ACS-IoT: A CNN-BiLSTM Model for Anomaly Classification in IoT Networks

Masters Project

By

Yue Guan

Supervised by:

Dr. Nasser Ezzati Jivan

Brock University, Canada

ACKNOWLEDGEMENT

I am grateful to God for His guidance and blessings throughout the completion of this work. I am immensely grateful to my supervisors, Dr. Nasser Ezzati Jivan and Dr. Morteza Noferesti for their unwavering support, patience, and extensive knowledge throughout my master's study and research. I would also like to express my deep appreciation to my friends, colleagues, and the entire university community at Brock University for their constant support and encouragement. My profound gratitude goes to my parents and family for their unconditional love and unwavering support throughout my academic journey. Lastly, I am indebted to the entire team at Brock University for creating a warm and friendly atmosphere and for their contributions in bringing this work to fruition.

Date: 12 June, 2023

Yue Guan

Abstract

This work proposes an Anomaly Classification System for IoT (ACS-IoT). The proposed system contains a pipeline of machine learning and deep learning algorithms for the effective classification of anomalies and their sub-types. Machine learning algorithms are adopted to distinguish between normal data and anomaly data. The deep networks, on the other hand, are used to perform anomaly-type classification. We propose the use of the Synthetic Minority Oversampling Technique (SMOTE) to address the data imbalance problem and Particle Swarm Optimization (PSO) as a feature selection mechanism to improve accuracy as well as execution time. The proposed system proved to be accurate as well as precise when tested on a publicly available IoT dataset.

Keywords: IoT, Network Attacks, Deep Network, Intrusion Detection, Anomaly Classification

Table of Contents

1. Introduction

Recent technological advancements have resulted in significant changes, particularly in the field of information and communication technology. These technological advancements have transformed and revolutionized every aspect of human life. The devices used daily are now connected through wired and wireless connections, resulting in an inter-connected internet-enabled network of devices. This internet-connected network of devices is commonly termed the Internet of Things (IoT) [15] and is transforming the whole concept of network and device communication.

The four-layer IoT architecture involves Perception, Network, Transport, and Application layers. As Figure 1 shows, in the Perception layer, the physical sensing devices, such as sensors and robots, incorporate the basic IoT hardware. IoT devices can consist of mobile and static devices and belong to multiple technological domains. The Network layer deals with the IoT protocol layers, which face corresponding challenges such as low-power, wireless, and reliable networking protocols.

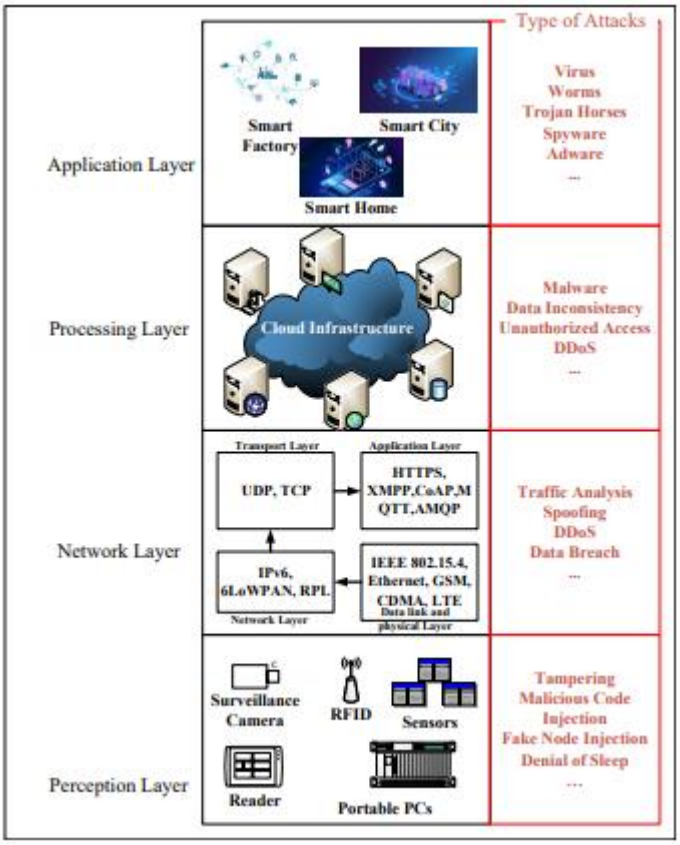

Figure 1. Potential Attacks on each IoT layer architecture

The processing layer receives data and processes them to provide services. Cloud computing is an essential part of IoT, which not only converges the servers but also processes increased processing power, analyses the valuable information obtained from the sensors, and even provides good storage capacity. The application layer involves protocols for specific environments such as smart homes, cities, and factories [1].

All the technological advancements in each IoT infrastructure layer bring particular security challenges. The massive number of variable devices connected through numerous links and using specific network protocols such as IPv6 over Low-power Wireless Personal Area Network (6LoWPAN) and Constrained Application Protocol (CoAP) introduce novel vulnerabilities and challenges and pose new security threats and attacks. The specific design of malware, viruses, and spyware for IoT applications are the fundamental challenges that should be solved to protect user privacy and service security.

The characteristics of IoT can be categorized into four major categories. This includes Automation, Heterogeneity, M2M protocols and Volume. Automation in IoT refers to the use of technological advancements to perform certain tasks and processes automatically in an IoT environment. Automation in IoT is achieved with the help of internet enabled sensors and devices, allowing them to communicate and exchange data. This data is analysed and used to trigger automated actions. In IoT context, heterogeneity refers to a network which consists of a variety of devices such as sensors, cameras, actuators, and gateways, using different protocols, each with its own unique characteristics and

capabilities. M2M (machine-to-machine) protocols are communication protocols that are designed to enable these devices to communicate with each other in an IoT network and the amount of data generated by these IoT devices is referred to as volume.

Moreover, those devices connected to the IoT network are not designed by default to be specifically a part of the connected network. This can make things even worse with respect of the security perspective as these devices are launched into the market without proper controls of security. Figure 2 shows an overview of a typical intrusion detection system present in the IoT architecture. Recent research in the filed of IoT has demonstrated that current IoT devices has several security flaws which leads to many vulnerabilities such as denial-of-service (DoS) attacks and distributed denial-of-service (DDoS) attacks. These attacks when generated on the IoT can lower the performance of the system and cause latency issues.

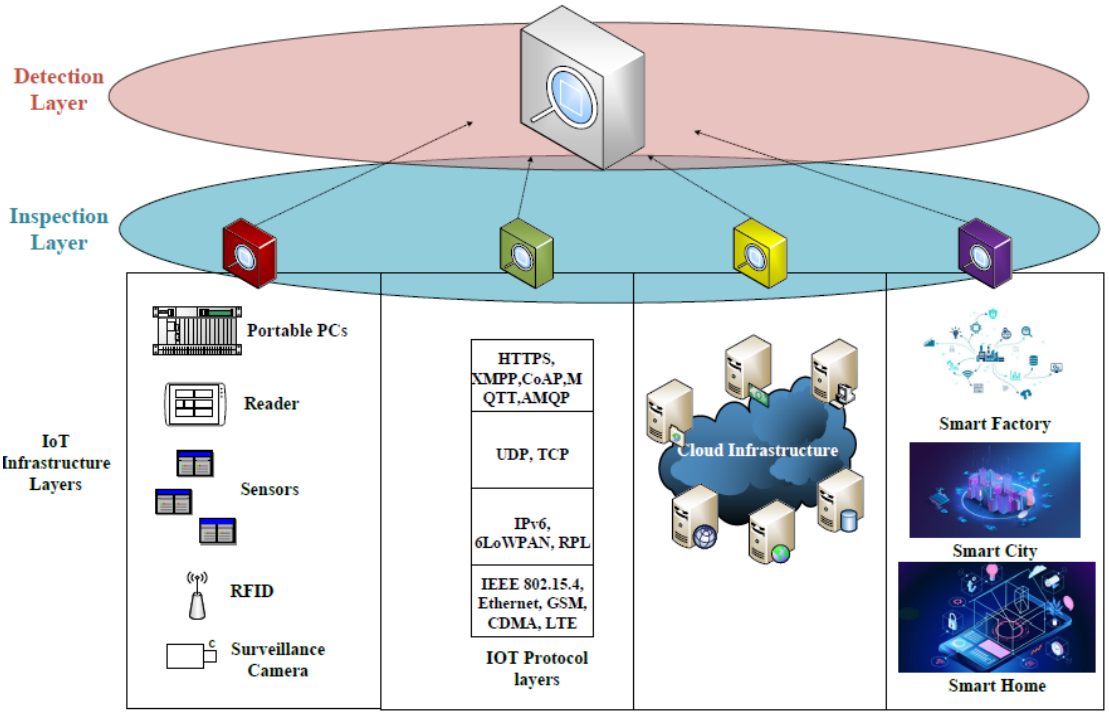

Figure 2. The Proposed Two-Layer Intrusion Detection System in IoT

To tackle all the issues in terms of attacks, anomalies, and intrusions, we have developed a security mechanism which could act to ensure that no device could be able to act maliciously by introducing attacks or intrusions. We propose an Anomaly Classification System for IoT (ACS-IoT) consisting of two tiers which analyses network traffic and packets to detect any intrusion or attack. The two tiers / layers include: (i) Inspection layer (ii) Detection Layer. Each layer has its own functionalities starting from data acquisition and pre-processing to complex machine learning based hybrid pipeline for binary and multi-class anomaly classification.

We also propose an efficient data preparation scheme for building better neural network models. Our data preparation scheme involves pre-processing of data including feature

selection using PSO, normalization and data balancing using SMOTE. The efficient data preparation scheme helps to optimize both training and detection of anomalies.

One of the major contributions of the proposed ACS-IoT is the use of both machine learning and deep learning algorithms for the anomaly classification as well as type of attack being generated on the network traffic. To check whether if a network packet is normal or anomaly, we employ a binary classifier called Decision trees. The reason for using machine learning based algorithm for binary classification is that it has less complexity and requires less computation cost and time.

On the other hand, for multi-class classification (sub-type of anomalies), we have proposed a hybrid deep learning network CNN-BiLSTM. The multi-classification helps us to not only identify whether it is anomaly traffic but also helps us to distinguish the type of attack being generated on the network traffic. The deep learning approach self-learns the features necessary to detect anomalies and can perform intrusion detection accurately. It has more computational cost as compared to shallow network but results in more accurate and precise results. We proposed most optimal hyper-parameter tuning of the ACS-IoT including loss function, optimizer, batch size and epochs. Our proposed set of values for each hyper-parameter involved in the system will aid and assist other intrusion detection systems to achieve maximum performance and resource utilization.

The proposed system possesses the characteristic of automation in such a way that different modules of the proposed pipeline work in an automated fashion. The data is first loaded from the database and then multiple pre-processing operations are applied to it automatically in a cascaded way till the final classification of anomalies and normal packets. This process requires no human intervention and the whole process is executed automatically. One other aspect of automation in the proposed system is that since the data is analysed automatically, it can generate alarms or trigger automated actions upon the detection of attacks or anomalies. The proposed system is also heterogeneous in nature as the dataset used in this study contains raw network packet files generated from multiple devices such as an AI-based speaker, an EZVIZ Wi-Fi Camera, along with different smartphones and laptops. However, all these devices were connected to the same wireless network using a smart home Wi-Fi router. Since the dataset used in this study was acquired from multiple devices, therefore, M2M protocols characteristic also holds in the proposed study. However, as amount of data that is used in this study is fixed, so volume does not directly relate to this research study.

Our main contributions can be summarized as follows:

- We propose an Anomaly Classification System for IoT (ACS-IoT) consisting of two tiers which analyses network data to detect any intrusion or attack.
- We propose an efficient data preparation scheme involving feature selection using PSO and data balancing using SMOTE for building better neural network models.

- We propose a machine learning and deep learning-based pipeline for anomaly classification as well as type of attack being generated on the network traffic.
- We further propose a hybrid deep learning network CNN-BiLSTM with most optimal hyper-parameter tuning to efficiently perform anomaly type classification.

2. Related Work

Several researchers in the recent past have used machine learning and deep learning algorithms to develop intrusion detection systems for IoT. These systems mainly used supervised learning algorithms to perform classification of normal traffic and abnormal traffic data. These systems were trained with some known abnormalities in the network which were then used at real time for intrusion detection.

In order to understand the relationship between normal data and abnormal data in IoT system, regression analysis has been widely used [2]. For simple scenarios, where the relationship was not complex, simple linear regression was employed. On the other hand, for complex situations, logistic regression analysis has been used \cite{li2019system}. Both linear and logistic regression proved to simple and easy solutions but only for binary anomaly classification.

Support vector machine is another well-known and commonly used classification technique used for intrusion detection and classification. [3] proposed a method based on SVM in which during training it tries to maximize the margin between normal IoT data and abnormal data. It also uses a kernel trick to enhance the performance in more complex situations but selecting the kernel function with higher polynomials increases the complexity of the model.

Belonging to the same category as regression analysis and support vector machine, the decision trees are also used by [4] to classify the IoT data with given labels under supervised learning domain. A tree is generated with each leaf node leading to a classification result. Each leaf node could represent different anomalies and can be pruned for efficient performance. However, the size of the tree grows enormously with the addition of large-scale data.

Algorithms from unsupervised learning domain have also been used by the researchers for anomaly classification [5]. One of the classical algorithms for unsupervised learning is k-means clustering algorithm. Unlike other algorithms, this algorithm does not require labelled anomalies data and can work on unlabelled data in an unsupervised way. [6] used K-means to create a group or cluster of anomalies and normal data. These clusters helped to detect and prevent intrusions and abnormal activities but selecting the value of K is a trickier and tedious task.

Several optimization algorithms such as genetic algorithms, ant optimization and swarm optimization have also been used in the past [7] for anomaly classification. To improve the performance of intrusion detection system and to achieve maximum performance out of it, optimization algorithms are employed to tune the parameters. Once optimally

tuned, these algorithms can achieve highly accurate results in intrusion detection and prevention.

Artificial Neural Networks have also been used extensively [8] in intrusion detection systems for anomaly classification. Artificial neural networks are the building blocks of the modern-day deep networks which are being used for detection, classification, segmentation, and many other tasks. Perceptron is the basic building block of any neural network and a combination of multiple perceptron's in layers forms a multiple layer perceptron which is also termed as neural network. Neural networks also make the foundation of modern deep neural networks.

Deep learning algorithms have also been used widely to develop intrusion detection systems for IoT [9]. Just like machine learning algorithms, these systems also used supervised learning algorithms to perform classification of normal traffic and abnormal traffic. Some of these systems also performed classification under unsupervised learning domain. One such algorithm is auto encoder which is a type of unsupervised neural network and is used in machine learning domain to perform compression, classification, and many other related tasks. It employs back-propagation algorithm to set the values of the output to be equal to the inputs. [10] employed it in an inverse transform encoder-decoder fashion to perform anomaly classification.

Restricted Boltzmann Machine is another type of deep neural network which is used for classification, detection, and regression problems. It is also termed as energy-based model and is a shallow learning algorithm. It constitutes of a two layered neural network which also acts as a building block of deep belief network. One layer is termed as hidden layer and the other layer is known as visible layer. [11] used it for anomaly classification in network traffic. The input was provided to a node in the visible layer which was multiplied by its corresponding weight and bias term. After passing through the activation function, this is passed to the hidden layer where same phenomenon is performed, and a backward pass is established. After multiple forward and backward passes, the input is reconstructed which is used for anomaly classification.

Other deep neural networks used for anomaly classification includes deep belief network and recurrent neural network. A deep belief network is an advanced form of restricted Boltzmann machine (RBM) in which each machine is placed in the form of a stack. An RBM layer can communicate in both the directions with its previous layer and subsequent layers. To perform the classification, the RBM layers end up in a SoftMax layer, like a conventional neural network. Alrawashdeh et al. [12] made the layers in a deep belief network to work in two modalities. It acts as a hidden layer for all its previous layers, and it also acts as input layer for all the subsequent layers. However, the first and final layers don't possess these modalities.

The literature review revealed that most of the existing systems have high computational complexity as complex deep learning models were employed even for binary classification cases. Also, the data imbalance issue was not tackled properly due to which

the model was biased toward the anomaly classes that had more training instances. Furthermore, appropriate feature selection or optimization strategies were not taken into consideration during the training process. The proposed system addresses all these concerns which are detailed in the following section.

3. Proposed Methodology for Anomaly Classification

The proposed Anomaly Classification System for IoT (ACS-IoT) consists of two layers to analyze IoT data for attack detection and classification. These two layers work in a cascaded fashion to inspect and detect anomalies present in the IoT environment. Each layer serves as an essential building block and a vital component to detect intrusions in the system.

Two Layered Approach

We follow a two layered approach in our proposed anomaly classification system for efficient performance and accuracy. These layers are as follows: (i) Inspection layer (ii) Detection Layer. Brief detail of each layer is as under:

Inspection layer:

The goal of the inspection layer is to acquire and collect data from IoT surrounding. It consists of devices that are connected to the IoT system such as; Mobile phones, Tablets, IPod's, Cameras, RF-ID and other different computing devices. These devices captures and acquires data from network surroundings and transmits them over to the IoT system through network layer.

The role of this layer is to connect multiple devices with each other and to receive data from multiple layers present in the IoT architecture. Each device of the IoT will act as an edge and that is why this layer can also be considered as an edge which will perform data reception and data transmission. The data is received and transmitted in the form of network packets. Each network packet has a transmission port ID from where the data or packet has been generated. It also has a destination port ID containing the identification information of the destination device and many other parameters which are shown in the dataset.

This layer also contains all the information related to machine learning and deep learning algorithms which are being used to perform anomaly detection and classification. This layer also handles the training and testing part of the algorithms. The training is performed on the data that is pre-processed and cleaned. After that the trained weights produced by the computation layer are used in the detection layer to extract further useful information.

Detection Layer:

The role of the detection layer is to use the trained classifier received from the inspection layer to perform anomaly detection and classification. The trained classifier is first

validated on the validation data to ensure higher accuracy and performance. It then performs binary classification and multi-class classifications to not only distinguish between normal and anomaly data but to also identify the type of attack being generated on the network.

The detection layer is basically a service-oriented application and helps the users to deploy service based application into the IoT system. The role of the detection layer is to perform application level functionalities on the overall architecture of the IoT system. It serves as a means to apply and deploy the trained and validated machine learning and deep learning model on the IoT architecture for the sake of protection and detection from different types of attacks and anomalies being generated on the system.

Data Preprocessing

A publicly available IoT dataset [5], consisting of intrusions and normal data is used to develop the system. The most recent and state-of-the art dataset is selected for this purpose. However, this dataset has a number of issues and imbalance data is one of them which is resolved in the preprocessing step along with feature selection to reduce computation cost.

Imbalance data present in the dataset can cause the model to identify minority classes as majority classes. This can cause the model to be biased and consequently degrades the model's performance and overall quality of the classification task. To overcome this issue, we have employed SMOTE oversampling method which synthetically generates samples of minority class [13].

Figure 3 shows the anomaly samples distribution of IoTID20 dataset, prior and after balancing the dataset. It can be seen that the dataset contains around 5.7% of the MITM intrusion samples, while Mirai intrusion has 66.47% part of the dataset. In order to overcome the data imbalance issue, SMOTE is applied which over sampled the minority class to balance it with the majority classes.

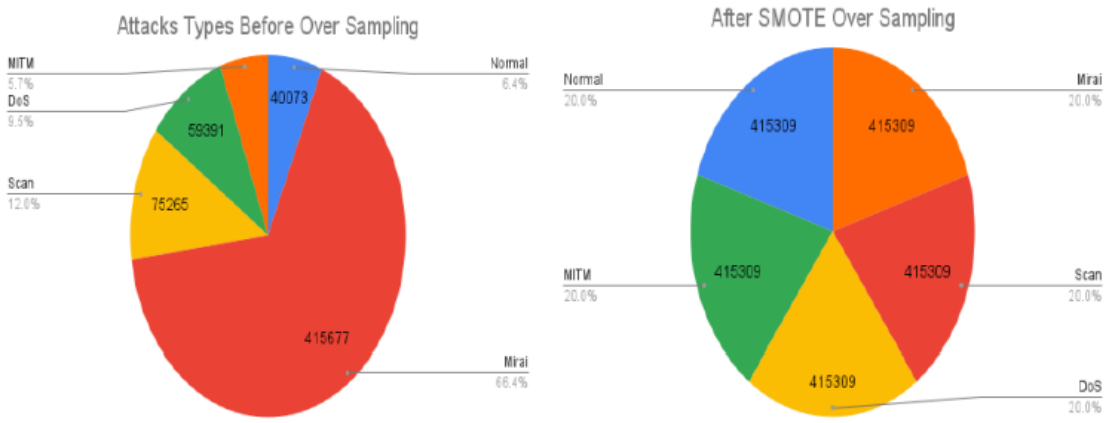

Figure 3. Anomaly samples distribution prior and after balancing

Another important function that is performed in the preprocessing step is that we have applied Particle Swarm Optimization (PSO) [14] as a feature selection technique in order

to select most important and distinguishing features from the input dataset. The feature selection is an important process to optimize the working of a system. A system without feature selection and reduction process try to employ all the features present in the dataset. Most of these features are not useful and does not play any role in the performance of the system. PSO not only reduced the computation cost but also saved the training time and memory. This also had an impact on the energy consumption of the system.

Machine learning and Deep Learning Pipeline

One of the major contributions of our proposed work is the use of both machine learning and deep learning pipeline for the anomaly classification as well as type of attack being generated on the network traffic. Figure 4 shows the basic relation between machine learning and deep learning pipeline. In order to check whether if a network packet is normal or anomaly, we employ a binary classifier called Decision trees. The reason for using shallow network for binary classification is that it has less complexity and requires less computation cost and time.

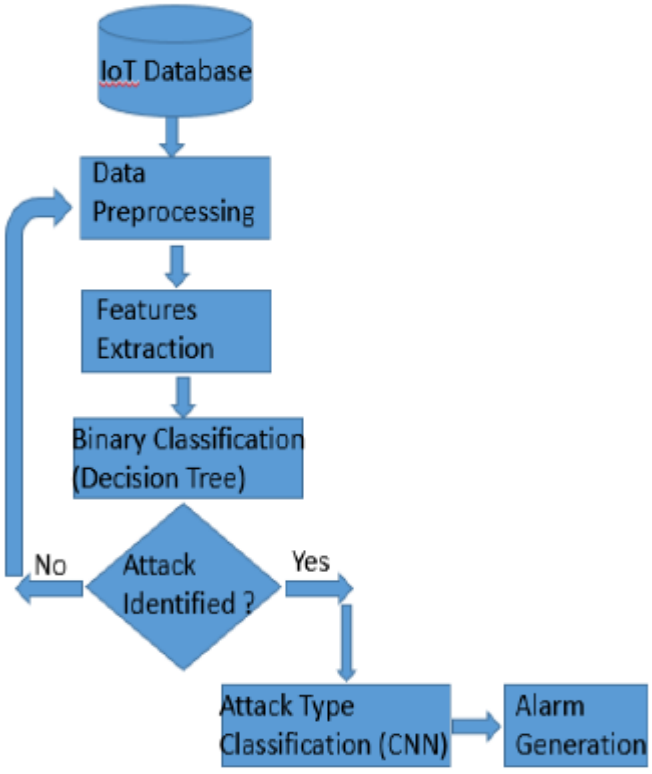

Figure. 4. Flowchart of machine learning and deep learning pipeline

On the other hand for multi-class classification, we have proposed a hybrid deep learning network CNN-BiLSTM with most optimal hyper-parameter tuning to efficiently perform anomaly type classification. The multi-classification helps us to not only identify whether it is anomaly traffic but also helps us to distinguish the type of attack being generated on the network traffic. It has more computational cost as compared to shallow network but results in more accurate and precise results.

The machine learning algorithms are beneficial in one way that they are simple in nature and are less compute and resource intensive as compared to deep learning algorithms. The training and inference time of machine learning algorithms is much lower as they are usually trained on small number of training samples. They have less number of parameters which could easily be learned from small training samples. But this limits their capability to solve only simple problems such as binary classification problems (normal vs anomaly) and cannot perform well on complex problems such as multi-classification (attack type classification).

On the other hand, deep learning algorithms have large training and inference times as they have large number of parameters and usually require large amount of training samples to learn underlying features present in the data. The deep learning models are data driven models and necessitate the inclusion of large number of training samples to understand varying feature representations present in the dataset. The downside of deep learning models is that they are highly compute intensive, require specialized compute resources to perform training and inference from data. But the advantage is that they can solve very complex problems such as multi-classification problems.

In our proposed machine learning and deep learning pipeline, we have gained the advantages of both these domains by employing them in a 2-layer based approach. In first layer we have employed Decision Tree classifier which is a machine learning based classifier and works efficiently for binary classification problems. It only tells that if there is an anomaly or not. It cannot classify the type of anomaly as it is a less compute intensive layer and only works as a filter to classify between normal and anomaly data. If there is no anomaly then the control will not pass on to second layer and the IDS will keep on monitoring the IoT network. This keeps the overall system lightweight and reduces the computation costs without compromising on the IoT vulnerability.

If the first layer detects an anomaly then the control moves on to the second layer. The second layer is a compute intensive layer and employs a hybrid CNN-BiLstm model to accurately identify the type of attack being generated on the IoT network. Since Decision Tree and other machine learning based classifiers were unable to accurately identify the type of attack being generated on the IoT network, therefore, we have to use a complex deep learning model. The proposed hybrid CNN-BiLstm model classifies the type of attack accurately with high precision and recall. Once the attack type is classified, appropriate security measures could be taken into consideration.

Model Training and Testing

To design most optimal intrusion detection system based on deep learning algorithms, it is crucial to select most appropriate hyper-parameter values along with its basic architecture. We tested with multiple configurations by varying different hyper parameters of the model which include: 1. Number of Layers 2. Loss Function 3. Number of Epochs and came up with the most optimal architecture. Multiple configurations help

to identify the layers sweet point of the model. The architecture of the proposed hybrid CNN-BiLstm model is shown in Figure 5.

Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 52, 64)	7872
max_pooling1d (MaxPooling1D)	(None, 10, 64)	0
batch_normalization (BatchNormalization)	(None, 10, 64)	256
bidirectional (Bidirectional)	(None, 128)	66048
reshape (Reshape)	(None, 128, 1)	0
max_pooling1d_1 (MaxPooling1D)	(None, 25, 1)	0
batch_normalization_1 (BatchNormalization)	(None, 25, 1)	4
bidirectional_1 (Bidirectional)	(None, 256)	133120
dropout (Dropout)	(None, 256)	0
dense (Dense)	(None, 5)	1285
activation (Activation)	(None, 5)	0

Figure. 5. Architecture of CNN-BiLstm model

We have a convolutional layer with 64 input kernels with a kernel size of 122. This layer employs 'relu' as an activation function. The following layer is the pooling layer which perform max pooling operation to reduce the resolution size. The pool size is selected to be 5 in this layer. After that batch normalization is performed which leads to the subsequent layer of bidirectional lstm layer with a total of 64 kernels in it. This follows to the alternating layers of max-pooling, batch normalization and bidirectional lstm layer leading to the fully connected layer for classification. This layer employs softmax as the activiation function to generate probabilities against the anomaly classes with a drop out value of 0.5.

Multiple loss functions for the tuning of the model for higher anomaly classification rates are tested. These loss functions include: 1. Jaccard Loss Models 2. Binary and Categorical Cross Entropy Loss Models 3. Mean Squared Error Loss Models. It was observed from the results that which loss function gives the best estimation for the model to improve as it has the highest validation or accuracy. So it was noted that Categorical Cross Entropy Loss Model provided the best results with 'adam' as an optimizer function. The overall model architecture comprises of 208,585 parameters from which 208,455 are trainable parameters and 130 are non-trainable parameters.

A train test split was made with a test size of 0.2. It means that 80 percent of the data was used to train the proposed hybrid CNN-BiLstm model and 20 percent of the data was used to test the model. Stratified Kfold cross validation was used during training, with a

batch size of 256. The loss and accuracy for training set as well as for validation was observed for each epoch. The model seemed to be converging with each and every epoch.

Hyper-parameter Optimization

We have performed hyper-parameter tuning of the system, so that it is efficiently tuned and perform better. The tuning is performed on the following parameters:

Loss Function:

The goal of the neural network during training is to reduce the loss function. This means to reduce the difference between computed and targeted values. We have mainly used three loss functions including Binary and Categorical Crossentropy, Mean Square Error and Jaccard. The metrics and loss plots for each of the aforementioned loss functions are calculated respectively. It was noted that Categorical Cross Entropy Loss Model provided the best results.

Optimizer:

We also used the optimizer functions including Gradient Descent, Stochastic Gradient Descent and Adam. We propose to use 'Adam' optimizer with Categorical Crossentropy as it is an efficient algorithm developed to reduce the loss function in the quickest way possible.

Batch Size:

We have divided the dataset in a number of mini-batches during training process. We suggest that 256 is most efficient batch-size for the training of the model. Batch size means that we will divide the input data into a number of batches and process each batch in parallel.

Epochs and Steps:

Epoch and steps corresponds to the number of times to run the model in the training phase. The execution of the model in the training phase is an iterative process. We can add and remove additional epochs, but the accuracy might not change much.

4. Implementation Details

This section describes the implementation details of our proposed system. The proposed system is executed on Google Colaboratory which facilitates the execution of arbitrary python code with GPU back-end. The compute resources used in this study consists of Nvidia K80 GPU with a memory of 12 GB. A total disk space of 358 GB was utilized to handle the data with 12 GB of RAM. The GPU memory clock was utilized at 0.82 GHz with a performance of 4.1 TFLOPS.

The dataset is first loaded into the system memory using Pandas library. After that the labels are extracted from the dataset and removal of strings are performed which have a datatype of 'object'. This is because the type object is mostly strings and does not contribute to the training of the model.

In pre-processing, we first check whether the dataset contains any infinite or NaN values, if yes, they have to be removed in order to produce efficient results. The dataset is huge and contains large values which eventually takes a lot of time to train the model. This is why we normalize the dataset so that each feature of the input would contain smaller values and produce efficient results.

The IP addresses are given in string format in the dataset. For using them as our feature data, we converted them into int or float data types. First we have converted the Source and Destination IPs into integer format using IP address library. After that feature values are assigned to label values as both of them are present in different containers.

After that, we converted different categories Cat column into numbers for which we have used dummies libraries dividing each category into different column. Then we have merged those converted categories with the input table and dropped the column of category which contains string values.

After performing all the basic necessary operations we dealt with the data imbalance issue present in the dataset. It was observed that the dataset contains around 5.7% of the MITM intrusion samples, while Mirai intrusion has 66.47% part of the dataset. To overcome this issue, SMOTE is applied which over sampled the minority class to balance it with the majority classes. SMOTE oversampling method synthetically generates samples of minority class. This is achieved by taking into account the feature space of samples present in the dataset and their inherent relationships.

The main working of SMOTE algorithm is described in Algorithm 1. The input parameters to the algorithm includes the number of minority class instances, the rate or percentage at which the oversampling needs to performed, the value of hyper-parameter 'K' for the nearest neighbour algorithm and the granularity parameter. After complete execution, the algorithm generates the synthetic over sampled data as an output. The oversample minority class is initiated with original minority class samples. Then the feature ranking mechanism is employed to calculate the order inducing variable 'y'. it is used in the IMOWAD distance measuring for the derivation of its weights. The IMOWAD distance function calculates the neighbourhood of size 'k' for each minority sample present in minority class. After that randomly selected samples are interpolated to generate synthetic samples. This process is repeated again and again to generate balanced samples against the minority classes.

Algorithm 1 Data balancing algorithm using SMOTE

0: **Input** ← minority instances (M_T); oversampling percentage ($N\%$); value for nearest neighbors (k); granularity parameter (α)
0: **Output** ← synthetic data (S_D)
0: S^* ← the minority class instances (M_T)
0: y ← mechanism to perform feature ranking
0: c ← granularity parameter (y, α)
0: P ← maximum value of cas 'r'
0: **for** $i \in T$ **do**
0: **for** i' ∈ T, i'≠ i **do**
0: IMOWAD(x_i, x_j, w); distance measure operator
0: **end for**
0: **for** k **do** ← 1 **to** N
0: x_k ← selects random samples from M_T
0: x'_k ← perform interpolation to generate synthetic data
0: T^* ← synthetic data added to S^*
0: S_D ← remove specific sample x_k from P
0: **end for**
0: **end for**=0

We have also applied Particle Swarm Optimization (PSO) \cite{shafipour2021particle} as a feature selection technique in order to select most important and distinguishing features from the input dataset. PSO not only reduced the computation cost but also saved the training time and memory. This also had an impact on the energy consumption of the system.

Particle Swarm Optimization is an optimization strategy in which the whole search space is optimized with the help of principal space method. The swarm in the PSO method is referred to the most optimized solution which is obtained after processing the particles present in the search space. The fitness of the swarm particle is measured through a fitness function which returns its output as either '1' or '0'. '1' indicated the higher fitness of a particle and it is highly probable to get selected. Each particle updates its position along with the velocity randomly in the search space to generate more features. The strength and effectiveness of the PSO algorithm highly relies on these randomly generated features and the search space.

The main working of PSO algorithm is described in Algorithm 2. The input parameters to the algorithm includes the fitness function and the total population. Also, the algorithm takes the total number of particles as input represented . After complete execution, the algorithm provides the selected features as an output. The objective of the fitness function is to measure the fitness of the particles. In each iteration, until the optimal features are not selected, the algorithm compares the fitness of the current particle with the most fittest particle of the whole population. After that, the values are updated accordingly. The whole process is repeated until the most optimal features have been achieved.

Algorithm 2 Feature selection using PSO

Input ← Fitness function as $A^n_{counter}$, Total population $Pop_{counter_{fittest}}$ Total number of particles T
Output ← Selected features
while (! Optimal features) **do**
  for (counter =1 to T) **do**
    if $A^n_{counter} > Pop_{counter_{fittest}}$ **then**
      Update $Pop_{counter_{fittest}} = A^n_{counter}$
      if fitness $A^n_{counter} > H_{fittest}$ **then**
        Update $H_{fittest}$ = fitness $A^n_{counter}$
      end if
    end if
    Update velocity $D^{t+1}_{il} = \{w * D^t_{il} + acc1 * r_{1i}(p_{il} - x^t_{il} + acc2 * r_{2i}(p_{gl} - x^t_{il})\}$
    Update position $D^{t+1}_{id} = \{D^l_{id} + D^{l+1}_{id}\}$
    Next particle in the population
  end for
  repeat until convergence
end while=0

After finalizing the preprocessing, we split the dataset into train and test set. For this purpose, we have used sklearn library. 70 percent of the dataset is used as training data while the remaining 30 percent is used as test data. After that training of the system is performed with highly tuned hyper-parameters to perform intrusion detection.

We train both machine learning and deep learning networks under supervised learning domain over large amount of data. The training phase extracts useful features from the data in order to distinguish between normal data and anomaly data. We train both binary classifiers and multi-class classifiers to not only distinguish between normal and anomaly data but also the type of attack being generated on the network by attacker.

Initially we build a binary classifier and then we create a multi-class classifier. In binary classifier we have two classes in the dataset. Normal and Anomaly. Since it still have the datatype as object and would produce incorrect results while training the model. We have to encode the label classes as well to convert them into integers. Since we have only two classes, the labels would be 0 and 1 for anomaly and normal respectively.

For multi-class classification, we have employed a hybrid deep learning network CNN-BiLSTM with most optimal hyper-parameter tuning to efficiently perform anomaly type classification. The multi-classification helps us to not only identify whether it is anomaly traffic but also helps us to distinguish the type of attack being generated on the network traffic. It has more computational cost as compared to shallow network but results in more accurate and precise results.

We have trained the Decision trees classifier using our training data for binary anomaly classification. The reason for using machine learning based algorithm for binary classification is that it has less complexity and requires less computation cost and time. For multi-class classification, we have employed deep learning network. It has more

computational cost as compared to shallow network but results in more accurate and precise results.

The deep neural network is a hybrid deep learning network of CNN-BiLSTM with most optimal hyper-parameter tuning to efficiently perform anomaly type classification. The proposed hybrid model has alternating layers on convolution, pooling, batch normalization and bidirectional LSTM. The convolutional layer has 64 filters with a kernel size of 122, employing relu activation function. A pool size of 5 is used in max pooling layer, while the bidirectional LSTM has 64 filters and the layers are repeated afterwards terminating to a Softmax activation layer. The final model employs Adam optimizer with categorical crossentropy as the loss function. The proposed hybrid CNN-BiLSTM model has a total of 208,585 parameters, from which 208,455 are trainable parameters and 130 are non-trainable parameters.

The trained classifier whether it is Decision trees classifier or hybrid deep network is tested on the test dataset and then it is used in the intrusion detection system to detect and classify anomalies.

5. Experimental Setup

This section provides an overview of the experimental setup being used in this work. We provide a detailed description of the dataset used in this research work by highlighting its advantages. We then describe the tools and technologies that are used for the development and execution of the proposed system. We then describe the evaluation parameters that are used to evaluate the proposed system.

Datasets

The dataset that is used in this work is New IoTID20 dataset [5], which is developed for anomalous activity detection in IoT networks. The dataset is publicly available in CSV format and is free to be used for research purposes. It contains different types of IoT attacks as well as their families.

One of the advantages of using this dataset is that it has a large number of features which are general in nature. Another advantage is that it also has a large amount of flow based features present in the dataset. All the features are also ranked with most of them as high rank features.

A number of anomalies and attacks with different types of network attacks present in the Internet of Things (IoT) environment for has been created in this dataset for academic purpose. The smart devices including some laptops or smart phones are used for data generation which were connected to the same wireless network.

The dataset consists of three major categories. One is binary which has a binary label distribution consisting of 40073 Normal instances and 585710 Anomaly instances. The second category has a category label distribution consisting of 40073 Normal instances, 59391 instances of DoS, 415677 instances of Mirai, 35377 instances of MITM and 75265 instances of Scan attack. The third category has a subcategory label distribution with

40073 Normal instances, 59391 instances of DoS, 55124 instances of Mirai Ack Flooding and some other attack categories.

All the attacks present in the dataset excluding Mirai Botnet category are the packets captured while simulating attacks. The tools which are used to simulate the attacks include Nmap. However, for the atatck category of Mirai Botnet, the packets for the attacks were generated by using a computer system which were further manipulated to make it appear as if it originated from the IoT device.

Tools and technologies

The tools and technologies that are used for the implementation and development of proposed anomaly detection system for IoT mainly consists of Python developmental language. A virtual environment is created for python code development in Anaconda. For deep learning model creation, Keras framework based on Tensorflow library has been used. We will also employ Pandas library for data manipulation and MatPlotlib for visualization of data and results.

Evaluation parameters

In order to evaluate the performance of the proposed system, a number of evaluation parameters will be used. These evaluation parameters will help to understand impacts of different hyper-parameters tested on the proposed model. These evaluation parameters will also help to compare the performance of the proposed system with existing models.

Confusion Matrix:

A confusion matrix is often used to describe the performance of a classification model on a set of test data that is unseen to the classifier. The confusion matrix is generated in the form of a table which is relatively simple to understand.

Accuracy:

Accuracy refers to the closeness of a measured value to a standard or known value. It means that the measurements for a given object are close to the known value, but the measurements are far from each other, then it means that the accuracy is without precision. Accuracy is calculated as:

Accuracy = TP+TN / TP+TN+FP+FN

Precision:

Precision is the quality of being exact. It refers to how close two or more measurements are to each other, regardless of whether those measurements are accurate or not. Precision is calculated as:

Precision = TP / TP+FP

Recall:

Along with precision, recall is also used as a basic performance measure. The precision measures the quality while recall is a measure of the quantity. Recall is calculated as:

Recall = TP / TP+FN

F1 Score:

F1 score is used to further elaborate the precision and recall and can be considered as the harmonic mean of precision and recall. It can be a good performance measure for imbalanced datasets. F1 Score is calculated as:

F1 = 2*TP / 2*TP+FP+FN

6. Results and Discussion

In this section, we present the results of our proposed system. We first present the results for binary classification case in which we are detecting and classifying whether there is anomaly data or normal data in the IoT network traffic.

After that we present the results for multi-classification case in which we are detecting and classifying not only anomaly data or normal data but also the type of anomaly present in the IoT network traffic. We present the results for both of our machine learning and deep learning based proposed methods.

It was observed during experimentation that the use of PSO as a feature selection mechanism proved to be an appropriate feature selection technique and helped reducing the training time of the system. The training time was improved for both binary classification case and also for multi-classification case.

Results for Machine Learning based Anomaly Classification}

Figure 6 shows the performance parameters of three different classifiers for the classification of whether there is anomaly present in the data or the data is in normal condition. It can be seen from the figure that decision tree is performing quite well for this task. It can be clearly seen that the performance of the decision trees is better than the naive Bayes classification and Neural network classifier.The accuracy and precision are more than 90%, which is quite good results for binary anomaly classification tasks.

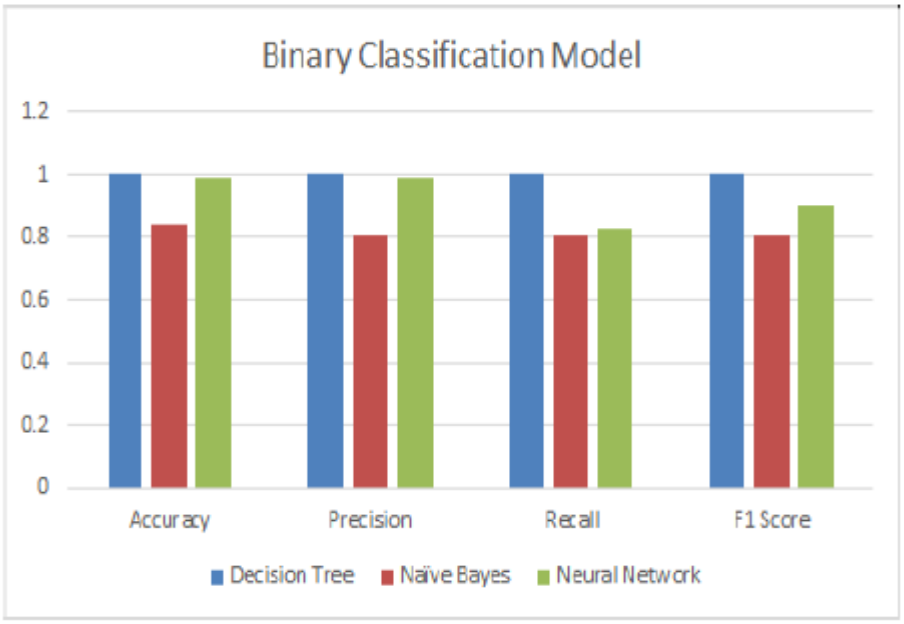

Figure 6 Performance parameters of three different binary-class classifiers

Figure 7 shows the accuracy of the neural network during training and Figure 8 shows loss over multiple epochs. As it can be seen from the figure that the accuracy of the system is increasing with each and every increase in the number of epochs. Especially till 0.975 that increase in accuracy is almost perpendicular, however, it reaches to a stable state after that and could not outperform decision tree after full training.

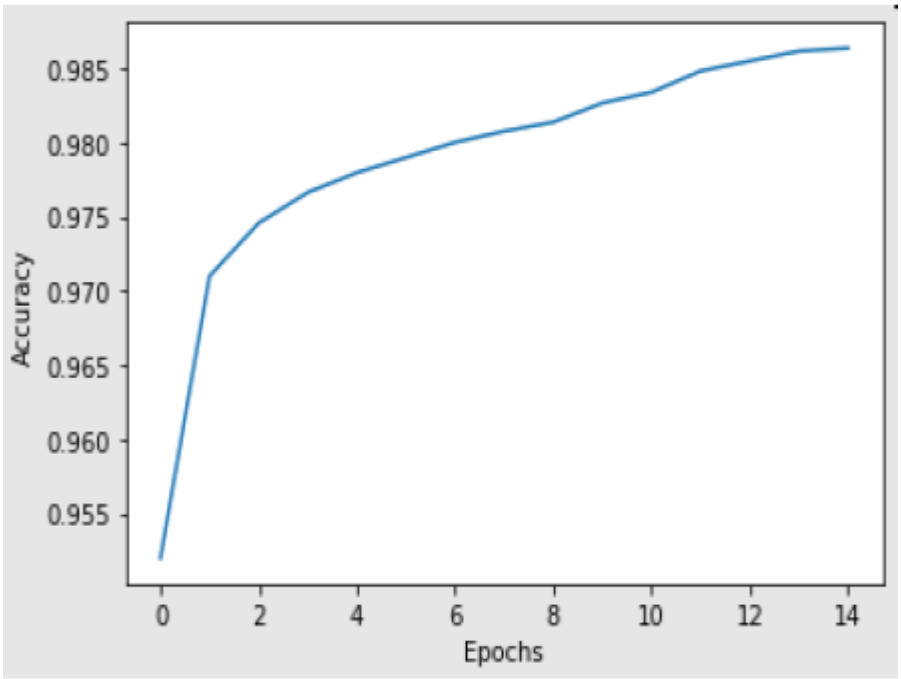

Figure 7 Accuracy Curve

The same behaviour can be observed in the loss figure 8 of the neural network during training. It can be noted from the figure that the loss of the network is decreasing with each and every epoch. It is also quite observable that the loss of the network is very

minimum after 14 epochs of iterations but again it maintained a a stable state after that and could not outperform decision tree.

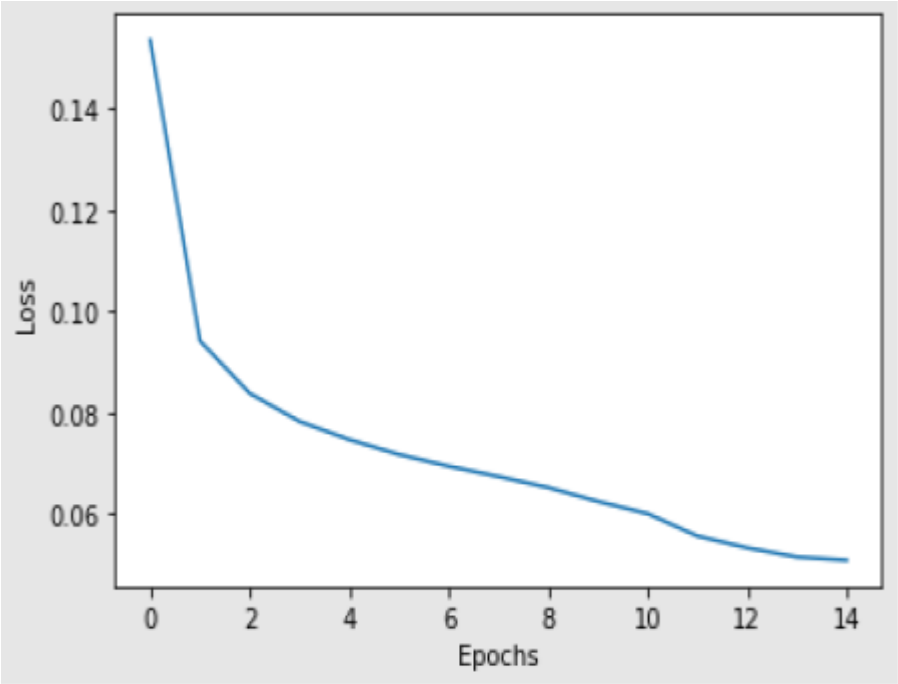

Figure 8 Loss Curve

Results for Machine Learning based Anomaly Type Classification

Figure 9 shows the performance parameters of three different classifiers for the classification of the type of anomaly present in the data. It can be seen from the figures that, again, the decision tree is performing quite well for this task. The accuracy and precision are close to 80% and the recall, F-measure, and F1 score are also quite good for multi-anomaly classification tasks.

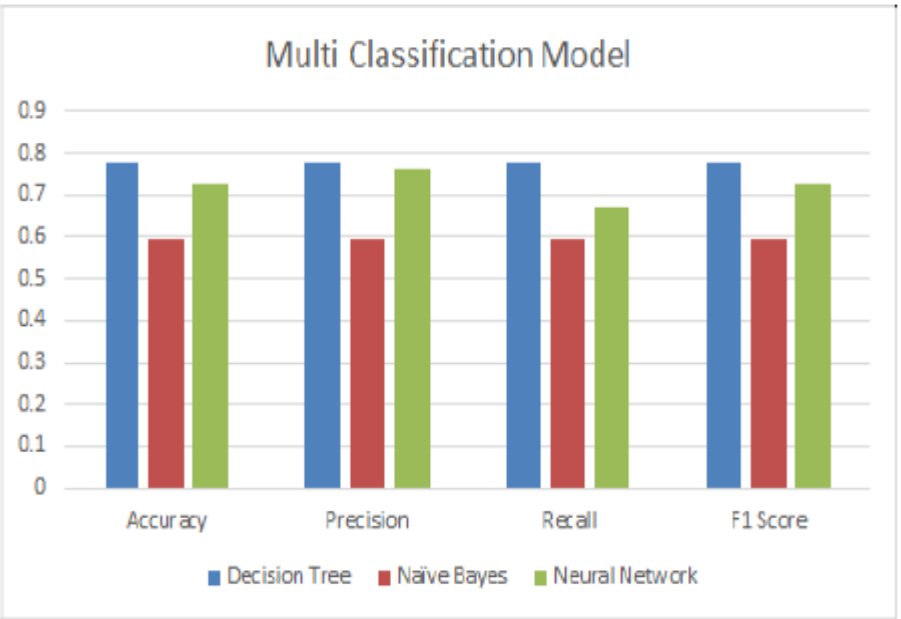

Figure 9 Performance parameters of three different binary-class classifiers

Figure 10 shows the accuracy of the neural network during training and Figure 11 shows loss over multiple epochs. As it can be seen from the figure that the accuracy of the system is increasing with each and every increase in the number of epochs. Especially till 0.725 that increase in accuracy is almost perpendicular, however, it reaches to a stable state after that and could not outperform decision tree after full training.

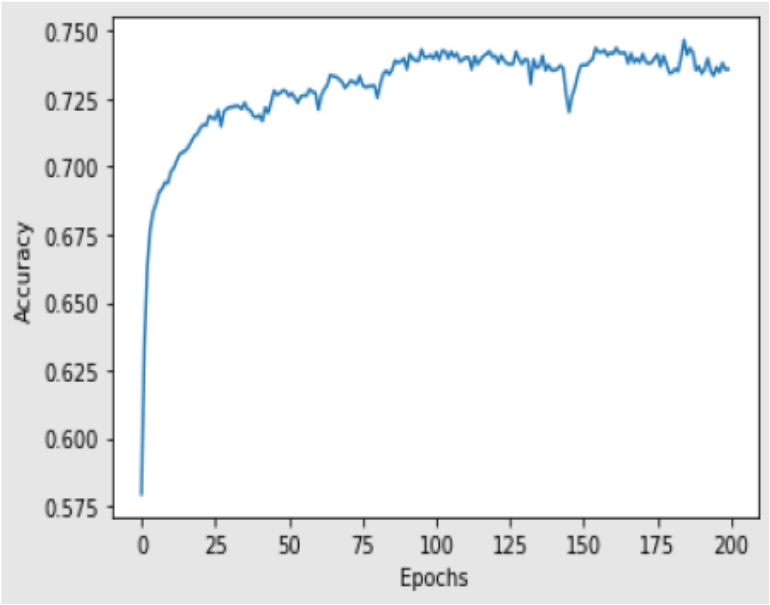

Figure 10 Accuracy Curve

The same behaviour can be observed in the loss figure 11 of the neural network during training. It can be noted from the figure that the loss of the network is decreasing with each and every epoch. It is also quite observable that the loss of the network is very minimum after 200 epochs of iterations but again it maintained a a stable state after that and could not outperform decision tree.

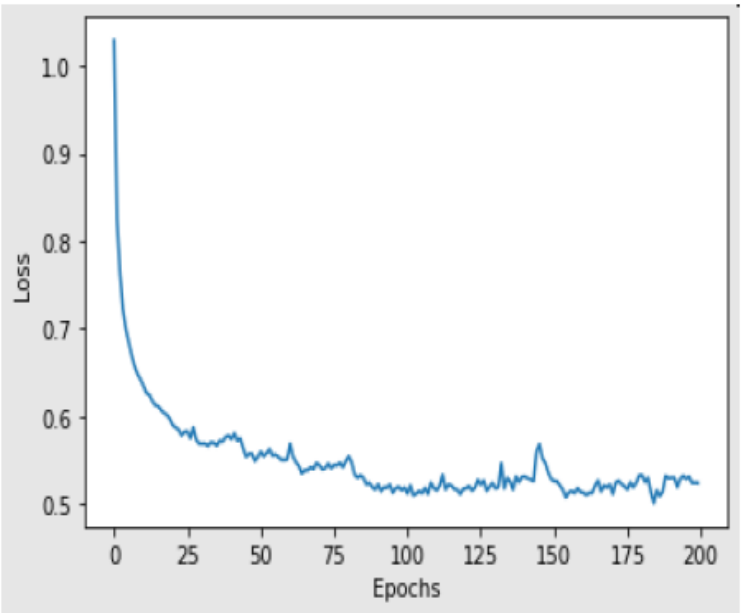

Figure 11 Loss Curve

Hybrid Deep Learning Model for Anomaly Classification

The deep learning based hybrid classifier CNN+BiLSTM is also trained for both binary case in which we have established that whether any anomaly is present in the data or if it is normal data. The deep learning based classifier performed quite well as can be seen in Figure 12.

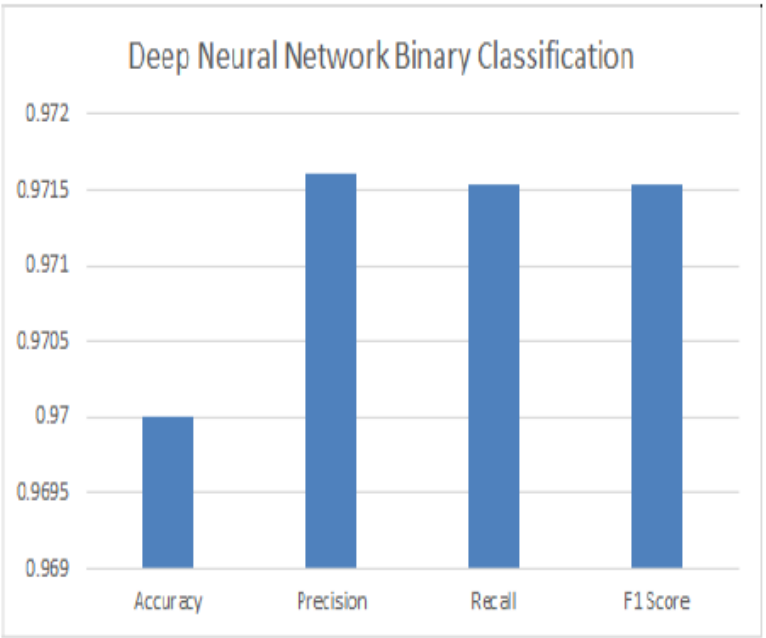

Figure 12 Performance parameters of hybrid deep learning model for binary-class classification

It can be observed that there is not much difference in the accuracy of deep learning network for binary classification when we compare it with the accuracy of machine learning network for binary classification. However, the difference between the training time of machine learning network is quite low as compared to deep learning network for

binary classification. Same behavior was observed for the remaining performance measures including precision, recall and F1 score. Therefore, for binary classification, machine learning based classifier was selected for the proposed system.

Hybrid Deep Learning Model for Anomaly Type Classification

As discussed earlier that the deep learning based hybrid classifier is also trained for both binary case in which we have established that whether any anomaly is present in the data or if it is normal data. The deep learning based classifiers performed quite well for anomaly type classification task as well, as it can be seen in Figure 13.

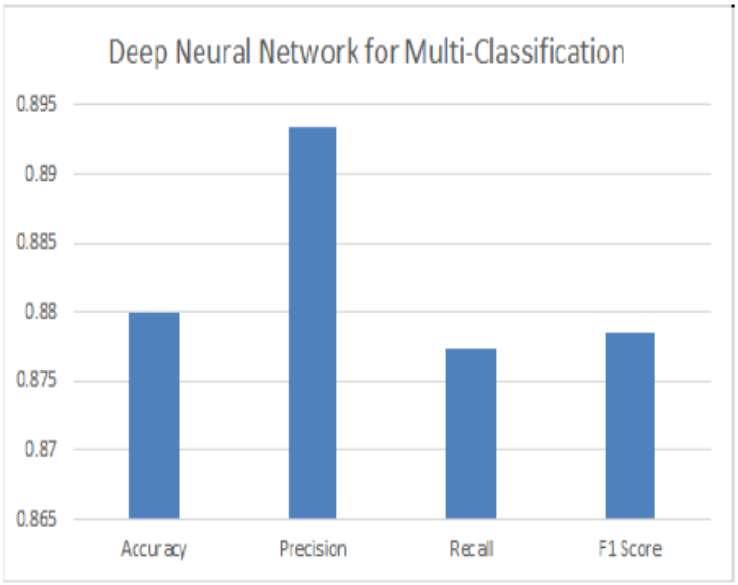

Figure 13 Performance parameters of deep learning model for binaryclass classification

It can be observed that there is a greater difference in the accuracy of deep learning network for multi-class classification when we compare it with the accuracy of machine learning based algorithm for multi-class classification. Although, there is difference between the training time of machine learning network as compared to deep learning network for multi-class classification but we are also achieving higher accuracy rates. Same behavior was observed for the remaining performance measures including precision, recall and F1 score. Therefore, for multi-class classification, deep learning based classifier was selected for the proposed system.

We have also performance comparison of the proposed model with state-of-the-art models for Multiclass classification. It can be observed from Figure 14 that there is a greater difference in the accuracy of the proposed model for multi-class classification when we compare it with the accuracy of state-of-the-art models for multi-class classification. The same behavior was observed for the remaining performance measures including precision, recall, and F1 score.

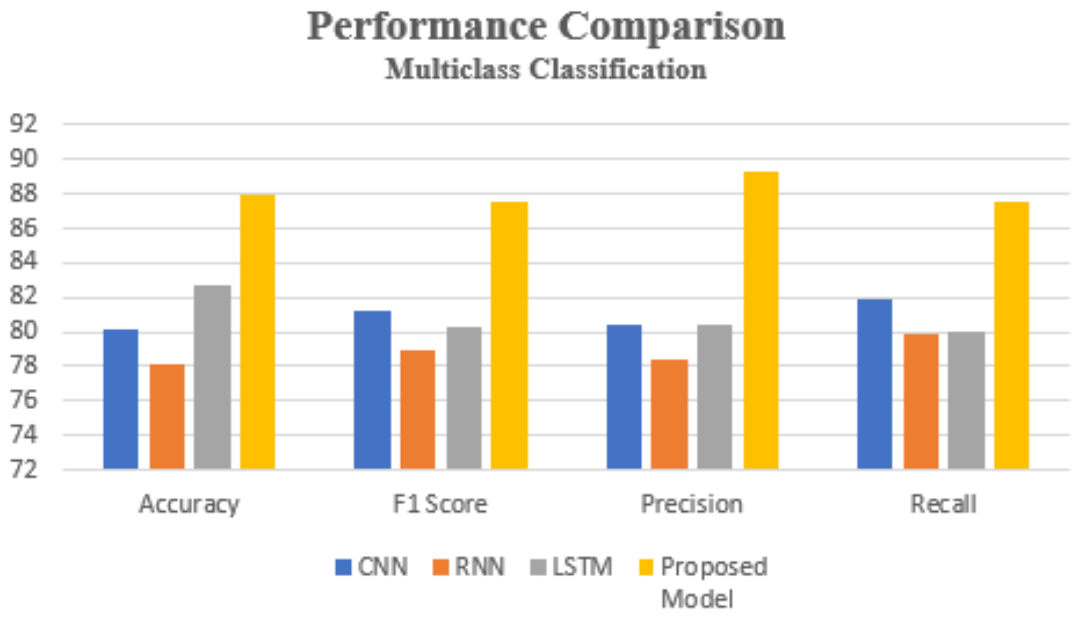

Figure 14 Performance comparison of deep learning models for multicass classification

7. Conclusion

In this work, we proposed an Anomaly Classification System for IoT (ACS-IoT) consisting of two layers to perform anomaly classification efficiently. Each layer has its own functionalities starting from data acquisition and preprocessing to complex machine learning based hybrid pipeline for binary and multi-class anomaly classification. We propose an efficient data preparation scheme for building better neural network models. Our data preparation scheme involves preprocessing of data including dimension reduction, feature selection and normalization.

One of the major contributions of our proposed work is the use of both machine learning and deep learning pipeline for the anomaly classification as well as type of attack being generated on the network traffic. The reason for using machine learning for binary classification is that it has less complexity and requires less computation cost and time. The deep learning based hybrid approach self-learns the features necessary to detect anomalies and is able to perform intrusion detection accurately.

We also propose most optimal hyper-parameter tuning of the intrusion detection system including loss function, optimizer, batch size and epochs. Our proposed set of values for each hyper-parameter involved in the system can assist other intrusion detection systems to achieve maximum performance and resource utilization.

The proposed system has a number of application not only in IoT but many other domains such as safe cities, smart cities and sensor networks. The proposed approach is also feasible for not only wireless systems but also for wired systems. With minor modifications and tweaks, the system can be deployed on large scale real life infrastructures. Companies can use this system to better analyze the types of attacks

being generated by the intruders, which can help the organization to improve their security mechanisms and employ more effective controls. The proposed system can also help the network administrators to monitor the traffic and identify bugs or anomalies present in their network device configuration.

8. Future Work

In the future we would like to extend this work by working on more complex datasets incorporating more real life attacks, anomalies and challenges. We would also like to work on other machine learning and deep learning based models for anomaly classification and intrusion prevention. One side which needs to be emphasized more is the optimal configuration of models specifically for the attacks and anomaly classification for intrusion prevention.

Another important aspect on which we would like to work in future is an optimal and efficient feature selection and dimension reduction strategy. We will propose and efficient approach which will not only reduce the number of features required for correct classification but will also improve the overall accuracy and performance of the system. This will also help to reduce the computation cost and time of the system.

Another important aspect is to use the hardware accelerators such as GPUs for the efficient performance of the system. We will develop a high performance platform which will be equipped with modern GPUs and will help to accelerate our proposed anomaly classification pipeline for intrusion detection and prevention. We intend to embed the proposed system before the router near to the firewall. This will enhance the performance and efficiency of the proposed system and will not let the malicious data to reach the user.

References:

[1] Mukherjee, S., Gupta, S., Rawley, O., Jain, S., 2022. Leveraging big data analytics in 5g-enabled IoT and industrial IoT for the development of sustainable smart cities. Transactions on Emerging Telecommunications Technologies.

[2] Fahim, M., Sillitti, A., 2019. Anomaly detection, analysis, and prediction techniques in iot environment: A systematic literature review. IEEE Access

[3] Li, F., Shinde, A., Shi, Y., Ye, J., Li, X.Y., Song, W., 2019. System statistics learning-based iot security: Feasibility and suitability. IEEE Internet of Things Journal

[4] Ever, Y.K., Sekeroglu, B., Dimililer, K., 2019. Classification analysis of intrusion detection on nsl-kdd using machine learning algorithms, in: International Conference on Mobile Web and Intelligent Information Systems, Springer

[5] Ullah, I., Mahmoud, Q.H., 2020a. A scheme for generating a dataset for anomalous activity detection in iot networks, in: Canadian Conference on Artificial Intelligence, Springer

[6] Liu, X., 2021. A real-time detection method for abnormal data of internet of things sensors based on mobile edge computing. Mathematical Problems in Engineering 2021.

[7] Yang, Z., Abbasi, I.A., Mustafa, E.E., Ali, S., Zhang, M., 2021. An anomaly detection algorithm selection service for iot stream data based on tsfresh tool and genetic algorithm. Security and Communication Networks

[8] Gaifulina, D., Kotenko, I., 2021. Selection of deep neural network models for iot anomaly detection experiments, Euromicro International Conference on Parallel, Distributed and Network-Based Processing

[9] [Shareena et al.(2021)Shareena, Ramdas, AP et al.] Shareena, J., Ramdas, A., AP, H., et al., 2021. Intrusion detection system for iot botnet attacks using deep learning. SN Computer Science

[10] Hou, Y., He, R., Dong, J., Yang, Y., Ma, W., 2022. Iot anomaly detection based on autoencoder
and bayesian gaussian mixture model. Electronics

[11] Coli, G.O., Aina, S., Okegbile, S.D., Lawal, A.R., Oluwaranti, A.I., et al., 2022. Ddos attacks detection in the iot using deep gaussian-bernoulli restricted boltzmann machine. Modern Applied Science

[12] Malik, R., Singh, Y., Sheikh, Z.A., Anand, P., Singh, P.K., Workneh, T.C., 2022. An improved deep belief network ids on iot-based network for traffic systems. Journal of Advanced Transportation 2022.

[13] Li, J., Zhu, Q., Wu, Q., Fan, Z., 2021. A novel oversampling technique for class-imbalanced learning based on smote and natural neighbors. Information Sciences

[14] Shafipour, M., Rashno, A., Fadaei, S., 2021. Particle distance rank feature selection by particle swarm optimization. Expert Systems with Applications.

[15] Ali, Z.H., Ali, H.A. and Badawy, M.M., 2015. Internet of Things (IoT): definitions, challenges and recent research directions. International Journal of Computer Applications