Indian Institute of technology
Guwahati
Guwahati-781039, Assam(INDIA)
Department of Computer Science and Engineering

Machine Learning and Deep Learning approach for Intrusion detection systems

Submitted By:
Shubham Kumar
170101064
8th Semester

Guided by:
Dr. Manas Khatua
Asst. Professor
Dept. of CSE

Sept 2020 - April 2021
A B.Tech Project report submitted for the course CS499

Contents

Abstract

Advancements in communication fields hugely increased the network size, which ultimately gives rise to more intrusions (attacks). There should be a mechanism to detect those attacks. IDS are used to secure networks from various attacks and they do so by checking for incoming and outgoing traffic on the network. Implementing an IDS is a challenging task and an acceptable IDS should have high detection accuracy, low false detection rate and should detect new attacks. Traditional IDS lacks accuracy and becoming more complex with time. Recently, many researchers are working on making IDS based on machine learning (ML) and deep learning (DL) methods to solve many issues related to traditional IDS. In this report, we first detail about IDS and what are the traditional methods that we have used in IDS and why they are not good at current times. Then we will see the need for AI-based methods to develop IDS and understand and evaluate various ML and DL-based methods that can be used in IDS. We will see the performance of various models in real-life based on some evaluating metrics. This report also give comparative analysis of accuracy, training and testing time of various ML and DL models.

1 Introduction

Recently, due to advancements in internet and communication fields, many organizations are moving their platform to the internet, which ultimately increasing network sizes. Networks are prone to various attacks as they contain sensitive information, So network security is a vital research domain. Network security domain deals with the detection and prevention of various novel attacks on the system. Firewall, antivirus software and intrusion detection system are some tools that are used to secure networks.

1.1 IDS

An intrusion detection system (IDS) is used to secure the network. IDS checks for any vulnerable attacks and informs the administrator if it found any deviation from normal behavior.

1.1.1 Classification of IDS

As depicted in Figure 1, the Classification of IDS can be done based on their deployment or detection methods.

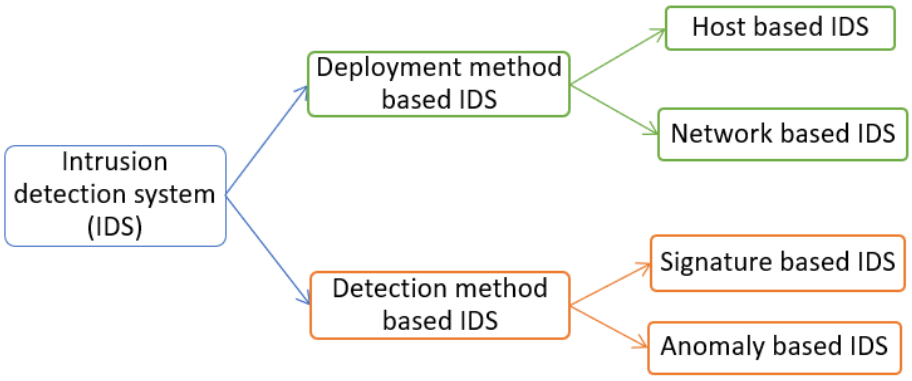

Figure 1: Classification of Intrusion Detection Systems (IDS)

- **Deployment Method based IDS** - IDS can be differentiated by their deployment in the secure network. Deployment based perspective IDS can be further sub classified as:-
  - **Host based IDS (HIDS)** - It is deployed on each single information device in the network. It monitors traffic on a single device and detects violations in the security policy on a single device only. It should be deployed on each device in the network, so this method is not good.

- **Network based IDS (NIDS)** - This is deployed on the network and monitors the incoming and outgoing data of all devices in the network. It detects intrusions on the network level and protects all devices in the network.

- **Detection Method based IDS** - IDS can be differentiated by the detection method used by IDS to detect intrusions. Detection based perspective IDS can be further sub classified as:-

  - **Signature based IDS (SIDS)** - The idea behind this method is to look for patterns. It stores patterns (signature) used by intrusions and matches these patterns with the incoming and outgoing traffic patterns to detect intrusions. It gives high accuracy while detecting known intrusions, but lacks accuracy against the unknown attacks.
  - **Anomaly detection based IDS (AIDS)** - **Anomaly** can be defined as a deviation from normal behaviour. These IDS try to estimate the normal behaviour of the system to be secured and report intrusion if any anomaly found in the traffic. They have benefited over the signature method based IDS these IDS as they can detect unknown attacks.

In this project we will see ML models that can be used with NIDS (Deployment method perspective) and AIDS (Detection methods perspective). A deployment of NIDS is depicted in Figure 2. NIDS copy all the traffic that comes and goes through the network and by monitoring all traffic it detects any intrusions in the network.

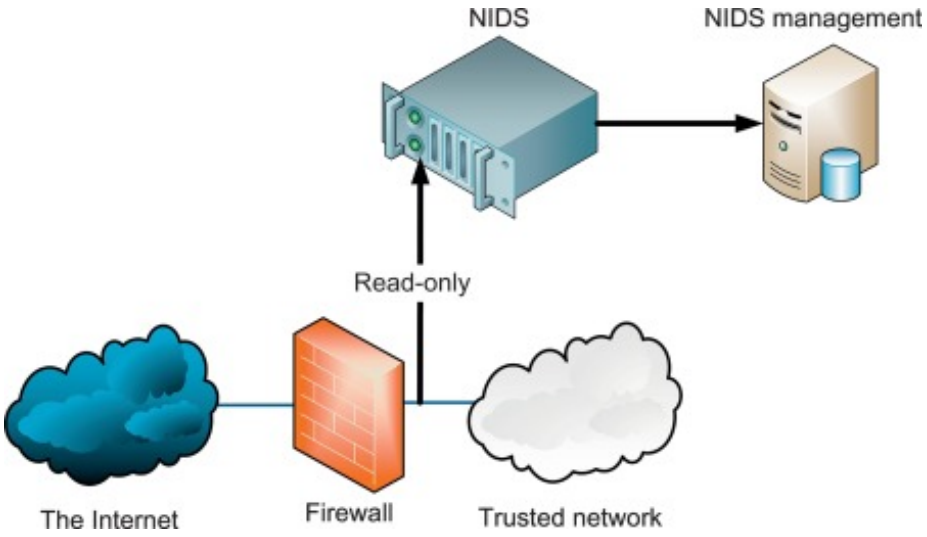

Figure 2: Deployment of NIDS

1.2 Traditional IDS and the need of machine learning in the IDS

Traditional IDS use mechanism based on Signature based methods or anomaly based methods, which has some limitations:-

- Real attacks are more often far less in number than false detection in the traditional IDS.
- Network suffers from noise and it increases false alarm rate.
- A signature method based IDS can only detect known attacks, they are not much useful against new attacks. Due to above reason they need software updates frequently.
- Anomaly based IDS can only detect deviation from normal behaviour which limits the variety of attacks that they can detect.
- Traditional IDS are unable to detect new attacks.

To solve these problems there should be a model which can automatically learn like humans to make a distinction between Normal traffic and Intrusion traffic. ML and DL methods are based on the idea of learning automatically based on past events. This idea of learning from features automatically can be used in designing IDS to overcome shortcomings of traditional IDS.

2 Related Work

IDS came into existence in 1980 when Jim Anderson[1] proposed the idea of IDS. In his report he focused on improving computer security. Since then many systems and models were proposed by researchers to secure networks. Due to the increasing network size specially wireless networks, traditional IDS are becoming more complex and fails in detecting new emerging attacks. The author of the article [2] discussed the challenges of IDS in wireless networks. They also proposed an anomaly-based detection method, but it lacked robustness. Due to shortcomings of traditional IDS, recently researchers have focused on using ML and DL algorithms in IDS.

In paper [3], authors studied and used the K-Means Clustering method on spark to identify whether the traffic is normal or attack behavior. Though they didn't use any feature selection technique, they got 10 anomalies for almost 400 thousand data on the KDDCup data set. Similarly, [4] used Mini Batch K-means combined with component analysis (PCA) to propose a clustering method for IDS.

Authors in [5] proposed a Machine Learning technique IDS based on a decision tree. They used a preprocessing algorithm to normalize the data and improved the detection accuracy. The authors also compared decision tree-based IDS with the Naïve Bayes method and the KNN method-based IDS on the KDDCUP99 data set. Similarly, [6] evaluated and compared the performance of Support Vector Machine, Decision Tree, Naïve Bayes, and Random Forest classifiers based IDS using Apache Spark.

Some researchers used unsupervised machine learning models to improve the efficiency of IDS. Zanero and Savaresi [7] proposed a two-tier architecture where first-tier is an unsupervised clustering algorithm to reduce the network packet payload size and the second tier is an already existing anomaly-based IDS. Lee and solfo [8] proposed data mining techniques to extract useful patterns of system features and use these patterns to generate classifiers to detect intrusions. They implemented two general data mining algorithms: the association rules algorithm and the frequent episodes algorithm. They also proposed an agent-based architecture for IDS to overcome the problem to compute and update detection agents continuously.

Deep learning is a branch of Machine Learning. Machine learning-based IDS heavily depends upon feature engineering to learn useful features from the traffic [9] while deep learning-based IDS do not rely on feature engineering and are good at learning complex features automatically [10]. The author of the article [11] proposed a neural-network-based IDS method. The author used neural networks to detect unusual activity. The author employed feedforward neural networks with the backpropagation training algorithm in his study. An article

[12] by Hongyu Liu and Bo Lan provides a brief overview of different ML and DL approaches that can be used in IDS.

In spite of the many shortcomings of existing IDS and extensive research in the Machine learning and Deep learning field, it is far from reality to implement ML-based IDS in a real-life scenario. Sommer and Paxson [13] studied the lack of deployment of Machine Learning based IDS. The authors explained why the network intrusion detection systems problem is different from other problems where Machine learning achieved high success rate. They claim that the task of detecting attacks has a fundamental distinction from these other applications, which makes this task significantly hard for the researchers to employ machine learning effectively in IDS. They identified various domains and domain-specific challenges in creating and deploying Machine Learning based IDS. They also proposed some guidelines that can be used for applying Machine Learning in Intrusion detection systems.

Training and Testing of ML and DL models are also challenging due to a lack of data. The data that is available is limited and available only for benchmarking purposes. KDDCUP99 [14] is one of the few data sets that are available and can be used to train the models. The KDDCUP99 data set was used in the KDDCUP contest in 1999, where the task was to create a model to classify data into 5 classes of attacks and normal traffic. The preprocessing of the data for the competition was done using MADAMID framework [15].

Since many researchers have proposed many methods ML and DL methods that can be used in IDS. But they lack robustness and this machine learning problem is different than other problems of this area. In this report we have evaluated performances of various models for 2 different scenarios to understand why ML-based IDS are not used more often in reality.

3 Machine Learning in IDS

Machine learning (ML) algorithms give computer models the capability to learn from experience and improve upon them without explicit programming. ML models learn from observations or training data and predict the output for a given input. These models, modify themselves to estimate correct values without any human intervention.
Machine learning algorithms used in IDS can be classified as:-

- **Supervised algorithms** - These types of models work with labeled data. They use labeled training data set to build a function, which can predict outcomes of input.
- **Unsupervised algorithms** - These are used with unclassified or unlabelled data. These algorithms try to build a function to represent the hidden structure of the unlabeled data.

3.1 Overview of some ML algorithms in IDS

3.1.1 Decision tree

This supervised Machine earning algorithm classify data based on some order. The model builds a tree with nodes, branches and leaf. Each node represents attribute, each branch represents a decision and each eaf represnt a class. Then final tree is built by selecting the best features and removing the unnecessary branches.

3.1.2 K-Nearest Neighbour (KNN)

These types of algorithms are some of the simplest supervised algorithms. These algorithms are based on the assumption that a sample has a higher chance to be in the same class, to which most of the neighbors belong. Thus, these models are highly influenced by the value of k.

3.1.3 Support vector machine (SVM)

This class of supervised algorithms, tries to find a max-margin separation hyperplane in the n-dimensional feature space. These can be used for linear as well as non-linear problems, although for solving non-linear problems they usually use kernel functions. These models can achieve high accuracy even with small-size training sets as separating hyperplane is determined by only a small number of support vectors.

3.1.4 Clustering

These unsupervised algorithms tries to divide data into different groups or clusters. They place highly similar data into the same group/cluster and less similar data into the different groups/clusters . Since these are unsupervised al-

gorithms, no prior labelling of data is needed. **K-means clustering** is a clustering algorithm, where K represents a number of clusters and means is the mean of the attributes. It uses distance to measure similarity between two data points. Data points having short distances has high chances of being in the same group. K-means algorithm is highly influenced by the value of K.

3.1.5 Naïve Bayes

This type of algorithm is based on the idea of conditional probability and the hypothesis of attribute independence. For each data, these classifiers calculate the conditional probabilities for different classes. Then classify each data into the class which has maximum probability . The conditional probability of data is calculated as in Formula (1).

$$P(X = x|Y = c_k) = \prod_{i=1}^{n} P(X^{(i)} = x^{(i)}|Y = c_k) \tag{1}$$

In reality attribute, the independence hypothesis is tough to satisfy, so this algorithm does not perform well in reality.

3.1.6 Linear Regression (LR)

Linear Regression is a type of logarithmic linear model. It uses parametric logistic distribution to compute the probabilities of different classes. The probability Calculation formula is shown in Formula (2).

$$P(Y = k|x) = \frac{e^{w_k * x}}{1 + \sum_{k}^{K-1} e^{w_k * x}} \tag{2}$$

$$k \in 1, 2, 3, ..., K - 1.$$

This model can be built easily.

3.1.7 Artificial neural network (ANN)

These supervised models are based on the working of the human brain. ANN has an input layer, many hidden layers, and an output layer. The adjacent layer units are fully connected. Though training ANN models are not easy due to their complex structure.

3.1.8 Ensemble methods

Every individual classifiers has strengths and weaknesses. The idea behind these models is to use several weak classifiers to create a single strong classifier by selecting using a voting algorithm.

4 Evaluation of some ML algorithms in IDS

In this section we will see performance of some ML based models in the detection of IDS.

4.1 Dataset

In this report for evaluation purposes, we will be using the KDD dataset, which was used in the UCI KDD1999 competition. The task of the competition was to develop IDS to detect various attacks. The dataset has a total of 4898430 different data instances. The dataset has 23 different classes of data with 22 types of attack. Each datum has 41 features. Table 1 shows instances of various data classes in the dataset.

In this Report for experiments we used a subset of dataset to train the model and other subset to test the model. For Machine learning models this ratio is 90:10 and for Deep Learning models this ratio is 70:30.

4.2 Evaluation metrics

In this report we will be using **Accuracy**, **Precision**, **Recall**, **F-Measure** and **Confusion Matrix**.

- Confusion matrix - Confusion Matrix provides information about predicted and actual classes of unlabeled data. For example, let's take the confusion matrix with n rows and m columns. Then the value in cell (I, j) can be seen as the total instances of the class i in unlabelled data which are predicted to be in class j.
- **Accuracy** - It represents the ratio of correctly classified instances to the total number of instances.

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

- **Precision** - The ratio of correctly predicted instances in the class to the total instances predicted to be in that class.

$$Precision = \frac{TP}{TP + FP}$$

- **Recall** - The ratio of correctly predicted instances in the class to the total instances of the class.

$$Recall = \frac{TP}{TP + FN}$$

- **F-Measure** - It is the harmonic mean of Precision and Recall.

$$F - Measure = 2\left(\frac{Precision \times Recall}{Precision + Recall}\right)$$

Attack class	Attack name	Instances
DOS	SMURF	2807886
	NEPTUNE	1072017
	BACK	2203
	POD	264
	TEARDROP	979
	LAND	21
U2R	Buffer Overflow	30
	Load Module	9
	PERL	3
	Rootkit	10
R2L	FTP Write	8
	Guess Password	53
	IMAP	12
	MultiHop	7
	PHF	4
	SPY	2
	Warez client	1020
	Warez master	20
PROBE	IPSWEEP	12481
	NMAP	2316
	PORTSWEEP	10413
	SATAN	15892
Normal	-	972780

Table 1: Various attacks with their instances

4.3 Analysis of some Machine learning (ML) models in IDS

In this report, we will analyze the performance of the ML model based on accuracy. There are some techniques known as 'Evaluation metrics', that are used in this report to compare the performance of various models. We have used K-

means, Random Forest, and Naïve Bayes to evaluate ML Models. We will first see results for 5 classes of attacks and normal. Then we will treat all intrusions in a single class. The next subsections give brief results obtained.

4.3.1 K-means

This is an unsupervised model so we cannot evaluate performance with the metrics discussed above. But we can get some idea from the various cluster value counts. Some conclusions from the results-

- On both cluster size 5 and 2, most of the data are clustered into a single cluster while training model.
- On testing data, this model fails badly and maps all values of testing data to a single cluster.
- Some feature selection techniques can help to overcome this problem.

4.3.2 Decision Tree

As explained earlier this is a supervised model that classifies data based on a series of rules. Evaluation Metrics of this model are described below for both classifications.

Class	DOS	NORMAL	PROBE	R2L	U2R
DOS	388031	230	4	0	0
Normal	3	96483	835	0	0
PROBE	26	273	3844	0	0
R2L	0	106	2	0	0
U2R	0	4	2	0	0

Table 2: Decision Tree algorithm Confusion matrix for cluster size = 5

Class	Precision	Recall	F-Measure	Support
DOS	1.00	1.00	1.00	388265
NORMAL	0.99	0.99	0.99	97321
PROBE	0.82	0.93	0.87	4143
R2L	0.00	0.00	0.00	108
U2R	0.00	0.00	0.00	6
Accuracy			1.00	489843
Macro avg	0.56	0.58	0.57	489843
Weighted avg	1.00	1.00	1.00	489843

Table 3: Decision Tree algorithm evaluation metrics for cluster size = 5

Class	Intrusion	Normal
Intrusion	391982	540
Normal	12	97309

Table 4: Decision Tree algorithm Confusion matrix for cluster size = 2

Class	Precision	Recall	F-Measure	Support
Intrusion	1.00	1.00	1.00	392522
Normal	0.99	1.00	1.00	97321
Accuracy			1.00	489843
Macro avg	1.00	1.00	1.00	489843
Weighted avg	1.00	1.00	1.00	489843

Table 5: Decision Tree algorithm evaluation metrics for cluster size = 2

4.3.3 Random forest

It is an ensemble method type classifier. It combines many other algorithm classifiers. Classifiers create many trees on any random subset of data and then combine the total votes of each tree to give a class of the test data. Evaluation Metrics of this model are described below for both the classifications.

Class	DOS	NORMAL	PROBE	R2L	U2R
DOS	388263	2	0	0	0
Normal	0	97321	0	0	0
PROBE	0	3	4140	0	0
R2L	0	4	0	104	0
U2R	0	5	0	0	1

Table 6: Random Forest algorithm Confusion matrix for cluster size = 5

Class	Precision	Recall	F-Measure	Support
DOS	1.00	0.99	1.00	388265
NORMAL	0.99	1.0	1.00	97321
PROBE	1.00	0.99	1.00	4143
R2L	1.00	0.96	0.98	108
U2R	1.00	0.17	0.29	6
Accuracy			1.00	489843
Macro avg	1.00	0.83	0.85	489843
Weighted avg	1.00	1.00	1.00	489843

Table 7: Random Forest algorithm evaluation metrics for cluster size = 5

Class	Intrusion	Normal
Intrusion	392509	13
Normal	0	97321

Table 8: Random Forest algorithm Confusion matrix for cluster size = 2

Class	Precision	Recall	F-Measure	Support
Intrusion	1.00	0.99	1.00	392522
Normal	0.99	1.00	1.00	97321
Accuracy			1.00	489843
Macro avg	1.00	1.00	1.00	489843
Weighted avg	1.00	1.00	1.00	489843

Table 9: Random Forest algorithm evaluation metrics for cluster size = 2

4.3.4 Naïve Bayes

As discussed in Section 2 this approach is based on a probabilistic graphical model. Here is an evaluation of this approach in IDS.

Class	DOS	NORMAL	PROBE	R2L	U2R
DOS	387935	262	0	0	68
NORMAL	27294	66590	787	98	2552
PROBE	2626	1158	98	0	261
R2L	5	43	19	0	41
U2R	1	1	0	0	4

Table 10: Naïve Bayes algorithm Confusion matrix for all attack types

Class	Precision	Recall	F-Measure	Support
DOS	0.93	0.99	0.96	388265
NORMAL	0.98	0.68	0.81	97321
PROBE	0.11	0.02	0.04	4143
R2L	0.00	0.00	0.00	108
U2R	0.00	0.67	0.00	6
Accuracy			0.93	489843
Macro avg	0.40	0.47	0.36	489843
Weighted avg	0.93	0.93	0.92	489843

Table 11: Naïve Bayes algorithm evaluation metrics for all attack types

Class	Intrusion	Normal
Intrusion	347174	45348
Normal	769	96552

Table 12: Naïve Bayes algorithm Confusion matrix for all intrusion and normal traffic

Class	Precision	Recall	F-Measure	Support
Intrusion	0.99	0.88	0.94	392522
Normal	0.68	0.99	0.81	97321
Accuracy			0.91	489843
Macro avg	0.84	0.94	0.87	489843
Weighted avg	0.93	0.91	0.91	489843

Table 13: Naïve Bayes algorithm evaluation metrics for intrusion and normal traffic

4.4 Accuracy and timing analysis

In this subsection, we will analyze the training and testing time of various algorithms and their accuracy. As shown in Figure 3, Random Forest training and testing time are much higher than other algorithms. The testing time of all the models is small and not differs from much. Though we cannot directly measure the accuracy of the K-Means algorithm so we have not included that in the measurement of accuracy. Testing accuracy of all the models in above 90%. Though accuracy alone cannot guarantee good performance of the model as we have seen in the confusion matrix that the models failed to detect U2R and R2L attacks.

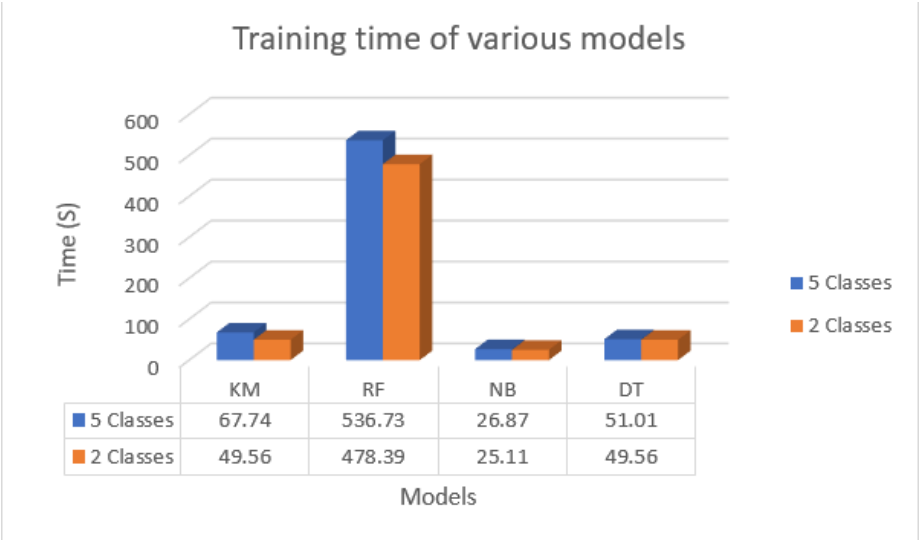

Figure 3: Training time of various models for both classifications

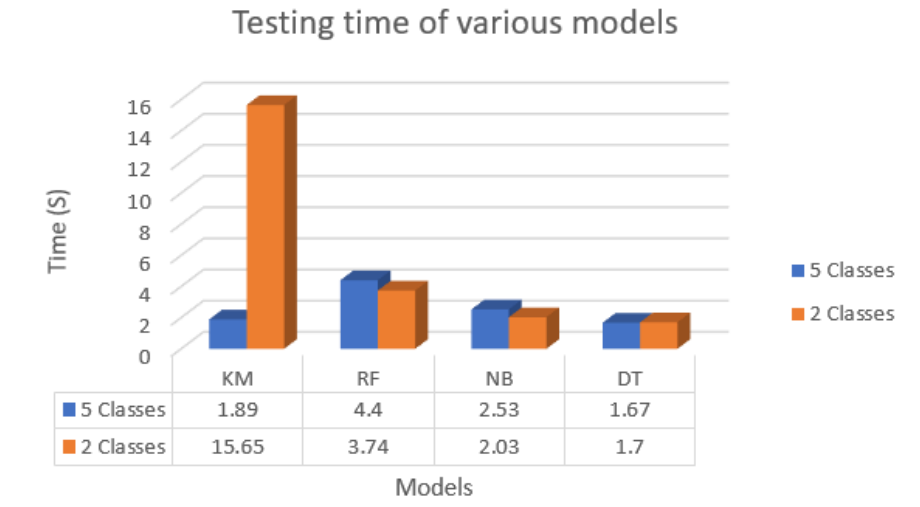

Figure 4: Testing time of various models for both classifications

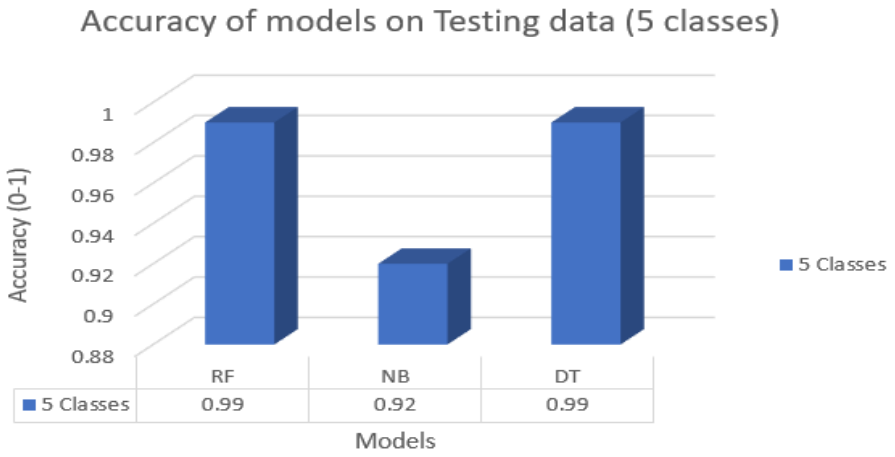

Figure 5: Accuracy of various models on testing (5 classes)

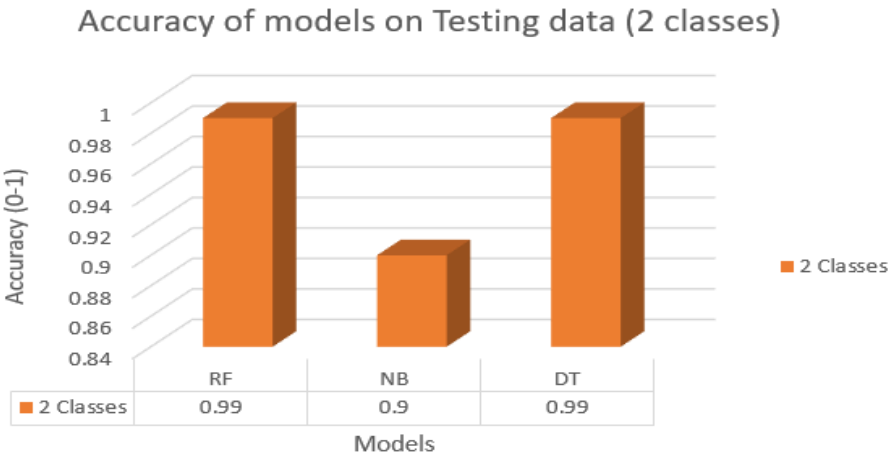

Figure 6: Accuracy of various models on testing (2 classes)

As depicted in Figure 3 training time of Random forest is quite high as compared to other models, which is due to its training model as it generates many decision trees to choose the optimal one. Though the accuracy of decision tree and the random forest is quite high compared to Naïve Bayes.

5 Deep Learning in IDS

Machine learning (ML) algorithms are widely used to develop an intrusion detection system (IDS). These ML techniques detect and classify cyber attacks at the network level and the host level. But these techniques have many challenges as malicious attacks are continually changing and are occurring at a very large rate. Due to the continually changing nature, it is needed to update the training dataset accordingly and train model with updated dataset. Deep learning techniques can be a feasible solution to overcome some of these problems as these models can extract features from the datasets and can classify and detect unforeseen attacks.
In this paper, we will discuss the architecture of deep Neural Networks and test some deep neural models for some network datasets.

5.1 Deep Learning Architecture

Deep learning is a branch of ML and these algorithms enable machines to learn automatically, somewhat as humans do. These models have multiple hidden layers to learn the features of the network. These algorithms extract features from the dataset automatically, and generate corresponding output based on the learned features. This property makes these models more powerful than ML models. In the next subsections, we will discuss some Deep Learning models.

5.1.1 Recurrent Neural Networks (RNN)

These Neural Networks remember the output from the previous step and fed that previous step output as input to the current step. Since in many cases previous output is required to generate current output so RNN came into existence. The hidden unit in these models is considered as memory elements. Figure 7 shows the Structure of RNN, where all W are same.

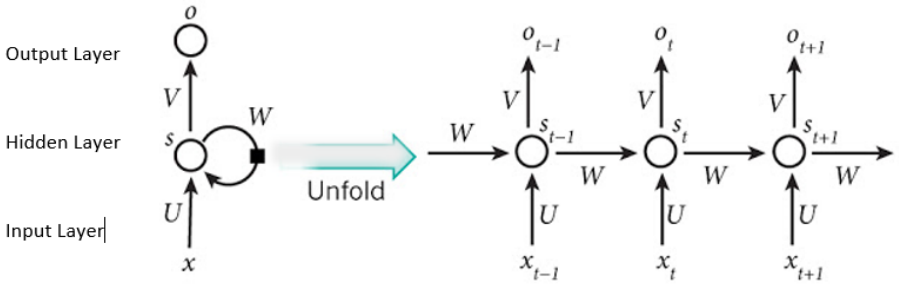

Figure 7: The structure of RNN

These models are widely used in various fields like music composition, Natural Language Processing, speech synthesis, etc. In IDS, RNN is used for feature classification and supervised classification. RNN can handle only small length sequence as they will from short-term memory loss for large sequences. [16] proposed RNN-based IDS. He used the NSLKDD dataset and proposed binary and multiclass classification of the dataset. He tested the model by using different numbers of hidden nodes and learning rates. Different scenarios of the number of hidden nodes and learning rates showed different results and accuracy. The proposed model performed better than many ML algorithms, but it took too much time in training and failed for many attack types whose data was less as compared to other classes. Some Other IDS based on RNN were also proposed by many researchers. [17] used GRU as the main memory with the multilayer perceptron and softmax classifier.

5.1.2 AutoEncoder (AE)

AutoEncoder is a popular unsupervised DL technique, which is based on the idea of learning the best features and the input should be as near as output as possible. It comprises an input, output, and hidden layers. Input and Output layers are of the same dimension and generally dimensions of the hidden layer are smaller than the input layer. The structure of the autoencoder is shown in Figure 8.

An autoencoder contains two symmetrical components, an encoder, and a

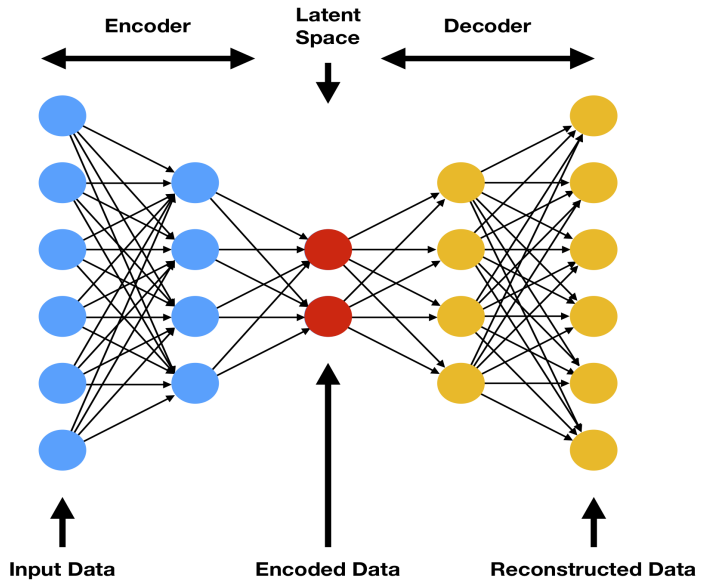

Figure 8: The structure of an autoencoder

decoder. The encoder is used to extract features from raw data and the decoder reconstructs the data from the extracted features. Training of the model includes reducing the error between the input of the encoder and the output of the decoder and stops when the decoder successfully reconstructs the data via extracted features. This training requires no supervised information. There are many variants of autoencoder like dancing autoencoders [18][19], Stacked autoencoder, Sparse autoencoder and variational autoencoder [20].
Many researchers proposed IDS based on autoencoders. [21] designed deep autoencoders and random forest ML technique-based IDS. [22] proposed an IDS based on SVM and stacked sparse autoencoder (SSAE). This model achieved fair accuracy in detecting U2R and R2L attacks. Though the proposed IDS lacks comparison in other classes of attacks.

5.1.3 Deep Neural Network (DNN)

This basic DL structure focuses on learning in multiple layers. This model has an input layer, an output layer, and many hidden layers. First, parameters are learned using unlabeled data in an unsupervised feature learning stage. Then, the network is tuned through the labeled data in a supervised learning stage. The unsupervised feature learning stage makes these models more accurate than other models. The basic structure of a DNN is shown in Figure 9.
These models have applications in modeling complex nonlinear functions. [23]

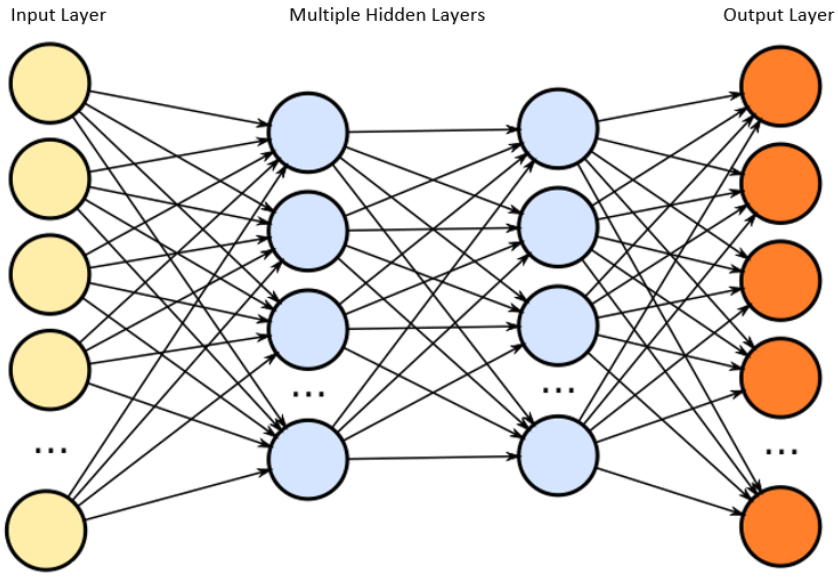

Figure 9: Structure of Deep Neural Network

proposed an IDS based on DNN. Proposed DNN in IDS contains four hidden layers which are used to classify the datasets. The Output layer had one fully connected layer and a softmax classifier was used to classify data. This model achieved high accuracy for all attack types except U2R due to less number of data points.[24] researched and evaluated Deep Neural Networks based IDS with adversaries. [25] studied and proposed scale-hybrid-IDS-AlertNet which is a hybrid scalable DNN framework . Scale-hybrid-IDS-AlertNet can be used for intrusion identification in both host and network levels.

5.1.4 Convolutional Neural Network (CNN)

The design of CNNs is inspired by the human visual system. This DL structure is recommended for data that is stored in arrays. This structure has an Input layer and stacked with alternate convolutional and pooling layers which are used for feature extraction. A fully connected layer and a softmax classifier are used in the classification layer. Since this structure mimics the human visual system, these structures made a significant success in the computer vision field. The basic structure of CNN is shown in Figure 10.

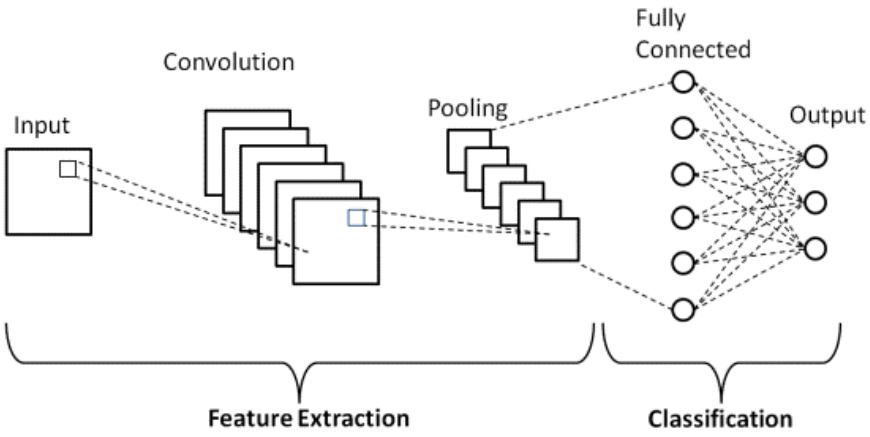

Figure 10: Structure of Convolutional Neural Network

Many researchers studied and proposed IDS based on CNNs. Researchers used CNNs in IDS for classification and supervised feature extraction. [26] proposed IDS based on CNN, which uses Principal Component Analysis and AE for Feature Extraction. Though this model also lacked accuracy for R2L and U2R attack types. [27] suggested a CNN and gcForest based IDS. [28] combined CNN and bidirectional long short-term memory in a deep hierarchy to propose an efficient IDS system. This model achieved slightly high accuracy, but the training time of this model is very high as compared to other models.

5.1.5 Restricted Boltzmann MachineRBM

RBM is named after Boltzmann distribution (an integral part of statistical mechanics). In these randomized neural networks, all units obey the Boltzmann distribution. These non-deterministic generative DL models have only *hidden* and *visible* nodes. Same layer units are not connected and different layer units are fully connected as shown in Figure 11.

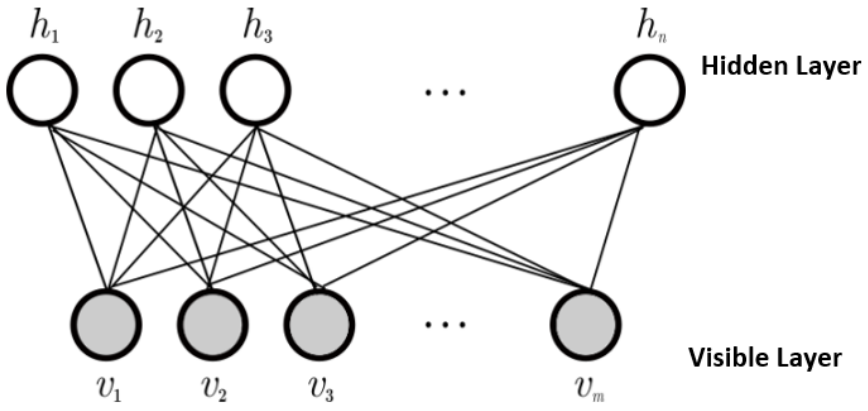

Figure 11: Structure of Restricted Boltzmann Machine

RBMs do not differentiate between the forward and backward directions, so they have the same weights in both directions. RBMs are unsupervised learning models trained by the contrastive divergence algorithm and are ususally applied for feature extraction or dancing.

5.1.6 Deep belief network(DBN)

Deep belief network consists of many Restricted Boltzmann Machines (RBM). RBMs are stacked in layers and classified by a softmax classifier. DBN is trained by using an unsupervised greedy layer-by-layer method, then fine-tuned by a supervised method to extract useful features. Figure 12 shows the Structure of DBN.

In IDS DBNs are used for feature extraction and classification tasks. [29] and [30] proposed IDS based on DBNs.

5.1.7 Generative Adversarial Network (GAN)

A GAN model consists of a generator and a discriminator. The generator is used to generate new content and the discriminator is able to distinguish be-

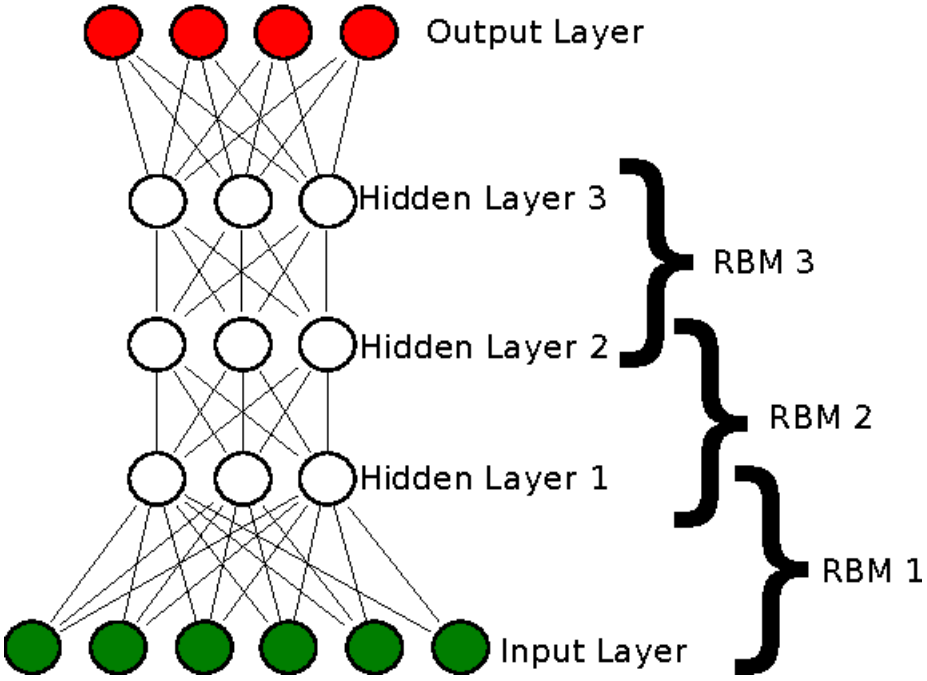

Figure 12: Structure of Deep belief network

tween generated data and real data. These models can be used to increase the accuracy of existing models.

5.1.8 Multilayer Perceptron(MLP)

It is a neural model which can be created by one or more dense layers. Each node is connected to the output of previous layer and each node is connected to the inputs of the next layer.

5.2 Testing of Sequential model

In this subsection, we will analyze a sequential model with a different number of hidden layers. For testing we are using keras and sequential model to build Multilayer perceptron. The dataset is the same that we used for analyzing ML algorithms. The input layer has 41 inputs. The first hidden layer has 12 nodes and uses the relu activation function. Other hidden layers use 8 nodes and relu activation function. The output layer uses the softmax activation function. Now we will discuss the confusion matrix and classification matrix for one, two, and three hidden layers.

5.2.1 One hidden layer

Here the number of hidden layers in the sequential model is one.

Class	DOS	NORMAL	PROBE	R2L	U2R
DOS	1164437	790	14	0	0
NORMAL	7	291652	51	0	0
PROBE	3	146	12097	0	0
R2L	0	294	1	21	0
U2R	0	16	0	0	0

Table 14: Confusion matrix for one hidden layer Sequential model and 5 classes

Class	Precision	Recall	F-Measure	Support
DOS	1.00	1.00	1.00	1165241
NORMAL	1.00	1.00	1.00	291710
PROBE	0.99	0.99	0.99	12246
R2L	1.00	0.07	0.12	316
U2R	0.00	0.00	0.00	16
Accuracy			1.00	1469529
Macro avg	0.80	0.61	0.62	1469529
Weighted avg	1.00	1.00	1.00	1469529

Table 15: Evaluation metrics for one hidden layer Sequential model and 5 classes

Class	Normal	Intrusion
Normal	291696	14
Intrusion	3750	1174069

Table 16: Confusion matrix for one hidden layer Sequential model and 2 classes

Class	Precision	Recall	F-Measure	Support
Normal	0.99	1.00	0.99	291710
Intrusion	1.00	1.00	1.00	1177819
Accuracy			1.00	1469529
Macro avg	0.99	1.00	1.00	1469529
Weighted avg	1.00	1.00	1.00	1469529

Table 17: Evaluation metrics for one hidden layer Sequential model and 2 classes

5.2.2 Two hidden layers

Here the number of hidden layers in the sequential model is two.

Class	DOS	NORMAL	PROBE	R2L	U2R
DOS	1164333	904	4	0	0
NORMAL	13	291674	23	0	0
PROBE	4	468	11774	0	0
R2L	1	313	2	0	0
U2R	0	16	0	0	0

Table 18: Confusion matrix for two hidden layers Sequential model and 5 classes

Class	Precision	Recall	F-Measure	Support
DOS	1.00	1.00	1.00	1165241
NORMAL	0.99	1.00	1.00	291710
PROBE	1.00	0.96	0.98	12246
R2L	0.00	0.00	0.00	316
U2R	0.00	0.00	0.00	16
Accuracy			1.00	1469529
Macro avg	0.60	0.59	0.60	1469529
Weighted avg	1.00	1.00	1.00	1469529

Table 19: Evaluation metrics for two hidden layers Sequential model and 5 classes

Class	Normal	Intrusion
Normal	291707	3
Intrusion	11747	1166072

Table 20: Confusion matrix for two hidden layers Sequential model and 2 classes

Class	Precision	Recall	F-Measure	Support
Normal	0.96	1.00	0.98	291710
Intrusion	1.00	0.99	0.99	1177819
Accuracy			0.99	1469529
Macro avg	0.98	1.00	0.99	1469529
Weighted avg	0.99	0.99	0.99	1469529

Table 21: Evaluation metrics for two hidden layers Sequential model and 2 classes

5.2.3 Three hidden layers

- Here the number of hidden layers in the sequential model is three.

Class	DOS	NORMAL	PROBE	R2L	U2R
DOS	1164483	755	3	0	0
NORMAL	29	291615	66	0	0
PROBE	2	491	11753	0	0
R2L	1	314	1	0	0
U2R	0	16	0	0	0

Table 22: Confusion matrix for three hidden layers Sequential model and 5 classes

Class	Precision	Recall	F-Measure	Support
DOS	1.00	1.00	1.00	1165241
NORMAL	0.99	1.00	1.00	291710
PROBE	0.99	0.96	0.98	12246
R2L	0.00	0.00	0.00	316
U2R	0.00	0.00	0.00	16
Accuracy			1.00	1469529
Macro avg	0.60	0.59	0.59	1469529
Weighted avg	1.00	1.00	1.00	1469529

Table 23: Classification report for three hidden layers Sequential model and 5 classes

Class	Normal	Intrusion
Normal	291705	5
Intrusion	38287	1139532

Table 24: Confusion matrix for three hidden layers Sequential model and 2 classes

Class	Precision	Recall	F-Measure	Support
Normal	0.88	1.00	0.94	291710
Intrusion	1.00	0.97	0.98	1177819
Accuracy			0.97	1469529
Macro avg	0.94	0.98	0.96	1469529
Weighted avg	0.98	0.97	0.97	1469529

Table 25: Evaluation metrics for three hidden layers Sequential model and 2 classes

5.2.4 Accuracy and timing analysis

In this subsection, we will analyze the training and testing time of three DL models that we have implemented. As shown in Figure 14 training time on DL models is very high as compared to ML models. Accuracy of all 6 models (3

models, each has two classifications) has an accuracy of over 99%.

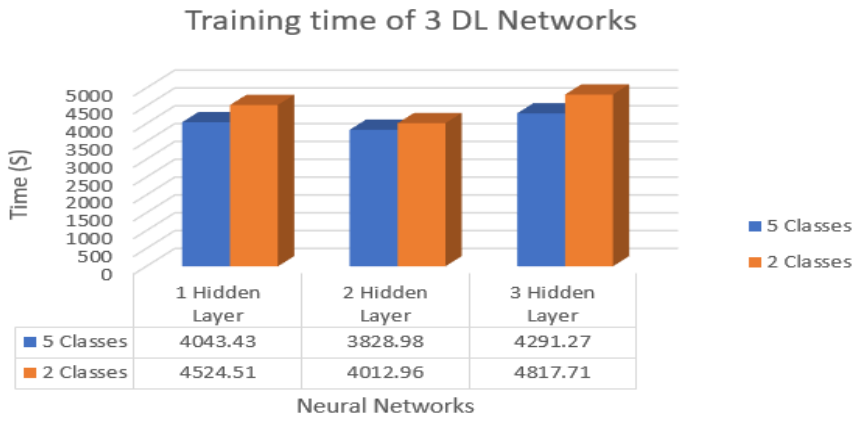

Figure 13: Training time of Three DL models for both classifications

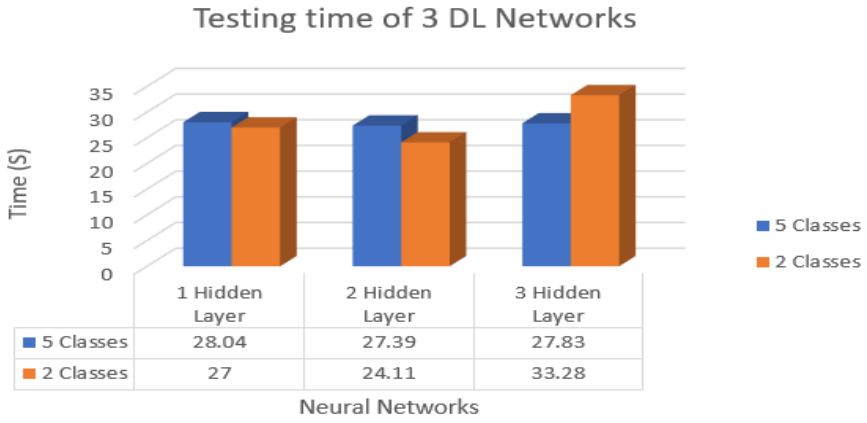

Figure 14: Testing time of Three DL models for both classifications

6 Conclusion

This project, explained the various aspects of Intrusions in networks and how networks can be exploited by intruders. Since Intrusions always keep themselves changing so it is necessary for an IDS to able to learn new features of intrusions. Here Machine learning and Deep learning approaches can play a big role as they can learn from experience and can even detect new attacks based on old attacks. In this report, we have seen many ML and DL techniques to tackle Intrusions. We analyzed some Machine learning and Deep learning models that can be used in IDS. Here are some conclusions that can be drawn from the observations-

- An ML algorithm can be better in terms of precision, but it may not be the best one due to false rate. No single algorithm can handle all types of attacks.
- ML algorithms in IDS should be applied according to the requirements of the network.
- ML Models can differentiate between Normal traffic and intrusion traffic, but lacks accuracy in differentiating various attacks.
- Deep learning models are better in terms of low false rate but they need a large training dataset as they fail to identify U2R and R2L type attacks due to low data points. Also, Deep learning models take more time in training.
- Better Accuracy does not mean that model is better as we have seen in the case of the Deep learning sequential model where accuracy was high but the model failed to identify R2L and U2R attacks. High Accuracy can also be due to the high difference in the number of data points of different classes.
- An increasing number of hidden layers did not decrease the false alarm rate. This can be due to over-fitting.

References

[1] James P. Anderson Co. Computer Security Threat Monitoring and Surveillance. *Technical report*, 1980.

[2] Yongguang Zhang, Wenke Lee, and Yi-an Huang. Intrusion detection techniques for mobile wireless networks. *ACM Wireless Networks Journal*, 9, 04 2003.

[3] Sevcan Korkmaz and Ferhat Karatas. Big data: Controlling fraud by using machine learning libraries on spark. *International Journal of Applied Mathematics, Electronics and Computers*, 6:1–5, 03 2018.

[4] K. Peng, V. C. M. Leung, and Q. Huang. Clustering approach based on mini batch kmeans for intrusion detection system over big data. *IEEE Access*, 6:11897–11906, 2018.

[5] Kai Peng, Victor C. M. Leung, Lixin Zheng, Shangguang Wang, Chao Huang, and Tao Lin. Intrusion detection system based on decision tree over big data in fog environment. *Wireless Communications and Mobile Computing*, 2018(4680867):10, 2018.

[6] Mustapha Belouch, Salah El Hadaj, and Mohamed Idhammad. Performance evaluation of intrusion detection based on machine learning using apache spark. *Procedia Computer Science*, 127:1–6, 2018. PROCEEDINGS OF THE FIRST INTERNATIONAL CONFERENCE ON INTELLIGENT COMPUTING IN DATA SCIENCES, ICDS2017.

[7] Stefano Zanero and Sergio Savaresi. Unsupervised learning techniques for an intrusion detection system. *Proceedings of the ACM Symposium on Applied Computing*, 1, 12 2003.

[8] Wenke Lee and Salvatore Stolfo. Data mining approaches for intrusion detection. 7, 02 1998.

[9] Maryam Najafabadi, Flavio Villanustre, Taghi Khoshgoftaar, Naeem Seliya, Randall Wald, and Edin Muharemagic. Deep learning applications and challenges in big data analytics. *Journal of Big Data*, 2, 12 2015.

[10] B. Dong and X. Wang. Comparison deep learning method to traditional methods using for network intrusion detection. In *2016 8th IEEE International Conference on Communication Software and Networks (ICCSN)*, pages 581–585, 2016.

[11] Alan Bivens, Chandrika Palagiri, Rasheda Smith, Boleslaw Szymanski, and M. Embrechts. Network-based intrusion detection using neural networks. *Intell. Eng. Syst. Artif. Neural Networks*, 12 2002.

[12] Hongyu Liu and Bo Lang. Machine learning and deep learning methods for intrusion detection systems: A survey. *Applied Sciences*, 9:4396, 10 2019.

[13] Robin Sommer and Vern Paxson. Outside the closed world: On using machine learning for network intrusion detection. pages 305–316, 01 2010.

[14] *http://kdd.ics.uci.edu/databases/kddcup99/task.html*, 1999.

[15] Wenke Lee and Salvatore J. Stolfo. A framework for constructing features and models for intrusion detection systems. *ACM Trans. Inf. Syst. Secur.*, 3(4):227–261, November 2000.

[16] C. Yin, Y. Zhu, J. Fei, and X. He. A deep learning approach for intrusion detection using recurrent neural networks. *IEEE Access*, 5:21954–21961, 2017.

[17] C. Xu, J. Shen, X. Du, and F. Zhang. An intrusion detection system using a deep neural network with gated recurrent units. *IEEE Access*, 6:48697–48707, 2018.

[18] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, page 1096–1103, New York, NY, USA, 2008. Association for Computing Machinery.

[19] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res.*, 11:3371–3408, December 2010.

[20] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[21] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi. A deep learning approach to network intrusion detection. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2(1):41–50, 2018.

[22] Dimitrios Papamartzivanos, Félix Gómez Mármol, and G. Kambourakis. Introducing deep learning self-adaptive misuse network intrusion detection systems. *IEEE Access*, 7:13546–13560, 2019.

[23] Yang Jia, Meng Wang, and Yagang Wang. An network intrusion detection algorithm based on new deep neural network. *IET Information Security*, 13, 08 2018.

[24] Z. Wang. Deep learning-based intrusion detection with adversaries. *IEEE Access*, 6:38367–38384, 2018.

[25] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman. Deep learning approach for intelligent intrusion detection system. *IEEE Access*, 7:41525–41550, 2019.

[26] Y. Xiao, C. Xing, T. Zhang, and Z. Zhao. An intrusion detection model based on feature reduction and convolutional neural networks. *IEEE Access*, 7:42210–42219, 2019.

[27] X. Zhang, J. Chen, Y. Zhou, L. Han, and J. Lin. A multiple-layer representation learning model for network-based attack detection. *IEEE Access*, 7:91992–92008, 2019.

[28] K. Jiang, W. Wang, A. Wang, and H. Wu. Network intrusion detection combined hybrid sampling with deep hierarchical network. *IEEE Access*, 8:32464–32476, 2020.

[29] N. Marir, H. Wang, G. Feng, B. Li, and M. Jia. Distributed abnormal behavior detection approach based on deep belief network and ensemble svm using spark. *IEEE Access*, 6:59657–59671, 2018.

[30] P. Wei, Y. Li, Z. Zhang, T. Hu, Z. Li, and D. Liu. An optimization method for intrusion detection classification model based on deep belief network. *IEEE Access*, 7:87593–87605, 2019.