Cloud based Network Intrusion Detection System using Deep Learning Algorithms

Archana C   
Dept. of Computer Science and Engineering  
PES UniversityBengaluru, India

T P Nandini  
Dept. of Computer Science and Engineering  
PES UniversityBengaluru, India

Chaitra H P  
Dept. of Computer Science and Engineering  
PES UniversityBengaluru, India

Dr. Sivaraman Eswaran  
Dept. of Computer Science and Engineering  
PES UniversityBengaluru, India

Khushi M  
Dept. of Computer Science and Engineering  
PES UniversityBengaluru, India

Prof. Prasad Honnavalli  
Dept. of Computer Science and Engineering  
PES UniversityBengaluru, India

***Abstract*—****The internet traffic and the number of attempts at malicious access to any organization have increased over the years. The attacker can take advantage of this and over flood the organization with dummy traffic and make the systems unresponsive. The current existing models of NIDS have a good prediction rate but somehow have high FPR. So the proposed model offers to use a DL algorithm to reduce the FPR. The model prefers DL algorithm over ML algorithm because DL learns new features which majorly define a model on its own without any human intervention, whereas ML requires the user to feed the important features to be considered by the model. Also with the increase in the internet traffic world wide, the influx of network traffic into any system has also been a major spike over the years. The proposed system offers a scalable solution using DL algorithms to increase the responsiveness of the NIDS during high loads, hence increasing the reliability. The experimental results shows that DNN with five hidden layers achieved the best accuracy of 94.12% and least accuracy of 88.75% achieved by a LSTM with two layers whereas highest achieved from ML algorithm is 86% using Random Forest.**

***Keywords—Deep Learning, Network Intrusion Detection System, Host Intrusion Detection System, Machine Learning, False Positive Rate***

# INTRODUCTION

Network Security has become an important aspect because computer networks are used extensively in all the fields. Various sensitive user data is handled by networks which are prone to various kinds of attacks. There are a variety of tools like firewall, antimalware, antivirus etc. used to detect and exploit attacks. These attacks are diverse and are evolving with sophisticated algorithms which makes it undetectable by traditional tools leading to data breach. These kinds of malicious attacks pose security challenges which makes it important to have a reliable and flexible IDS.

An Intrusion Detection System (IDS) is a software application or a device that monitors a network for any suspicious activity and an alert is issued to the management system on discovery of such abnormal events. IDS can be classified into the following categories: Network Intrusion Detection system (NIDS) monitors the network traffic flowing in and out of the entire system to detect traces of malicious activity. It is usually placed at the crucial points where all the traffic traverses. Host Intrusion Detection System (HIDS) monitors a particular host’s/system’s important OS files, application logs, system calls etc. for malicious activities.It can be installed on any individual devices such as desktop, servers. IDS uses the following methods to identify the attacks: Signature based Intrusions can be detected by classifying access to the systems into predefined signature, patterns, malicious activities. Signatures need to be updated regularly and it is accurate in finding known types of attacks. Anomaly based analyses the normal behaviour of the network and alerts when there’s a deviation.It is used to detect unknown attacks. Hybrid based can be achieved by using a combination of signature based and Anomaly based and has low positive rate and high accuracy.

Anything that violates the CIA triad must be caught by the Intrusion Detection System. NIDS usually has a large false positive rate and classical ML classifiers have high computation rate with high FPR because they learn the characteristic of simple TCP/IP features locally whereas, Deep learning is a complex subnet of machine learning that learns hierarchical feature representations and hidden sequential relationships by passing the TCP/IP information on several hidden layers making it reduce the false positive rate. Since the DL algorithms require extensive computational resources, the advent of GPUs and cloud‐based platforms can ease the way for the implementation of DL‐based methods.

To enhance the performance of IDS many studies are focusing on applying Machine Learning (ML) techniques both supervised and unsupervised. Some of the ML algorithms used are Naive bayes, Support Vector Machine (SVM), Random forest, Decision Trees, K-Nearest Neighbour (KNN) , Decision Trees etc. IDS built using ML have achieved a high accuracy with small amounts of input data. However if the large dataset is used it’s very time consuming and has high latency for large dataset.There are some issues with these classical algorithms. ML techniques fail in multi class classification because of more number of features. There are difficulties such as overfitting, inducing high bias due to redundant or irrelevant features. Most of the studies have used old publicly available dataset which doesn’t cover today's wide range of attacks. There is also the issue of scalability, where the model can take a long time to respond if it is bombarded with requests.

So the proposed idea is to use a deep learning based Network Intrusion Detection System which will be hosted on the cloud to classify malicious activities. The proposed method trains a model using DL algorithms that learns about the characteristics of a malicious activity using a pre-existing dataset and with each detection improves and optimizes the model. It will be hosted on the cloud and be configured as microservices which can be scaled depending on the network traffic. The proposed model uses the UNSW-NB15 dataset for NIDS. The signature based IDS has evolved over the years into anomaly based IDS to detect anomalous behaviour posed by the attacker. However, these attacks are always evolving in quality and both quantity. With increase in the internet traffic world wide , the influx of network traffic into any system has also seen a major spike over the years. The attacker can take advantage of this and over flood the system with dummy traffic and make the systems unresponsive.

In such environments having a single running instance of an IDS to alert the organization about a potential attack in real-time can be quite challenging. The proposed system offers a scalable solution using DL algorithms to increase the responsiveness of the IDS during high loads, hence increasing the reliability. The IDS can be scaled up or down depending on the traffic. This way there won’t be concerns regarding investment on infrastructure and wastage of resources when the load is high and low respectively.

The model prefers DL algorithm over ML algorithm because DL learns new features which majorly define a model on its own without any human intervention, whereas ML requires the user to feed the important features to the model. Moreover, DL performs the optimization like deciding on the weights for each input on its own to improve the accuracy of the model. The proposed model has three instances on the cloud for database operations, classification

and re-training of the model in frequent time-intervals respectively.

1. LITERATURE STUDY

Authors have presented the model which is designed and implemented using Recurrent Neural Network[1]. Performance of the model is studied for both multiclass classification and binary classification and how the accuracy is impacted by different combinations of the learning rate and number of neurons. NSL-KDD dataset is used to compare the performance of multiclass classification using different Machine Learning algorithms like Naive Bayes, Support Vector Machine, multi layered perceptron. There are two parts in this Recurrent Neural Network Intrusion Detection System implementation namely Forward Propagation and Backward Propagation. Updating the weights by passing the accumulated residuals is the responsibility of Backward Propagation. Evaluation is done through metrics like Accuracy which defines true positive and true negative over all the four values namely true positive, true negative, false positive, false negative. So, Binary Classification achieved the highest accuracy with 0.1 learning rate and number of hidden nodes as 80. Multi-class classification achieved the highest accuracy with 0.5 learning rate and number of hidden nodes as 80.

A Deep Neural Network model with one input layer, five hidden layers and an output layered architecture on the KDD99 dataset as the hybrid intrusion detection system approach is presented by Vinayakumar *et al.*[2]. ReLU is used as an activation function, stochastic gradient as optimization method and cross entropy as loss function. The challenges faced in this paper are the dataset preprocessing, algorithm development, efficiency and scalability. The Deep Neural Network is trained using backpropagation. All layers are fully connected and the number of neurons in the hidden layers is varied depending on the dataset. The last layer is the classifier layer. Binary classification uses sigmoid as an activation function. Multi-class classifiers use sigmoid as it’s activation function. Batch normalization and regularization were used to avoid overfitting. For KDDCup 99 and NSL KDD datasets, most of the Deep Neural Network topologies showed training accuracy upto 93% and False positive rate was close to 0% in many cases.

The main challenge faced by Stefan Dlugolinsky *et al.* [3] is the harmony between large scale data processing and Deep Learning techniques. Most of the current intrusion detection systems are capable of responding to network activities in near real time. But these are largely reactive solutions.However, there is a lack of a proactive solution. This paper proposes a proactive solution. The dataset used in an existing system ZEEK/Bro real data. Training data has less time index than testing data. Big data stack comprising Apache Spark, Arrow and Parquet is used to cope with the offline development phase of the model. This model has three components namely intrusion detection system module,data processing module and proactive forecasting module. This model uses ZEEK/Bro net IDS. Data processing module manages offline and online processing and feature engineering. Data is stored in order of timestamp. Feature engineering involves feature selection and extraction, data cleaning, and data transformation. Interaction with the data repository and Model building is controlled by this module. Proactive forecasting Module produces Deep Learning Model using Machine Learning methodologies in the development phase. There is no particular winning model, Long Short term memory (Gradient recurrent Unit) derivatives are the best candidates in terms of quality and performance.

The model by coupling filter based feature selection with Feed Forward Deep Neural Networks (FFDNNs) algorithm is proposed by Yanxia Sun *et al.*[4]. NSL-KDD dataset is used. They have compared their results with the Machine Learning algorithms namely Support Vector Machine, Naive Bayes, K-Nearest Neighbor and Decision Tree. In Feature Extraction, authors have used a filter model which depends on the data’s inherent nature rather than the classifier used. First step of the architecture consists of the separation of raw data. Split the dataset into evaluation and reduced training sets and have used an Evaluation dataset for validating the training process. The test contains different data from the training and validation sets. Process of two-way normalization and feature transformation on the data constitutes the second step of this model. The last step of the architecture is training and testing of the models using FFDNNs and the feature extraction unit-feedforward deep neural network information gain(IG) based algorithm is used to rank all the features which is a function of the feature extraction unit. Training is done in the following steps: forward propagation using ReLu. Backpropagation of errors (stochastic descent) and updating the weights and biases using cost function. For the multiclass and binary classification problems, the FFDNNs models with full and a Feature Extraction Unit-reduced feature space achieved superior performance compared to other Machine Learning classifiers.

A model which uses a hybrid of Convolutional Neural Network and weight-dropped Long Short term memory is presented by Ahmad alsanad *et al.*[5]. Hybrid Deep Learning model is used to efficiently detect network intrusions based on Convolutional Neural Network and a weight-dropped Long short term memory network. The deep Convolutional Neural Network is used to extract necessary and meaningful features from the Intrusion Detection System big data and weight dropped long short term memory to retain long-term dependencies among extracted features, This is done to prevent overfitting on recurrent connections. Authors have used the deep convolutional neural network to extract the meaningful features from the network data traffic, exploiting its speed due to its weight sharing property and have used the Long short term memory network to retain long-term dependencies among extracted features and to avoid the gradient vanishing problem and have also used the drop-connect regularization technique on the hidden-to-hidden weight matrices within the Long short term memory to avoid the overfitting problem. Optimization is done based on the trial and error. Dataset used is UNSW-NB15 (has over 100GB of real time network traffic data) and ISCX2012. The accuracy of the model was calculated to be 96.97%. The average execution time of the model is very low, that is an average of 0.002383ms for one instance.This makes it more effective for real-time intrusion detection systems. This shows that this model can be employed on bigger datasets and be used in real time Intrusion Detection Systems too.

Khalil Dajani *et al.*[6] have proposed an intrusion detection system based on the convolutional neural network to reinforce the security of the internet. CICIDS2017 dataset is used in this study. The technique used is Convolutional Neural Network with Relu activation function. The reason for using Convolutional Neural Network is it could potentially learn the much more complex characteristics of some modern cyberattacks, which other neural networks find difficult to capture. Convolutional Neural Network uses shared weights with fewer parameters to optimize with less time overhead. slightly decreases the degree of freedom which helps to avoid overfitting. Convolutional Neural Network can achieve better generalization on the classification of cyberattack samples enabling it to potentially detect innovative attacks. The model achieved overall accuracy of 98% with False positive rate 1.02. The proposed model provides better multi-class classification results than some other recently proposed Convolutional Neural Network based Intrusion Detection System because of the innovative data pre - processing method. The model not only offers protection against the known attacks with a high Decision Rate and low False Alarm Rate, but also shows potential at detecting innovative attacks. Some sample augmenting techniques that are used in the Convolutional Neural Network models to solve the issue of insufficient samples of minority attacks.

A hierarchical Convolutional Neural Network + Recurrent Neural Network called LuNet, is proposed by Hui Guo *et al.*[7] to detect intrusions on a large scale network. Authors have used NSL-KDD and UNSW-NB15 dataset. Reasons for using Recurrent Neural Network is Recurrent Neural Network remembers the previous features or elements for predicting the future element. Recurrent Neural Network creates a loop which helps in the persistence of these types of information. Reasons for using Hybrid Architecture is to avoid the information loss due to different learning focuses of Convolutional Neural Network and Recurrent Neural Network, they have synchronized both Convolutional Neural Network and Recurrent Neural Network to learn the input data at the same granularity. It uses Convolutional Neural Network to learn spatial features in the traffic data and Long Short term memory for temporal features. Compared with other state-of-the-art techniques, this can significantly improve the validation accuracy and reduce the False positive rate for Network Intrusion Detection System.

Farrukh Aslam Khan *et al.*[8] have used a stacked Autoencoder (AE) based two staged Deep Learning model by using soft-max for classification. The initial stage is used for binary classification where it classifies a network traffic as attack or benign and gives a probability value. The probability value is fed as an input to the second stage along with the initial features to classify into different classes of attacks. The proposed model consists of two stages. Both the stages use Deep Neural Network because of its high performance and speed which makes it suitable for real time classification. It comprises a Deep Stacked Autoencoder with two hidden layers and a softmax layer on top for classification. This model is evaluated on two benchmark datasets: UNSW-NB15 dataset and KDD99 dataset. For UNSW-NB15 the model achieved 89.711% accuracy and 0.1018 False Alarm Rate in the initial stage by extracting 10 abstract features and 89.13% accuracy with 0.7495 False Alarm rate in the final stage with 5 abstract features. This paper shows that Rather than using classical linear transformation algorithms for feature extraction using Deep Learning techniques which are able to extract abstract features is a better option.

In order to enhance the detection rate of the intrusion detection system, [9] mainly focuses on false positive and false negative performance metrics. Dataset used in this paper is the KDD dataset. The implemented experiments demonstrate that the decision table classifier achieves the lowest value of false negative while the random forest classifier achieves the highest average accuracy rate.

A genetic algorithm based exhaustive search and fuzzy C - means clustering algorithm is used in Genetic Convolutional neural network[10]. The main aim is to select an improved feature subset which can be used for Network Intrusion Detection System. They have presented three layers for constructing features using genetic algorithms, CNN and Fuzzy C - means to improve the final detection performance. Bagging classifier is used to select the CNN structure. The dataset used for training and evaluating their model is NSL - KDD. This paper proposes a genetic algorithm based exhaustive search and fuzzy C - means clustering algorithm to select an improved feature subset which is used for Network Intrusion Detection System. Total of 33 features are selected from the model 28 from Genetic Algorithm and 5 features from Fuzzy C - means clustering. Out of three CNN models selected from the model selection phase 98.24% is the highest accuracy and 0.52 FPR is achieved. The proposed model has two parts namely feature extraction and classification. Feature selection is an important aspect for building an Intrusion Detection System for real time usage. This paper has employed a combination of different algorithms to extract best possible features.

1. DATASET

The dataset used is UNSW-NB15[11] and it was collected at the Australian Center for Cyber Security(ACCS) in 2015 by the research group of cyber security. It is a network intrusion dataset. The raw network packets of the UNSW-NB15 data set are created by the IXIA PerfectStorm tool. The training set consists of 175,341 records. Testing set contains 82,232 records from the different types, attack and normal. UNSW-NB15 contains approximately 2.5 million records.

Using two different simulation periods of 15 hours and 16 hours, the dataset was generated. This dataset consists of both synthesized attack activities and real modern normal activities and of the network traffic. TCP dump is utilised to capture 100GB of the raw traffic. It contains nine families of attacks, namely, Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode and Worm.

The columns in the dataset are enumerated as shown in Table 1.

Table 1: Description of features in UNSW-NB15 dataset

|  |  |
| --- | --- |
| **NAME** | **DESCRIPTION** |
| scrip | source IP address |
| sport | source port number |
| dstip | destination ip address |
| proto | transaction protocol |
| dur | record total duration |
| Sload | source bits per second |
| Dload | Destination bits per second |
| Spkts | Source to destination packet count |
| Dpkts | destination to source packet count |
| dsport | destination port number |
| attack\_cat | the name of each attack category |
| Label | 0 for normal and 1 for attack records |

# 

# DESIGN AND IMPLEMENTATION

The proposed architecture is to build the Intrusion Detection System model as microservices on the cloud. The Intrusion Detection System will classify the input data instance as normal or different types of attacks based on the features of the initial stage. Using Neural Network models like Convolutional Neural Network, Long Short Term Memory or hybrid models with appropriate activations at each layer, the model will perform better than classic Machine Learning algorithms.

Fig 1 represents the proposed architecture.Microservice is a service-based application development. Here, big applications will be divided into smaller independent service units. It is the process of implementation of SOA (service oriented architecture) by dividing the whole application as a collection of the interconnected services in which each service will serve only one business need.

The model hosted on cloud will be such that, there will be three instances which will run the services for training, testing and storing the data as docker containers. We are using Docker Containers, REST APIs over postman and load balancer.

Docker is a tool which is designed to make the developers easier for creation, deployment, and running applications by using containers. Containers are the ones which allow a developer to package up an application which contains all of the parts it needs, namely libraries and other dependencies, and deploy it as one package. REST stands for REpresentational State Transfer. It is a kind of architecture which is based on web standards and it uses HTTP as its Protocol. REST revolves around a resource in which every component is treated as a resource and a resource is accessed by a common interface using HTTP as its standard methods. For testing these REST APIs, we are using postman. Postman is a software testing Application programming Interface platform which is used to build, test, design, modify, and document APIs.

Across multiple targets, the incoming traffic is automatically distributed by the load Balancer. EC2 instances and IP addresses are the examples of incoming traffic. It monitors the registered targets’ health, and it routes the traffic only to the healthy targets. It scales the load balancer as the incoming traffic changes over time. Vast majority of workloads are automatically scaled by the load balancer. For clients, it serves as a single point of contact. It increases the availability of our application. We can add one or more listeners to the load balancer.

Fig 2 represents the Flowchart of Methodology. The model will get the incoming traffic features from the training dataset. The traffic is fed to the neural network model which extracts features and trains the model to classify traffic as attack or benign. The testing dataset records are used to verify the performance of the model such as accuracy, precision etc.

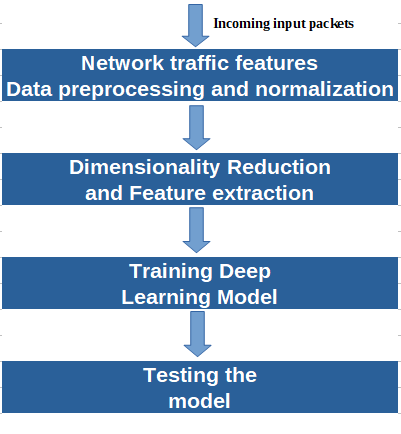


Fig 2: Flowchart of Methodology

A two stage model is used in this study: Initial stage is to extract minimal meaningful features which best represent the data. Next stage is to classify the input data instance as normal or different types of attacks based on the features of the initial stage. Neural Networks like Deep Neural Networks and some other models are used with appropriate activation functions at each layer. During optimization, hyperparameter tuning like learning rate, number and depth of hidden layers are used, number of epochs can be fixed after trial and experimentation. In an optimization algorithm, Learning Rate is a tuning parameter which determines the step size that needs to be taken at every iteration when we are moving towards a minimum of a loss function.

In between the input and output of the algorithm in regular neural networks, a hidden layer is located, where the function applies weights to the inputs and it directs those through an activation function as the output. Nonlinear transformations of the inputs need to be performed on the hidden layers which will enter into the network.

The training and testing are implemented as a flask app in localhost. Both can be accessed through REST APIs over Postman. Flask is considered as a popular Python web framework, it means that it is a third-party python library which is used for developing web applications. It is an API of Python that allows to build up web-applications.

Batch normalization is used to improve the accuracy. By normalizing the input of each layer in the network, not only on the input layer, Batch Normalization significantly reduces the training time. It is a technique which is used for very deep neural networks training which standardizes the inputs to a layer for each of the mini-batch. This will have the effect of stabilizing the learning process. This dramatically reduces the number of training epochs which are required to train deep networks. Batch Normalization is usually added after the activation function of the output layer or before the activation function of the input layer.

The Intrusion Detection model will be hosted on the cloud (EC2 instances). There are three microservices: Training, Testing and Database. The testing/classification of packets

can be done by sending REST APIs to the testing microservice. The testing microservice will in turn send this

now classified packet into the database to ensure “real time”

data is being stored. There will be weekly updates of the

training microservice on the data in the database. We have hosted the Network Intrusion Detection System model on EC2 instances as microservices. We have configured the load balancer and docker containers accordingly and also accessed the model through REST APIs. Depending on the amount of data to be processed by the docker containers , the number of their running instances will be either scaled-up or scaled-down accordingly.

Initially we have tried with Machine Learning algorithms for better comparison. The machine learning models implemented are Logistic Regression, Naive Bayes, K - Nearest Neighbor, Random Forest, Decision Trees and Adaboost. The models have been built using functions from the sklearn library. First step is preprocessing the dataset which will be used for model implementation. All the categorical data is converted to numerical using the sklearn label\_encoder function. To remove all the correlated columns pandas correlation function with pearson correlation coefficient is used. We were able to reduce 42 features to 29 which can highly improve the performance of the model.

Some of the basic Machine Learning classifiers are implemented such as Logistic Regression, Random Forest etc. Using logistic function, by estimating the probabilities, Logistic regression measures the relationship between more than one independent variable and the categorical dependent variable. Accuracy of this model came out to be 70%. Naive Bayes basic assumption is that the features are independent which may not be true in many cases and reduces the performance. Accuracy of this model came out to be 75%. The K - Nearest Neighbor algorithm is a simple, supervised machine learning algorithm which can be used in solving both regression and classification problems. KNN might result in a low prediction stage for larger dataset. Accuracy of this model came out to be 78%. Decision trees are using multiple algorithms to decide to split a single node into two or more sub-nodes. The major limitation of the Decision tree is that if we change a small amount of data, then it will lead to a large change in the structure of the decision tree. Accuracy of this model came out to be 86%. Extending the Decision Trees, Random Forest is implemented which will help in reducing overfitting in the decision trees which in turn helps in increasing the accuracy. Accuracy of this model came out to be 86%.

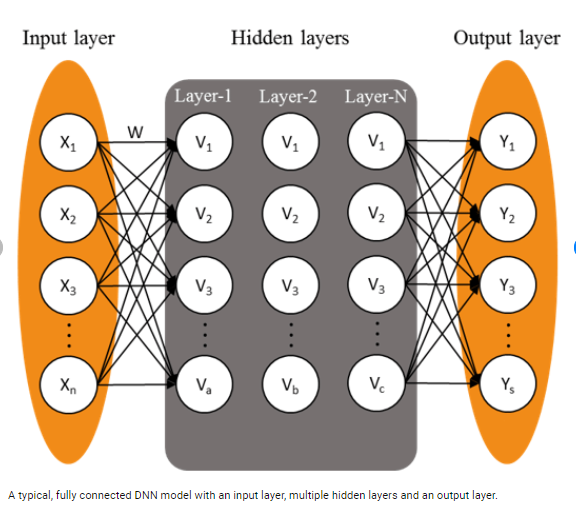
Deep Learning models are implemented. Initially, Long Short Term Memory with different layers are used, with ReLu as an activation function and using stochastic gradient descent as the optimization function. Long Short - Term Memory is an improvement on Recurrent Neural Network. Recurrent Neural Network suffers from vanishing gradient problems so the layers won’t learn much and will forget the information it has learned. When we have a long sequence to process and predict the output, the Recurrent Neural Network may forget some of the starting important information hence suffers from short memory. It contains three gates namely Forget gate, Input gate and Output gate. The rectified linear activation function(ReLU) is a piecewise linear function. It will output the input directly if it is positive, or, it will output zero. Allowing models to learn faster and perform better, it overcomes the vanishing gradient problem. The results of the LSTM with different hidden layers are mentioned in Table 3.

Gated Recurrent Units are improvised versions of standard recurrent neural networks. To solve the vanishing gradient problem of a standard Recurrent Neural Network, Gated Recurrent Unit uses so called update gate and reset gate. These two gates are nothing but two vectors which decide what information should be passed to the output. Gated Recurrent Unit is similar to a Long Short term memory. Gated Recurrent Unit will not use cell state, instead it will use the hidden state to transfer information. The results of the GRU with different hidden layers are mentioned in Table 3.

CNN has one or more convolutional layers which are used mainly for classification, image classification segmentation and also for other auto correlated data. A convolution is basically sliding a filter over the input. CNN is made up of a set of neurons, where each layer is fully connected to all neurons in the layer before. Fully - connected layer is the output layer which represents the predictions. The result of the CNN is mentioned in Table 3.

A Deep Neural Network is a kind of Artificial Neural Network with a number of hidden layers between the input and the output layers. Complex non - linear relationships can be modeled by Deep Neural Networks. The aim of Deep Neural Networks is to receive a set of inputs, perform progressively complex calculations on them, and give output to solve real world problems like classification.

Deep Neural Networks use the gradient descent method for optimizing the network and minimising the loss function. Every node in the output and the hidden layers has their own classifiers. For further activation, inputs are taken by the input layer and passed on its scores to the next hidden layer and this continues till the output is reached. This kind of progress from the left to the right from the input to the output in the forward direction is called forward propagation. Every node in the Deep Neural Network is a perception that mimics a neuron in a biological neural network. The prediction of accuracy depends on its weights and biases. The result of the DNN is mentioned in Table 3.

 Fig 3: Deep Neural Network model

# EXPERIMENTAL RESULTS

Cloud Implementation is done by creating an AWS EC2 instance with two containers using flask framework for making API calls. And compared the results of different Machine Learning and Deep Learning Models on the UNSW-NB15 dataset.

Table 2:Accuracy of different Machine Learning Algorithms

|  |  |
| --- | --- |
| **MACHINE LEARNING ALGORITHMS** | |
|  | **Accuracy** |
| **Logistic Regression** | **70%** |
| **Naive Bayes** | **75%** |
| **K - Nearest Neighbour** | **78%** |
| **Decision Tree** | **86%** |
| **Random Forest** | **86%** |

Table 2 shows the accuracy of different Machine Learning Algorithms. Among the different Machine Learning models implemented, Random Forest and Decision Tree achieved the highest accuracy of 86%.

Table 3:Accuracy of different Deep Learning Algorithms

|  |  |
| --- | --- |
| **DEEP LEARNING ALGORITHMS** | |
|  | **Accuracy** |
| **LSTM with 1 hidden layer** | **88.81%** |
| **LSTM with 2 hidden layers** | **88.75%** |
| **LSTM with 3 hidden layers** | **88.78%** |
| **GRU with 1 hidden layer** | **88.79%** |
| **GRU with 2 hidden layers** | **88.79%** |
| **GRU with 3 hidden layers** | **88.76%** |
| **CNN with 2 convolutional layers** | **93.11%** |

Table 3 shows the accuracy of different Deep Learning Algorithms. From Table 3, it seems that adding more number of hidden layers to LSTM and GRU did not improve the accuracy whereas that was not the case with DNN. The LSTM and GRU achieved around 88.78%, 88.79% accuracy respectively and Deep Neural Network with five hidden layers achieved the highest accuracy of 95.02%.

Table 4: DNN with different hidden layers

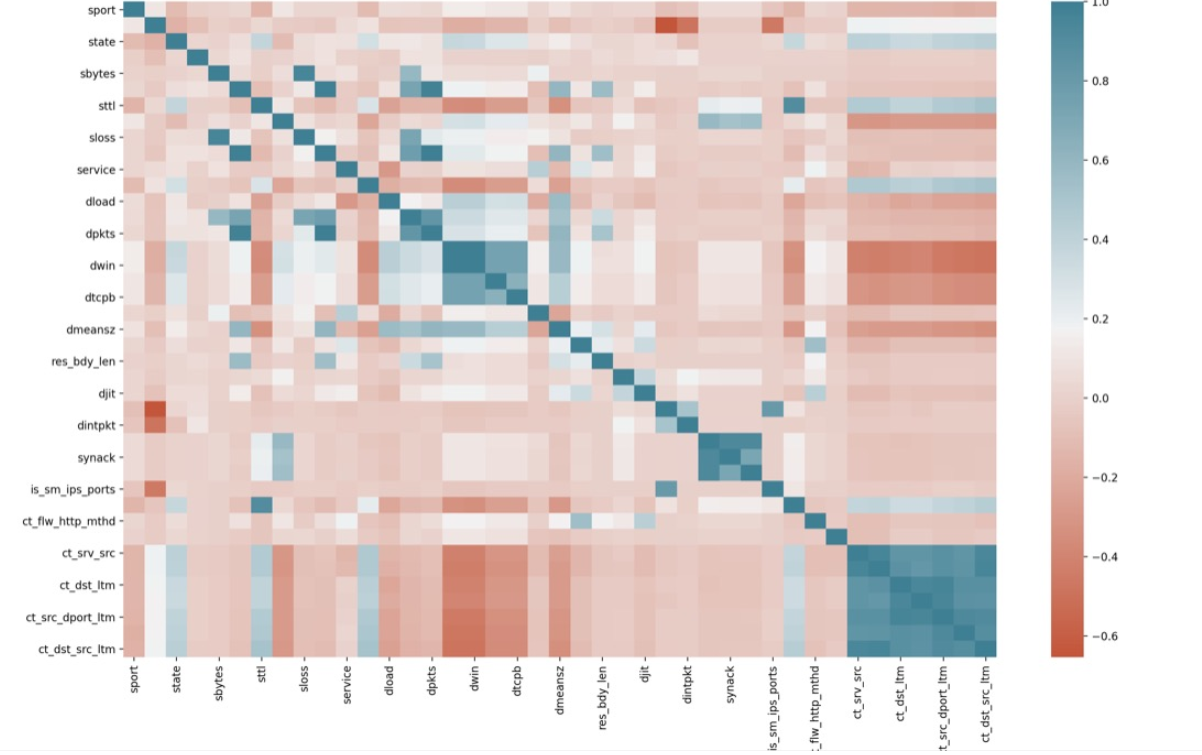
|  |  |  |
| --- | --- | --- |
| **DNN WITH DIFFERENT HIDDEN LAYERS** | | |
|  | **Accuracy** | **PRECISION** |
| **1 hidden layer** | **93.71%** | **84.96%** |
| **2 hidden layers** | **93.81%** | **72.90%** |
| **3 hidden layers** | **94.43%** | **85.32%** |
| **4 hidden layers** | **95.02%** | **85.92%** |

Table 5: Confusion Matrix for the DNN model

|  |  |  |
| --- | --- | --- |
|  | **Predicted benign** | **Predicted attack** |
| **Actual benign** | 196299 (TN) | 3511 (FP) |
| **Actual attack** | 7355 (FN) | 21440 (TP) |

Table 4 represents the confusion matrix for the Deep Neural Network model. True positive tells that the traffic is predicted as an attack correctly by the model. True negative tells that the traffic is predicted as a benign correctly by the model. False positives tell that the model predicted an attack but they are benign. False negatives tell that the model predicted as benign but actually they are the attacks.

Comparing Table 2 and 3, it is noted that using Deep Learning over Machine Learning appreciably increases the performance of the model for larger datasets.



# CONCLUSION

Instances of an AWS are used to run the model on the cloud and configure the docker containers accordingly. We have implemented different Deep Learning Models like Deep Neural Networks, Long Short Term Memory and Gated Recurrent Unit. The Performance Metrics are calculated for the proposed model and comparing it with the other studies. The Deep Neural Network model achieved good accuracy. Using batch normalization has significantly improved the accuracy of the model. This model can offer real time detection provided, the input to the model for testing is given according to the format specified and follows necessary. A proactive model which can detect future attacks may be implemented in future for prior detection of attacks.

##### References

1. Chuanlong Yin, Yuefei Zhu, and Xinzheng He, “A Deep Learning Approach for Intrusion Detection using Recurrent Neural Networks”, 2017
2. R. Vinayakumar, Mamoun Alazab, K. P. Soman, Prabaharan Poornachandran, Ameer Al-nemrat, Sitalakshmi Venkatraman, “Deep Learning Approach for Intelligent Intrusion Detection System'', 2019
3. Giang Nguyen, Stefan Dlugolinsky, Viet Tran, Alvaro Lopez Garcia, “Deep Learning for Proactive Network Monitoring & Security Protection”, 2020
4. Sydney Mambwe Kasongo, Yanxia Sun “A Deep Learning method with filter based feature extraction for wireless intrusion detection system”, 2020
5. Mohammed Mehedi Hassan, Abdu Gumaei, Ahmed Alsanad, Majed Alrubaian, Giancarlo Fortino, “A hybrid Deep Learning model for efficient network intrusion detection in big data environment”, in 2020
6. Samson Ho, Saleh Al Jufout, Khalil Dajani, Mohammed Mozumdar, “A Novel Intrusion Detection Model for Detecting Known and Innovative Cyberattacks Using Convolutional Neural Network”, 2021
7. Peilun Wu, Hui Guo, “A Deep Neural Network for Network Intrusion Detection”, 2019
8. Farrukh Aslam Khan (Senior Member, IEEE), Abdu Gumaei , Abdelouahid Derhab, and Amir Hussain , “A Novel Two-Stage Deep Learning Model for Efficient Network Intrusion Detection”, IEEE 2019.
9. Mohammad Almseidin, Maen Szilveszter Kovacs and Mouhammd Alkasassbeh, “Evaluation of Machine Learning Algorithms for Intrusion Detection Systems”
10. Minh Tuan Nguyen, Kiseon Kim, “Genetic Convolutional Neural Network for intrusion detection systems”, 2020
11. UNSW-NB15 Dataset: unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-Datasets/

[12]https://www.researchgate.net/figure/A-typical-fully-connected-DNN-model-with-an-input-layer-multiple-hidden-layers-and-an\_fig1\_334379585

## 