Anomaly Detection in IoT network Using Deep Neural Networks

IT-801 MAJOR PROJECT-II THESIS

SUBMITTED IN PARTIAL FULFILMENT OF THE
REQUIREMENTS FOR THE AWARD OF THE DEGREE OF

MASTER OF TECHNOLOGY
IN
INFORMATION SYSTEMS

Submitted by:

Deepanshu Singhal
2K19/ISY/07

Under the supervision
of
Dr. Jasraj Meena
Assistant Professor

DEPARTMENT OF INFORMATION TECHNOLOGY
DELHI TECHNOLOGICAL UNIVERSITY (Formerly
Delhi college of Engineering)
Bawana Road, Delhi-110042

(2019-2021)

CANDIDATE'S DECLARATION

I, Deepanshu Singhal, Roll No. 2K19/ISY/07 student of Master of Technology, Information Systems, hereby declare that the Research Problem Formulation titled "**Anomaly Detection in IoT network Using Deep Neural Networks**" which is submitted by me to the Department of Information Technology, Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of the degree of Master of Technology is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of any Degree, Diploma Associateship, Fellowship, or other similar title or recognition.

Deepanshu

Place: New Delhi

Deepanshu Singhal

Date: June 27, 2021

ANNEXURE - I

DECLARATION

I Deepanshu Singhal 2k19/ISY/07 hereby certify that the work which is presented in the Research Problem Formulation ISY5202 entitled "**Anomaly Detection in IoT network Using Deep Neural Networks**" in fulfillment of the requirement for the award of the Degree of Master of Technology in Information Systems and submitted to the Department of Information Technology, Delhi Technological University, Delhi is an authentic record of my own, carried out during a period from January to May 2020, under the supervision of **Dr. Jasraj Meena.**

The matter presented in this report has not been submitted by me for the award of any other degree of this or any other Institute/University. The work has been published/accepted/communicated in SCI/ SCI expanded/SSCI/Scopus indexed journal OR peer-reviewed Scopus indexed conference with the following details:

Deepanshu

Student(s) Roll No.: 2K19/ISY/07
Name and Signature: Deepanshu Singhal

SUPERVISOR CERTIFICATE

To the best of my knowledge, the above work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere. I, further certify that the publication and indexing information given by the students is correct.

03/07/2021
Dr. Jasraj Meena
Supervisor Name and Signature

Place: Dausa, rajasthan

Date: 03/07/2021

ACKNOWLEDGEMENT

I am very thankful to **Dr. Jasraj Meena** (Assistant Professor, Department of Information Technology) and all the faculty members of the Department of Information Technology at DTU. They all provided me with immense support and guidance for the project.

I would also like to express my gratitude to the University for providing us with the laboratories, infrastructure, testing facilities, and environment which allowed us to work without any obstructions.

I would also like to appreciate the support provided to us by our lab assistants, seniors, and our peer group who aided us with all the knowledge they had regarding various topics.

Deepanshu Singhal
2K19/ISY/07

Abstract

Internet of things (IoT) is a continuous flow of information among many small power embedded machines that work based on the Internet to link with one another. It is anticipated that the IoT will be extensively installed and will find usage in numerous areas of life. The need for IoT has recently engrossed vast attention, and administrations are keen about the data generated from the devices by arraying such systems. On the conflicting side, IoT has several safety and confidentiality worries for the users that put a break on its rise. Outbreak and irregularity uncovering in the Internet of Things (IoT) setup are some of the mounting concerns in the area of IoT. With bigger usage of the Internet of things(IoT) everywhere, threats and outbreaks are also increasing. In our paper, we have used machine learning and deep neural models to detect different kinds of outbreaks and abnormalities on IoT machines. Results were based on the estimation of efficiency given by precision, accuracy, f1 score, recall, and ROC Curve. Our model resulted in 99.43% accuracy using a deep neural network.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS, ABBREVIATIONS AND NOMENCLATURE

IoT	Internet Of Things
ML	Machine Learning
LR	Logistic regression
SVM	Support Vector Machine
DS2OS	Distributed Smart Space Orchestration System
ROC	Receiver Operating Characteristic Curve
DoS	Denial of Service
D.P.	Data Type Probing
M.C.	Malicious Control
M.O.	Malicious Operation
S.C.	Scan
S.P.	Spying
W.S.	Wrong Setup
N.L.	Normal

Chapter 1

INTRODUCTION

In the modern era, the propagation of the Internet of Things has been extensively growing in this world. The figure of linked loT machines had already touched 27 billion in 2017 and the numbers will grow exponentially on request of the market, so it is been anticipated to touch about 125 billion in 2030 [1]. With the intensifying urgency and development in the Internet of Things (IoT), the IoT prototypes are getting complex each day. People are getting habituated to data-focused structure, and this is taking the study on ML techniques through IoT. There are many systems built on these two techniques that are currently used in many areas of human life presently. In medicine, it is used in disease detection with the help of X-Ray, interpretation of ECG, pattern matching in genomic information. Moreover, these systems are useful in these services. The increasing complication in IoT substructures is nurturing uninvited susceptibility to its schemes.

A usual message occurrence in the confined system is restricted to confined nodes or minor confined domains but the outbreak in IoT structure surges in excess to the greater area and shows shattering results on IoT data[2]. Therefore, it is necessary to have a safe IoT setup for safety from cyberattacks. The safety steps we are using turn into susceptibility to the susceptibility of IoT machines concerning time. Mostly for some investors and business persons, information is the currency for their profitmaking. Vulnerability in IoT devices makes an entrance for the invader to gather personal material from any significant union [3]. In signature built [4] technique, outbreaks and irregularity are formerly kept in a databank. Furthermore, this structure is verified at specific interval breaks alongside the databank. On the other hand, this procedure produces overhead in handling, and it can be exposed to unfamiliar concerns. The data analysis-centered method works efficiently compared to other methods and the difficulty that arises from unidentified threats can be overcome. Therefore, we will be using data analysis grounded methods.

This research aims to develop a smooth, secure, and trustworthy IoT constructed setup that can experience its weakness, have a safe firewall in opposition to all attacks, and improve according to the situation robotically. We are using a Machine learning and Deep Neural Network constructed explanation that can sense and protect the device from an anomalous state that it can experience.

1.1 MOTIVATION

The concept of the noticeable Internet-of-Things (IoT) concept is intended to develop the value of current life. IoT resolutions, for instance, meaningfully improve the day-to-day habits of old and disabled folks, thus swelling their self-government and confidence. Implantable and wearable IoT gadgets record and extract necessary records to allow immediate emergency notifying to upturn patient's odds of existence. This developing expertise is also being leveraged to decrease reply epochs in responding to sudden fitness events such as unexpected toddler death syndrome during sleep.

The security concern impacting the Internet-of-Things (IoT) archetype has lately concerned noteworthy responsiveness from the research community. To this end, numerous studies were put onward talking about numerous IoT-centric issues comprising threat modeling, intrusion detection systems, and emerging technologies. The main objective of the system is to detect vulnerability so that our system can be smart, secured, and reliable. Here, Machine Learning and Deep Neural Networks-based explanations are suggested which can sense and defend the device when it is in the anomalous state.

1.2 PROBLEM STATEMENT

The chief aim of this project is to contribute to an understanding of different machine learning and neural network algorithms to classify the categories of the attack that occurs in an IoT network. We proposed a system that can detect eight different types of attacks.

1.3 THESIS STRUCTURE

This thesis is organized as follows:

- Introduction (Chapter 1): This chapter highlights the importance of IoT and the threats IoT is facing.
- Background (Chapter 2): This chapter discusses the previous work done in this field and the technologies used in this thesis.
- Proposed Methodology (Chapter 3): This chapter discusses the dataset, the process used to develop the proposed model.
- Result and Discussion (Chapter 4): This chapter discusses the results obtained by implementing different algorithms.
- Conclusion and Future Work (Chapter 5): This chapter discusses the conclusion of results obtained and the future work that can be done in this field.

Chapter 2

BACKGROUND

2.1 RELATED WORK

Pahl et al. [5] have primarily made a detector and firewall to detect abnormality in microservices of IoT sites. They used K-Means and BIRCH Clustering methods [9] for dissimilar micro-services in their work. 96.3% accuracy was achieved by their model.

An On and Off detector was proposed by Liu et al. [6] that detects a malicious node in industrial IoT sites. Here On and Off are the states of IoT networks. They used the light probe routing method to develop their model. They used Information Ratio to evaluate their model and achieved a .80 information ratio.

Diro et al. [7] proposed the discovery of attacks using fog-to-things architecture. They made two models, one was a shallow neural network and the other was a deep neural network. They focused to classify four different classes of attack and anomaly. They achieved 98.27% accuracy for the deep neural network model and 96.75% for the shallow neural network model respectively.

Anthi et al. [8] proposed an intrusion detection system for the IoT. Different Machine Learning classifiers were used to classify network scanning probing and Denial of service attacks. To produce the dataset, network traffic is taken for four successive days through Wireshark software.

Pajouh et al. [10] proposed an archetypal for intrusion detection established on a two layer dimension reduction and two tier classification module. The prototype was aimed to classify malicious events for example Remote to Local (R2L) attacks and User to Root (U2R). Component and linear discriminate analyses were used for dimension reduction. NSL-KDD dataset was used for the research. Naive Bayes and the Certainty Factor type of K-Nearest Neighbor were used for detecting malicious events with the two-tier classification module.

D'Angelo et al. [11] used U-BRAIN(Uncertainty-managing Batch Relevance-based Artificial Intelligence) on Real Traffic Data (from Fredrico II University of Napoli) and binary NSL-KDD dataset. The U-Brain is an active archetypal worked on numerous machines which can handle misplaced data.

Ukil et al. [12] conferred the discovery of irregularities in health system analytics constructed on IoT. An archetypal of cardiac anomaly detection by using a smartphone was also presented in this research. For abnormality discovery in health systems, IoT devices, biomedical signal analysis, medical image analysis, and predictive analytics, and big data mining were used.

Table1. Previous Related Work Description

Author and Year	Dataset	Classification Type	Method Used	Evaluation Scheme
Pahl.et.al.[5] 2018	Synthetic data	Multiple	K-Means algorithm Birch clustering	Accuracy=96.3
Liu.et.al.[6] 2018	Synthetic data	Two	Light Probing Algo	Information ratio=.80
Diro.et.al.[7] 2018	NSL_KDD data	Multiple	Neural Network.	Accuracy=98.2
Anthi.et.al.[8] 2018	Synthetic	Two	Naive Bayes method	N.A
Brun.et.al.[9] 2018	Own dataset	Multiple	Random Forest	N.A
Pajouh.et.al.[10] 2016	NSL_KDD	Two	K-means neighbor	Information ratio = 84.8

D'Angelo.et.al.[11] 2015	NSL_KDD
	Real-time traffic
		Two	U-BRAIN method	Accuracy = 94.1
				Accuracy = 97.4

Our Research	DS2OS	Multiple	Deep Neural Network	Accuracy= 99.4

2.2 KEY TERMS & CONCEPTS

2.2.1 Machine Learning Algorithm

Machine learning is the foremost important division of artificial intelligence. Its the knowledge of attaining systems to work without being explicitly automated [13]. It is so persistent nowadays that you undoubtedly use it loads of times a day. It's a branch of artificial intelligence that offers a machine to repeatedly study and progress from knowledge [14]. The course of knowledge starts with interpretations of data, for example, direct experience, or instruction, to search for patterns in records and predict improved results in the upcoming samples that we provide [15,16].

Supervised learning comprises each job in which the algorithm can use the input and output data. Input data is defined as the outside data that the algorithm can use, such as characteristic data and metadata, although output data are the precise tags of the class attribute [17].

Additionally, machine learning models are usually categorized by their primary learning approaches, which are recognized by the number of implications the system can achieve. They are classified as:

- Rote Learning defines the approach that old-style systems use. No inference is performed and it’s the responsibility of the programmer to directly implement all their knowledge because the model is not capable to learn or make any conclusions from the specified data[18].
- Learning from examples is mostly used for learning strategies as it gives more flexibility and gives a system to develop unknown skills or to predict patterns [16]. Learning from cases is a method that is frequently used in grouping and data mining jobs to forecast the class tag of novel information entries founded on an active set of recognized examples [18].

Following are the utmost used machine learning algorithms that we will concisely describe:

- Support Vector Machine (SVM)
- K-Nearest Neighbour (KNN)
- Decision Tree (DT)
- Logistic Regression (LR)
- Random Forest (RF)

2.2.1.1 Support Vector Machine (SVM)

SVM is a supervised machine learning algorithm that is appropriate for regression and classification. However, it is commonly used to decipher classification glitches. In SVM, what we do is, examine and plot information in any n number of dimensional space (where n symbolizes the total features you have) value of a specific coordinate in space would be the value of each feature. Now to achieve classification we need to search for a hyperplane that can equally differentiate two classes, either class 0 or class 1, and the distance between the hyperplane and successive classes should be uniform that is the best classification hyperplane solution we would prefer to achieve that.

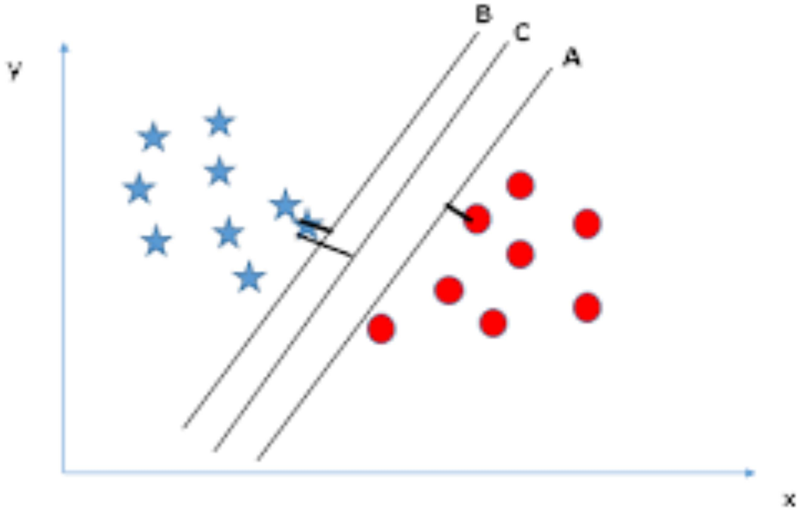


Figure 1: Basic Structure Of SVM [18]

In figure 1, the parting of A is much superior to B and C because it differentiates the dual classes in an added specific way. Correctly SVM makes one or many hyperplanes in an n-dimensional plane. The initial effort in the course of piercing the information is repeatedly, to attempt to linearly distinct the information into the equivalent tags. Steinwart and Christmann remark two chief worries, about this method [19]:

- The records might not be linearly distinguishable or not linearly distinguishable at all.
- The second worry is the likelihood of overfitting the SVM. To overcome this, information has to be pre-processed to classify defects and receive misclassifications. Else, the correctness morals of the SVM will be awed and result in more flawed arrangements for upcoming actions.

2.2.1.2 K-Nearest Neighbor (KNN)

K-Nearest algorithm Neighbour is considered as, an inactive learning process that classifies datasets in the context of their similarity with neighbors. K-Nearest neighbor process is an easy kind of supervised algorithm. Its method is for the situation, that after a novel training dataset arrives, it reviews a category of training information ordered in the structure and classifies them given a novel preparing example [8]. The only trouble with this approach is its great computational rate for classifying individually novel training information after arriving at the model. An additional downside is that the novel prototype will be observed as similar given the qualities of the novel training strategy that are considered similar and are earned.

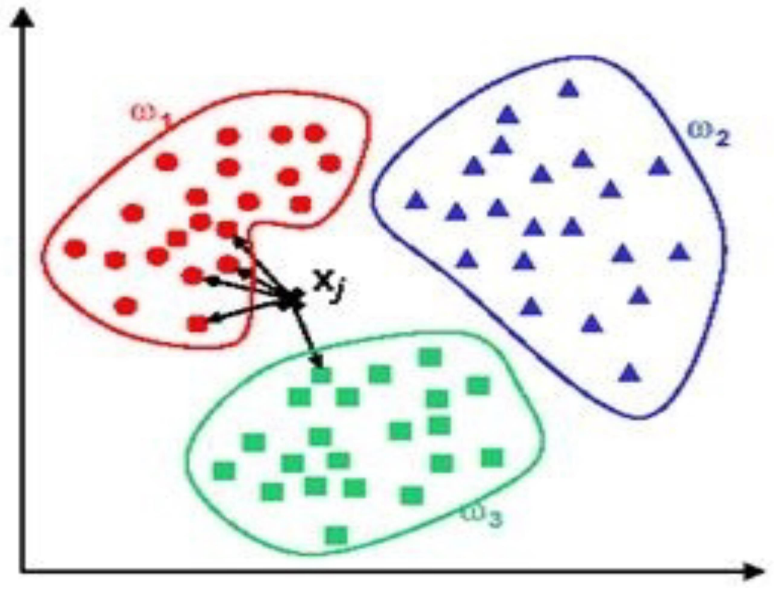


Figure 2: Basic Structure Of KNN [20]

A number point is categorized by a mainstream division of its neighbors, with the number point being allocated to the course of greatest mutual between its K-nearest neighbors calculated by a distance purpose. The utmost public distance formula used in KNN is:

$$d(x_i, y_i) = \sqrt{\sum_{r=1}^{n}(a_r(x_i) - a_r(x_j))^2}$$

In this procedure, we get that entirely training information is in n-dimensional space. The adjacent neighbour separates among double points utilizing the normal order of the separation of two points [20,21].

2.2.1.3 Decision Tree

A Decision Tree is an organization method that emphasizes a simply comprehensible picture form and it is a commonly used learning method. Decision Trees practice datasets that contain characteristic paths. A Decision Tree stands on iteratively piercing the dataset on the trait that splits the records possibly in the dissimilar present classes till a convinced halt principle is touched [22].

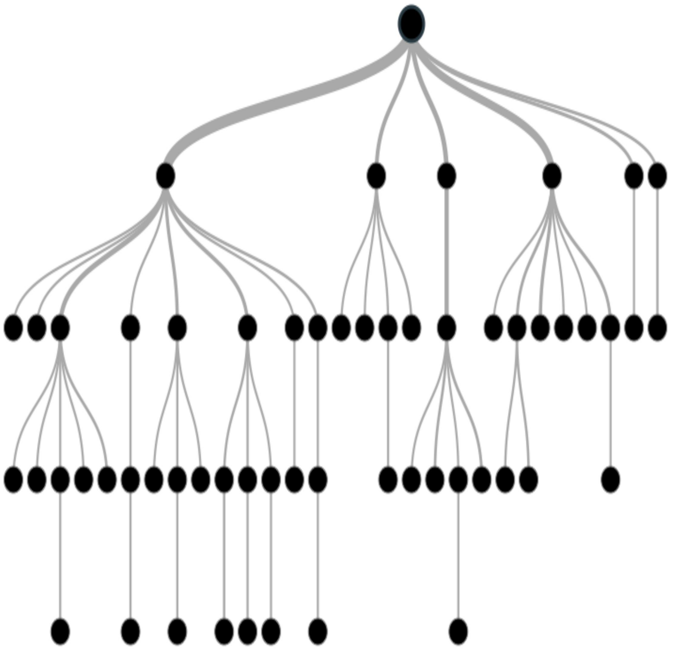


Figure 3: Basic Structure of Decision Tree [23]

The main disadvantage of this decision tree technique is that if the exercise dataset must have numerous structures it will not offer decent performances [24, 25] and overfitting is a drawback too. To escape the overfitting of the exercise dataset, two public methods are typically used:

- Stop the training as soon as the classifier fits the data perfectly.
- The most used process is to pre-prune the tree so that we can stop it from reaching its full size. Pruning of the induced decision tree. We can achieve it by a verge test for the feature excellence metric.

The chief benefit of using decisions is their clarity. The grouping of an example to a specific lesson is effortlessly comprehensible. One more feature is that the tree works well for definite structures.

2.2.1.4 Logistic Regression

Logistic regression is used for classification problems based on modeling possibility having binary likely results. It includes a reliance on a variable that is used to discover the likelihood of victory or failure of an occurrence. LR doesn't attempt to fit a hyperplane or straight line, the logistic regression model practices the logistic function (sigmoid function) to embrace the linear equation output between 0 and 1. It attempts to model the provisional chance of the class label to give its opinion. LR the utmost used Machine Learning algorithms for two-class sorting. It’s a simple process that you can practice as a routine model, it is easy to use and it will do sufficient enough in numerous jobs. Consequently, each Machine Learning engineer must be aware of their ideas [17].

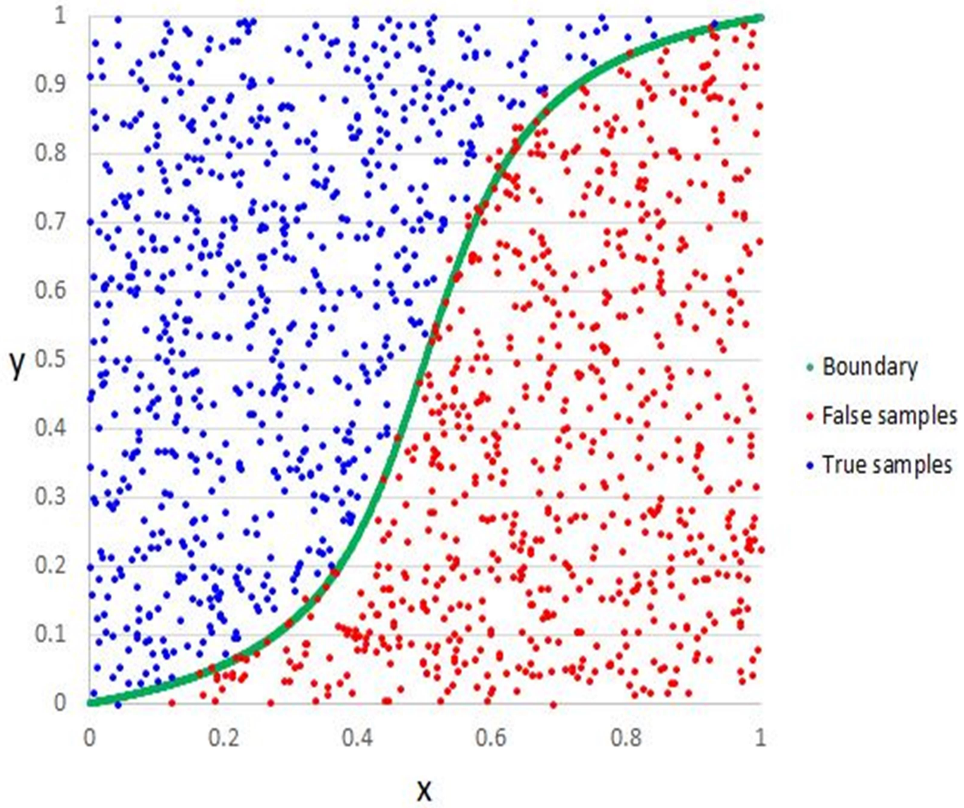


Figure 4: Logistic Regression Example [26]

2.2.1.5 Random forests

Random forest is "Ensemble Learning" which compelling the Decision Tree algorithm many epochs. RF is called the cooperative learning method since it customs numerous choice trees recognized as Forest and takes a decision based on majority. The decision tree regression algorithm splits the node based on the intensity level. The complete idea of the bagging method stands for overall results enhancement by the combined effort of all the learning models.

Random forests are a mixture of tree forecasters such that every tree depends on the standards of a chance course tested self-sufficiently and by the similar delivery for all trees in the forest. The simplification mistake for forests meets a boundary as the number of trees in the forest develops big [27].

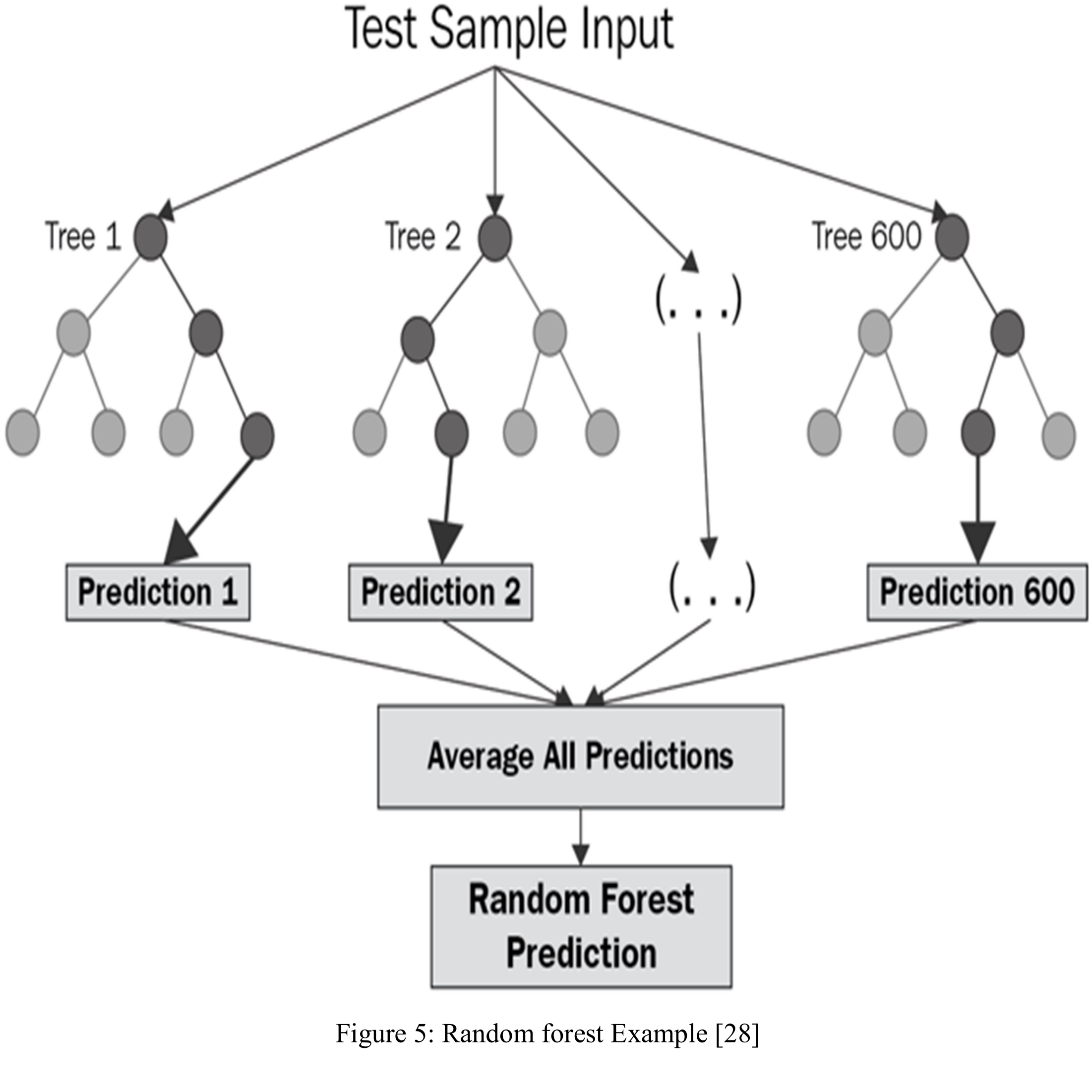


Figure 5: Random forest Example [28]

2.2.2 Deep Neural Network (DNN)

The basic components in deep neural networks [Figure 5] are the layers. There are three basic films in a neural network i.e. input layer, hidden layer, and output layer. The input layer consists of neurons, which are equal to the number of inputs. There can be an extra hidden level that lets the neural network archetypal non-linear functions. . DNN is taught using a backpropagation learning algorithm. The number of outputs is based on the activation function applied on the output layer. In this paper, we have made a model, one having softmax as the activation function on the output layer and ReLU on input and hidden layers.

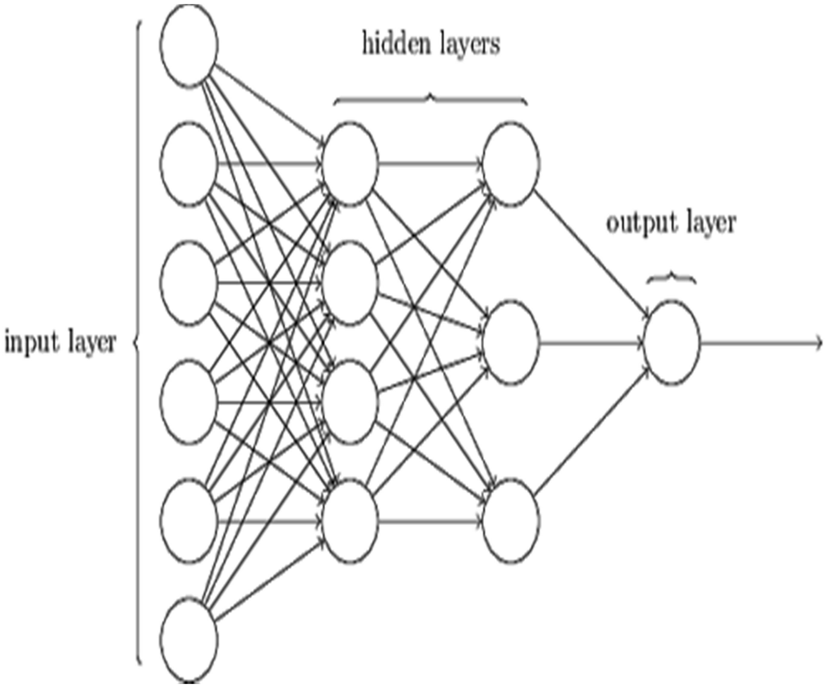


Figure 6: Basic Deep Neural Network Architecture [29]

Basics of Neural Networks

It works like a human brain. A “neuron” in a neural system is a scientific task that gathers and categorizes data conferring to a precise manner. The network tolerates a robust similarity to numerical approaches by way of arc fitting and regression examination [30].

It contains interrelated nodules. Each nodule is a perceptron and is similar to numerous creased regressions. The perceptron headlongs the indicator shaped by a numerous lined reversion into activation purpose that might be nonlinear [30].

Application of Neural Networks

Neural networks likewise increased pervasive acceptance in commercial requests such as predicting and advertising research answers, scam discovery and hazard valuation. These are mainly used with applications for economic tasks, creativity preparation, and creation care [30].

A neural network assesses price statistics and extracts openings for creating trade conclusions grounded on information study. The links can differentiate refined non-linear interdependencies and arrays extra methods of practical examination cannot. It remains the process that substance. It is the well-prepared contribution information on the directed pointer that eventually controls the level of achievement of a neural network [30].

2.2.2.1 Rectified Linear Unit (ReLU) Activation Function

ReLU function has developed efficiently in fast-tracking the size of the whole training task. Old-style neural networks have used Tanh or Sigmoid. However, the overhead activation functions are suffering from vanishing gradient problems [31]. ReLU is used as a replacement activation function that solves the problems confronted by the tanh or sigmoid activation functions.

The activation state of a neuron is decided by using an appropriate Activation Function, which calculates a weighted sum of predictors and adds bias to it. ReLU is the very popular activation function in the current scenario of the deep learning platform and is a widely preferred choice in most CNNs. The idea of using activation is to add a non-linearity factor into the neuron output.

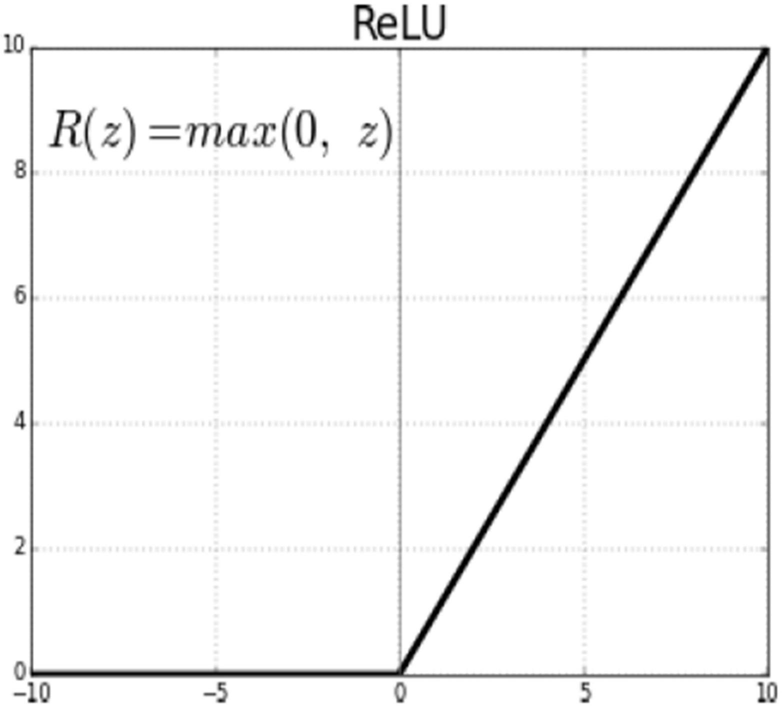


Figure 7: ReLU activation function [29]

The derivative, along with the function itself, is monotonic. The only issue arises in handling negative values, which output zero immediately and lower the ability of the model to fit properly on training data. Hence, there is no appropriate mapping for negative values according to the nature of the ReLU function.

2.2.2.2 Softmax Activation Function

This task is applied by way of the activation function on the output level of neural network prototypes that forecast a multinomial possibility delivery. That is, softmax for more than 2 class problem classification where a class association is necessary on more than two class labels. Since we are playing with probabilities here, the scores returned by the softmax function will sum up to 1. The class with the highest confidence score will be predicted.

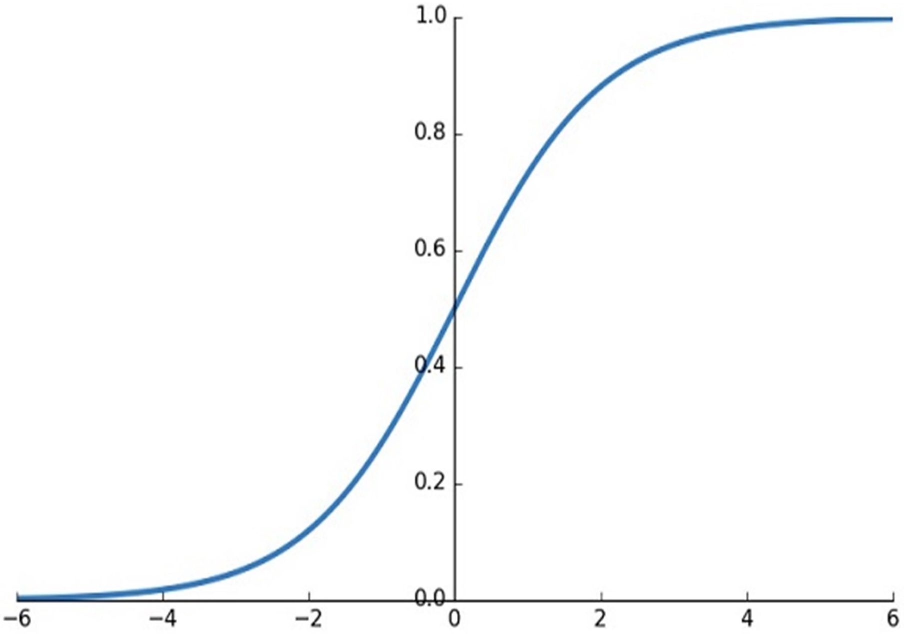


Figure 8: Softmax activation function [33]

It a good activation function that fits records as logits in possibilities that equals to one. It produces a course that indicates the possibility allocations of a list of probable goods. It is also a crucial component in deep learning grouping jobs [34].

2.2.2.3 Adam Optimizer

Adam optimizer is a self-adaptive learning degree optimization system that is been aimed exactly for teaching neural networks. Originally available in 2014, it was offered by a precise high-status conference for deep learning experts [35]. The research enclosed about encouraging figures and displaying enormous performance improvements in terms of rapidity of preparation a model [35].

The algorithms control the authority of the adaptive learning rate system to discover specific education amounts for each factor. The advantages of Adagrad [36], that works fine with situations with sparse gradients but brawls in nonconvex optimization of the neural network model and RMSprop that challenges to decide about the difficulties of Adagrad and suits fine in online situations [37].

2.2.3 Keras

Keras [14] is a high level API that is useful for making neural network models, established in python, that can sit on top of Tensor Flow. It empowers research at ease and quicker. In this research paper, Keras was used with TensorFlow for developing the anomaly detection model using the deep learning approach.

2.2.4 Tensor Flow

Tensor Flow [15] is an open source tool that allows safe construction and installing prototypes of machine learning and deep learning techniques. It is used with both the CPU or GPU backing and also chains connection with all the operating system mostly used.

2.2.5 Learning Curves

- Train Learning Curve: Learning curve calculated from the training dataset that gives an idea of how well the model is learning [41].
- Validation Learning Curve: Learning curve calculated from a hold-out validation dataset that gives an idea of how well the model is generalizing [41].

Chapter 3

PROPOSED METHODOLOGY

3.1 FRAMEWORK AND DATASET DESCRIPTION

The dataset was taken from Kaggle [38] which is an open-source by Pahl et al. [5]. They used a simulated IoT atmosphere by using DS2OS for creating artificial records. The structural design is a group of microservices that interconnect using the MQTT algorithm. The dataset contains 3,57,951 mocks and 13 features. The dataset is having 3,47,934 Standard data and 10,016 irregulars data and includes 8 classes. Two feature, "Accessed Node Type" has 147 missing data, and "Value" has 2049 missing value. The dataset description and type of attacks are shown in table 2 and table 3 respectively.

Table 2: Description of Features in the dataset.

SL .NO.	Features.	Data Types.
1.	Source ID.	Text
2.	Source Address.	Text
3.	Source Type.	Text
4.	Source Location.	Text
5.	Destination Service Address.	Text
6.	Destination Service Type.	Text
7.	Destination Location.	Text
8.	Accessed Node Address.	Text
9.	Accessed Node Type.	Text
10.	Operation.	Text
11.	Value.	Continuous
12.	Timestamp.	Discrete
13.	Normality.	Text

Table 3: Occurrence of attacks in the dataset.

Attacks	Occurrence in Dataset
DoS	5780
D.P.	342
M.C.	889
M.O.	805
S.C.	1547
S.P.	532
W.S.	122

1. Denial of Service (D.o.S): This outbreak is achieved by inundating the target appliance with traffic, or sending it data that activates a crash. The invader directs many indistinct packets so that the target gets overflow and the system gets into a deadlock state.
2. Data Type Probing (D.P.): In this, an untrusted nodule composes dissimilar data types than

a planned data type.

3. Malicious Control (M.C.): Due to software weaknesses at times the hacker can access a legal session or check the packets.

4. Malicious Operation (M.O.): These are normally triggered by malware. Malware is used to getting access to personal information.

5. Scan (S.C): Occasionally the information is picked up over the device by skimming, and in this practice rarely the information can get tainted.

6. Spying (SP): Invader exploits the susceptibilities of the device, and a sideway door is used to get into the machine.

7. Wrong Setup (W.S.): Sometimes due to the wrong setup, the records can also get corrupted.

8. Normal (N.L.): Accurate records are known as Normal data.

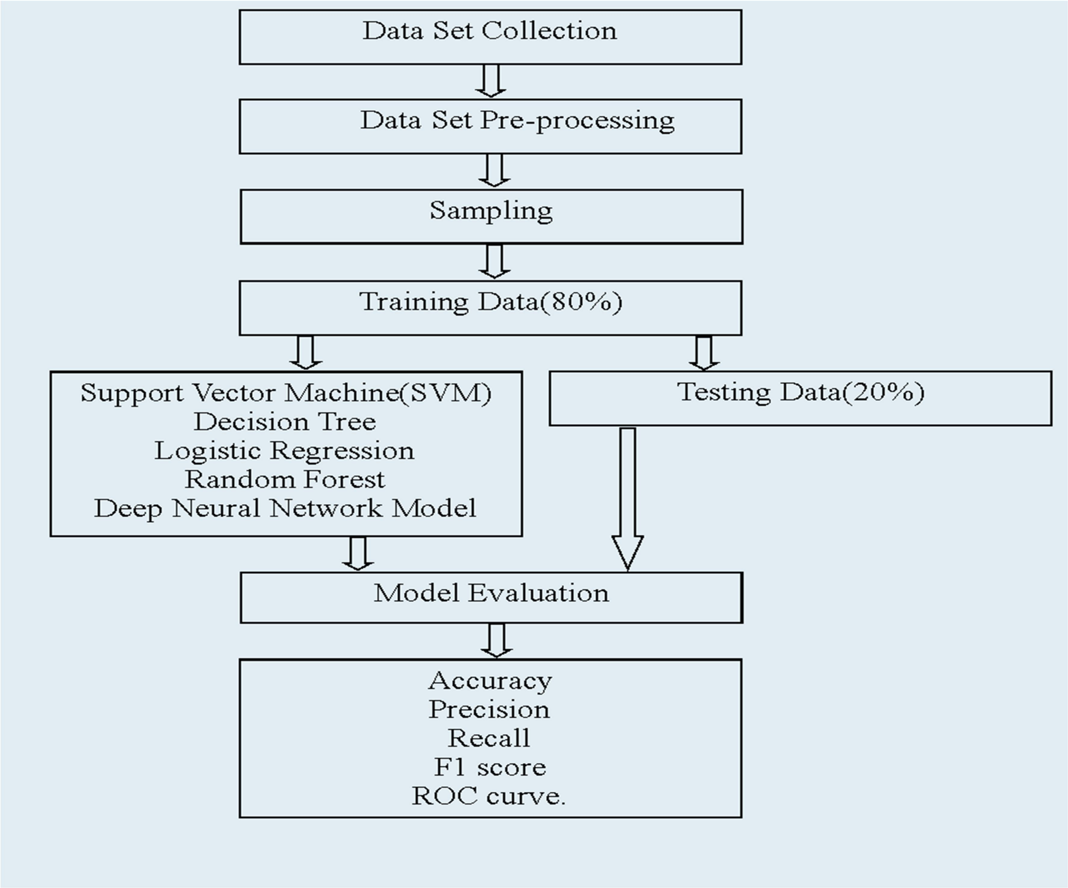


Figure 9: Proposed Overall Architecture

The overall architecture (Figure 8) is an amalgamation of numerous autonomous steps.

- Step1: In this, the data was collected from open-source and was precisely perceived to discover the kinds of data.
- Step2: In this step, Data pre-processing was applied to the dataset that we got in Step1. There are various steps involved in this process. The first data is cleaned. After this, these features are made and vectors are made.
- Step3: Sampling of data is done in this step. Feature vectors are made by converting the data. After this two sets are made of eighty and twenty ratios into training and testing.

- Step4: We used a training dataset to train our model, and a test dataset was used to test our model accuracy.
- Step5: Different evaluation metrics were used to test our model.

3.2 DATA PREPROCESSING

The primary job in this study was to style the dataset for our model. So to achieve this task, the chief task was to take care of missing data. In this dataset, two columns contain a missing value. Accessed_Node column and Value column have anomalous data because of irregularity upturned in data. Accessed_Node is having 148 rows with “NaN” data portrayed as, “Not a Number. Accessed_Node_Type, have a category feature, and removing these 148 rows will affect the valuable data, therefore, the “NaN” value is replaced by some malicious figure. The value feature is also having unexpected data which are not continuous. These we have converted these altered values into substantial continuous data that will help improve the accuracy. Unpredicted values like True, Ten, False, none in the Value column are swapped by significant values 1,10,0,0 respectively.

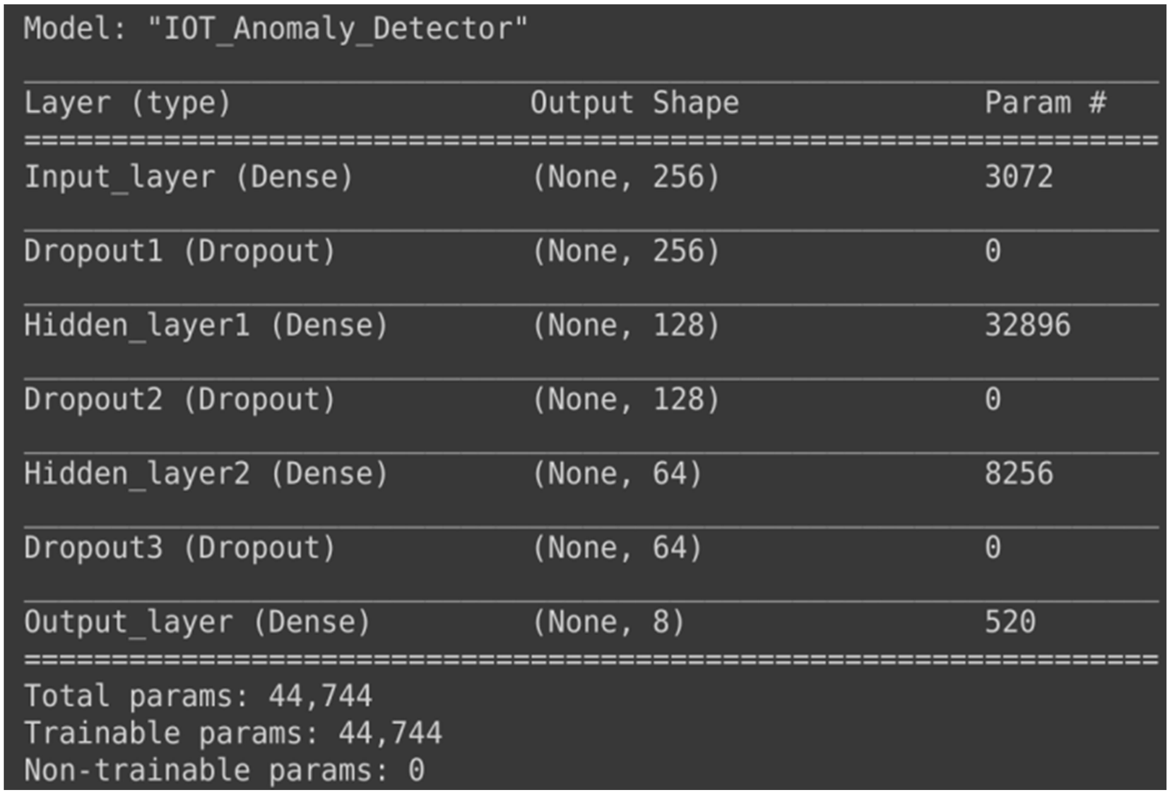
In the feature selection, the timestamp column has been removed from the dataset because there is a negligible correlation with the forecaster variable normality. In the feature engineering process, the first step is to find the feature type in the dataset. There are two kinds of data, Categorical and Numerical. Categorical Data has two values Ordinary and Nominal, whereas the Numerical dataset has two values Discrete and Continuous.

Now the main step is to convert nominal category data into vectors. There are many processes to convert Category figures into vectors. The two famous techniques are Hot and Label Encoding. We have used the label encoding process in our experiment.

3.3 NEURAL NETWORK MODEL ARCHITECTURE

The proposed deep neural network model(Figure 9) comprises of total 4 layers. The initial layer is the input layer which is having 256 neurons. The input layer is followed by two hidden layers. The first hidden layer is having 128 neurons and the second one is having 64 neurons. The last one is the output layer which will give us 8 different outputs. We have used ReLU as an activation function on input and hidden layers. For our output layer, we have used the Softmax activation function. We have used a dropout of 20% in our hidden layers to avoid overfitting.

We have a total of 11 inputs that are passed to the first layer. As we have used 256 neurons for our first layer, therefore the total number of parameters becomes ((256*11)+256) which is equal to 3072 parameters. Now for the first hidden layer, the input is 256, therefore the total number of parameters for the first hidden layer becomes ((128*256)+128) which is equal to 32,896. For the second hidden layer, the input is 128, therefore the total number of parameters for the second hidden layer becomes ((128*64)+64) which is equal to 8,256. For our output layer, the input is 64, therefore the total number of parameters for the output layer becomes ((64*8)+8) which is equal to 520.



```
Model: "IOT_Anomaly_Detector"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
Input_layer (Dense)          (None, 256)               3072
_________________________________________________________________
Dropout1 (Dropout)           (None, 256)               0
_________________________________________________________________
Hidden_layer1 (Dense)        (None, 128)               32896
_________________________________________________________________
Dropout2 (Dropout)           (None, 128)               0
_________________________________________________________________
Hidden_layer2 (Dense)        (None, 64)                8256
_________________________________________________________________
Dropout3 (Dropout)           (None, 64)                0
_________________________________________________________________
Output_layer (Dense)         (None, 8)                 520
=================================================================
Total params: 44,744
Trainable params: 44,744
Non-trainable params: 0
```

Figure 10: Proposed Model Architecture

3.4 TECHNOLOGY STACK

- TEXT EDITOR - Text editors used: Google colab, Jupyter Notebooks
- PROGRAMMING LANGUAGE - Python 3.7
- LIBRARIES/MODULES - Numpy, NLTK, Scikit-learn, Pytorch
- OS VERSION - WINDOWS 10

Google Colab is a free cloud administration, in light of Jupyter Notebooks for AI learning and research. It extends a platform for runtime completely configured to deep learning and complimentary access to a robust GPU. We can implement deep learning modules with Google Colab on the free Tesla K80 GPU, utilizing Tensorflow, PyTorch, and Keras. The utility of Google Colab for this project:

1. Free GPU support.
2. It provides common access to remote users & developers sharing Jupyter Notebooks and other files, likewise Google docs.
3. Main Python collections, like Scikit-Learn, TensorFlow, Matplotlib, etc. are pre-installed.
4. It is developed on top of the Jupyter Notebook.
5. It allows the training of deep-learning models free of cost from anywhere in the world.

3.5 EVALUATION PROCESS

Subsequently after the typical Feature Engineering, Selection, and, actualizing a model and attaining nearly yield in types of a likelihood or a course, the following stage is to discover how compelling is the model given some metric utilizing test data sets. Distinctive execution measurements are utilized to assess diverse Machine Learning Algorithms which are

- Confusion Matrix
- Accuracy
- Precision
- Recall
- F1-Measure
- ROC-curve

3.5.1 Confusion Matrix

It is very conspicuous and the most naive metrics used for the conclusion of the correctness and precision of the algorithm. It is used for grouping issues wherever the outcome can remain between binary sorts of modules. The best method to judge the outcome of a machine learning model is the confusion matrix, also called contingency table that differentiates among true positive, false positive, true negative, and false-negative predictions [39].
It is used to predict the accuracy of a process. A 2-D array is made which helps to predict the correctness of a prototype on a dataset for which we know the true value. True Positive, False Positive, False Negative and True Negative is used to measure the accuracy for multiple from the calculated confusion matrix

- True Positive (T.P.): If the example is positive and is categorized as positive.
- False Negative (F.N.): If the example is positive but is categorized as negative.
- True Negative (T.N.): If the example is negative and is categorized as negative.
- False Positive (F.P.): If the example is negative but is categorized as positive.

	Predicted	Predicted
Actual	TP	FP
Actual	FN	TN

3.5.2 Accuracy

It is a metric for assessing grouping models. It is the portion of likelihoods that our prototype got accurate. It indicates the fraction of precise findings with the complete network traffic.

- Accuracy = ((True Positive + True Negative) / (True Positive + True Negative + False Positive + False Negative)).

3.5.3 Precision

Precision rate similarly known as positive predictive value is well-defined as the comparative quantity of properly classified as correct examples between all as true categorized examples [40]. A precision worth of 1 means that each client categorized truly uses the service and vice versa. It indicates the fraction of precise findings of irregularities with the total number of authentic records as an irregularity.

- Precision = (TP / (TP + FP)). The ratio of a total of true positives equated to positives it states.

3.5.4 Recall

Recall value likewise known as sensitivity is well-defined as the comparative quantity of true categorized cases among all true cases. It indicates the fraction of precise findings of irregularities with the total number of authentic records as an irregularity

- Recall = (TP/(TP + FN)). The ratio of positives in the prototype it rights equated to the real numeral of positives.
-

3.5.5 F1 Score

We would prefer not to carry both Precisions and Recall each period we build a prototype for deciding cataloging issues. So we can get a unique score using F1-Score that sorts both. Precision (P) and Recall(R). The F1-score wishes to syndicate the reports of recall and precision employing the harmonic mean amongst them.
This is an important metric used for the calculation of interference discovery prototype representing Precision and Recall,

- F1 Score = (2*True Positive /(2*True Positive + False Positive + False Negative)). The average recall and precision is the F1 score.
-

3.5.6 ROC curve

A receiver operating characteristic curve also known as a ROC curve is a graph presenting the outcome of a category prototype at all classification thresholds. This curve plots two parameters:

- True Positive Rate (T.P.R.)

  TPR= (True Positive /(True Positive + False Negative))

- False Positive Rate (F.P.R.)

  FPR= (False Positive /(False Positive +True Negative))

Chapter 4

RESULTS AND DISCUSSION

We have used different machine learning algorithms and we proposed a deep neural network model for the dataset. We performed five cross-validations on machine learning algorithms. Figure 10,11,12,13 shows the accuracy results for the training and testing of machine learning algorithms. From the results obtained it can be concluded that Random Forest and Decision Tress performed slightly better than Logistic Regression and SVM.

For test accuracy, we can observe that for Logistic Regression the accuracy was not good for the first two folds but the accuracy increased after the third fold. The test accuracy for SVM was slightly better than Logistic regression but after the fifth fold, it almost becomes equal to Logistic Regression.

Test accuracy for Decision Tress declined in the second fold but after the third fold it started increasing and by the fifth fold accuracy was better than SVM and Logistic Regression. The accuracy of Random Forest kept on increasing by every fold. After the fifth fold, we can observe that the accuracy was nearly equal to the Decision Tree.

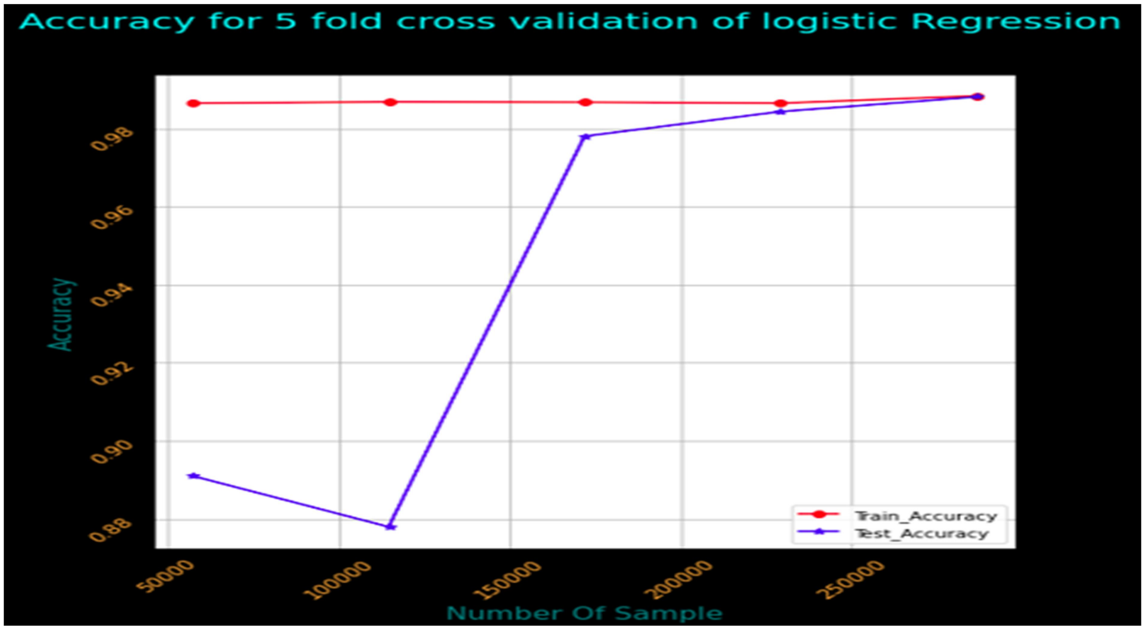


Figure 11: Training and Test accuracy for Logistic Regression for 5 fold cross-validation

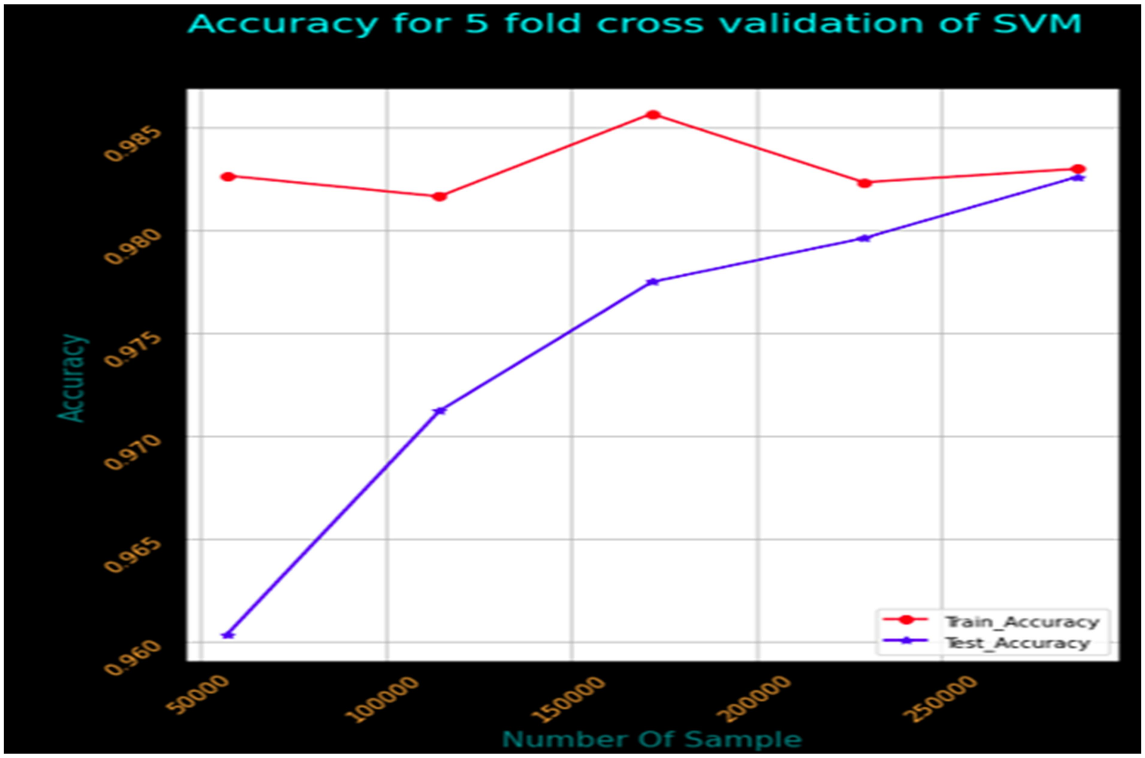


Figure 12: Training and Test accuracy for SVM for 5 fold cross-validation

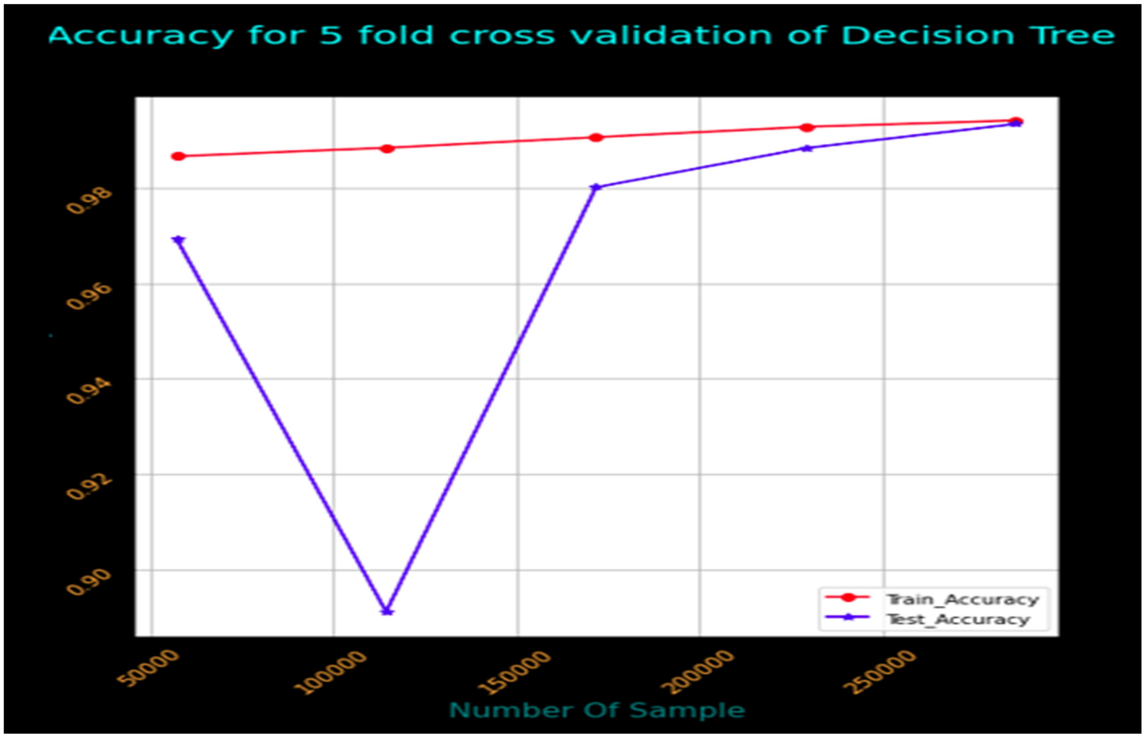


Figure 13: Training and Test accuracy for Decision Tree for 5 fold cross-validation

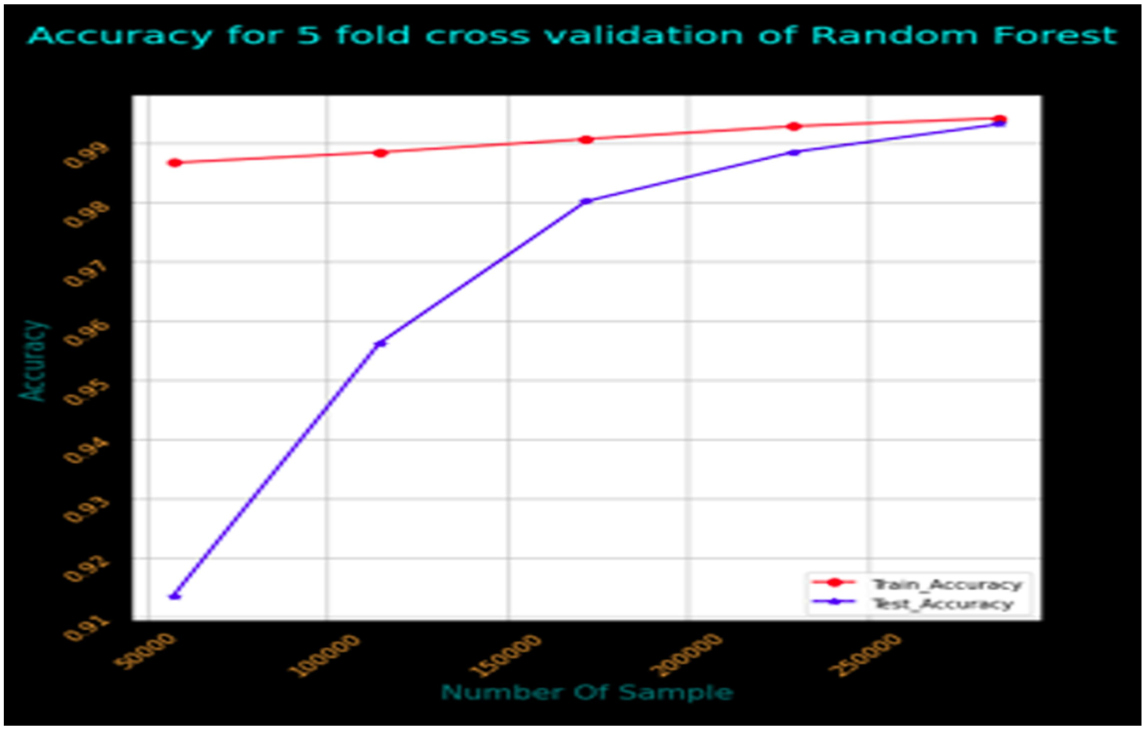


Figure 14: Training and Test accuracy for Random Forest for 5 fold cross-validation

The statistics of Accuracy, Precision, F1-Score, and Recall for machine learning and deep neural networks are given in Figures 14,15,16,17. From the results obtained we can observe that our proposed model performed the best. Logistic Regression and SVM performed almost the same. Random Forest and Decision Tress performed better than Logistic Regression and SVM.

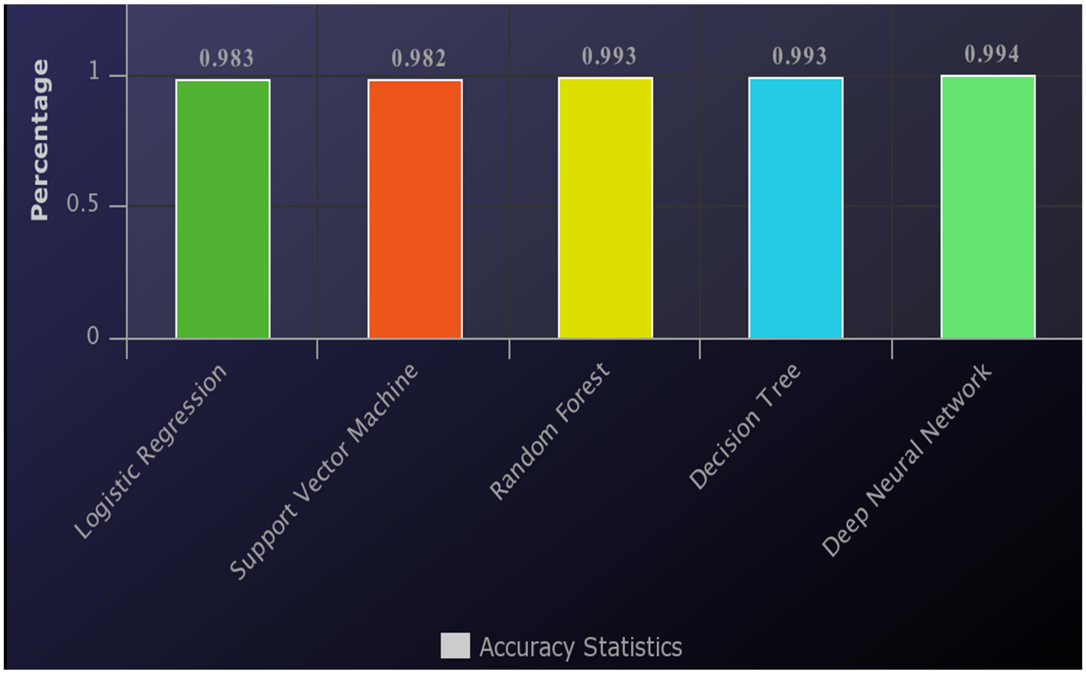


Figure 15: Accuracy Statistics

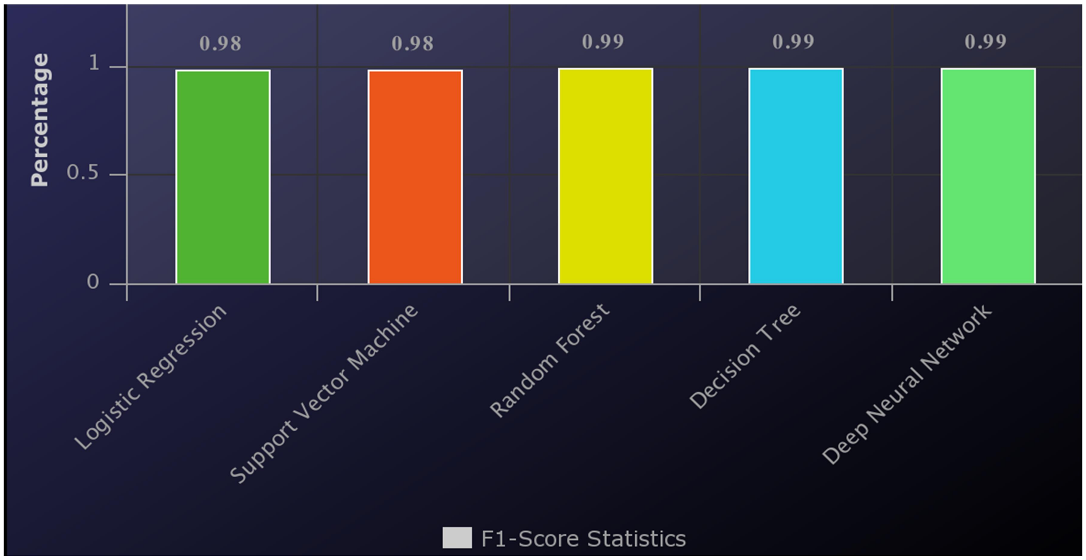


Figure 16: F1-Score Statistics

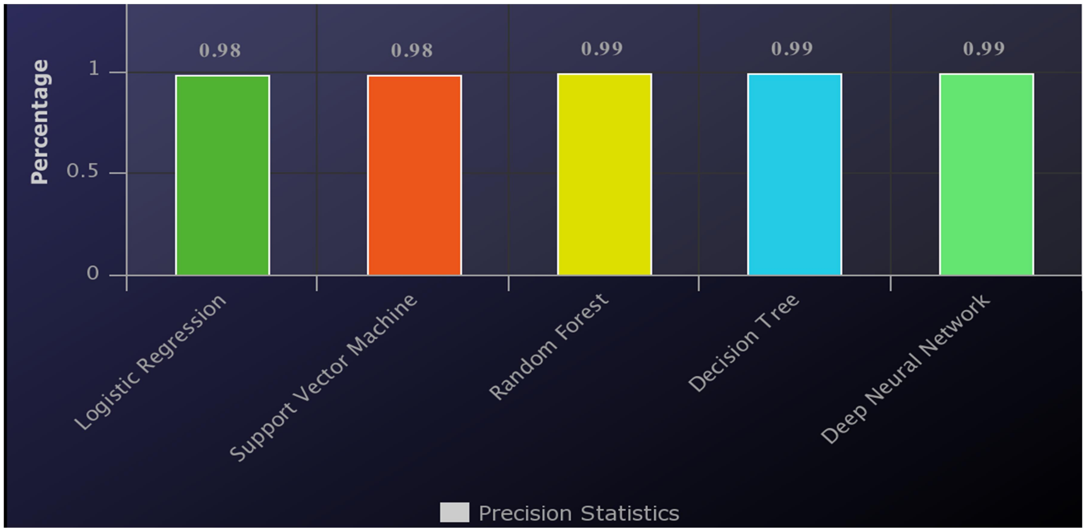


Figure 17: Precision Statistics

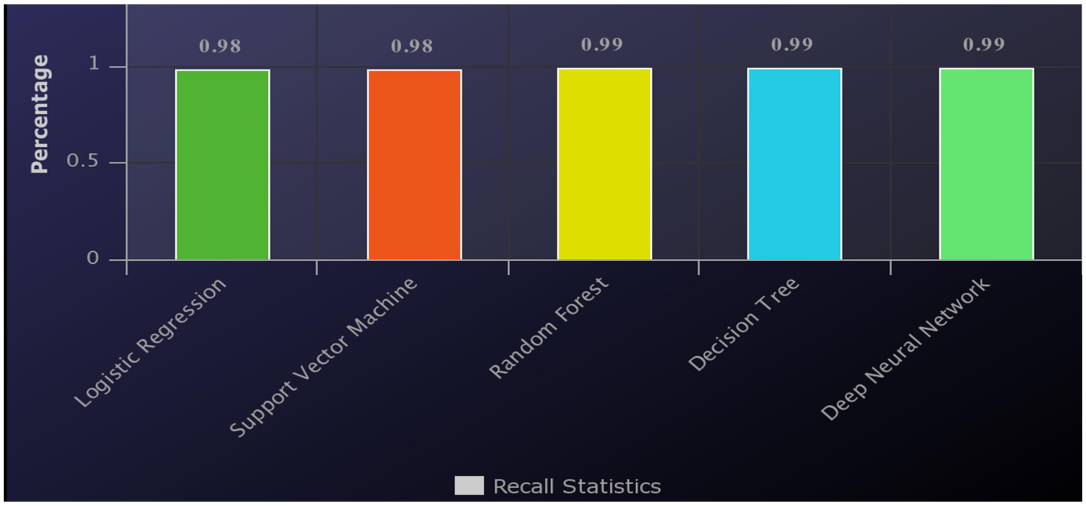
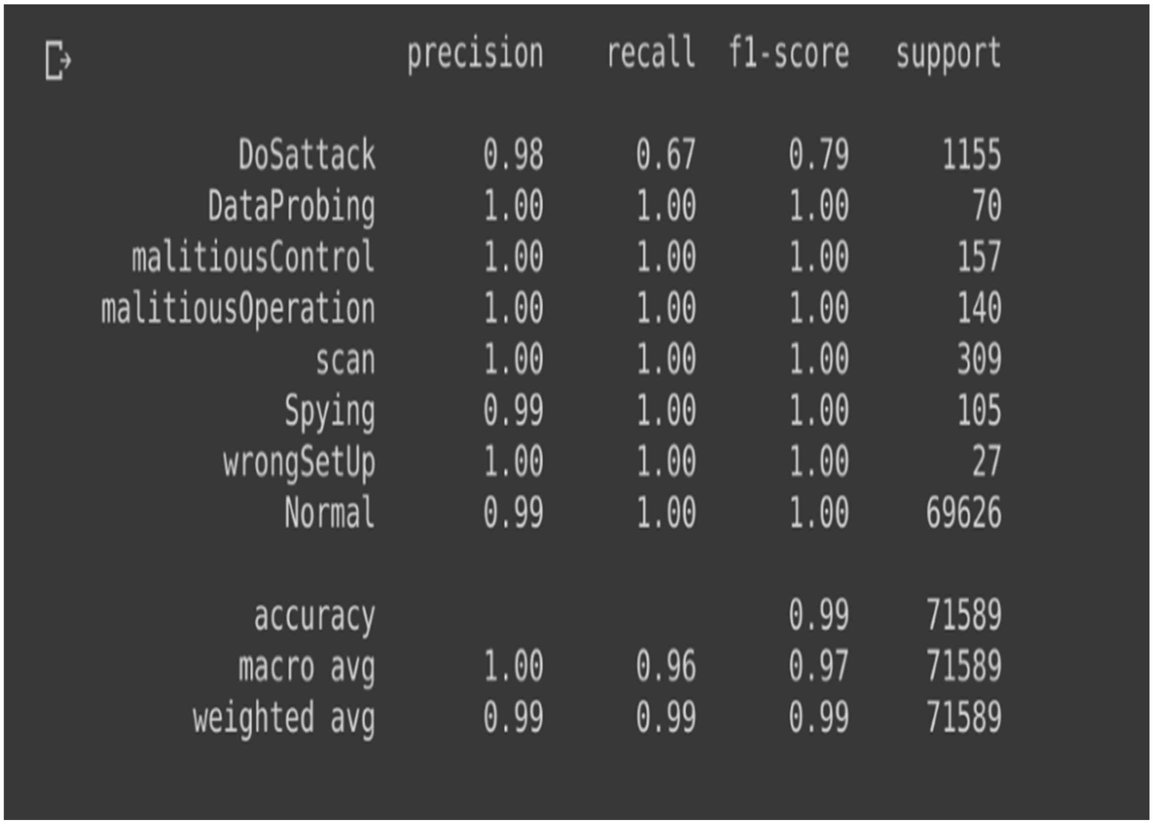


Figure 18: Recall Statistics

Table 4 and Figure 17 shows the evaluation results that we got for our dataset. The table and figure show the performance evaluation of the dataset.

Table 4: Evaluation metrics of Different Machine Learning Algorithms.

Evaluation_Classifiers.		**L.R.**	**S.V.M.**	**S.D.T.**	**R.F.**
Training.	Accuracy.	.983	.982	.993	.993
	STD(+/-).	.0012	.0015	.00081	.00081
	Precision.	.98	.98	.99	.99
	Recall.	.98	.98	.99	.99
	F1 Score.	.98	.98	.99	.99
Testing.	Accuracy.	.983	.982	.993	.993
	STD(+/-).	.0012	.0015	.00081	.00081
	Precision.	.98	.98	.99	.99
	Recall.	.98	.98	.99	.99
	F1 Score.	.98	.98	.99	.99



```
                    precision    recall  f1-score   support

          DoSattack       0.98      0.67      0.79      1155
        DataProbing       1.00      1.00      1.00        70
   malitiousControl       1.00      1.00      1.00       157
 malitiousOperation       1.00      1.00      1.00       140
               scan       1.00      1.00      1.00       309
             Spying       0.99      1.00      1.00       105
         wrongSetUp       1.00      1.00      1.00        27
             Normal       0.99      1.00      1.00     69626

           accuracy                           0.99     71589
          macro avg       1.00      0.96      0.97     71589
       weighted avg       0.99      0.99      0.99     71589
```

Figure 19: Evaluation metrics for the proposed model

From table 4 and figure 17 we can conclude that our proposed model gave the best results as compared to other machine learning algorithms.

Table 5: Confusion Matrix of Logistic Regression

	DOS.	DP.	MC.	MO.	SC.	SP.	WS.	NL.
DOS.	775	0	0	0	0	0	0	403
DP.	0	36	0	0	0	0	0	27
MC.	0	0	164	0	0	0	0	5
MO.	0	0	0	78	0	0	0	77
SC.	5	0	2	0	125	0	2	171
SP.	0	0	0	0	16	0	0	104
WS.	0	0	0	0	0	0	28	0
NL.	34	0	1	1	2	0	0	69533

From table5 we can see that LR predicted 775 correct DOS results in and 403 were misclassified. 36 DP were predicted accurately while 27 were miss classified. 164 MC were predicted correctly while 5 were misclassified. 78 MO were predicted accurately while 77 were misclassified. 125 SC were predicted accurately while 178 were misclassified. 0 SP were predicted accurately while 120 were misclassified. 28 WS were predicted accurately while 0 were misclassified. 69533 were predicted accurately while 38 were misclassified.

Table 6: Confusion Matrix of SVM

	DOS.	DP.	MC.	MO.	SC.	SP.	WS.	NL.
DOS.	775	0	0	0	0	0	0	403
DP.	0	0	0	0	0	0	0	63
MC.	0	0	10	0	0	0	0	159
MO.	0	0	0	33	0	0	0	122
SC.	0	0	2	0	125	0	2	171
SP.	0	0	0	0	16	0	0	104
WS.	0	0	0	0	0	0	28	0
NL.	34	0	1	1	2	0	0	69533

From table6 we can see that SVM predicted 775 correct DOS results in and 403 were misclassified. 0 DP were predicted accurately while 63 were miss classified. 10 MC were predicted correctly while 159 were misclassified. 33 MO were predicted accurately while 122 were misclassified. 125 SC were predicted accurately while 175 were misclassified. 0 SP were predicted accurately while 120 were misclassified. 28 WS were predicted accurately while 0 were misclassified. 69533 were predicted accurately while 38 were misclassified.

Table 7: Confusion Matrix of DT

	DOS.	**DP.**	**MC.**	**MO.**	**SC.**	**SP.**	**WS.**	**NL.**
DOS.	775	0	0	0	0	0	0	403
DP.	0	63	0	0	0	0	0	0
MC.	0	0	169	0	0	0	0	0
MO.	0	0	0	155	0	0	0	0
SC.	0	0	0	0	305	0	0	0
SP.	0	0	0	0	0	120	0	0
WS.	0	0	0	0	0	0	28	0
NL.	18	0	0	0	2	0	0	69551

From table7 we can see that DT predicted 775 correct DOS results in and 403 were misclassified. 63 DP were predicted accurately while 0 were miss classified. 169 MC were predicted correctly while 0 were misclassified. 155 MO were predicted accurately while 0 were misclassified. 305 SC were predicted accurately while 0 were misclassified. 120 SP were predicted accurately while 0 were misclassified. 28 WS were predicted accurately while 0 were misclassified. 69551 were predicted accurately while 20 were misclassified.

Table 8: Confusion Matrix of RF

	DOS.	**DP.**	**MC.**	**MO.**	**SC.**	**SP.**	**WS.**	**NL.**
DOS.	775	0	0	0	0	0	0	403
DP.	0	63	0	0	0	0	0	0
MC.	0	0	169	0	0	0	0	0
MO.	0	0	0	155	0	0	0	0
SC.	0	0	0	0	305	0	0	0
SP.	0	0	0	0	0	120	0	0
WS.	0	0	0	0	0	0	28	0
NL.	18	0	0	0	0	0	0	69553

From table8 we can see that RF predicted 775 correct DOS results in and 403 were misclassified. 63 DP were predicted accurately while 0 were miss classified. 169 MC were predicted correctly while 0 were misclassified. 155 MO were predicted accurately while 0 were misclassified. 305 SC were predicted accurately while 0 were misclassified. 120 SP were predicted accurately while 0 were misclassified. 28 WS were predicted accurately while 0 were misclassified. 69553 were predicted accurately while 18 were misclassified.

Table 9: Confusion Matrix Of Proposed Model

	DOS.	**DP.**	**MC.**	**MO.**	**SC.**	**SP.**	**WS.**	**NL.**
DOS.	771	0	0	0	0	0	0	384
DP.	0	70	0	0	0	0	0	0
MC.	0	0	157	0	0	0	0	0
MO.	0	0	0	140	0	0	0	0
SC.	0	0	0	0	309	0	0	0
SP.	0	0	0	0	0	105	0	0
WS.	0	0	0	0	0	0	27	0
NL.	16	0	0	0	0	1	0	69609

From table9 we can see that our proposed model predicted 771 correct DOS results in and 384 were misclassified. 70 DP were predicted accurately while 0 were miss classified. 157 MC were predicted correctly while 0 were misclassified. 140 MO were predicted accurately while 0 were misclassified. 309 SC were predicted accurately while 0 were misclassified. 105 SP were predicted accurately while 0 were misclassified. 27 WS were predicted accurately while 0 were misclassified. 69609 were predicted accurately while 17 were misclassified.

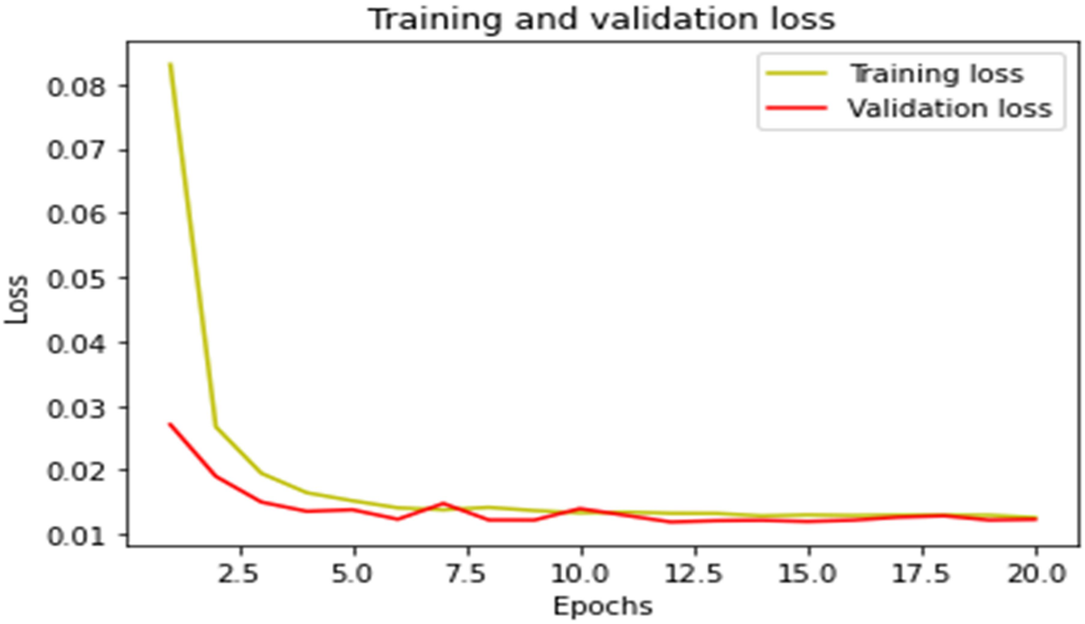


Figure 20: Comparison of Training and validation loss of Proposed Model

From figure 18 we can see, both the curves converge and have no gap between the two lines. The validation loss curve is stable. Hence we can predict that it is a good fit.

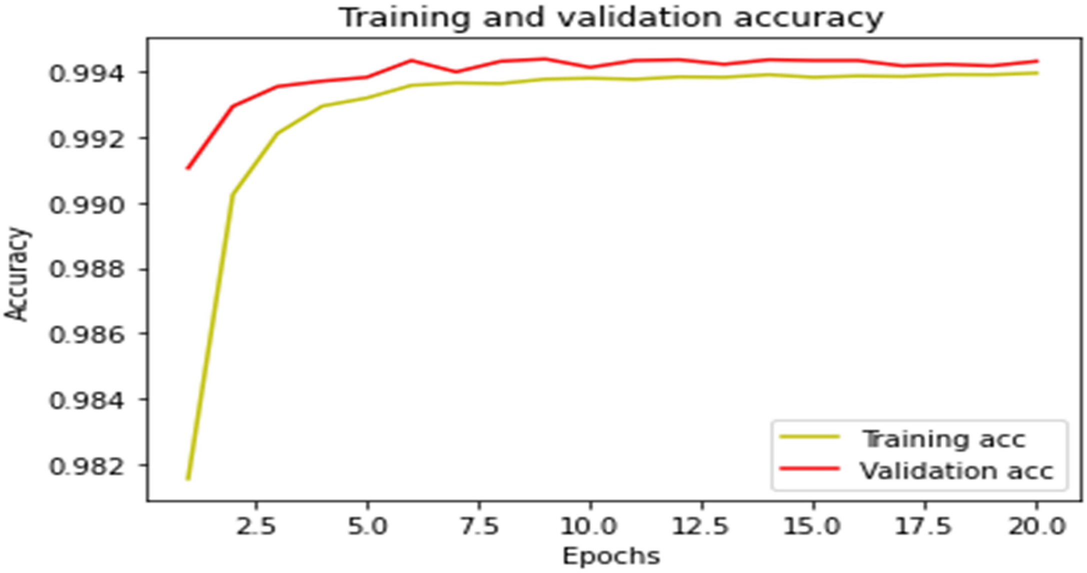


Figure 21: Comparison of Training and validation Accuracy of Proposed Model

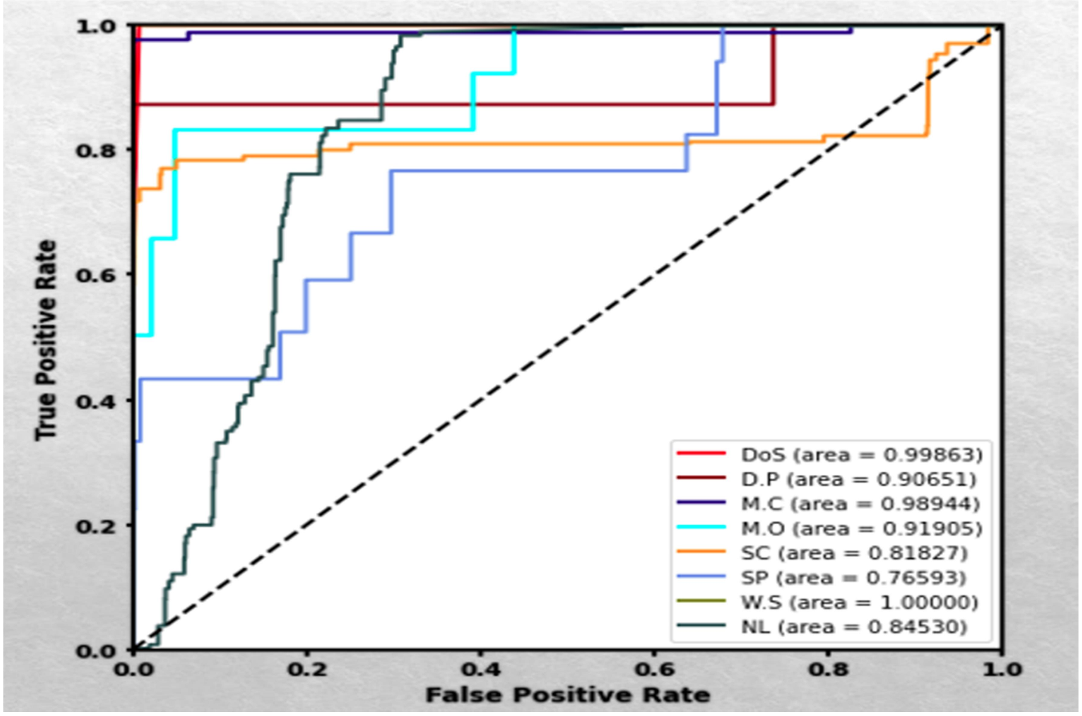


Figure 22: ROC of LR

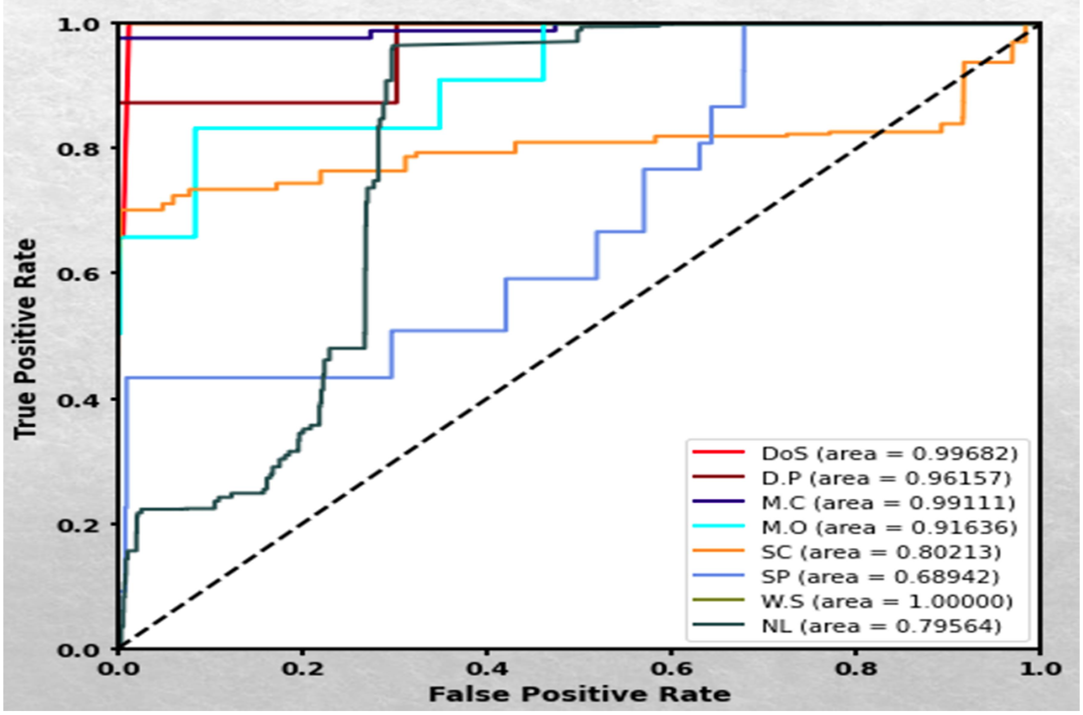


Figure 23: ROC of SVM

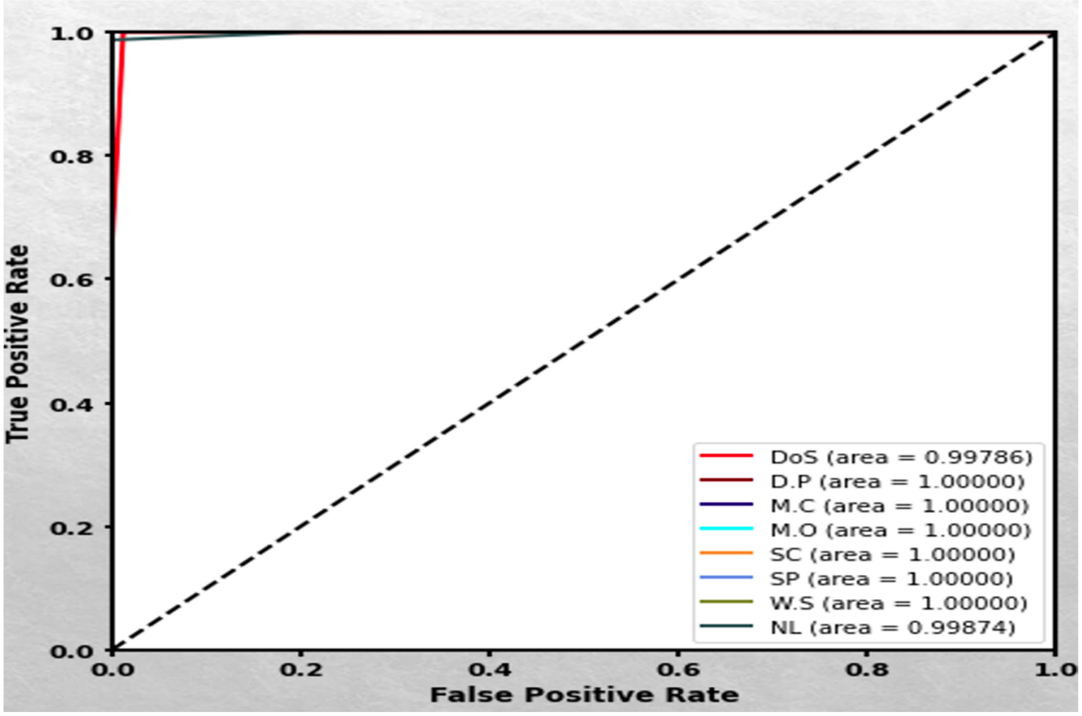


Figure 24: ROC of SVM

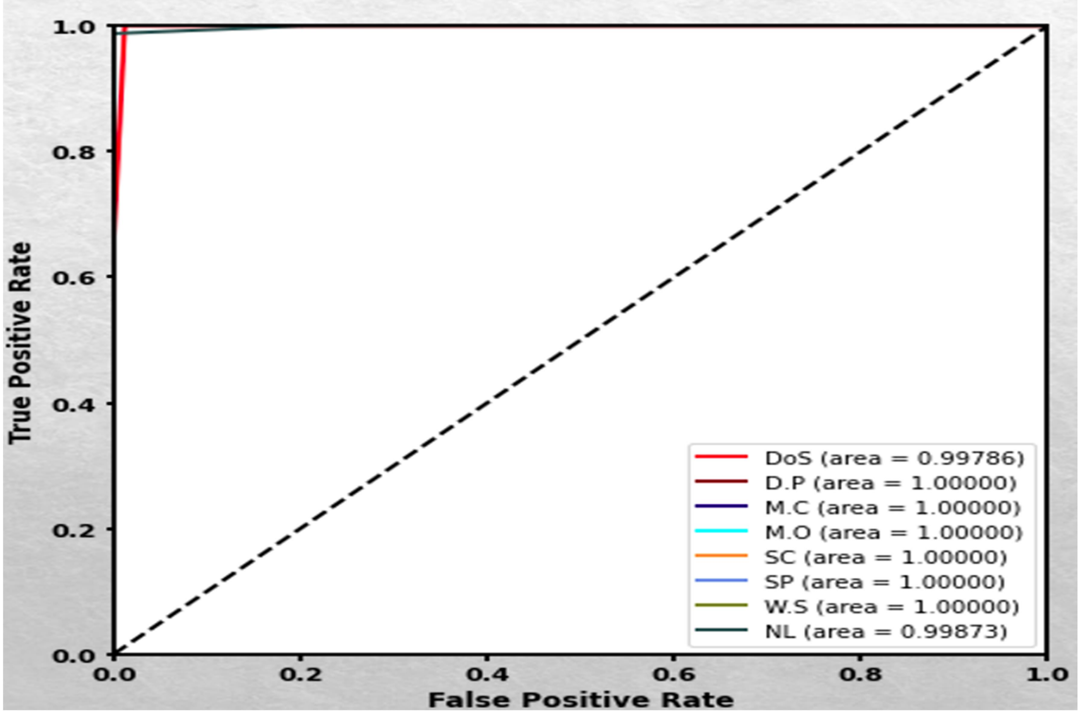


Figure 25: ROC of RF

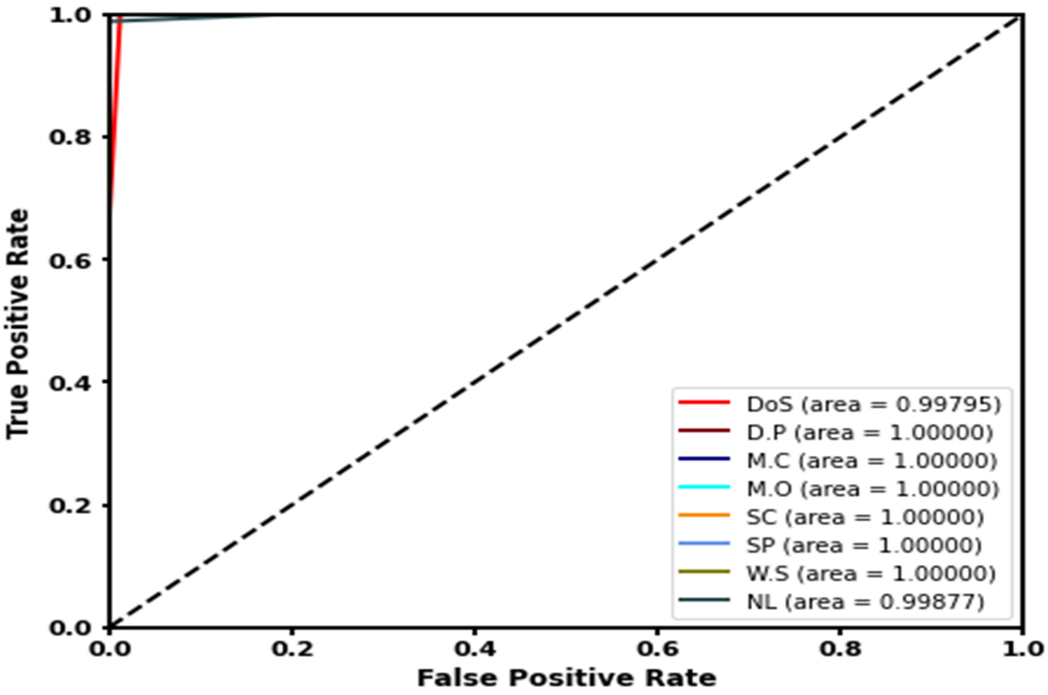


Figure 26. ROC curve of Proposed Model

So after looking at all these results we can conclude that our proposed model that is based on deep neural networks has produced the best results as compared to machine learning results. The accuracy was also maximum as compared to other models.

Chapter 5

CONCLUSION AND FUTURE WORK

5.1 Conclusion

Based on the full study we can conclude that for machine learning techniques Random Forest performed better than other techniques. Our proposed model that is based on a deep neural network performed better than machine learning techniques. We achieved an accuracy of 99.43% which is better than all other techniques used. By looking at the confusion matrix, we can conclude that our proposed model accurately predicts the attack as compared to other machine learning techniques.

5.2 Future Work

In this research work, our study is grounded on simulated atmosphere data. In the event of real-time information, there might be some other complications. More experiential research is required on this issue concentrating on real time datasets. In the IoT system, micro-services perform inversely at dissimilar intervals which may cause abnormalities in common performance in IoT services, therefore, producing an irregularity. Additional research is required to understand these difficulties in an additional detailed way.

REFERENCES

[l] J. Howell. Number of connected iot devices will surge to 125 billion by 2030, ihs markit says - ihs technology. [Online]. Available: https://technology.ihs.com/596542/, last accessed: 02/06/2020.

[2] H.H. Pajouh , R. Javidan , R. Khayami , D. Ali , K.-K.R. Choo , A two-layer dimension reduction and two-tier classification model for anomaly-based intrusion detection in iot backbone networks, IEEE Trans. Emerg. Top. Comput. (2016) .

[3] I. Poyner, R. Sherratt, Privacy and security of consumer IoT devices for the pervasive monitoring of vulnerable people. (2018).

[4] A .A . Diro , N. Chilamkurti , Distributed attack detection scheme using deep learning approach for internet of things, Future Gen. Comput. Syst. 82 (2018) 761–768 .

[5] M.-O. Pahl , F.-X. Aubet , All eyes on you: distributed multi-dimensional IoT microservice anomalydetection, in: Proceedings of the 2018 Fourteenth International Conference on Network and Service Management (CNSM)(CNSM 2018), 2018 . Rome, Italy

[6] X. Liu , Y. Liu , A. Liu , L.T. Yang , Defending on–offattacks using light probing messages in smart sensors for industrial communication systems, IEEE Trans. Ind. Inf. 14 (9) (2018) 3801–3811 .

[7] A .A . Diro , N. Chilamkurti , Distributed attack detection scheme using deep learning approach for internet of things, Future Gen. Comput. Syst. 82 (2018) 761–768 .

[8] E. Anthi, L. Williams, P. Burnap, Pulse: an adaptive intrusion detection for the internet of things (2018).

[9] O. Brun , Y. Yin , E. Gelenbe , Y.M. Kadioglu , J. Augusto-Gonzalez , M. Ramos , Deep learning with dense random neural networks for detecting attacks against IoT-connected home environments, in: Proceedings of the 2018 ISCIS Security Workshop, Imperial College London. Recent Cybersecurity Research in Europe. Lecture Notes CCIS, in: 821, 2018 .

[10] H.H. Pajouh , R. Javidan , R. Khayami , D. Ali , K.-K.R. Choo , A two-layer dimension reduction and two-tier classification model for anomaly-based intrusion detection in iot backbone networks, IEEE Trans. Emerg. Top. Comput. (2016).

[11] G. D'Angelo , F. Palmieri , M. Ficco , S. Rampone , An uncertainty-managing batch relevance-based approach to network anomaly detection, Appl. Soft Comput. 36 (2015) 408–418

[12] A. Ukil , S. Bandyoapdhyay , C. Puri , A. Pal , Iot healthcare analytics: The importance of anomaly detection, in: Proceedings of the 2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA), IEEE, 2016, pp. 994–997 .

[13] Ivens Portugal, Paulo Alencar, Donald Cowan(2017),The Use of Machine Learning Algorithms in Recommender Systems:A Systematic Review, S0957-4174(17) 30833-3, 10.1016/j.eswa.2017.12.020

[14] Kotsiantis, Sotiris B., I. Zaharakis, and P. Pintelas. "Supervised machine learning:A review of classi_cation techniques." Emerging artificial intelligence applications in

computer engineering 160, 2007.

[15] C. Sammut and G. I.Webb, eds.,Encyclopedia of Machine Learning.Springer,2010.

[16] Aurangzeb Khan, Baharum Baharudin, Lam Hong Lee, Khairullah khan, Review of Machine Learning Algorithms for Text-Documents Classification," Journal of Advances in Information Technology, Vol. 1, No. 1, February 2010.

[17] Z. Zheng, "Constructing conjunctions using systematic search on decision trees,"Knowledge-Based Systems, vol. 10, pp. 421-430, 1998.

[18] https://www.analyticsvidhya.com/wp-content/uploads/2015/10/SVM_3.png

[19] Padraig Cunningham and Sarah Jane Delany, k-Nearest Neighbour Classifiers,Technical Report UCD-CSI-2007-4 March27, 2007.

[20] https://www.mathworks.com/matlabcentral/mlc-downloads/downloads/03faee64-e85e-4ea0-a2b4-e5964949e2d1/d99b9a4d-618c-45f0-86d1-388bdf852c1d/images/screenshot.gif

[20] Steck, H. (2013). Evaluation of recommendations: rating prediction and ranking. (pp. 213220). doi:10 . 1145 / 2507157.2507160

[21] S. B. Kotsiantis, I. D. Zaharakis, P. E. Pintelas, learning: a review of classification and combining techniques,"Springer Science and Business Me dia B.V.,2007

[22] J. Su and H. Zhang,fast decision tree learning algorithm," in AAAI, pp500505, AAAI Press, 2006

[23] https://littleml.files.wordpress.com/2012/01/screen-shot-2012-01-23-at-10-00-17-am1.png

[24] L. Rokach and O. Z. Maimon, Data mining with decision trees: theory and applications,vol. 69 of Series in machine perception and arti_cial intelligence. World Scienti_c Publishing Co., 2008.

[25] George Dimitoglou, James A. Adams, and Carol M. Jim," Comparison of the C4.5 and a Naive Bayes Classifier for the Prediction of Lung Cancer Sur vivability"

[26] http://storage.ning.com/topology/rest/1.0/file/get/2808358994?profile=original

[27] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 532

[28] https://miro.medium.com/max/2612/0*f_qQPFpdofWGLQqc.png

[29] Michael A. Nielsen, "Neural Networks and Deep Learning", Determination Press, 2015

[30] https://www.investopedia.com/terms/n/neuralnetwork.asp

[31] Y. Bengio, P. Simard and P. Frasconi, “Learning Long-term dependencies with gradient descent is difficult”, IEEE transactions on neural networks, 1994, pp. 157-166.

[32] https://miro.medium.com/max/357/1*oePAhrm74RNnNEolprmTaQ.png

[33] https://themaverickmeerkat.com/img/softmax/sigmoid_plot.jpg

[34] https://medium.com/data-science-bootcamp/understand-the-softmax-function-in-minutes-f3a59641e86d

[35] https://www.iclr.cc/archive/www/doku.php%3Fid=iclr2015:main.html

[36] John Duchi, Elad Hazan, and Yoram Singer. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. Journal of Machine Learning Research, 12:2121–2159, 2011

[37] https://towardsdatascience.com/adam-latest-trends-in-deep-learning-optimization-6be9a291375c

[38] M.-O. Pahl, F.-X. Aubet, DS2OS traffic traces, 2018, (https://www.kaggle.com/francoisxa/ds2ostraffictraces)

[39] Gar_eld (2014), Value Based performance metrics.

[40] Michele Fratello, DPControl, Salerno, Italy Roberto Tagliaferri, Universitiesa degli Studi di Salerno, Salerno, Italy,(2018),Elsevier Decision Trees and Random forests.

[41] https://machinelearningmastery.com/learning-curves-for-diagnosing-machine-learning-model-performance/

LIST OF PUBLICATION OF THE CANDIDATE'S WORK

[1] Research Paper accepted for IEEE conference on "Anomaly Detection in IoT network Using Deep Neural Networks" in 2021 IEEE INTERNATIONAL CONFERENCE ON COMPUTING, POWER AND COMMUNICATION TECHNOLOGIES (GUCON 2021).

Status: Accepted

Conference Date: 24-26 September 2021

Paper Topic: Anomaly Detection in IoT network Using Deep Neural Networks

Paper ID: 140