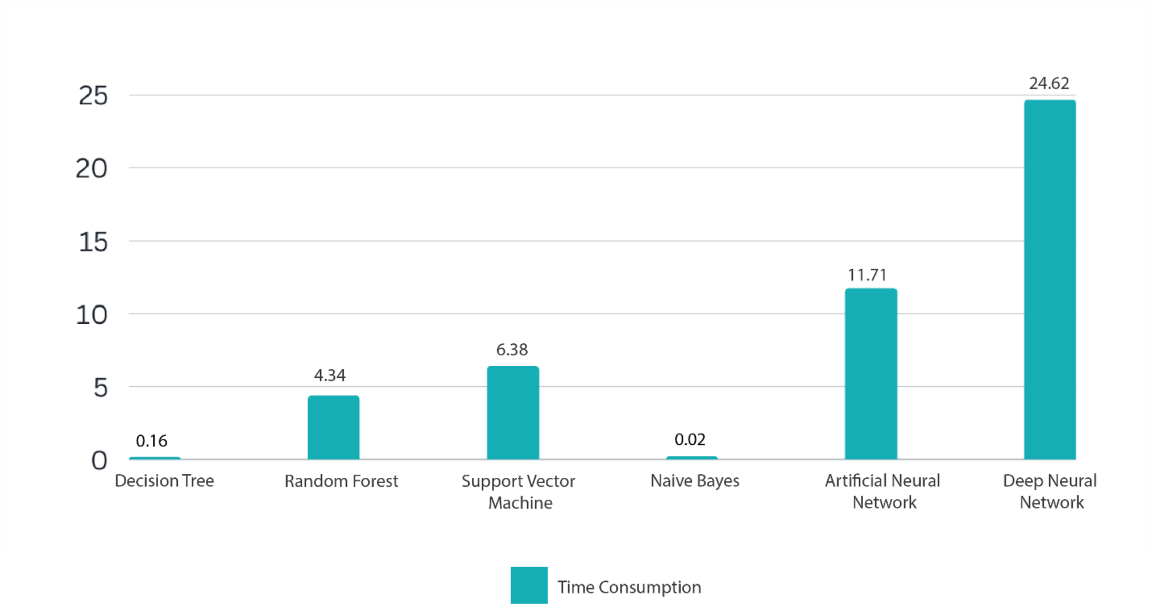

Figure 4.1.3: Time consumption for training on LUFlow dataset

The confusion matrix in figure 4.1.4, each model showed the difference between the true label and the predicted label. The Decision Tree almost showed a perfect prediction while Naïve Bayes could not predict correctly. Other algorithms also showed a good result alongside the Decision tree.

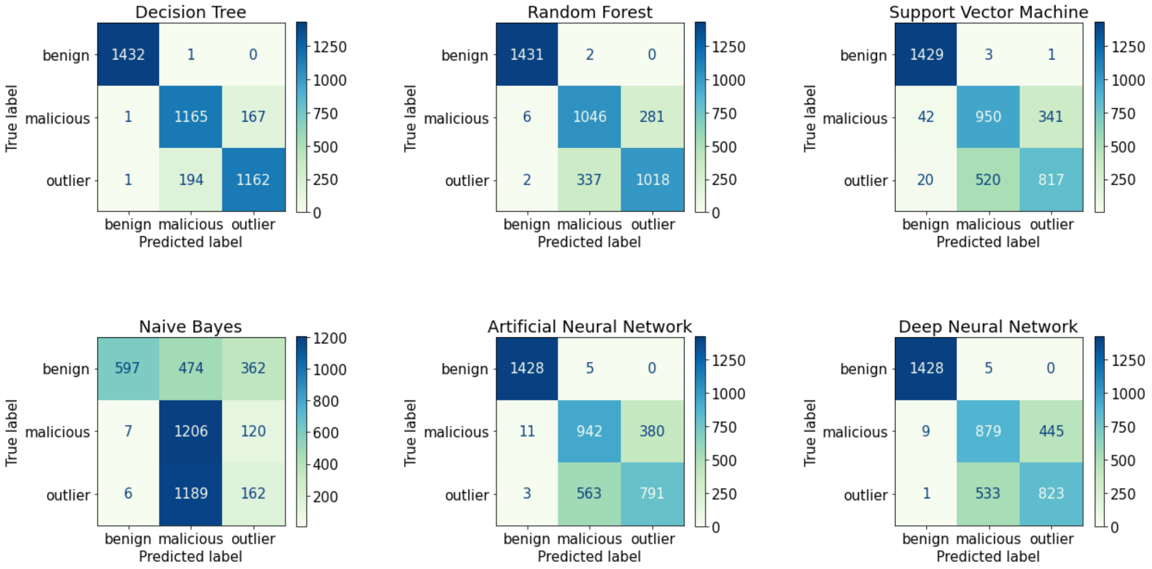

Figure 4.1.4: Confusion matrix of each model on LUFlow dataset

From table 2 to table 7, the performance of each algorithm has been shown. The harmonic mean of accuracy and recall is given by the f1-score. The scores for each class indicate the classifier's accuracy in categorizing data points in that class when compared to all other classes. The number of samples of the true response that fall into that class is the support.

TABLE 4.1.1: PERFORMANCE OF DECISION TREE

	Precision	**Recall**	**F1-Score**	**Support**
Benign	0.9986	0.9993	0.9990	1433
Malicious	0.8566	0.8740	0.8652	1333
Outlier	0.8743	0.8563	0.8652	1357
Accuracy			**0.9117**	4123
Macro Avg	0.9099	0.9099	0.9098	4123
Weighted Avg	0.9118	0.9117	0.9117	4123

TABLE 4.1.2: PERFORMANCE OF RANDOM FOREST

	Precision	**Recall**	**F1-Score**	**Support**
Benign	0.9944	0.9986	0.9965	1433
Malicious	0.7552	0.7847	0.7697	1333
Outlier	0.7837	0.7502	0.7666	1357
Accuracy			**0.8477**	4123
Macro Avg	0.8445	0.8445	0.8443	4123
Weighted Avg	0.8477	0.8477	0.8475	4123

TABLE 4.1.3: PERFORMANCE OF SUPPORT VECTOR MACHINE

	Precision	**Recall**	**F1-Score**	**Support**
Benign	0.9584	0.9972	0.9774	1433
Malicious	0.6449	0.7127	0.6771	1333
Outlier	0.7049	0.6021	0.6494	1357
Accuracy			**0.7752**	4123
Macro Avg	0.7694	0.7707	0.7680	4123
Weighted Avg	0.7736	0.7752	0.7724	4123

TABLE 4.1.4: PERFORMANCE OF NAÏVE BAYES

	Precision	**Recall**	**F1-Score**	**Support**
Benign	0.9787	0.4166	0.5844	1433
Malicious	0.4204	0.9047	0.5740	1333
Outlier	0.2516	0.1194	0.1619	1357
Accuracy			**0.4766**	4123
Macro Avg	0.5502	0.4802	0.4401	4123
Weighted Avg	0.5589	0.4766	0.4420	4123

TABLE 4.1.5: PERFORMANCE OF ARTIFICIAL NEURAL NETWORK

	Precision	**Recall**	**F1-Score**	**Support**
Benign	0.9903	0.9965	0.9934	1433
Malicious	0.6238	0.7067	0.6627	1333
Outlier	0.6755	0.5829	0.6258	1357
Accuracy			**0.7667**	4123
Macro Avg	0.7632	0.7620	0.7606	4123
Weighted Avg	0.7682	0.7667	0.7655	4123

TABLE 4.1.6: PERFORMANCE OF DEEP NEURAL NETWORK

	Precision	**Recall**	**F1-Score**	**Support**
Benign	0.9930	0.9965	0.9948	1433
Malicious	0.6203	0.6594	0.6393	1333
Outlier	0.6491	0.6065	0.6270	1357
Accuracy			**0.7592**	4123
Macro Avg	0.7541	0.7541	0.7537	4123
Weighted Avg	0.7593	0.7592	0.7588	4123

TABLE 4.1.7: ACCURACY EVALUATION OF DIFFERENT MODELS

Algorithms	Decision Tree	Random Forest	Support Vector Machine	Naive Bayes	Artificial Neural Network	Deep Neural Network
Accuracy	91	85	78	48	76	77

The accuracy evaluation of different models have been demonstrated in table 8. As discussed earlier, Decision Tree showed the highest accuracy in training and testing the LUFlow dataset. Random Forest also showed impressive accuracy, but its training time is higher than the Decision Tree and Naïve Bayes.

Deep Neural Network also showed a good performance with an accuracy of 77%, but its time consumption for training the dataset is 24.62 sec. So, considering the performance of all the algorithms, Decision Tree is the best algorithm to identify the labeled data as benign, malicious, or outlier.

4.2 Comparative Analysis

TABLE 4.2: COMPARATIVE ANALYSIS

Author	Title	Publishing Year	No of Features	Algorithm / Method	Accuracy
Our work	Flow-based Network Intrusion Detection System using Decision Tree Over Big Data	Not yet	16 features	Decision Tree	91%
D. Souza et a.l [2]	A new method for flow-based network intrusion detection using the inverse Potts model	2021	88 features	Energy-based Flow Classifier	78%
U. M. Fahad et al. [5]	A two-stage flow-based intrusion detection model for next-generation networks	2018	9 features	One-class support vector machine	NM
A. H. M. Mahmuddin et al. [6]	An Overview of Flow-Based and Packet-Based Intrusion Detection Performance	2011	9 features	Support Vector Machine, Decision	77% and 87%

	in High Speed Networks			Tree	
A. Guleria et al. [7]	Decision Tree Based Algorithm for Intrusion Detection	2016	41 features	C4.5 decision tree	75%
Z. S. P. Tarwireyi et al. [9]	Ensemble Learning Approach for Flow-based Intrusion Detection System	2019.	14 features	Ensemble Classifier	99%

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 Conclusion

This paper presents a flow-based network intrusion detection system that uses a decision tree. The decision tree allows for the use of fewer characteristics while yet providing adequate accuracy in a reasonable amount of time. This model showed an outstanding accuracy of 91% in identifying if the attack was benign, malicious, or an outlier. In this proposed approach, we have also used Random Forest (RF), Naive Bayes (NB), Support Vector Machine (SVM), Deep Neural Network (DNN), and Artificial Neural Network (ANN) to compare with the Decision tree. Decision Tree outperformed all other algorithms. Among these algorithms, Nayes Bayes showed the lowest accuracy result, although it has the lowest time consumption.

Considering the advantages presented, we believe Decision Tree to be a good algorithm for performing the flow-based Network Intrusion Detection System (NIDS).

As we have been able to accurately classify benign, malicious, or outlier attacks in the NIDS with a decision tree, we will try to improve the accuracy and lower the time consumption in the future.

5.2 Future Work

A more comprehensive investigation of flow-based Network Intrusion Detection Systems (NIDS) will be performed on different machine learning algorithms. We are already working on an Energy-based Flow Classifier (EFC), which is a new flow-based classifier for network intrusion detection. Finally, we will run thorough research to be capable of identifying different kinds of attacks in network intrusion detection systems.

5.3 Limitations

Although we have done this research perfectly, our work has some limitations. These limitations are given below:

- This detection should have been done using artificial intelligence
- Data should have been collected from our university
- More machine learning algorithms should have been used for comparison
- Feature selection method should have been used as not all features are essential.

APPENDIX

Abbreviation

NIDS = Network Intrusion Detection System

ML = Machine Learning

IDS = Intrusion Detection System

EFC = Energy-based Flow Classifier

DT = Decision Tree

CTI = Cyber Threat Intelligence

SVM = Support vector Machine

NB = Naive Bayes

ANN = Artificial Neural Network

DNN = Deep Neural Network

REFERENCES

[1] S. Williams, "SecurityBrief," 14 Sep 2022. [Online]. Available: https://securitybrief.com.au/story/hands-on-intrusion-attempts-up-50-year-over-year-report.

[2] P. C. FT, d. Souza, M. M., G. J. J., B. M. and M. , "A new method for flow-based network intrusion detection using the inverse Potts model," *IEEE Transactions on Network and Service Management*, no. 1, pp. 1125-1136, 2021.

[3] "Wikipedia, the free encyclopedia," 23 October 2022. [Online]. Available: https://en.wikipedia.org/wiki/Decision_tree.

[4] R. Mills, "kaggle," [Online]. Available: https://www.kaggle.com/datasets/mryanm/luflow-network-intrusion-detection-data-set.

[5] U. M. Fahad, M. Sher and Y. Bi, "A two-stage flow-based intrusion detection model for next-generation networks," *PloS one,* no. 1, p. p.e0180945, 2018.

[6] A. H. M. Mahmuddin and A. A. Mazari, "An overview of flow-based and packet-based intrusion detection performance in high speed networks," *Proceedings of the International Arab Conference on Information Technology,* no. 1, pp. pp. 1-9, 2011.

[7] R. K. M. S. Devi and A. Guleria, "Decision Tree Based Algorithm for Intrusion," *International Journal of Advanced Networking and Applications,* no. 1, p. p.2828, 2016.

[8] S. S. L. S. C., G. B., N. and K. G. Suni, "Decision Tree: A Machine Learning," *International Journal of Innovative Technology and Exploring Engineering (IJITEE),* no. 1, pp. pp.1126-1130, 2019.

[9] Z. S. P. Tarwireyi and M. Adigun, "Ensemble Learning Approach for Flow-based," *2019 IEEE AFRICON,* no. 1, pp. pp. 1-8, 2019.

[10] A. B. W. Jian and M. Shafiq, "Intrusion Detection by Using Hybrid of Decision Tree," *International Journal of Hybrid Information Technology,* no. 1, pp. 201-208, 2016.

[11] D. J. and C. Thomas, "Intrusion detection using flow-based analysis of Network Traffic," *International Conference on Computer Science and Information Technology,* no. 1, pp. pp. 391-399, 2011.

[12] "Wikipedia," 3 January 2023. [Online]. Available: https://en.wikipedia.org/wiki/Traffic_flow_(computer_networking).

[13] . N. Brownlee, C. Mills and G. Ruth, October 1999. [Online]. Available: https://www.ietf.org/rfc/rfc2722.txt.

Flow based intrusion detection system

ORIGINALITY REPORT

25%	17%	14%	14%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Daffodil International University Student Paper	2%
2	www.webology.org Internet Source	1%
3	Submitted to University of Strathclyde Student Paper	1%
4	bdm.unb.br Internet Source	1%
5	eprint.iacr.org Internet Source	1%
6	Submitted to Visvesvaraya Technological University, Belagavi Student Paper	1%
7	Submitted to Nepal College of Information Technology Student Paper	1%
8	Submitted to Liverpool John Moores University Student Paper	1%