National College of Ireland

Anomaly Detection in a Network Intrusion using a Software-defined Network and Deep Learning.

MSc Research Project
Research Project/Internship

Abiodun Ali
Student ID: x19209347

School of Computing
National College of Ireland

Supervisor: Vikas Sahni

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Abiodun Ali……

Student ID: x19209347…….

Programm: ……MSCCYBE……Year:2021-2022..

Module: …… Research Project/Internship……

Supervisor: …….. Vikas Sahni …….

Submission Due Date: …August 15, 2022…….

Project Title: Anomaly Detection in a Network Intrusion using a Software-defined Network and Deep Learning. ……

Word Count: ……6248…… **Page Count**……20……

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: ……Abiodun Ali……

Date: ……27/06/2022……

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Anomaly Detection in a Network Intrusion using a Software-defined Network and Deep Learning.

Abiodun Ali
Student ID: x19209347

Abstract

Hackers utilize a multi class group of network threats to elude the security mechanisms of networks. The deficiencies of network systems combined with the evolving methods of attack have created opportunities for hackers to exploit network systems. The motivation for the research work originated from the need to identify the best algorithm of Advanced Machine Learning (AML) that will detect network threats.
The study used three different Deep Learning models to investigate the performer at identifying cyber-threats in networks. The Accuracy of the Cu-DNN model across all classes was 97.40%, that of the Cu-GRU model was 98.24%, while the accuracy of the Cu-DNNGRU was 99.11%. The precision of the Cu-DNN model was the lowest with 96%, the Cu-GRU model was 98.64%; and the Cu-GRU model was 98.47%. The Cu-GRU model recorded the lowest Recall with 98.3%, the Cu-DNN model is 98.6%, while the highest Recall score was recorded for the Cu-DNNGRU model with 99.2%. Other performance metrics assessed were the F1-score, false detection rates, and true detection rates. The study concluded that the Cu-DNNGRU was the best of the three models at detecting network threats.

Keywords—SDN, threats detection, attacks, Advanced ML, Deep Learning

1 Introduction

Computer Networks play a pivotal role in modern organisational structure; hence the need to keep them secure, which is a major challenge in today's interconnected world. Networks are the basis of communication in IT infrastructure, as they are made up of various types of networks. However, networks are a to numerous vulnerabilities, which is a perfect precursor for malicious attacks. Commonly, the security of networks can be breached with various attacks, including the Denial of Service (DoS), Distributed Denial of Service (DDoS), Man in the Middle, and SQL injections. In recent times, Machine Learning (ML) and Deep Learning (DL) are being used to mitigate these attacks, and the trending of these techniques is the DL, which provides effective training solutions to the problem.

A standard Neural Network (NN) consists of many simple, connected processors called neurons, each producing a sequence of real-valued activations. Input neurons are activated through sensors perceiving the environment; other neurons are activated through weighted connections from previously active neurons. Some neurons may influence the environment by triggering actions. Learning or credit assignment is about finding weights that make the NN exhibit desired behaviour, such as driving a car.

Depending on the problem and how the neurons are connected, such behaviour may require long causal chains of computational stages, where each stage transforms (often in a non-linear way) the aggregate activation of the network. DL is about accurately assigning credit across many such stages. DL became practically feasible to some extent through the help of Unsupervised Learning (UL). The 1990s and 2000s saw improvements to purely supervised DL. In this millennium, deep NNs have attracted widespread attention, mainly by outperforming alternative machine learning methods such as kernel machines in numerous important applications (Fonseca, Mota & Passito, 2012).

1.1 Anomalies in the network using SDN based DL Detection

SDN has been recognized as an adaptable, profitable, and versatile infrastructure that altered the legacy network architecture with a complicated one. The IoT security is yet considered to be a significant area that gained an extraordinary attention. One of the ideal defence systems from anomaly is practicing the exploitation of SDN for protecting the IoT from DDoS attack. DL-based approaches overtook the present machine learning techniques when applied to a range of classification issues, as they provide greater efficiency in feature extraction, including extractions on huge datasets (Bawany, Shamsi & Salah, 2017). This scheme can identify the traffic, irrespective of the recently known activity.

The efficiency as well as the security of the SDN-enabled detection can be improved with the self-learning ML technique, which is the DL technique. Thus, DL can find associations within the data and produce an enhanced presentation of the information, making it a better technique for anomaly detection. Furthermore, the technique can be utilized in the detection of multi-class threats, which causes improved detection accuracy.

The research question here becomes “how to come up with a solution that provides a high rate of detection accuracy with the lowest rate of false alarms?” and “how to develop a system which addresses evolving network threats. Our thorough study will process these questions and provide state-of-the-art solutions, which will answer the queries listed.

In the research, I will follow the below listed objectives for the purpose to tackle the stated research question:

i. How efficiently the intrusion detection systems could detect the threats/intrusion issues in computer networks?

ii. Considering that False positives are the major issue in trained IDSs, how can this issue be addressed?

iii. How can the highest rate of detection accuracy be achieved, and the evolving nature of threats be handled?

The principal research process for this work is to perform literature review and look for drawbacks and issues. Limitation documentation and organization to conclude problems by applying a regulated approach is the initial step toward providing novel and efficient solution. The study will first review and observe various IDS techniques as well as their features. Based on acquired understanding, a novel approach for Network threats classification will be designed to categorize attack classes for the determination of attack identification and detection. The second phase of this research will be studying upon existing Anomalies in network detection using SDN-enabled detection approaches and identification of limitation based on a broad review of current practices as well as academic research.

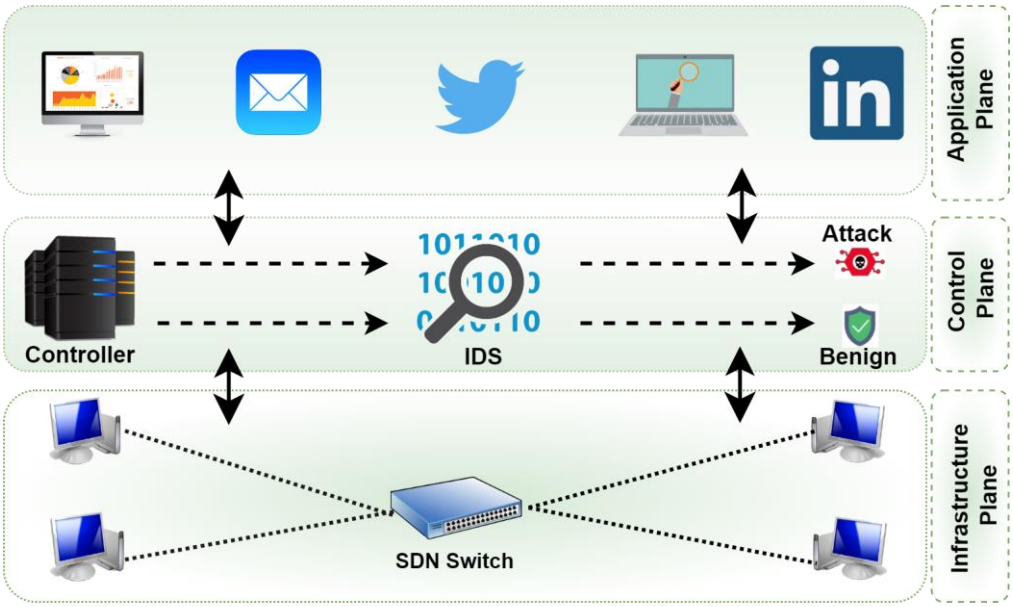

Figure 1.1: SDN based Intrusion Detection System (IDS). UT Computer Science (2022). An Architectural Evaluation of SDN Controllers.

1.2 Project Rationale

The gradually evolving cyber threats have become more sophisticated, with evolving network threats for destroying the network of the targeted organisation. For instance, a simple network threat can deny the alternative authentic users' access and restrict the services. Sometimes, attackers invade a network to establish a network threat on one or many targets with shared or distributed computing resources, to overwhelm the resources of the entire system and make the retrieval of information difficult for the owner. Thus, network threat is an emerging security issue that impacts IT infrastructures and services, to the disadvantage of both the networks users and providers (Roman, Zhou & Lopez, 2013).

A network-attack will affect the entire network, which can affect the performance of the organisation and the society at large, as most society have become dependent on IT infrastructures. Therefore, strict security measures must be in place to extenuate security risks. Resultantly, the cyber-security in the networks of laptops have become one of the major analysis issues in recent time (Ali & Li, 2019). To disrupt the conventional safe operation of a network or gain financial reward, the attackers target the various parts of a network. These parts represent the confidentiality, integrity, and convenience of the knowledge.

Researchers use various techniques to safeguard network devices, including databases, from the network attackers. Combining packet sampling and DL based applications, Software Defined Networking (SDN) has gained considerable traction among researchers. SDN gives a centralized view of the entire infrastructure at a global level, with programmable control-plane and data-plane independently (Shafi et al., 2018; Tang et al., 2018).

The description of certain attacks of the network for SDN enabled devices has been stated from the most common threat for instance, network threats. This threat provides an indication to the network-attack that makes the entities or legitimate users unprivileged to resources or

services. DL, which is an advanced form of machine learning (AML) will be used to tackle these attacks, as it has been broadly adopted and utilized in various fields of data-science, including natural processing, classification of images, and in big data for the performance of object recognition. The neural networks consisted in the DL scheme are based on the architecture that is deeply layered legitimizing hierarchic and automatic features learning (Abeshu & Chilamkurti, 2018). This research will propose network threats detection system based on DL models.

2 Related Works

This section gives a detailed description of works done in the domain of networks for handling the evolving threats and attacks.

2.1 Machine Learning

According to Diro & Chilamkurti (2018), ML should be utilized for cyberspace intrusion detection; although, the full-scale implementation of the strategy requires proper planning and effort. The threat detection, regardless of the network feature extraction, is considered as a major issue, the extracted features are not sufficient to provide a suitable way of representing the network data in such a way to give necessary and precise patterns(Liu and Yan, 2019). The sending and receiving of the number of bytes can be given as an example of features, the number of errors along with the connection duration are a few of the parameters that are manually engineered from the captured packing dump through the network traffic pattern (Diro & Chilamkurti, 2018a).

Probably, the features that are crafted manually may not be considered as the necessary features of Intrusion Detection System (IDS). Thus, numerous robust patterns are the actual nature of the instances of the network that can be captured utilizing the programs and algorithms of feature engineering (Diro and Chilamkurti, 2018a). The human generated errors included in the feature selection process along with the limitations of ML algorithms, ML endeavours to prevent error that can occur precisely determine the evolving network threats. Due to these limitations, the implementation of security in the security concerning organisations by acquiring the techniques of ML have been reduced (Liu et al., 2019; Fonseca, 2012).

However, ML-based solutions are still widely adopted by security organisations, and the issues of high rate of false alarms have been partly solved with low detection accuracy systems. It is difficult for traditional ML-based solutions to detect new attacks as they are less effective to the recent ones. These detection difficulties are caused by the system design and development flaws, which can be solved by the application of the plasticity of DL (Fonseca, 2012; Liu et al., 2019).

2.2 Network Traffic

The research summarization security technologies is carried out based on the conventional network attack detection. The rate of late detection and lack of intelligence at tackling the net traffic and modern network threat detection is high. When the network traffic behaves abnormally or in a different manner, the network threat may introduce. The detection of the

network activity and the performance of the targeted threats is required to prevent the network environment from being destroyed

The technique involves identifying the validity of the network source address. When the internet traffic is filtered to identify the source IP address, which might not be valid and creating a routing table between the source address and the routing equipment access port will map them together to be able to figure out the valid address (Fonseca, 2012). It is essential to determine the continued presence of the threats in the network by the association of the IP address of the internet traffic and the linked port address. Further, to understand and detail the instances of the threats in the network. Fonseca (2012) stated that when the SDN network detects the network threats, the switch will miss its connection to the upper controller, then check out and link to the standby controller. The negative impact of the network threats could be decreased on the temporary basis by this strategy, but it cannot prevent the threats in an efficient way.

Xu (2007) suggested that a system of detection of network traffic by the tracking of the source IP addresses through the Hidden Markov Models (HMM) is another method to look into as a system. A profile is created for regular traffic using the HMMs as the frequencies of new IP addresses, and the detectors are organized among the network nodes or near the network threat sources. A detection mechanism proposed the malevolent actions (Berral, 2008). The collaboration of the nodes in the network and sharing of the observations of the local traffic improves the overall traffic and information retrieval, and native Bayes algorithm is independently used by the node. The network threat detection, using neural networks had been performed, after acquiring data from each network packet. Abdelsalam et al. (2018) utilized the predictive models and classifiers of deep learning, to monitor the anomalous network traffic.

2.3 Existing Techniques

Manikopoulos (2002) utilized SDN based detection method on Neural Networks by taking benefit of Apache Spark cluster. Hsieh (2016) implemented an SDN-based neural network detection system with hadop, and Zhao et al. (2015) utilized Long Short Term Memory (LSTM) for network threat detection. If the network threats had been possessed by the backup controller, then SDN network would be unreachable completely.

Yuan (2017) recommended as a method of predicting normal traffic flow. By the mechanism of detection of the derived flow features and the defined threshold, Cisco's NetFlow Technology system utilized by the model for tracking network traffic. Therefore, operating on the network traffic is the necessity of this technique and a great amount of work is needed. The suitability of the test is though nearly linked with the professional experience of the researchers. Huawei1 introduced an SDN-based anti-DDoS cloud-cleaning scheme utilizing analytical techniques of large data (Fonseca, 2012). The incorporation of the security protection of the SDN network from the root of network threats and the data traffic analysis

is in more than 60 dimensions; yet this approach has a greater workload and has a greater dependency on relevant hardware and software.

Myung-Sup Kim (2004) provided a flexible and process-independent system for the protection of OpenFlow networks from the SDN based network threats; and recommended the Flood Defender, which was recognized by three innovative strategies: table-miss technology, packet filtering and maintenance of flow table. However, the device should have the dependency on the victim's neighbours. The switches of the neighbour should be enough in order to save bandwidth.

The description of the network threat monitoring and protection given above provide the results of bad detection performance as the technique seems to be increasingly, error or weakness report, weak adaptability, etc., utilizing the traditional techniques correspondingly. The machine learning-based intrusion detection system can be better understood, and the improvement of a few common features could be made possible from the current intrusion behaviours. Therefore, it has also become difficult to detect the threats of the intrusion data with sophistication and a wide range of functionality by utilizing the general methods of machine learning (ML).

The application of deep learning can be accomplished due to its advanced learning capacity at identifying network threats. A structure has been introduced by Bawany et al. (2017) to identify lightweight network threats. By the use of the algorithm of deep learning self-organized mapping for describing the attack source, 6 tuple fields of the network threat attribute have been developed by utilizing SDN traffic analytics tool. Gao (2017) has proposed an SDN-based network detection system that was based on deep learning. Difficulties arise by the approaches of low-rate attacks detection, as they seem like the legitimate network traffic of the victim.

Furthermore, it is necessary to produce the network threats on the target machines over time or else the infrastructure of the network would not be malevolent. The network packet sequences have been utilized by our detection strategy and have become capable of learning. The apparent and legitimate distinction flows among the threats. It helps in finding and tracing the repetitive patterns of network threats in long-term traffic series. Niyaz (2016) utilized a central flow-level control and the tracking technique to counter crossfire threats. When the implementation of the load balancing among the connections and routes is done in this classifier, the defender tends to hold onto the net execution devoid of blocking. If the massive traffic level had been attained, then the protector needs to track the origin of the traffic blocking and traffic rate limits. The ML techniques had been developed by the researchers to reduce the network threats in SDN (Ashraf & Latif, 2014). The controller of SDN accomplishes the analysis of traffic and the determination of the guidelines for the modification built in switches is achieved.

Abdelsalam et al. (2018) used the NSL-KDD dataset to detect the numerous threats by utilizing an algorithm of machine learning (ML) such as Decision Tree Bagging Ensemble,

Voting Ensemble utilizing KNN, Random Forest, Trapping and enhancement of decision trees and attained the accuracy of 94%. However, it is what stating that the NSL-KDD has become an outdated dataset and is not appropriate for a networking environment. Therefore, it has become important to utilize an up-to-date dataset for the practical application in a networking environment. The threats addressed by Wei Wang (2017) include Hydra-FTP, Java-Meterpreter, Hydra-SSH, and Web shell threats. For attack classification, the application of MLP and sparse matrix has been performed and pre-processing of data has been done that incorporates the feature mining along with feature selection. However, W. Wang (2017) comparison between CNN-1D and CNN-2D had been performed by the author for the detection of the anomalies of data packets where CNN-2D achieved the higher cataloguing accuracy of about 98.67%. The comparison between Abdelsalam et al. (2018) and Wang (2017) illustrates that higher accuracy had been attained by utilizing CNN-2D; hence, it is more efficient.

Thamilarasu & Chawla (2019) illustrated the detection of the network threats was performed in the networking environment utilizing CICID2017 dataset and CNN and LSTM along with Hybrid approach of CNN+LSTM in the research as the detection processes. A dense layer has been utilized by each model as last layer with the function of sigmoid activation and the author has performed no feature selection and classification. Roopak (2019) searching the root issues of users and network threats have been handled by the author and had been tested by utilizing NSL-KDD along with UNSW-NB15 datasets. The detection algorithms utilized include deep auto-encoders and DFFNN. The comparisons between various algorithms have been performed, including CVT, DMM, DBN, f-svm, TANN, RNN and DNN. Al-Hawawreh et al. (2018) includes RNN, CNN and stacked RNN that has been utilized in network threats, port scanning, network scan through TCP, UDP and ICMP detection.

Pal (2019) used Deep Neural Network (DNN) and Deep Boltzman (DBM) to identify malevolent actions through the binary classification and an applied 5-layers of DL model. Further, deep learning was recommended to detect emerging threats including the discovery of replicated device IDs, spoofing and Sybil attacks.

3 Research Methodology

The lightweight cyberattack detection is required due to the prone-to-attacks based nature of network devices. To achieve this thing in the organisational open environment, it is necessary to deploy the distribution of detection sensors for network devices in IT environments. In this section, the discussion of the proposed model, DL algorithms, and expected dataset is discussed and the Anomaly detection using SDN based threat detection model is proposed. This section provides the overview of testing and training phase of the project. Moreover, the proposed algorithm along with other state-of-the-art algorithms is discussed in detail.

Deep Neural Network (DNN), Gated Recurrent Units (GRU) and the Graphics Processing Unit (GPU) enabled Cu-DNNGRU are used as the classification algorithms to construct models used for the prediction of network threats reaching the computer network. For every classifier, four models are constructed separately and trained using more than 80 network

traffic features. The number of features used are same for every model and optimized based on the accuracy results.

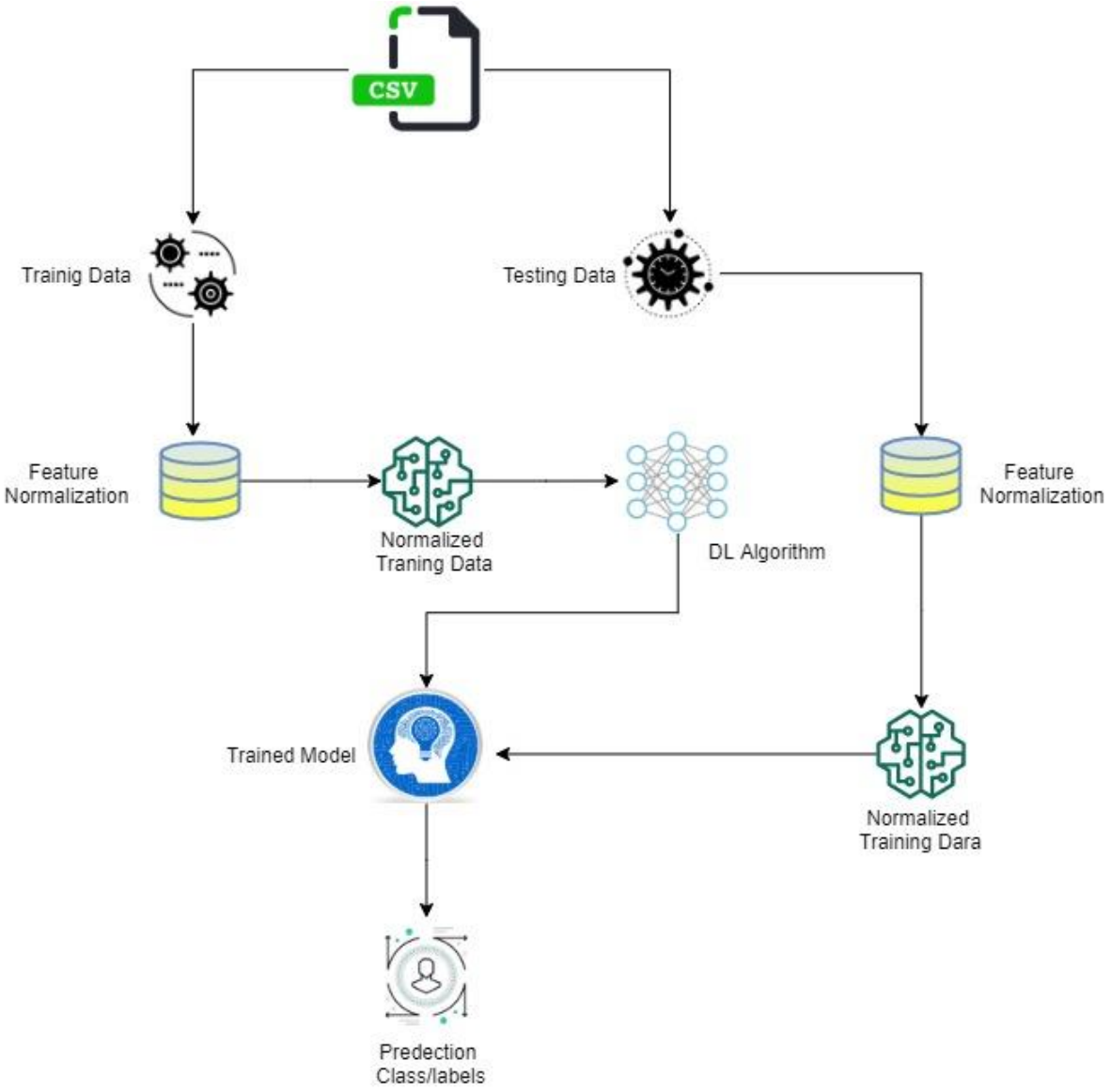

Figure 3: Training and Testing Model for the project.

Proposed model architecture

The proposed model is implemented on the basis of above given architecture (Figure 3). The model is designed using two phases of Machine/Deep Learning. The training and testing phase, the overhead architecture describes the overall working of the proposed approach. Beginning of the Comma Separated Value (CSV) of the dataset, the model divides the data into two distinct sections, (i.e., training and testing). The data contained for training purpose will then be used for feature learning and development of the proposed system, it will be normalized using preprocessing phase explained in the next sections. The standardized data will then be provided to the proposed system's algorithm for the purpose of performing the model learning to be prepared for the detection of upcoming threats. Trained system will then provide its predicted outcomes regarding attacks and normal instances. The mentioned

process will be accomplished many times corresponding to the number of epochs given by computer programmer. After completing the training phase, its iteration or epochs, the testing data will be used to test the model performance. The testing data is the data kept untouched and unknown for trained model. The model accepts these instances as new network traffic and provide exact evaluation ratio of selected metrics.

3.1 GPU enabled DNNGRU

DNNGRU is a predefined hybrid model of deep learning (DL) which is picked and proposed as a best classifier for network threat detection in this study. The DNNGRU model is the efficient algorithm, which by default combines the two state-of-the-art algorithms for achieving the maximum level of accurate detections. The model has been trained, the model using DNNLGRU layers with dense layers and then tested the system for determining the attack type. The model of DNN and GRU both is stated below.

3.2 Deep Neural Networks (DNN)

DNN - a Deep Neural Network is same as an Artificial Neural Network (ANN) with numerous layers among the input as well as output layers. The DNN finds the right scientific control to transform the contribution to the output, regardless of whether it be a straight relationship or a non-direct relationship. The system travels through the layers determining the likelihood of each output. For instance, a DNN that is prepared to perceive hound breeds will go over the given picture and compute the likelihood that the dog seen in the image is a definite breed. The user can survey the outcomes and select which probabilities the system should show (over a specific edge and so forth.) and return the proposed label. For each scientific control as such is viewed as a layer, and complex DNN has numerous layers, thus the name “Deep” networks. DNNs can display complex as well as relationships that are non-linear.

3.3 Layered Neural Networks

DNN architectures produce compositional representations where the objective is stated as a layered configuration of primitives. The additional layers empower composition of features after subordinate layers, possibly modelling composite data with a smaller number of units than a correspondingly performing narrow network.

3.4 Gated Recurrent Units (GRU)

Another normally utilized DL classifier to identify a DDoS attack is GRU or Gated Recurrent Units. The aim of a (GRU) is to provide a mechanism with gating functionality in recurrent neural networks (RNN), like a long-short-term-memory (LSTM) unit yet without the gate of an output. The GRU's attempt to take care of the disappearing gradient issue that can accompany using reset and update gates. A GRU can be viewed as a variation of the (LSTM) unit because both have a comparable plan and produce equivalent outcomes now and again. GRU's can take care of the disappearing gradient issue by utilizing reset and update gates. The reset gate controls the data that streams out of memory and the updating gate controls the data that streams into memory. The reset and update gates are two vectors that choose which data will get or given to the output. They can be prepared to keep information from the past or remove the data, which is unnecessary for the prediction.

3.5 Evaluation Metrics

The appropriate graphs as well as figures have been used for the evaluation of the overall results of all four deep learning models. Various well-known and standard metrics, including

Accuracy, Recall, Precision, FPR, as well as F1-score, etc. have been used for analyzing the performance of each model. The results that have been utilizing these metrics are compared with each other for proving the versatility of the proposed method. Furthermore, our proposed scheme has been compared with the available modern detection systems for providing the best results.

The performance metrics used for evaluating the DL classifiers have been defined as following:

i. Accuracy is the percentage of the correct prediction made by the trained model.

i. Recall is the ratio of the true positives which have been predicted accurately.

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

i. Precision includes the evaluation of the probability of the accuracy of the positive prediction.

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

i. F1-score is bending the precision as well as recall in order to obtain the average weight for acquiring relative values like 1 and 0.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Whereas TP like True Positive and TN True Negative indicate the accurately predicted values while False Negatives (FN) and False Positives (FP) denotes the misclassified incidents.

i. The True Positive Rate (TPR) is the ratio of the correct predictions.

i. False Positive Rate (FPR) represent the number of instances classified in class X whereas associated to a specific class, in addition to the instances which are not included in class X.

4 Design Specification

The techniques and framework that underlie the implementation as well as the associated requirements have been identified and described in this section. A word-based description of the functionality of the model must be included if the new algorithm has been proposed.

4.1 Proposed Methodology

This step provides a description about the implementation phase of the proposed scheme. This phase provides a brief description of the backend architecture of the training as well as testing system. The rare contributions of the dataset chosen have been employed for an embedded layer of deep learning-based algorithms as depicted in **Figure 3;** the dataset consists of a total of 80 features of benign as well as malicious traffic. The system conceptually functions from beginning to end in a way that the pre-processed data in CSV along with the extracted features of the network traffic have been used as an input and is provided to the layered neurons' model. The input layer is the first layer in which the activation function RELU is applied and the data is used for model training, multilayer perceptron (MLP) have been utilized as the hidden layers which are arranged in descending

order for the learning of the system to attain the highest as well as improved levels of accuracy. The output layer with activation function SOFTMAX consists of the 4 neurons for providing multi class threat detection. The features as well as the categories of the attack have been provided by the detection phase for the regulation of the learning process. It consists of two phases such as training phase and testing phase. The model is trained in the beginning by utilizing the training data that had been split into (80%) of the complete dataset. When the phase of data preprocessing as well as feature normalization have been completed, the deep learning model is applied over the normal data for processing the training phase, the iterations have been set for the pattern learning accordingly. When the training process have been completed, the trained data would then be utilized for the testing phase whereas the 20% of the data has been stored untouched while splitting for predicting the results as well as performance evaluation.

4.2 Dataset Description

The utilization of the modern as well as state-of-the-art network traffic CSE-CIC-IDS2018 dataset for the model phases like training as well as testing. The dataset is available publicly in CSV format for purposes of research and its distribution has been arranged in two kinds of traffic such as benign and malicious. Malicious traffic consists of the traces from the 14 kinds of patterns of network attack. Three types of classes of attack as well as one benign class have been extracted for the implementation of our proposed scheme. The image provided below consists of the dataset testbed, real-life benign as well as malicious traffic data that has been collected by utilizing the displayed environment.

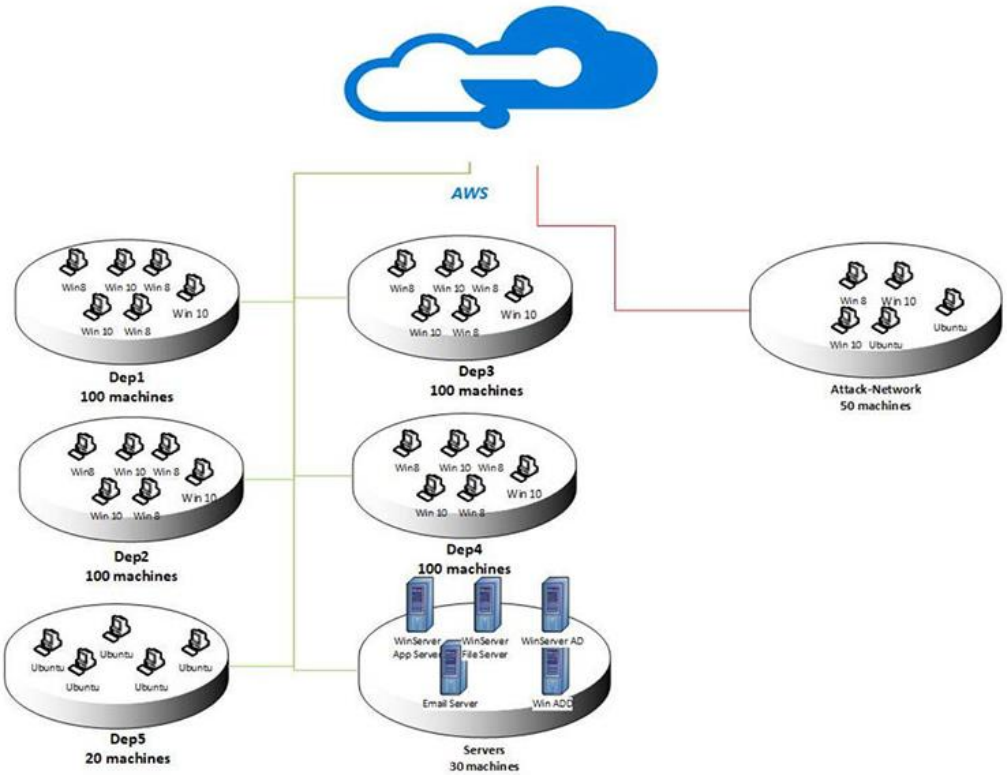

Figure 4.2: Dataset Testbed
Network traffic CSE-CIC-IDS2018 dataset for the model phases.

Eighty features for every record in a dataset is captured, and CIC flowmeter has been utilized for capturing the information from the real-life network flows. The 80 best features are selected with greater accuracy and lowest false alarms for evaluating the detection model.

4.3 Data Pre-processing

The major step in the Artificial Intelligence is the pre-processing of data before utilizing it. The captured data should be ready prior to the implementation of the proposed algorithm, the dataset document usually consists of various categorical, numeric, as well as Non values. The primary pre-processing of data in most cases is completed and exclusively of the attack as well as benign class labels. The system training as well as testing of the models should be completed in numeric values, therefore, all the categorical values have been converted while the data pre-processing phase.

The data, time, IP's as well as all the data that does not consist of numeric values would be pre-processed by utilizing the python libraries as well as code at first. If the Non values and the empty fields in the dataset have remained unhandled might be a result of bad training. Hence, the Non values as well as empty fields could be removed and dropped by using the python functions. Furthermore, the python **MinMax/Standard** scalar has been performed for scaling of data and the other step include splitting the data by range. The data has been split in the ratio of 80:20 as well as saved in the variables as the training and testing sets. Training data would later be utilized in the phase of implementation of the model for the purpose of training, the testing data would be untouched and later utilized for the testing and predictions of model.

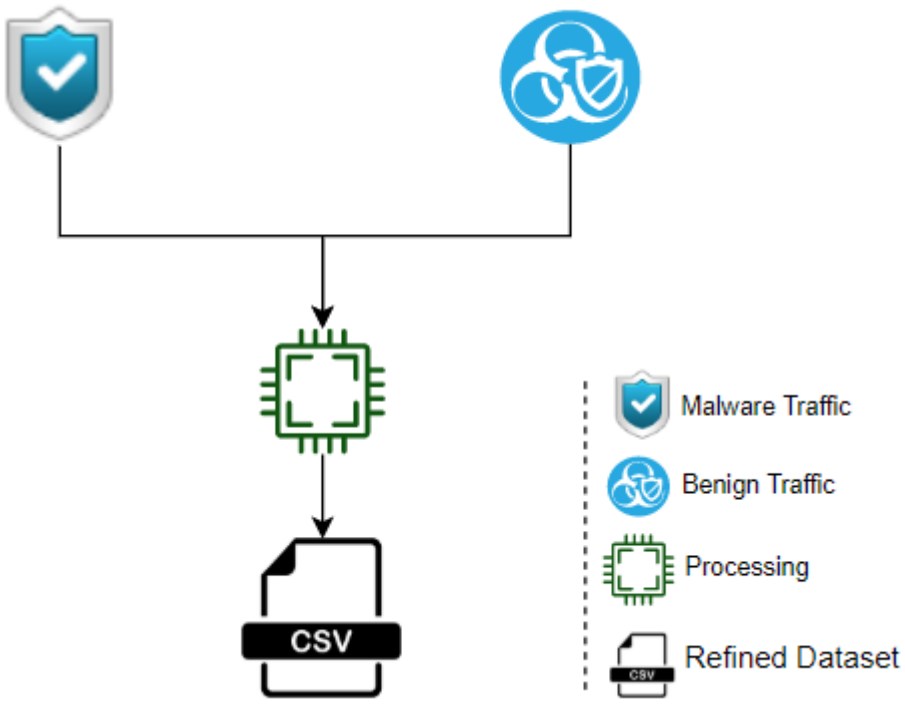

Figure 4.3: Dataset Pre-processing

Figure 4.3 indicate various steps of pre-processing to remove categorical values. Dataset with categorical values, lead to the bad training and for handling such types of values such as attack labels have been converted to numeric values.

5 Implementation

All the technologies as well as tools that have been utilized in the implementation phase for developing the model are described in this section.

5.1 Python

Python is a high-level, object-oriented programming language that has been utilized for numerous purposes, including data visualization, artificial intelligence, data analysis, programming applications, as well as machine learning. Humans can easily comprehend the python codes which make it very simple to construct the models of machine learning. A readable as well as precise code is provided which consumes plenty of time for implementation of the AI and the ML algorithms. It is very important to have a well structured as well as tested environment to help the developers to come about with the best coding solutions. With the rich pile of the python technology, it contains an extensive range of libraries set for artificial intelligence as well as machine learning.

6 Evaluation

This chapter provides the complete results of all the three DL models with the relevant figures. The performance of each model is analysed by choosing some of the standard metrics like Accuracy, Precision, Recall, FPR, and F1-score etc. The results utilizing these metrics have been compared to each other to prove the superiority of the proposed model. Hence, we have done a comparison between the available modern detection systems for providing excellent results.

6.1 Accuracy of DL Models

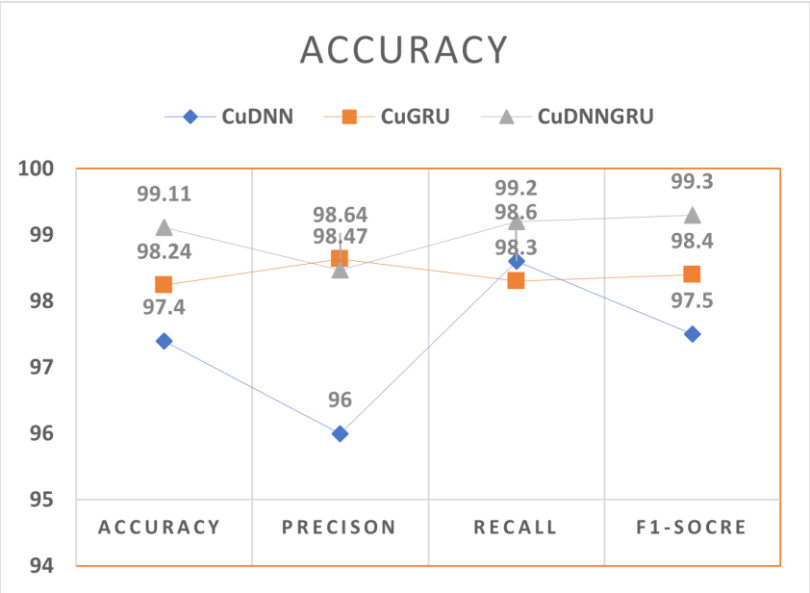

Figure 6.1: Accuracy, Precision, Recall, F1-Score

The Accuracy of the Cu-DNN model across all classes was 97.40%, that of the Cu-GRU model was 98.24%, while the accuracy of the Cu-DNNGRU was 99.11%. All models can be regarded as good, because their accuracy was above 70.00% (Figure 6.1). Accuracy metric in DL is the number of correct predictions divided by the total predictions multiplied by 100. The precision of the Cu-DNN model was the lowest with 96%, while the precision of the Cu-GRU model was 98.64%. The precision of the Cu-DNNGRU model was slightly lower than those of the Cu-GRU model with 98.47%. The Precision metric in DL is a measure of the correct positive predictions of the model.

The Recall metric in DL is a ratio of the number of true positives and the total number of elements, which actually belong to the positive class. The Cu-GRU model recorded the lowest Recall with 98.3%. The Recall of the Cu-DNN model is 98.6%, while the highest Recall score is recorded for the Cu-DNNGRU model with 99.2%. The F1-score of the Cu-DNN, Cu-GRU, and Cu-DNNGRU models are 97.5%, 98.4%, and 99.3% respectively. The F1-score is a measure of the accuracy of the model on the dataset. Thus, the Cu-DNNGRU model is the most accurate of all the models tested.

6.2 Discovery of False Detection Rates

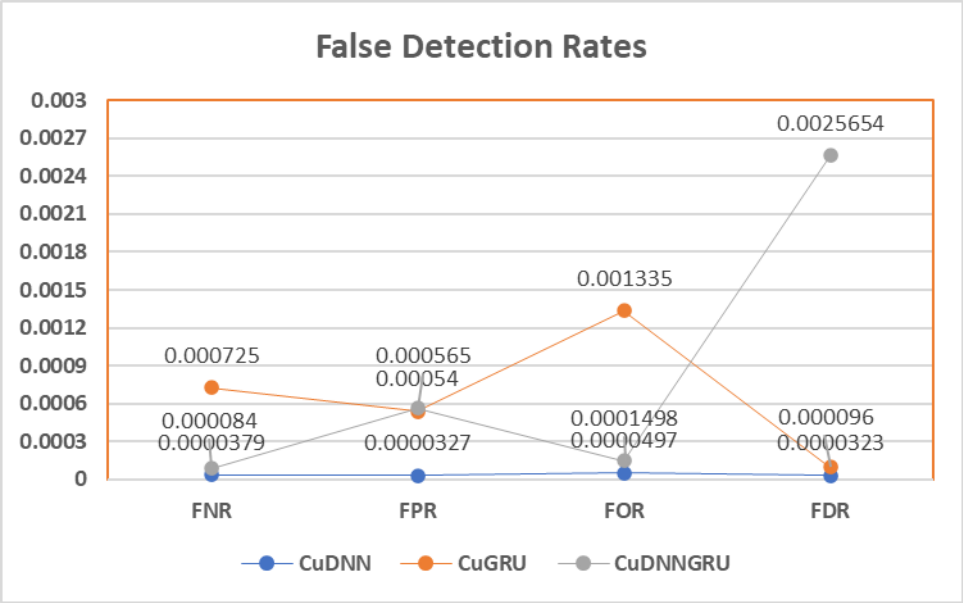

Figure 6.2: FNR, FPR, FOR, and FDR

The False Positive Rate (FPR) is the rate at which the model incorrectly predicts the positive class. In Figure 6.2, the Cu-GRU model recorded an FPR score of the Cu-DNN model was 0.00003, the FPR score of the Cu-GRU model is 0.00054, and that of the Cu-DNNGRU model was 0.00057. The False Negative Rate (FNR) is the rate at which the model incorrectly predicts the negative class. Thus, the Cu-DNN model gave the lowest FNR score of 0.000038, the FNR score of the Cu-DNNGRU model was 0.000084, and that of the Cu-GRU was 0.000725 (Figure 6.2).

The False Discovery Rate (FDR) is the rate of elements with a positive test result for which the true condition is negative. The Cu-GRU model recorded an FDR score of the Cu-DNN model was 0.000032, the FDR score of the Cu-GRU model is 0.000096, and that of the Cu-DNNGRU model was 0.002565 (Figure 6.2). The False Omission Rate (FOR) is the rate of elements with negative test result for which the true condition is positive. In Figure 6.2, the Cu-DNN model gave the lowest FOR score of 0.000050, the FOR score of the Cu-DNNGRU model was 0.000150, and that of the Cu-GRU was 0.001335.

6.3 True Detection Rates

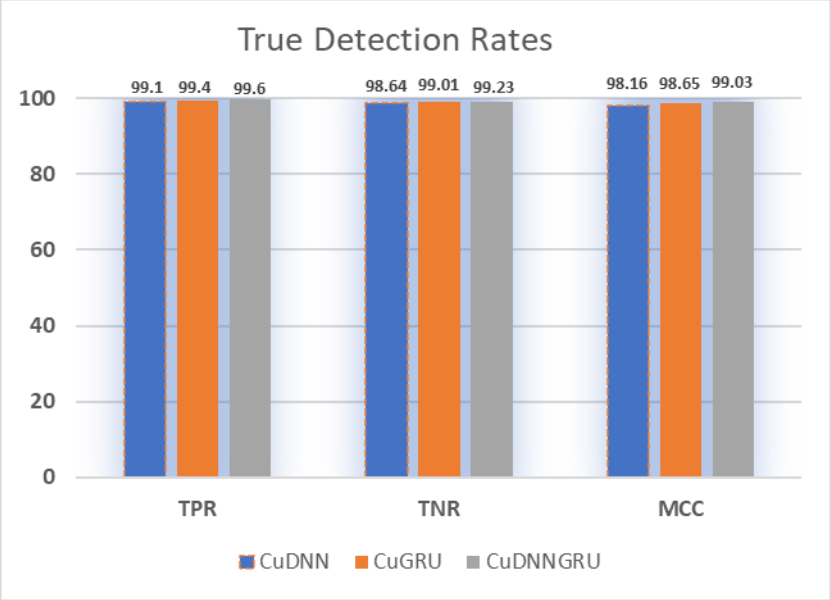

Figure 6.3: TPR, TNR, MCC

The True Positive Rate (TPR) is the probability that an actual positive element will test positive. In Figure 6.3, the TPR for the Cu-DNN, Cu-GRU, and Cu-DNNGRU models are 99.1%, 99.4%, and 99.6% respectively. The True Negative Rate (TNR) is the probability that an actual negative element will test negative, and the TNR recorded for the Cu-DNN, Cu-GRU, and Cu-DNNGRU models are 98.64%, 99.01%, and 99.23% respectively. The Matthews Correlation Coefficient (MCC) is a metric used to measure the performance of a classification model. The MCC score recorded for the Cu-DNN, Cu-GRU, and Cu-DNNGRU models are 98.16%, 98.65%, and 99.03% respectively.

6.4 Time Efficiency of DL Models

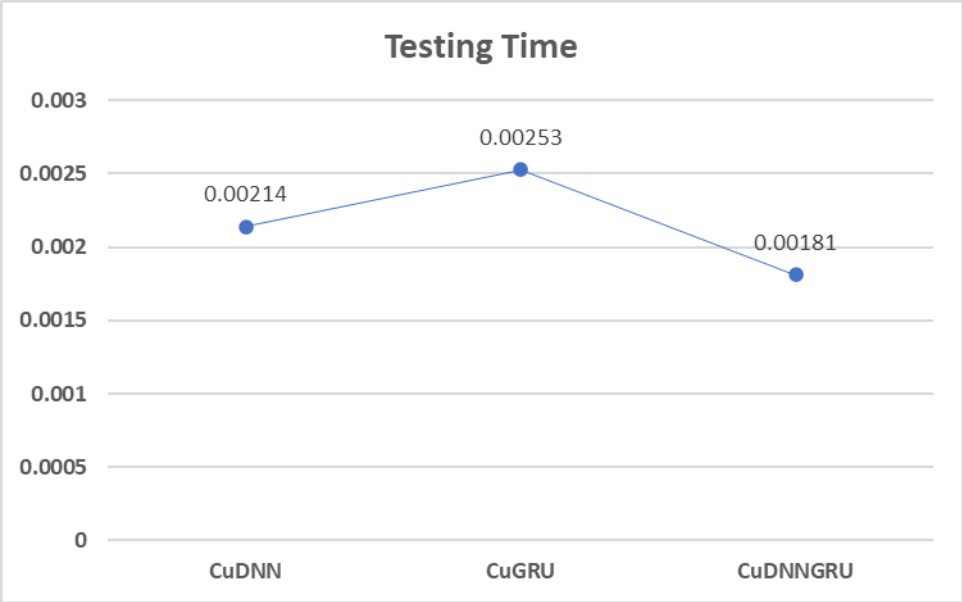

Figure 6.4: Test Time comparison

Figure 6.4 shows the time efficiency of each DL model used for the project. The Cu-DNNGRU model had the lowest testing time of 0.00181, the Cu-DNN model had a testing time of 0.00214, while the Cu-GRU had the highest time of 0.00253.

7 Discussion

The overall accuracy of the three models is presented in Figure 6.1, and the proposed model outperformed the other models, as it recorded high values of Precision, Recall, and F1-score of 98.64%, 99.62%, and 99.3% respectively. During the implementation phase of the proposed scheme, an average differentiation ratio of training as well as testing accuracies was achieved and an increase in the ratio of accuracy was observed. On the other hand, the Cu-DNN and Cu-GRU were less efficient as they produced results with less than 98.7% in all metrics.

Additionally, other metrics were analysed. For instance, FPR, FNR, FOR, and FDR as illustrated in Figure 6.2. The proportion of incorrectly classified positive samples gives the FNR. The FPR also called false alarm rate or FAR characterises the ratio among the negative samples that are incorrectly classified and the total amount of negative samples. The FDR measures supplement the NPV and PPV, correspondingly. As illustrated, the values of FPR, FNR, and FDR of the proposed scheme are in the range of 0 and 0.00003 which is appropriate for the network threats detection in organisational networks.

Furthermore, the TNR, TPR, as well as the MCC were used to understand the performance of Cu-DNNGRU model, as depicted in Figure 6.3. TNR denotes to the ratio of perfectly classified negatives, which shows that the greater value indicates the better the performance trained system. In contrast, TPR is the ratio of perfectly classified positives. While MCC represents the ratio of negative and positive classification consequences which ultimately reflects the proportion between the values of TN and TP. The Cu-DNNGRU model obtained

a high True Positive Rate (TPR) of 99.6% than other two models. The proposed model also outperforms other two models in terms of time efficiency by consuming the lowest time in predicting actual results as depicted in Figure 6.4.

7.1 Future Work

A future work might compare the performance of a Cu-DNNGRU model with a hybrid model of Cu-BLSTM and Cu-DNNGRU to investigate the best model for a proactive network defence.

7.2 Conclusion

This paper provides a description about the threats and attacks posed onto the networks. The hackers try to utilize the network threats of the multi class group for preventing the security systems. Hence, a lightweight SDN based threat detection system has been proposed. Three models were compared to check which models produce promising results. Our proposed scheme Cu-DNNGRU have been compared with the other two models (i.e., DNN and GRU). The proposed scheme provided as the highest accuracy of 99.11%, which outperformed the other two models. This is evident that our proposed model is efficient in detecting evolving network threats to secure networks.

References

AL-Hawawreh, M., Moustafa, N. & Sitnikova, E. (2018). Identification of malicious activities in industrial internet of things based on deep learning models. Journal of Information Security and Applications, 41, 1-11.

Abdelsalam, M., Krishnan, R., Huang, Y. & Sandhu, R. (2018). Malware Detection in Cloud Infrastructures Using Convolutional Neural Networks. 162-169.

Abeshu, A. & Chilamkurti, N. (2018). Deep Learning: The Frontier for Distributed Attack Detection in Fog-to-Things Computing. IEEE Communications Magazine, 56, 169-175.

Ali, S. & Li, Y. (2019). Learning Multilevel Auto-Encoders for DDoS Attack Detection in Smart Grid Network. IEEE Access, 7, 108647-108659.

Ashraf, J. & Latif, S. (2014). "Handling Intrusion and DDoS Attacks in Software Defined Networks using Machine Learning Techniques," In NSEC 2014, Rawalpindi, Pakistan, Nov. , pp. 55–60.

Basumallik, S., Ma, R. & Eftekharnejad, S. (2019). Packet-data anomaly detection in PMU-based state estimator using convolutional neural network. International Journal of Electrical Power & Energy Systems, 107, 690-702.

Bawany, N. Z., Shamsi, J. A. & Salah, K. (2017). DDoS Attack Detection and Mitigation Using SDN: Methods, Practices, and Solutions. Arabian Journal for Science and Engineering, 42, 425-441.

Berral, J. L., Poggi, N., Alonso, J., Gavalda, R., Torres, J., & Parashar, M. (2008). Adaptive distributed mechanism against flooding network attacks based on machine learning. . In Proceedings of the 1st ACM workshop on Workshop on AISec ACM, , 43-50.

Chockwanich, N., & Vasaka Visoottiviseth. (2019). “Intrusion Detection by Deep Learning with TensorFlow.”. 2019 21st International Conference on Advanced Communication Technology (ICACT). IEEE.

Conti, M., Dehghantanha, A., Franke, K. & Watson, S. (2018). Internet of Things security and forensics: Challenges and opportunities. Future Generation Computer Systems, 78, 544-546.

Cooke, E., Jahanian, F., & McPherson, D. (2005). The Zombie Roundup: Understanding, Detecting, and Disrupting Botnets. SRUTI, 5, 6-6.

Diro, A. & Chilamkurti, N. (2018a). Leveraging LSTM Networks for Attack Detection in Fog-to-Things Communications. IEEE Communications Magazine, 56, 124-130.

Diro, A. A. & Chilamkurti, N. (2018b). Distributed attack detection scheme using deep learning approach for Internet of Things. Future Generation Computer Systems, 82, 761-768.

Fonseca P, B. R., Mota E, Passito A. (2012). A replication component for resilient OpenFlow-based networking[C]. IEEE Network Operations & Management Symposium. IEEE,, 933-939.

Gao S, P. Z., Xiao B, Hu A, Ren K. (2017). FloodDefender: protecting data and control plane resources under SDN-aimed DoSAttacks[C]. . INFOCOM 2017-IEEE Conference on Computer Communications, IEEE, , 1-9.

Hsieh, C. J., & Chan, T. Y. (2016, May). Detection DDoS attacks based on neural-network using Apache Spark. In 2016 international conference on applied system innovation (ICASI) (pp. 1-4). IEEE.

Javeed, D., Gao, T., Khan, M. T., & Ahmad, I. (2021). A hybrid deep learning-driven SDN enabled mechanism for secure communication in Internet of Things (IoT). Sensors, 21(14), 4884.

Javeed, D., Gao, T., Khan, M. T., & Shoukat, D. (2022). A hybrid intelligent framework to combat sophisticated threats in secure industries. Sensors, 22(4), 1582.

Khan, M. T., Akhunzada, A., & Zeadally, S. (2022). Proactive Defense for Fog-to-Things Critical Infrastructure. IEEE Communications Magazine.

Liu, H., Lang, B., Liu, M. & Yan, H. (2019). CNN and RNN based payload classification methods for attack detection. Knowledge-Based Systems, 163, 332-341.

Lopez-Martin, M., Carro, B., Sanchez-Esguevillas, A. & Lloret, J. (2017). Network Traffic Classifier With Convolutional and Recurrent Neural Networks for Internet of Things. IEEE Access, 5, 18042-18050.

Manikopoulos, C. A. P., S. (2002). Network intrusion and fault detection: a statistical anomaly approach. . IEEE Communications Magazine, 40, 10,, 76-82.

Mcdermott, C. D., Petrovski, A. V., & Majdani, F. (2018). Towards situational awareness of botnet activity in the internet of things. . 2018, June, Institute of Electrical and Electronics Engineers.

Mendez Mena, D., Papapanagiotou, I. & Yang, B. (2018). Internet of things: Survey on security. Information Security Journal: A Global Perspective, 27, 162-182.

Myung-Sup Kim, H.-J. K., Seong-Cheol Hong, Seung-Hwa Chung and J. W. HONG. (2004). A flow-based method for abnormal network traffic detection. IEEE/IFIP Network Operations and Management Symposium (IEEE Cat. No.04CH37507), , pp. 599-612.

Niyaz Q, S. W., Javaid AY. (2016). A deep learning based DDoS detection system in software-defined networking (SDN)[J]. . arXiv preprint arXiv:1, 611.07400, .

Pal, S. (2019). Limitations and Approaches in Access Control and Identity Management for Constrained IoT Resources. In 2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), March, pp. 431-432.

Pour, M. S., Mangino, A., Friday, K., Rathbun, M., Bou-Harb, E., Iqbal, F., Shaban, K. & Erradi, A. (2019). Data-driven Curation, Learning and Analysis for Inferring Evolving IoT Botnets in the Wild. 1-10.

Roopak, M., Tian, G. Y., & Chambers, J. (2019, January). Deep learning models for cyber security in IoT networks. In 2019 IEEE 9th annual computing and communication workshop and conference (CCWC) (pp. 0452-0457). IEEE.

Roman, R., Zhou, J., & Lopez, J. (2013). On the features and challenges of security and privacy in distributed internet of things. Computer Networks, 57(10), 2266-2279.

Shafi, Q., Basit, A., Qaisar, S., Koay, A. & Welch, I. (2018). Fog-Assisted SDN Controlled Framework for Enduring Anomaly Detection in an IoT Network. IEEE Access, 6, 73713-73723.

Tang, T. A., Mhamdi, L., Mclernon, D., Zaidi, S. A. R., & Ghogho, M. (2018), June. Deep recurrent neural network for intrusion detection in sdn-based networks. In 2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft), (pp. 202-206). IEEE.

Thamilarasu, G. & Chawla, S. (2019). Towards Deep-Learning-Driven Intrusion Detection for the Internet of Things. Sensors (Basel), 19.

UT Computer Science (2022). An Architectural Evaluation of SDN Controllers. Available at: https://www.cs.utexas.edu/~samehdi/Architectural-Eval-SDN-Controllers.pdf. (Accessed on: 20 July 2022)

Ujjan, R. M. A., Pervez, Z., Dahal, K., Bashir, A. K., Mumtaz, R. & González, J. (2020). Towards sFlow and adaptive polling sampling for deep learning based DDoS detection in SDN. Future Generation Computer Systems, 111, 763-779.

Wang, W., Zhu, M., Wang, J., Zeng, X., & Yang, Z. (2017, July). End-to-end encrypted traffic classification with one-dimensional convolution neural networks. In 2017 IEEE international conference on intelligence and security informatics (ISI) (pp. 43-48). IEEE.

Wang, W., Zhu, M., Zeng, X., Ye, X., & Sheng, Y. (2017, January). Malware traffic classification using convolutional neural network for representation learning. In 2017 International conference on information networking (ICOIN) (pp. 712-717). IEEE.

Xu, X., Sun, Y., & Huang, Z. (2007). Defending DDoS attacks using hidden Markov models and cooperative reinforcement learning. . In Pacific-Asia Workshop on Intelligence and Security Informatics Springer, , 196-207.

Yuan, X., Li, C., & Li, X. (2017). DeepDefense: Identifying DDoS Attack via Deep Learning. . In 2017 IEEE International Conference on Smart Computing (SMARTCOMP) IEEE, , 1-8.

Zhao, T., Lo, D. C.-T. & Qian, K. (2015). A Neural-Network Based DDoS Detection System Using Hadoop and HBase. 1326-1331.