INTRUSION DETECTION USING A NEURAL NETWORK

A MAJOR PROJECT REPORT

Submitted by

Anshumaan Mishra [Reg No: RA1811028010076]

Under the Guidance of

Dr. VIGNESHWARAN P

(Associate Professor, Department of Networking and Communications)

in partial fulfillment of the Requirements for the Degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE ENGINEERING WITH SPECIALIZATION IN CLOUD COMPUTING

DEPARTMENT OF NETWORKING AND COMMUNICATIONS

COLLEGE OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR- 603 203

MAY 2022

BONAFIDE CERTIFICATE

Certified that this project report titled "**Intrusion Detection using a Neural Network"** is the bonafide work of **Anshumaan Mishra [Reg No: RA1811028010076]** who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion for this or any other candidate.

Dr VIGNESHWARAN P	**Dr ANNAPURNAI PANAIYAPPAN K**
SUPERVISOR	**HEAD OF THE DEPARTMENT**
Associate Professor	Professor & Head
Department of Networking and Communications	Department of Networking and Communications

Signature of Internal Examiner	Signature of External Examiner

Own Work Declaration

Department of Networking and Communications

SRM Institute of Science & Technology

This sheet must be filled in (each box ticked to show that the condition has been met). It must be signed and dated along with your student registration number and included with all assignments you submit – work will not be marked unless this is done.

To be completed by the student for all assessments

Degree/ Course : Bachelors of Technology/Computer Science and Engineering with specialization in Cloud Computing

Student Name : Anshumaan Mishra

Registration Number: RA1811028010076

Title of Work : Intrusion Detection using a Neural Network

I / We hereby certify that this assessment compiles with the University's Rules and Regulations relating to Academic misconduct and plagiarism**, as listed in the University Website, Regulations, and the Education Committee guidelines.

I / We confirm that all the work contained in this assessment is my / our own except where indicated, and that I / We have met the following conditions:

- Clearly references / listed all sources as appropriate
- Referenced and put in inverted commas all quoted text (from books, web, etc)
- Given the sources of all pictures, data etc. that are not my own
- Not made any use of the report(s) or essay(s) of any other student(s) either past or present
- Acknowledged in appropriate places any help that I have received from others (e.g. fellow students, technicians, statisticians, external sources)
- Compiled with any other plagiarism criteria specified in the Course handbook / University website

I understand that any false claim for this work will be penalised in accordance with the University policies and regulations.

DECLARATION:
I am aware of and understand the University's policy on Academic misconduct and plagiarism and I certify that this assessment is my / our own work, except where indicated by referring, and that I have followed the good academic practices noted above.
If you are working in a group, please write your registration numbers and sign with the date for every student in your group.

ACKNOWLEDGEMENT

We express our humble gratitude to **Dr C. Muthamizhchelvan**, Vice-Chancellor, SRM Institute of Science and Technology, for the facilities extended for the project work and his continued support.

We extend our sincere thanks to Dean-CET, SRM Institute of Science and Technology, **Dr T.V.Gopal**, for his invaluable support.

We wish to thank **Dr Revathi Venkataraman**, Professor & Chairperson, School of Computing, SRM Institute of Science and Technology, for her support throughout the project work.

We are incredibly grateful to our Head of the Department**, Dr K. Annapurani Panaiyappan K.** Professor, Department of Networking and Communications, SRM Institute of Science and Technology, for her suggestions and encouragement at all the stages of the project work.

We want to convey our thanks to our program coordinator **Dr M.B. Mukesh Krishnan, Associate Professor and** Panel Head, **Dr L Anand** Assistant Professor, and Department of Networking and Communications, SRM Institute of Science and Technology, for their inputs during the project reviews and support.

We register our immeasurable thanks to our Faculty Advisor**, Dr Vaishnavi Moorthy,** Assistant Professor, Department of Networking and Communications, SRM Institute of Science and Technology, for leading and helping us to complete our course.

Our inexpressible respect and thanks to my guide, **Dr Vigneshwaran P,** Associate Professor, Department of Networking and Communication, SRM Institute of Science and Technology, for providing me with an opportunity to pursue my project under his/her/their mentorship. He provided me with the freedom and support to explore the research topics of my interest. His passion for solving problems and making a difference in the world has always been inspiring.

We sincerely thank the Networking and Communications Department staff and students, SRM Institute of Science and Technology, for their help during our project. Finally, we would like to thank parents, family members, and friends for their unconditional love, constant support, and encouragement.

Anshumaan Mishra

ABSTRACT

Communication between devices in a network requires transporting data from one device to another. For this reason, data packets are present in the network carry some data from sender to receiver and vice versa. For example, the three-way TCP handshake process needs to complete in order to establish a link with the sender and receiver ports of two devices. But an attacker can leverage processes like TCP handshake to extract information from the target device. They can find out about the operating system, which service is being run on the port of the target device, number of open ports etc. To perform reconnaissance an attacker sends data packets using unique protocols, this generates malicious traffic in the network. Attackers may try their best to hide the traffic generated by them by making their traffic look benign. To find the malicious traffic inside a network a network intrusion detection system is put in place to detect abnormal traffic. In this work we have coded a neural network to identify the malicious traffic present in a network dataset. The dataset we used have multiple instances of abnormal traffic, our Neural Network seeks to learn and predict network samples. Neural networks are shown to have higher accuracy compared to most of the traditional machine learning models due to its architecture and complex computations. The deep learning algorithm we have used is a feed forward neural network with two hidden layers and an output layer. To analyse model performance, we use performance metrics like precision, recall and f-measure which are calculated using the values obtained in the confusion matrix. We use a Precision-recall curve at various thresholds to see at what threshold the model has the strongest performance.

Keywords: Artificial Neural Network, Deep Neural Network, Intrusion Detection

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

ML	Machine Learning
DOS	Denial of Service
AI	Artificial Intelligence
ANN	Artificial Neural Network
DL	Deep Learning
DNN	Deep Neural Network
AWS	Amazon Web Services
ICMP	Internet Control Message Protocol
MLP	Multi-Layer Perceptron
ReLU	Rectified Linear Unit
VM	Virtual Machine
TCP	Transmission control Protocol
$\mathbf{W_x}$	Weight associated with input X
RF	Random Forest

CHAPTER 1

INTRODUCTION

1.1 DENIAL OF SERVICE

Denial of Service attacks occur due excessive pressure on a device to process requests simultaneously which leads to the inability of the device to function appropriately. A DOS attack can cause a system to malfunction for a period of time as long as the attacker wants if no protection mechanism is put in place DOS attacks are preformed not for many purposes such as harming a service so as to damage the service provider financially and reputationally. For example, Amazon Web Services, the cloud giant was hit by a gigantic DOS attack which targeted a specific unidentified user using a technique called Connectionless Lightweight Directory Access Protocol (CLDAP) reflection. This technique exploits the vulnerable third-party CLDAP servers and magnifies the quantity of data sent to the victim’s IP address. The attack continued for three days and at its zenith sent 2.3 terabytes per second. There also a variety of DOS attacks that occur using different protocols such as UPD, ICMP etc. Many mechanisms are available to counter these kinds of attacks which is discussed in the existing methodologies.

1.2 EXISTING METHODOLOGIES

The main provocation behind this research was to come up with an efficient method for the detection of abnormal traffic. Existing DOS detection mechanism include the following.

1.2.1 ANOMALY DETECTION TECHNIQUE

This method finds the normal activity profile of the system. This profile is used to compare all activities performed on system with this profile and detect strange behaviour of the activity. If found, an alarm is raised for the event, indicating an intrusion event.

1.2.2 SIGNATURE DETECTION TECHNIQUE

This method saves the attack pattern / signature. Every event that occurs on the system has its own pattern and is compared to the stored data. An alarm is triggered as soon as a match is found. Unable to detect unknown event (signature unknown)

1.2.3 PROTOCOL SPOOFING TECHNIQUE

This This Stateful log analysis is the process of identifying deviations by comparing a given profile of the definition of benign log activity for each log state to the recorded events. Stateful log analysis works better than anomalous and signature-based detection methods. Stateful Protocol Analysis affects the network layer, application layer, and transport layer, used by the provider's predefined specification settings to identify appropriate protocol and application deviations. In contrast to anomalous-based detection, stateful protocol analysis is based on a universal profile developed by the manufacturer that specifies how to use or not to use a particular protocol. "Stateful" protocol analysis conveys that the IDS can identify and track the condition of network, transport, and application protocols.

1.3 MOTIVATION

Evolving DOS attack can bring a lot of harm to the existing website services, network services. Denying access to a user or completely overloading a server the financial cost of causing a DOS attack is massive. Over the last 20 years, with rapid changes in DOS methodologies, there has been a growth of attacks like botnet. One can see DDOS attacks happening in recent times which led to extortion of hundreds of thosands of dollars from victims. The surprise attack on Estonian Government websites in 2020 was an example of the above. A need to stop these using latest methods is required. In today’s world, Deep learning Algorithms have shown a lot of applications in the field of cybersecurity. Hence our work involves the use of artificial neural network to solve this challenge.

1.4 OBJECTIVE

The main objectives of this research work are had been listed out below:

- Collection of an appropriate dataset
- Pre-processing the dataset, to remove all odd occurrences such as missing values, dataset imbalance, duplicate data removal, and much more.
- Selecting the important features on which the DL model can be trained to give better results.
- Splitting the dataset into the training and the testing class to implement Deep Learning Algorithms on it.
- Computing the Evaluation metrics on the Deep Neural Network model.

CHAPTER 2

LITERATURE REVIEW

2.1 PREVIOUS WORK

Algorithms like MLP have been used previously to determine malicious traffic [1]. This method proposes a DoS detection method based by leveraging Multilayer Perceptron. The author uses several metrics to successfully identify flooding attacks and classify them as inbound or outbound attacks. MLPs are trained with metrics taken from various types of passive measurements in the network. This can improve DoS detection performance. One of the latest datasets, DoS attack dataset CIC IDS 2017 dataset is used in the experiment [2]. During research with the dataset is segregated into different splits and the best split is found for each algorithm that is RF and MLP. Performance of RF and MLP are looked at and it is shown that RF gives improved results than MLP

2.2 WORK RELATED TO SIGNATURE BASED DETECTION

Just as a signature made by a person is unique for identify network intrusion attacks a signature-based IDS uses a set of rules which help it decide if a pattern related to intrusion is recognised or not. Their main advantage is the high precision and low number of false positives. It can detection network attacks in the cloud. The poor configuration of a system can led to some of the known attacks go unnoticed. Like traditional networks, it cannot be used to detect unknown attacks in the cloud. Unknown attacks being zero-day attacks that have been developed from a newly discovered vulnerability in the victim network. Since finding a fox for the novel vulnerability discovered takes time, potential victim will be defenceless to the zero day attack signature detection is not recommended in every use case. The approach shown in [2-5] uses a signature-based intrusion detection system to detect intrusions into the VM.

2.3 WORK RELATED TO ANOMALY BASED DETECTION

Anomaly-based approaches collect data on legitimate user behaviour for a while and using statistical examinations on those user choices it decided whether the traffic is benign. It has the advantage of being able to detect previously undiscovered attacks. One work has used provided anomaly-based solution to prevent intrusion in real time system, which analyses protocol-based attack and multidimensional traffic [6]. [7] announced a lightweight intrusion detection system for real-time, efficient and effective detection of intrusions. In this paper, for the detection of coordinated attacks, data mining is used.

2.4 ASSOCIATE RULE BASED DETECTION

Some attacks are created by using existing attacks or variants of known attacks. The signature pre-algorithm [23] can be used to detect such signatures or attacks. It detects a frequent subset of a particular set of attacks, including some characteristics of the original attack. H. Han et al. [8] propose the use of data mining for network-based intrusion detection technology. This method involves, an algorithm that learns from the signature(s) of existing DOS attacks and generates a signature to detect misuse. However, the limitation of the proposed algorithm is that it takes a long time to search the database and generate the signature. The author of [9] solved the database search time issue investigated in [8].

2.5 OTHER METHODS USED IN RESEARCH

H. Zhengbing, et al [11] has reported a lightweight intrusion detection system for constant, productive, and compelling detection of intrusions. In this paper, conduct profiles and data mining procedures are naturally kept up with to recognize facilitated attacks. J. Cannady [12] presents a three-layer neural organization to track down maltreatment in the organization. The capacity vectors utilized in [12] are nine organization capacities, for example, source port, destination port, and so forth Notwithstanding, the exhibition as far as the accuracy of interruption discovery is extremely low. Grediaga.et.al. [13] thought about the pace of persistent interruption discovery with MLP and Self Organising Maps (SOMs) and demonstrated that SOM has a more noteworthy accuracy than ANN. Yet, the accuracy of this technique can be better by trimming it with the other delicate processing advancements. Selective utilization of ANN-based IDS may not be a productive assault identification arrangement in the cloud as it requires a quick assault discovery system. One methodology proposed in [14] utilizes ANN-based inconsistency identification strategies for cloud conditions. This requires additional preparation models and more opportunities to successfully recognize intruders. The fuzzy connection rules displayed in [15] are utilized to perceive interruptions into the organization progressively. Two rulesets are created and arranged online from the preparation information. The examination work is obtained from the packet header. This methodology is utilized for enormous scope assaults like DoS/DDoS. H. Gong et al. [1] propose network-based interruption recognition utilizing genetic algorithms. In this methodology, a signature-based algorithm creates a signature to identify abuse. Notwithstanding, the downside of the proposed algorithm is that it consumes a large chunk of the day to look through the data set and create the signature. The creator of [17] utilized seven characteristics of the recorded packet in class esteems and numbers. They utilized a trust-based system for straightforward and adaptable wellness capacities. The produced rules are utilized to recognize interruptions in the organization. This white paper utilizes the quantitative characteristics of the organization to create classification rules. This further develops acknowledgment and accuracy. Nonetheless, the restrictions of this methodology are the best issue.

Lu et al. the authors of [18] proposed strategies to recognize misuse and oddities through a blend of genetic and fuzzy algorithms. Fuzzing is used to remember quantitative boundaries for intrusion discovery, and genetic algorithms are utilized to track down the ideal boundaries for the presented mathematical fuzzy capacity. To work on the presentation of IDS, the authors [19] introduced a methodology utilizing a mix of ANN, Decision Tree and Naive Bayes classifier classifier for three separate datasets. Independent results for every classifier are made and consolidated utilizing multiple combination techniques. This methodology uses any classifier to work on the general exhibition of the IDS. To further develop the execution of IDS, the authors in [20] introduced a methodology that uses a mix of Decision Tree classifiers, Naïve Bayes, and Artificial Neural networks on three separate arrangements of information input. GA is more proficient in design coordinating, yet it is done with a goal in mind rather than the overall way [21]. Affiliation rule-based IDSs are just productive against related assaults. Nonetheless, the effectiveness of IDSs dependent on affiliation rules relies upon the information base you use.

CHAPTER 3

PRELIMINARY DISSCUSION

3.1 VARIOUS INTRUSION ATTACKS

The objective of network security is to prevent damage to resources present in the network, prevent downtime of services provided by these resources and ensure data integrity and confidentiality. We discuss the types of attacks in further subsections.

3.1.1 DENIAL OF SERVICE

This attack is performed to disrupt the services of a system. For instance, a web application is hit with a DOS attack when too many users try to log in and the absence of a load balancer causes the website to load and process data slower. If it is an e-commerce website users may not be able to see the products, payments will not be completed successfully, loss of data, etc.

3.1.2 PROBE

Before attacking the network, an attacker may do a little investigation about the devices present in the network. Probing attacks are very normal methods of gathering information about the kinds and quantities of machines associated with a network, and a host can be attacked to decide the sorts of software introduced and additionally applications utilized. A Probe attack is viewed as the first phase in a genuine attack to think twice about the attack vectors present for hosts in the network. Albeit no particular damage is brought about by this phase, they are viewed as genuine dangerous abnormalities to organizations since they may acquire valuable data for dispatching a coordinated attack.

3.1.3 USER TO ROOT (U2R)

The purpose of this attack is to gain confidential user information which would help control or exploit significant organization assets. Utilizing a social engineering approach or sniffing credentials, the attacker can get to a typical client record and afterward exploit some weakness to acquire the advantage of being an administrator.

3.1.4 REMOTE TO USER (R2U)

This type of attack is performed in-game superuser access in a device connected to the same network as the attacker. These types of attacks are called R2L. The attackers use the tire and error method to find out the passwords, this can be done using brute-forcing or using automated scripts, etc. One of the sophisticated ways includes an attacker using a network monitoring tool to capture the password before attacking the system idiot

3.1.5 BOTNETS

These kinds of attacks are very traditional but still occur even today. online PCs, particularly those with a high-transmission capacity association, have turned into a helpful objective for attackers. Attackers can deal with these PCs employing immediate exploitation. The most common attacks suggest sending files containing a malicious payload that exploits a vulnerable PC, for instance, an unpatched eternal blue vulnerability in windows 7. For the most part, these attacks are led through automated software which helps them select their targets with ease. The necessity for dispatching direct attacks is that publicly accessible services on the designated PCs contain software vulnerabilities.

3.2 DATASETS USED

In much of the work, datasets such as NSL-KDD, KDD 99, and KDD+ have been used. The oldest dataset KDD 99 has been frequently used for anomaly detection due to the high number of attack samples as compared to attack-free samples. However, this dataset has a few redundant attack samples which may increase bias while training the machine learning models. NSL-KDD solves the problem of redundancy and data imbalance. Algorithms trained with this dataset generally tend to have a lower bias as compared to KDD 99 and perform well while detecting attacks that are underrepresented in the NSL-KDD dataset. KKDTest+ is a more refined version of KDD 99.

3.3 ALGORITHMS PRESENTED

In this survey, a myriad of algorithms had been used to detect anomalies. In the literature presented Most of them are supervised models used for the prediction of malicious attacks. Since the classification of attacks performed by most of the authors in this survey is multi-class algorithms that perform better in multiclass classification have provided higher accuracies. We describe the various techniques used by authors in this survey.

3.3.1 SUPERVISED LEARNING ALGORITHMS

Supervised algorithms map an input to output and learn about the function that helps it in the mapping process. To learn this, function the supervised algorithm needs historic data. In this scenario that historic data is the network datasets (KDD 99, NSL-KDD, KDD Test+, etc.) SVM uses a hyperplane to separate data samples of one class from another. SVMs can perform well even with limited scope training sets. In any case, SVMs are susceptible to noise near the hyperplane. The dimensions of the hyperplane depend upon the number of features, if there are two features there is only a single hyperplane. SVM kernel is used to provide complex data transformations which help in distinguishing data samples with different labels. KNN has also been used in intrusion detection. It uses an imaginary boundary line to classify data. In the event that the vast number of a sample's neighbours fit in the same class, the sample has a high probability of having its place in that class. The parameter k incredibly impacts the functioning of KNN models. The more modest k is, the higher the risk of overfitting is high. Conversely, the bigger k is, the lesser the chance of overfitting. The Naïve Bayes calculation uses conditional probability from Bayes Theorum. Naïve Bayes classifier determines the conditional probability for each sample for different classes. When the attribute independence hypothesis is satisfied, the ideal result is reached by Naïve Bayes. The conditional probability formula is calculated as shown in equation (1).

$$P(Y = ck) = \prod_{i=1}^{n} \quad P(Y = ck) \qquad (3.1)$$

The decision tree algorithm is mainly used for classification. The tree-like structure, which makes it easy to understand and automatic removal of inappropriate and redundant features makes it a top choice for classification problems. The learning system requires feature selection, generation of a tree, and tree pruning. During the training of the model, the tree generates child nodes from root nodes after selecting relevant features. Progressive algorithms, for instance, the Random Forest and the extreme gradient boosting (XGBoost), contain many decision trees stacked or clustered together. Clustering depends on similarity among data, grouping profoundly comparable data into one cluster and assembling less-comparable data into various other clusters. Different from classification, clustering is a kind of unsupervised learning. Past knowledge about labelled data is required for these types of algorithms however external knowledge is required. Therefore, the requirements for the dataset are moderately less. K-means is an example of a clustering algorithm, where the number of clusters is denoted by K and the mean of attributes is denoted by means. K-means clustering algorithm involves distance as a similarity measure criterion. K-means uses a centroid-based algorithm, which has a centroid for every cluster. The main motive of this algorithm is to reduce the sum of the distances of the data samples and their corresponding cluster. Data samples are split into K number of clusters. Less distance between two data points increases the chances of them being in the same class. Each classifier can be distinguished based on its advantages and limitations. One method is to add classifiers together to form a stronger classifier. Hybrid classifiers consist of different stages each having its classifier model. From past research, it is evident that ensemble and hybrid classifiers were better as compared to singular classifiers so more and more research has been done using these classifiers. For better performance choosing the specific classifier is very important.

3.3.2 DEEP LEARNING ALGORITHMS

From 2015 to now, more research on deep learning-based IDSs has been underway. There is no need for feature engineering. The data set provides an ample number of data samples that can be used by the deep learning models to learn the features. As a result, deep learning approaches can be used from start to finish. When dealing with massive datasets deep learning models have advantages as compared to shallow models. A typical deep learning model has three layers an input, hidden, and output layer. The input layer has neurons equal to the features of the labeled data. During the training system, algorithms utilize unknown elements in the input distribution to extricate features, bunch protests, and find valuable data patterns. Optimization strategy hyperparameter selection and network architecture are important factors while determining the performance of the neural network. The activation function is calculated using the values of the hyperparameters it is shown by equation (2).

$$a^i = g(x^i W_X + b_x) \tag{3.2}$$

In the formula above there is a weight W_X that is associated with the input matrix x^i and b_x is the bias.

3.3.3 SEQUENCE MODELS

Sequence models usually are employed to find patterns and learn from those patterns through features they learn. One of the widely used sequence models is Recurrent Neural Network (RNN). Each neuron in the recurrent neural network is called a unit which takes into consideration current input and the information from the previous input while making a decision. The attributes of sequential data are contextual; analysing disconnected data from the sequence makes no sense. To obtain relevant information, each unit in an RNN gets not only the current state yet additionally past states. This trademark makes RNNs often experience the ill effects of exploding or vanishing gradients. Actually, standard RNNs manage only limited-length sequences. To tackle the long-term dependence issue, many RNN variants have been proposed, like long transient memory (LSTM). LSTM's solve the problem of exploding/vanishing gradients. They retain more information which helps in better prediction however they are not utilized as much as RNN for intrusion detection.

3.3.4 HYPERPARAMETERS

These are values used for setting the configuration for the neural network there are many types of parameters that are discussed further.

3.3.4.1 NUMBER OF NEURONS

In every deep learning model, there are neurons present indie input-output and hidden layers according to the data set the arrangement of neurons must be taken into consideration. For instance, the data samples in the data set will be converted into a matrix consisting of binary numbers so the number of input neurons should be equal to the features of the data set if it is a labelled data set. If there is a multiclass classification, then the output neurons will be based on the number of classes.

3.3.4.2 LEARNING RATE

It is the adjustment value for the weights provided which the input so to make the deep neural network converge. If the learning rate is high the model executes faster however the convergence chance is low. It is the converse for a small learning rate.

3.3.4.3 OPTIMIZER

They are a pre-set configuration of methods that would help neural networks in learning with ease and reducing loss function. Although there are many optimizers used, for classification purposes this survey will cover a few of them.

- Gradient descent is a first-order optimization algorithm is used by backpropagation neural, also in classification and regression problems. It reduces the loss function by altering the weights the main benefit of this optimizer is that it’s easy to implement however it requires a large Ben a larger data set is provided for training.

- Stochastic gradient descent overcomes some of the disadvantages of the gradient descent optimizer in stochastic gradient descent the derivative is taken one at a time which requires less memory when loading larger datasets.
- Min-Batch Gradient Descent is a modification of the original gradient descent optimizer and for a vendor standard gradient descent and the stochastic gradient descent optimizers. One of its advantages is that it updates the model parameters after dividing the data set into batches. It also doesn't require a high amount of memory. However, this is not useful if better meters such as the learning rate need to be constant.

CHAPTER 4

SYSTEM ANALYSIS

4.1 CHAPTER OVERVIEW

DOS attacks are common in many institutions and using Deep Learning for intrusion detection is an upcoming method. We have used deep learning algorithms identify malicious behaviour and features which help us in doing so. The problem with the existing methodologies is that they cannot keep up with new methods of DOS attacks that are taking place and have become obsolete.

4.1.1 DEEP NEURAL NETWORKS

Deep learning was motivated by a function of the brain called artificial neural networks (ANNs). A perceptron is a unit of neural network that performs calculations to identify the characteristics of the input layer. Perceptron, also known as an artificial neuron, was developed in the 1960s by researchers Frank Rosenblatt, inspired by Walter Pitts and Warren Sturgis1[10]. Neural

networks outperform the conventional machine learning models. The advantages are:

- Non-linearity
- Variable interactions
- Customizability

The neural network begins with perceptron receiving inputs, then it multiplies them by a set of weights before moving them through an amplification mechanism to generate an output. Both classification and regression problems can be solved with neural networks.

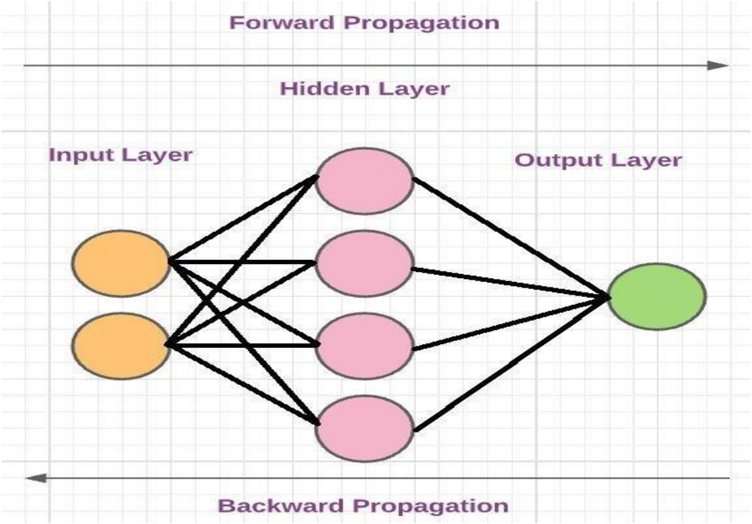


Figure 4.1 - Architecture of ANN with one hidden layer

There are three layers in a neural network:

- **Input Layer**: data is fed as input into the first layer. This layer serves as a connection between the exterior and interior. This layer does not do any computations. It sends the information it has gathered to the next layer.
- **Hidden Layer**: This layer sits between the input and output layers and understands the abstraction aspect of each neural network. Both computations on the features entered as input take place in this layer, which then transfers the effects to the output layer.
- **Output Layer**: This layer computes the values provided by the hidden layer and brings the values between an expected range of values

In neural networks, the activation function is very important in deciding whether to activate or fire a particular neuron by introducing non-linearity and calculating the weighted sum to add a bias. In the "deep learning" model, the rectified linear unit is one of the most frequently used activation functions in the hidden layer. This is the fastest learning feature that provides high success and excellent results. Deep learning is superior to other activation functions such as Sigmoid and SoftMax in terms of accuracy and generalization. Softmax or Sigmoid is used later in the output. This keeps the properties of the linear model virtually as a linear function, simplifying optimization.

4.1.2 ReLU

Rectified Linear Unit activation function provides an output or gradient if the inputs are positive, else it provides 0 which deactivates the neurons present in the region. The term "leaky ReLU" was coined to describe a solution to this problem. Instead of setting the ReLU activation function to 0 when the inputs are negative. It fixes the dying ReLU problem and does not have zero slope parts.

Optimizers are used to minimize losses to change neural network properties such as weights and learning rates. The steepest descent optimizer and the adaptive optimizer are two types of optimizers. Adam is the best performing optimizer. It is more effective to train the neural network effectively and in a short time using Adam. A complex learning speed optimizer is used for sparse data. The MiniBatch Gradient Descent is the best choice of gradient descent optimizer.

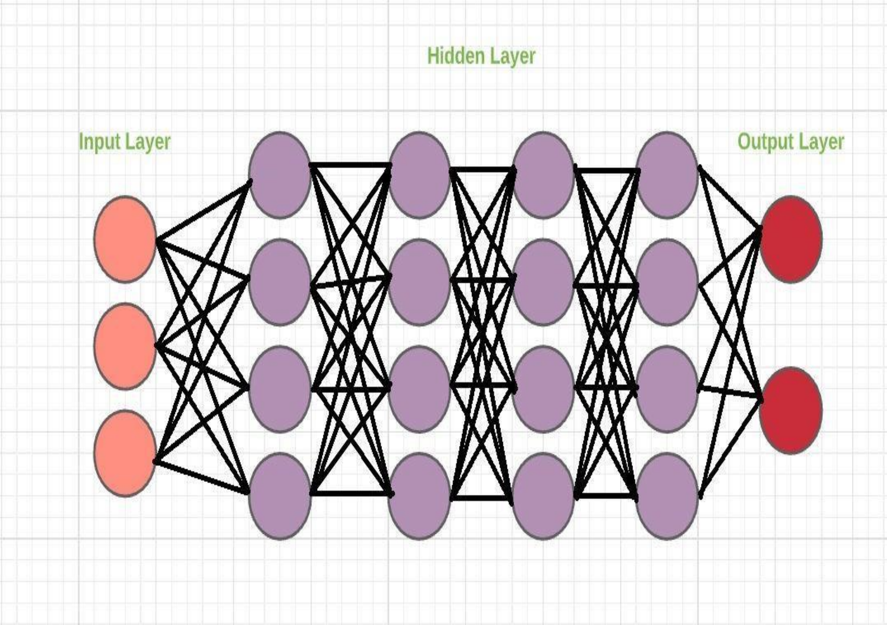


Figure 4.2 - DNN with four hidden layers

4.1.3 DROPOUT REGULARIZATION

It is used in the neural network to avoid overfitting. At each update of the training process, it selects the outgoing edges to 0 at random. It aids in the regularization of the established deep neural network model. In a neural network, dropout is introduced per layer. Dropout can be used on any or more of the network's hidden layers, as well as the input layer.

CHAPTER 5

SYSTEM DESIGN

5.1 SYSTEM ARCHITECTURE

The flow of the overall System Architecture can be understood through the points below:

- The foremost step, while creating this system was to define the problem allowed by searching for an appropriate dataset. After finalizing the Dataset and performing various pre-processing to draw insights.
- We use a standard scaler to bring the values between 0 and 1. This helps the Neural network to learn better.
- We plot the feature importance and then use random forest classifier to choose 5 features.
- The dataset was split, into the training and the testing sets using the model selection's function called the train-test-split from the sklearn library, with a test size of 0.3. This led to the division of the dataset into two parts, with 70% train data in one set and 30% test data of the data in one set.
- The last block shows the statistics we use to measure the performance of our model. It can be influenced by the activation function, optimizer and other sections of the architecture.
- We plot a confusion matrix at different thresholds and calculate the precision recall value for those thresholds
- We add a precision-recall graph to check these values at different thresholds

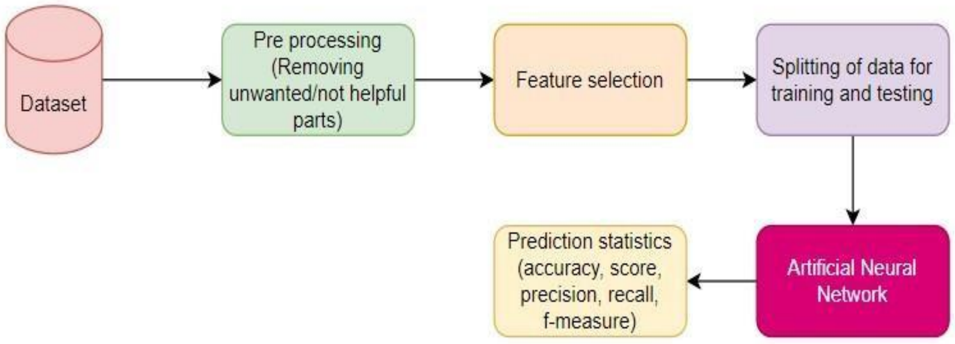

Dataset
Pre processing (Removing unwanted/not helpful parts)
Feature selection
Splitting of data for training and testing
Artificial Neural Network
Prediction statistics (accuracy, score, precision, recall, f-measure)

Figure 5.1 – Modules of the system

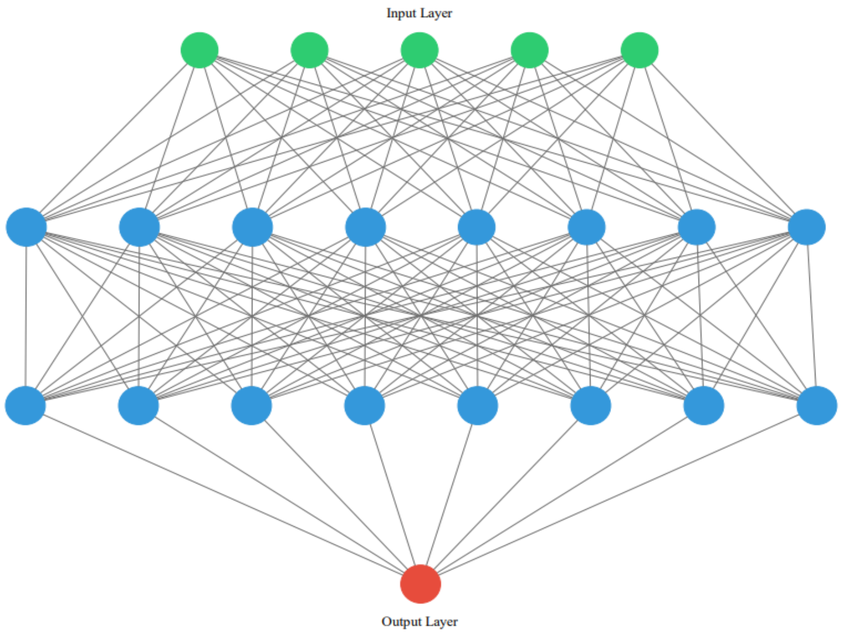


Figure 5.2 - Our Neural Network Architecture

From the figure 5.2 it is clear that the green-coloured neurons are the input neurons which take the input of 5 features, In the two hidden layers we have added 8 neurons each and one output layer (red) as this is a binary classification. For a multiclass classification this output layer will be increased by one if the number of labels is 3.

CHAPTER 6

RESULTS AND ANALYSIS

6.1 DATASET

Our dataset contains numerous values for rows and columns. It has traffic accumulated after multiple attacks were ran on it in a safe environment. It has around 42 columns which denote the features of the dataset all of which are not very useful. We use methods like recursive feature elimination to remove least contributing features.

6.2 FEATURES

To remove a lot of features and use only 15 features we have chosen to eliminate features using recursive features elimination where we have used a random forest algorithm to remove least contributing features.

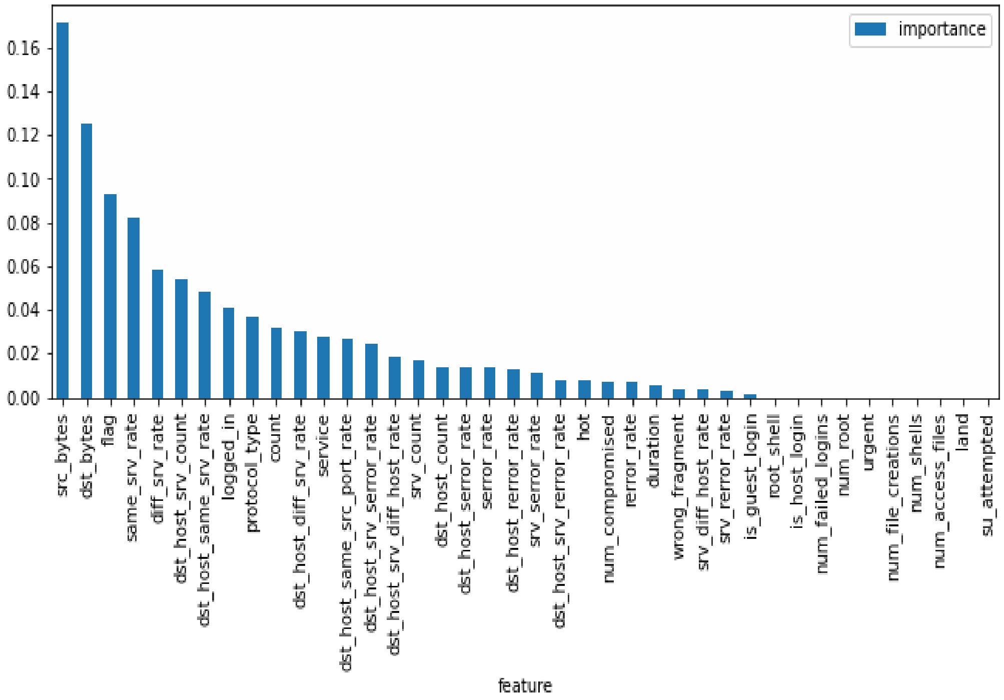
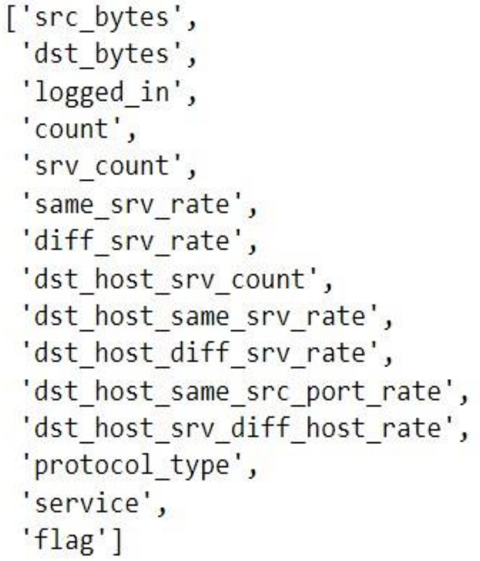


Figure 6.1- shows feature hierarchy in contribution

In the figure above we have found the importance of the features below.



```
['src_bytes',
 'dst_bytes',
 'logged_in',
 'count',
 'srv_count',
 'same_srv_rate',
 'diff_srv_rate',
 'dst_host_srv_count',
 'dst_host_same_srv_rate',
 'dst_host_diff_srv_rate',
 'dst_host_same_src_port_rate',
 'dst_host_srv_diff_host_rate',
 'protocol_type',
 'service',
 'flag']
```

Figure 6.2 - shows all top 15 features

We have chosen the top five features for our work namely:

- Source bytes: Size of the network packet sent from the source
- Destination Packet: Size of the Network packet sent from the destination to source
- Flag: Status of the connection between the machines/operating systems
- Same service rate: percentage of connections to the same service
- Destination hosts same service rate: percentage of connections between same ports running the same service

6.3 EVALUATION METRICS

Scoring metrics are very important in building powerful models. Building a predictive model follows a constructive feedback mechanism. Metrics play an important role in helping to improve by distinguishing between model results. This makes it easy to achieve a model with the desired characteristics. The various metrics used in this task for comparative analysis are described below.

6.3.1 CONFUSION MATRIX

It is an NXN matrix, where N is the number of classes that are being predicted. For our dataset, the value of N is 2, as it is a binary classification problem.

$$ConfusionMatrix = \begin{bmatrix} TP & FN \\ FP & TN \end{bmatrix}$$

Figure 6.3 Sample Confusion Matrix

True positive is the number of cases correctly identified, false positive is the number cases wrongly identified, False negative is the number of cases correctly identified as benign, True negative is the number of cases wrongly identified as malicious.

6.3.2 ACCURACY

It is a highly useful metric and can be implemented to measure the performance of both binary and multi-class classification problems. It shows the percentage of cases with true findings out of the total number of cases investigated.

$$\mathrm{ACCURACY} = \frac{\mathrm{TN+TP}}{\mathrm{TN+TP+FN+FP}} \tag{6.1}$$

6.3.3 PRECISION

This metric is often used for estimation where a high level of certainty is needed in the forecast. It shows what percentage of expected positives are true positives.

$$\mathrm{PRECISION} = \frac{\mathrm{TP}}{\mathrm{TP+FP}} \tag{6.2}$$

6.3.4 RECALL

Recalls are the percentage of properly classified true positive results. Used when you need to capture as many negatives as possible. Sensitivity is another name for it.

$$\mathrm{RECALL} = \frac{\mathrm{TP}}{\mathrm{TP+FN}} \tag{6.3}$$

6.3.5 F-MEASURE

The F1 score is a harmonic means of precision and recall that is always between 0 and 1. This is most often used where both precision and recall are needed. For a classifier, it strikes a balance between accuracy and recall.

$$F_1 = 2 \times \frac{\text{PRECISION} * \text{RECALL}}{\text{PRECISION} + \text{RECALL}} \tag{6.4}$$

Table I contains the True positive, True negative, false positive and false negative for threshold starting from 0.6 to 0.9. Table I is important for determining the relationship false positive and the threshold. As the threshold increase the wrongly identified samples decrease. However. the wrongly identified benign network (false negatives) samples have been increasing as the threshold has increased. With 0.9 being the highest threshold we can see the false positive and true positive values to be 0. This is due to reduction in number of cases identified as malicious to be null at 0.9 threshold.

TABLE I Confusion matrix

Threshold	True Negative	False positive	False negative	True positive
0.6	6814	1431	87	9302
0.7	6890	1355	161	9228
0.8	7192	1053	1339	8050
0.9	8245	0	9389	0

TABLE II results obtained after using the Confusion Matrix

		METRICS		
Probability threshold	Class type	Precision	Recall	F-Measure
0.7	Normal	0.89	0.81	0.85
	Anomaly	0.85	0.91	0.88
0.8	Normal	0.84	0.83	0.84
	Anomaly	0.85	0.86	0.86
0.9	Normal	0.75	0.87	0.81
	Anomaly	0.87	0.74	0.80

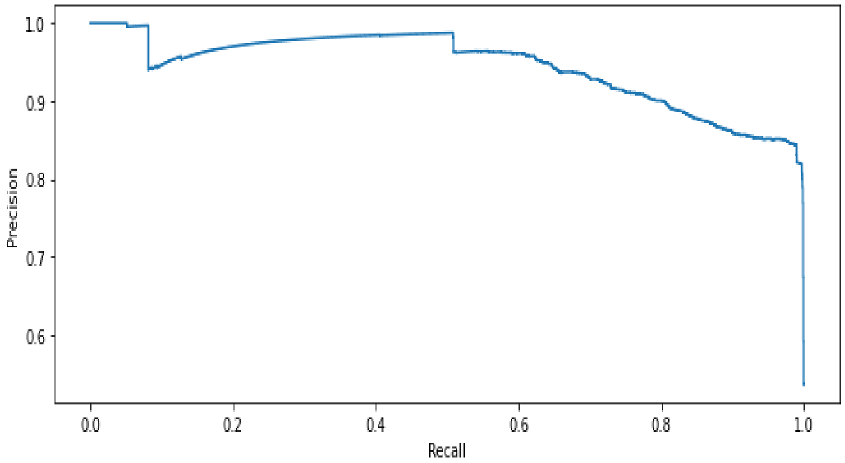


Figure 6.4 Precision-Recall Curve

The curve tells us different values of precision and recall. Apart from providing a relation between the two values for different thresholds, it also tells us about the performance of the neural network, High recall ideally means a low false negative rate, while a system having high precision would have a low false positive rate. In the case of having high recall but low precision in a system, returns a large number of results, but high number of samples are incorrectly predicted. A system providing high precision and a downsized recall returns few results, but a great number of its predicted labels are predicted correctly when analysed with training labels. We compare our precision, recall and f-measure values obtained at 0.7 threshold with the work of other's recent research.

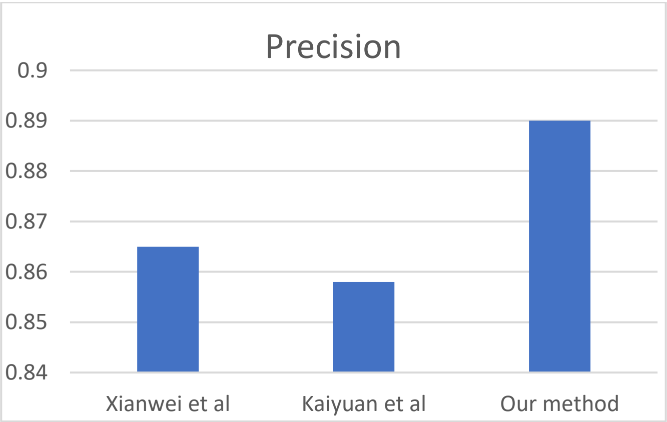


Figure 6.5 Comparing precision

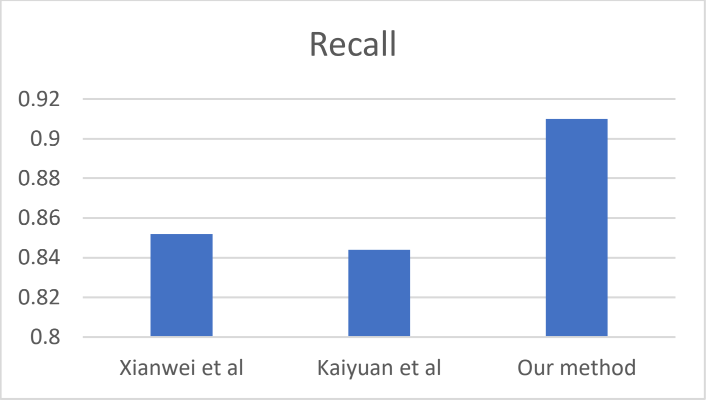


Figure 6.6 Comparing Recall

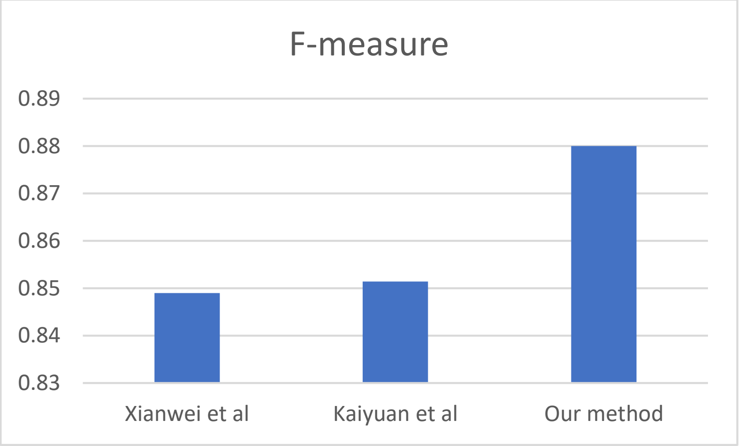


Figure 6.7 Comparing F-measure

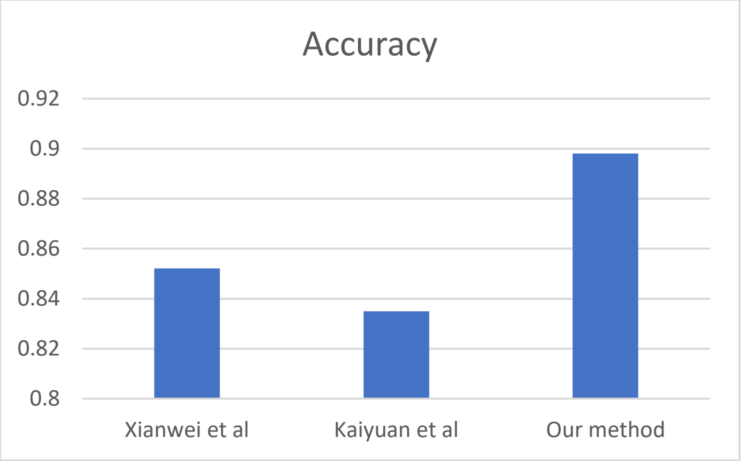


Figure 6.8 Comparing Accuracy

While our accuracy. Our work performs a binary classification and does not provide the name type of network attack occurring, it can only sense malicious intent in the network and thus for future work a classifier which can accurately pinpoint a type of attack is desired. Thus, future work regarding accurate detection of the type of attack is required. Binary classifiers do identify malicious intent but a proper classification of these attacks.

CHAPTER 7

CONCLUSION AND FUTURE WORK

The broad application of neural networks is well known and has shown great progress in the history of Intrusion Detection System. The feedforward neural network properly classifies regular traffic and tests its ability to detect attacks. Neural networks have been found to recognize the known attacks used during neural network training. From the history of research in this field, we can concur that having a signature-based IDS is not useful in today's ever-growing field of cybersecurity. Due to development in zero day attacks a signature-based IDS cannot be implemented in the first place as it would repeatedly be needed to be updated. Neural Networks have been created to prefect existing machine learning algorithms, a strength that has led many researchers to use NNs a lot more than ML algorithms. We also found that neural networks can detect unknown attacks that have never used during the training phase. These results mean that neural networks are an important finding for detecting new attacks. The DNN has provided an accuracy of 89.9% which is very high.

In the future we wish to use more efficient algorithms to find more detailed attacks on the system. Our work was based on binary classification but did not tell us about the properties of specific attacks due to dataset limitations but in the future, we wish to work on a multi class classification problem which has several attack features in it. Such a classification would benefit in the discovery of weak areas in the network through the identification of the attack.

REFERENCES

[1] Wang, Meng, Yiqin Lu, and Jiancheng Qin. "a dynamic mlp-based ddos attack detection method using feature selection and feedback." Computers & Security **vol.** 88 2020. DOI:10.1016/j.cose.2019.101645

[2] D. J. Brown, B. Suckow, and T. Wang.”a survey of intrusion detection systems. department of computer science”, University of California, San Diego. **vol**. 146. DOI: 10.5120/ijca2016910839

[3] S. Roschke, C. Feng, and C. Meinel, “an extensible and virtualization compatible ids management architecture,” Fifth International Conference on Information Assurance and Security, **vol**. 2, 2009, pp. 130-134.DOI: 10.1016/j.protcy.2012. 10.110

[4] A.bakshi, and B. Yogesh, “securing cloud from ddos attacks using intrusion detection system in virtual machine,” Second International Conference on Communication Software and Networks, **vol**. 123 2010, pp. 260- 264 DOI: 10.1109/ICCSN.2010.56

[5] C. C. Lo, C. C. Huang, and J. Ku, “cooperative intrusion detection system framework for cloud computing networks,” First IEEE International Conference on Ubi-Media Computing, **vol**. 23 2008, pp. 280-284 DOI:10.1109/ICPPW. 2010.46

[6] T. Dutkevyach, A. Piskozub, and N. Tymoshyk, “real-time intrusion prevention and anomaly analyze system for corporate networks,” 4th IEEE Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications. **vol**. 136, 2007, pp. 599-602. DOI: 10.1016/j.jnca.2012.05.003

[7] H. Zhengbing, S. Jun, and V. P. Shirochin, “an intelligent lightweight intrusion detection system with forensic technique,” 4th IEEE Workshop on Intelligent Data Acquisition and Advanced ComputingSystems: Technology and Applications, **vol**. 36,2007, pp.647-651 DOI: 10.1016/j.jnca.2012.05.003

[8] H. Han, X. L. Lu, and L. Y. Ren, “using data mining to discover signatures in network- based intrusion detection”, Proceedings of the First International

Conference on Machine Learning and Cybernetics, Beijing, **vol**. 1, 2002 pp. 1-12. DOI: 10.1155/2018/4071851

[9] H. Zhengbing, L. Zhitang, and W. Jumgi, "a novel intrusion detection system (nids) based on signature search of datamining," WKDD First International Workshop on Knowledge discovery and Data Ming, **vol 28** 2008, pp. 10-16.

[10] Manav Mehra, Sameer Saxena, Suresh Sankaranarayanan, Rijo Jackson Tom M.Veeramanikandan, "IOT based hydroponics system using deep neural networks", **vol**. 155 2018. DOI:10.1016/j.compag.2018.10.015.

[11] Zhengbing, H., Jun, S., & Shirochin, V. P. "An intelligent lightweight intrusion detection system with forensics technique". In *2007 4th IEEE Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications* **vol. 11** 2007 pp. 647-651. IEEE. DOI:10.1109/JSYST.2015. 2418434

[12] Cannady, J. "Artificial neural networks for misuse detection". In *Proc. of the 1998 National Information Systems Security Conf. 1* pp. 443-456.

[13] Grediaga, Á., Ibarra, F., García, F., Ledesma, B., & Brotóns, F. (2006, May). "Application of neural networks in network control and information security". In *Int. Symposium on Neural Networks* Springer, Berlin, Heidelberg **vol.** 3973 2006 pp. 208-213. DOI: 10.1007/11760191_31

[14] Vieira, K., & Schuler, A. westphall, C.: "Intrusion detection techniques in grid and cloud computing environment". *Proc. of the IEEE IT Professional Magazine*. 2012 DOI:10.1109/MITP.2009.89

[15] Han, H., Lu, X. L., & Ren, L. Y. "Using data mining to discover signatures in network-based intrusion detection". In *Proc.. Int. Conf. on Machine Learning and Cybernetics* **vol**. 1. 2002, pp. 13-17. IEEE.

[16] Gong, R. H., Zulkernine, M., & Abolmaesumi, P. (2005, May). "A software implementation of a genetic algorithm-based approach to network intrusion detection". In *Sixth Int. Conf. on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing and First ACIS Int. Workshop on Self-Assembling Wireless Network* 2005 pp. 246-253. IEEE. DOI: 10.1109/ SNPD-SAWN.2005.9

[17] Dhanalakshmi, Y., & Babu, I. R. 2008. “Intrusion detection using data mining along fuzzy logic and genetic algorithms”. *Int. Journal of Computer Science and Network Security*, **vol**. *8* 2008, pp. 27-32. DOI: 10.1.1.133.5130

[18] C. Katar, “Combining multiple techniques for intrusion detection,” *Int. Journal of Computer Science & Network Security*, **vol**. 6, 2006, pp. 208–218,

[19] Botha, M., Von Solms, R., Perry, K., Loubser, E., & Yamoyany, G. (2002, September). “The utilization of artificial intelligence in a hybrid intrusion detection system”. In *Proc. of the 2002 annual research Conf. of the South African institute of computer scientists and information technologists on Enablement through technology* **vol**. 10 2002 pp. 149-155. DOI: 10.1.1.120.4282

[20] Li, L., Yang, D. Z., & Shen, F. C. “A novel rule-based Intrusion Detection System using data mining”. In *2010 3rd Int. Conf. on Computer Science and Information Technology* **vol**. 6 2010, pp. 169-172. IEEE. DOI: 10.26438/ijcse/v6i5.203208

[21] Botha, M., Von Solms, R., Perry, K., Loubser, E., & Yamoyany, G. “The utilization of artificial intelligence in a hybrid intrusion detection system”. In *Proc. of the 2002 annual research Conf. of the South African institute of computer scientists and information technologists on Enablement through technology* **vol**. 13 2002 pp. 149-155. DOI: 10.1.1.120.4282

APPENDIX

DATA PREPROCESSING AND EXPLORATION

```
import numpy as np import pandas as pd
 import seaborn as sns import matplotlib.pyplot as plt from scipy import stats
```

```
#'num_outbound_cmds' is a redundant column so remove it from both train & test datasets
train.drop(['num_outbound_cmds'], axis=1, inplace=True) test.drop(['num_outbound_cmds'],
axis=1, inplace=True)
```

```
extract numerical attributes and scale it to have zero mean and unit variance
cols = train.select_dtypes(include=’float64','int64']).columns
sc_train = scaler.fit_transform(train.select_dtypes(include=['float64','int64']))
sc_test = scaler.fit_transform(test.select_dtypes(include=['float64','int64']))
```

EXTRACTIN GAND ENCODING

```
# extract categorical attributes from both training and test sets
cattrain = train.select_dtypes(include=['object']).copy()
cattest = test.select_dtypes(include=['object']).copy()

# encode the categorical attributes
traincat = cattrain.apply(encoder.fit_transform)
testcat = cattest.apply(encoder.fit_transform)

# separate target column from encoded data
enctrain = traincat.drop(['class'], axis=1)
cat_Ytrain = traincat[['class']].copy()
```

FEATURE HIERARCHY AND EXTRACTION

```
# create the RFE model and select 15 attributes
rfe = RFE(rfc, n_features_to_select=15)
 rfe = rfe.fit(train_x, train_y)

# summarize the selection of the attributes
feature_map = [(i, v) for i, v in itertools.zip_longest(rfe.get_support(), train_x.columns)]
selected_features = [v for i, v in feature_map if i==True]
```

TRAINING AND MODEL ARCHITECTURE

```
from sklearn.model_selection import train_test_split

X_train,X_test,Y_train,Y_test =
train_test_split(train_x,train_y,train_size=0.70, random_state=2)

#Fitting Models

# Importing the Keras libraries and packages import keras from keras.models import Sequential
from keras.layers import Dense

# Initialising the ANN
classifier = Sequential()

# Adding the input layer and the first hidden layer
classifier.add(Dense(units = 8, kernel_initializer = 'uniform', activation = 'relu', input_dim = 15))

# Adding the second hidden layer
classifier.add(Dense(units = 8, kernel_initializer = 'uniform', activation = 'relu'))

# Adding the output layer
classifier.add(Dense(units = 1, kernel_initializer = 'uniform', activation = 'sigmoid'))

# Compiling the ANN
classifier.compile(optimizer = 'adam', loss = 'binary_crossentropy', metrics = ['accuracy'])

# Fitting the ANN to the Training set
classifier.fit(X_train, Y_train, batch_size = 10, epochs = 10)
```

PERFORMANCE METRICS

```
from sklearn.model_selection import cross_val_score
from sklearn import metrics
from keras.wrappers.scikit_learn import KerasClassifier
accuracy = metrics.accuracy_score(Y_train, yhat_train)
confusion_matrix = metrics.confusion_matrix(Y_train, yhat_train)
 classification=metrics.classification_report(Y_train,yhat_train) print('Accuracy:',accuracy)
print('Confusion Matrix: ',confusion_matrix) print('Classification',classification)
```

Chapter

ORIGINALITY REPORT

8%	3%	6%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Hongyu Liu, Bo Lang. "Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey", Applied Sciences, 2019 Publication	2%
2	Shreekh Wankhede, Deepak Kshirsagar. "DoS Attack Detection Using Machine Learning and Neural Network", 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), 2018 Publication	1%
3	openaccess.city.ac.uk Internet Source	1%
4	Submitted to University of Hertfordshire Student Paper	1%

5	haipozeras.com Internet Source	<1%
6	www.coursehero.com Internet Source	<1%
7	cyberleninka.org Internet Source	<1%

8	docs.h2o.ai Internet Source	<1%
9	Giuliano Losa, Antonio Barbalace, Yuzhong Wen, Ho-Ren Chuang, Binoy Ravindran, Marina Sadini. "Transparent Fault-Tolerance Using Intra-Machine Full-Software-Stack Replication on Commodity Multicore Hardware", 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS), 2017 Publication	<1%
10	www.worldscientific.com Internet Source	<1%
11	R. R. Janghel. "Breast cancer diagnosis using Artificial Neural Network models", The 3rd International Conference on Information Sciences and Interaction Sciences, 06/2010 Publication	<1%
12	studentsrepo.um.edu.my Internet Source	<1%
13	360cyber.co Internet Source	<1%
14	Mustafa Altaha, Jae-Myeong Lee, Muhammad Aslam, Sugwon Hong. "Network Intrusion Detection based on Deep Neural Networks for the SCADA system", Journal of Physics: Conference Series, 2020 Publication	<1%

15	ijirset.com Internet Source	<1%
16	lib.buet.ac.bd:8080 Internet Source	<1%
17	www.pluralsight.com Internet Source	<1%
18	Mrudul Dixit, Rajashwini Ukarande. "Network Traffic Intrusion Detection System Using Fuzzy Logic and Neural Network", International Journal of Synthetic Emotions, 2017 Publication	<1%
19	Vinay Arora, Rohan Leekha, Raman Singh, Inderveer Chana. "Heart sound classification using machine learning and phonocardiogram", Modern Physics Letters B, 2019 Publication	<1%
20	Chirag Modi, Dhiren Patel, Bhavesh Borisaniya, Hiren Patel, Avi Patel, Muttukrishnan Rajarajan. "A survey of intrusion	<1%

detection techniques in Cloud", Journal of Network and Computer Applications, 2013

Publication

Exclude quotes On Exclude matches < 3 words Exclude bibliography On

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Deemed to be University u/s 3 of UGC Act* 1956)

Office of Controller of Examinations

REPORT FOR PLAGIARISM CHECK ON THE DISSERTATION PROJECT REPORTS FOR UG/PG PROGRAMMES (TO be attached in the dissertation/ project report)

1	Name of the Candidate (IN BLOCK LETTERS)	ANSHUMAAN MISHRA
2	Address of the Candidate	Horniman Circle, Kala Ghoda Mobile Number: 9619048802
3	Registration Number	RA1811028010076
4	Date of Birth	04/12/2000
5	Department	Department of Networking and Communications
6	Faculty	Dr Vigneshwaran Pandi
7	Title of the Dissertation/ Project	INTRUSION DETECTION USING A NEURAL NETWORK
8	Whether the above project/ dissertation is done by	Individual
9	Name and address of the Supervisor / Guide	Mail ID :vignesp@srmist.edu.in Mobile Number : 9994194794
10	Name and address of the Co-Supervisor / Co- Guide (if any)	Mail ID .• Mobile Number :
11	Software Used	
12	Date of Verification	
13	Plagiarism Details: (to attach the final report from the software)	

Chapter	Title of the Chapter	Percentage of similarity index (Including self-citation)	Percentage of similarity index (Excluding self-citation)	% Of plagiarism after excluding Quotes, Bibliography, etc.,
1	INTRODUCTION	6%	6%	6%
2	LITERATURE SURVEY	1%	1%	0%
3	PRELIMINARY DISCUSSION	7%	6%	3%
4	SYSTEM ANALYSIS	1%	0%	1%
5	SYSTEM DESIGN	4%	4%	0%
6	RESULT AND ANALYSIS	6%	6%	3%
7	CONCLUSION AND FUTURE WORK	8%	8%	8%
Appendices				

I / We declare that the above information have been verified and found true to the best of my / our knowledge.

Signature of the Candidate	Name & Signature of the Staff Who uses the plagiarism check software? Dr. Vigneshwaran P

Name & Signature of the Supervisor/Guide **Dr. Vigneshwaran P**	Name & Signature of the Co-Supervisor/Co-Guide
Name & Signature of the HOD	