SDN BASED DDoS DETECTION USING TEMPORAL CONVOLUTIONAL NETWORK

A Project Report

Submitted by

Ms. SUBUHANA N

REG NO : TKM20MEAI14

SEMESTER : IV

In partial fulfillment for the award of the degree of

MASTER OF TECHNOLOGY
IN
Mechanical Engineering (Artificial Intelligence)

Under the guidance of
Prof. SUMOD SUNDAR

Thangal Kunju Musaliar College of Engineering
Kerala

JULY 2022

DECLARATION

I undersigned hereby declare that the project report " SDN BASED DDoS DETECTION USING TEMPORAL CONVOLUTIONAL NETWORK", submitted for partial fulfillment of the requirements for the award of degree of Master of Technology of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by me under supervision of Prof. Sumod Sundar. This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the university and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other university.

Place: Kollam
Date: SUBUHANA N

Thangal Kunju Musaliar College of Engineering
Centre for Artificial Intelligence

C E R T I F I C A T E

This is to certify that, this report titled ***SDN BASED DDoS DETECTION USING TEMPORAL CONVOLUTIONAL NETWORK*** is a bonafide record of the **Project** presented by **SUBUHANA N(TKM20MEAI14)**, under our guidance and supervision, in partial fulfillment of the requirements for the award of the degree, **M.Tech in Mechanical Engineering (Artificial Intelligence)** in **APJ Abdul Kalam Technological University** .

Project coordinator & Internal Supervisor

Prof. Sumod Sundar
Assistant Professor
Centre for Artificial Intelligence

Head of the Department

Dr. Imthias Ahamed
Professor & HOD
Centre for Artificial Intelligence

Internal Examiner

External Examiner

ACKNOWLEDGEMENT

A successful project is a fruitful culmination of efforts by many people, some directly involved and some others indirectly, by providing support and encouragement. Firstly I would like to thank the almighty for giving me the wisdom and grace for making my project a successful one. I thank him for steering me to the shore of fulfillment under his protective wings

I express my sincere gratitude to **Dr. T A Shahul Hameed**, Principal of TKMCE, and **Dr. Imthias Ahamed**, Professor and Head of the Department, Centre for Artificial Intelligence, TKMCE, for their constant support and encouragement throughout the project work.

I would like to express my heartfelt thanks to our project coordinator cum my guide **Prof. Sumod Sundar**, Assistant Professor, Centre for Artificial Intelligence, TKMCE, for his expert guidance and cooperation. With a profound sense of gratitude, I would like to thank **Dr.Santhi Natarajan**, Honorary Professor, for her immense encouragement. I would like to express my gratitude to **Rajeev Azhuvath**, Mentor, Tata Consultancy Services (TCS),Tata Consultancy Services (TCS), for his expert guidance, and cooperation. I also extend my thanks to the entire faculty and staff members of the Centre for AI, TKMCE, who has encouraged me throughout this work.

I also express my thanks to my loving parents, brother and friends, for their support and encouragement in the successful completion of this project work.

SUBUHANA N

Abstract

Software-Defined Networking (SDN) is an emerging network architecture to overcome the weaknesses of traditional networks. The main goal of SDN is to separate the control and data planes and make network management more effortless. The network intelligence is logically separated and centralized at the control layer. Distributed Denial-of-Service (DDoS) is one of the most prevalent and sophisticated threats. DDoS attack attempts to disrupt the available services of a target machine by sending massive malicious requests from a large number of hacked computers called botnets. Accurate detection and prevention of DDoS attacks are necessary to protect network systems. In this work, we propose and demonstrate the design, implementation, and testing of detecting DDoS attacks using a Temporal Convolutional Network (TCN). The proposed TCN model is evaluated using the DDoS attack SDN dataset. Feature Selection is made using Neighboring Component Analysis (NCA) and XGBoost feature selection. The performance of the TCN model is compared to other state-of-the-art Machine Learning and Deep Learning techniques. The supervised Machine Learning algorithms used for the comparison are Logistic Regression, Gaussian Naive Bayes, SVM, Gradient Boosting, and XGBoost with Auto Encoder extracted features. The Deep Learning techniques used for DDoS detection in an SDN environment are DNN, CNN, LSTM, BiLSTM and Tab Net. The experimental result shows that tree-based classifiers such as Gradient Boosting and XGBoost perform very well in the case of supervised Machine Learning algorithms. The Deep Learning algorithms significantly improve the performance of DDoS detection. TCN model outperforms all other models and gives the best accuracy of 99.48 %. The TCN model with XGBoost feature selection improves the performance of the model and yields 99.57 % accuracy. The results are validated using another comprehensive SDN dataset called INSDN dataset. The TCN model gives 99.98 % accuracy in the INSDN dataset and implies that TCN performs very well in both datasets. We obtained a definite improvement in DDoS detection compared to other benchmarking methods. TCN model with XGBoost feature selection offers much confidence in protecting SDN networks. In the initial experiments silhouette score from K means clustering is used to analyze the similarities between the classes in the SDN DDoS attack dataset.

Contents

List of Figures

List of Tables

ABBREVIATIONS

SDN	Software Defined Network
DDOS	Distributed Denial-Of-Service
TCN	Temporal Convolutional Network
NCA	Neighboring Component Analysis
XGBoost	eXtreme Gradient Boosting
SVM	Support Vector Machine
LIME	Local Interpretable Model-agnostic Explanations
PCA	Principal Component Analysis
TSNE	T-distributed Stchastic Neighbor Embedding
DNN	Deep Neural Network
CNN	Convolutional Neural Network
LSTM	Long Short Term Memory
BiLSTM	Bi Directional Long Short Term Memory

Chapter 1

INTRODUCTION

1.1 General Background

In the era of exponentially growing digitalization, the role of emerging information and communication technologies in our day-to-day life is undeniable. In addition, it raises new security concerns. These systems are vulnerable to various cyber threats and attacks. SDN is a unique network paradigm that promises highly dynamic network architecture. The major benefit of SDN is that it separates the control and data planes and makes the network more versatile and easier to manage [1]. A single remote computer called a controller can handle the entire system. Many industrialized enterprises are implementing SDN technology in their network environments.

The SDN architecture consists of three layers; Infrastructure, control and application. The infrastructure layer is integrated with the various network components to form the underlying network for routing network traffic. It acts as the network's physical layer that communicates with the virtualized network configured through the control panel. The control layer is the land of the control panel of the SDN network. It contains various protocol modules and acts as the intelligence of your network infrastructure. This layer is where most of the business-based logic is written. The application layer is responsible for the open-source platform for developers building innovative applications such as improving network topology and network statics.

There are many attack vectors that can exploit the SDN network [2]. DDoS is the most prevalent and sophisticated attack. DDoS attacks attempt to overwhelm the victim's available resources from servicing legitimate users by sending many malicious requests using botnets. Attackers use Botnets created from devices called zombies hijacked by internet hackers. DDoS attacks are carried out with many machines, so it is complicated to detect and block. The frequency and severity of DDoS attacks constantly increase and can have fatal effects on many network services. Many companies such as Amazon, Facebook, Twitter, and GitHub suffered from DDoS and suffered huge losses. So that, accurate detection and prevention of DDoS attacks are the most critical problems for network service providers.

According to the cenralized architecture of SDN, DDoS targeting the SDN network can be classified into three types; DDoS against the infrastructure layer, DDoS targeted at the control layer, and DDoS threatened at the application layer. The goal of the DDoS attack at the infrastructure level is to overwhelm one or a few specific network devices at the infrastructure level. As a result, the attacker instructs a series of zombies that connect to

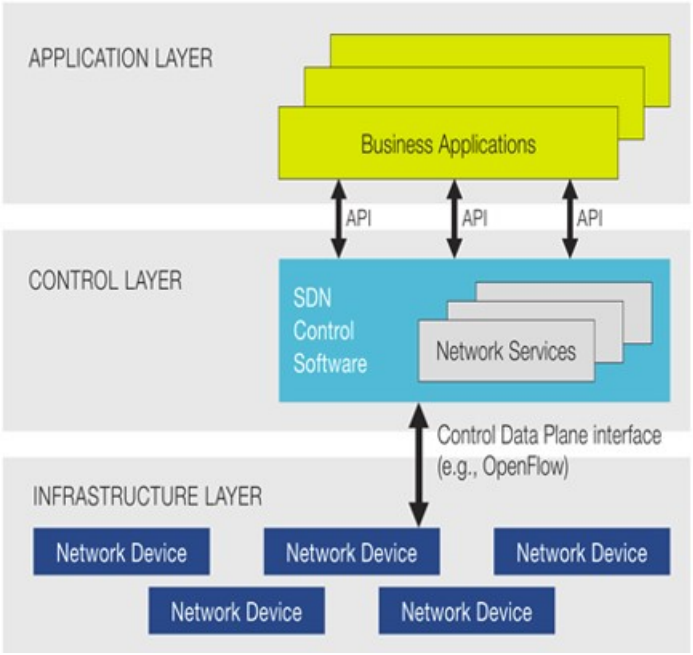

Figure 1.1: SDN architecture.

targeted network devices to send DDoS packets. In contrast, DDoS attacks that target the control layer attempt to overwhelm the controllers compute, storage, or bandwidth resources. Therefore, the adversary usually causes all or most network devices to send messages to the controller by ordering zombies to send many new packets to those network devices. DDoS attacks threaten the application layer; the target is the application server. Therefore, the adversary will order the zombies to create many new specific packages for all or most SD-Ncompatible switches. Eventually, the application that handles the ARP packets will receive many incoming packet messages, which will cause the application servers to compute, memory, and bandwidth resources to be exhausted. The differences between DDoS attacks that target the infrastructure layer, control layer and application layer are that they have different attack targets and patterns. Therefore, we need to detect DDoS attacks and protect our systems accurately.

The first line of defence against these attacks is the Network Intrusion Detection System (NIDS). In recent years, many approaches using Machine Learning techniques have been proposed to detect DDoS attacks [3]. Machine Learning algorithms such as SVM, Naïve bayes, Gradient boosting and XGBoost are used for DDoS detection. The extensive network data makes intrusion detection problems susceptible to machine learning methods. Classical Machine Learning algorithms struggle to work with large amounts of data due to their limited ability to learn features. Recently, Deep Learning has achieved great success in many different applications, such as Face recognition, image processing and natural language translation [4]. Deep Learning can extract raw features from data without human intervention. It can achieve high-performance rates by automatically finding the correlation in the raw data. Therefore, with the advent of Deep Learning-based models, the accuracy of attack detection

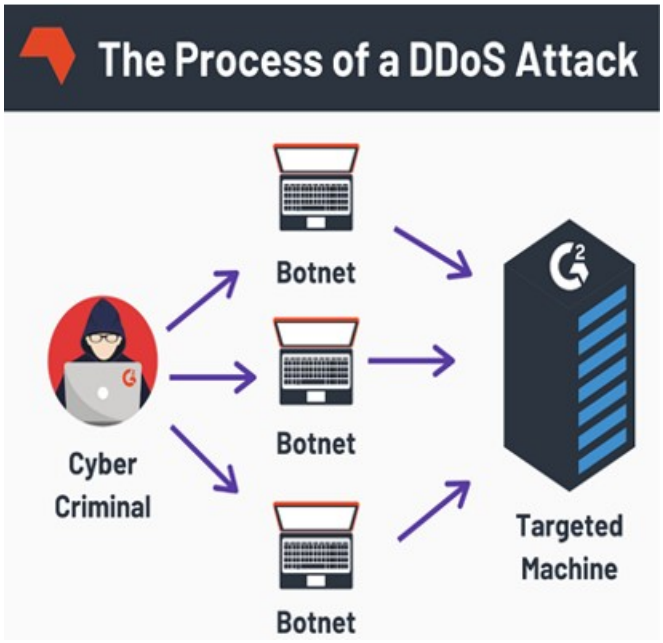

Figure 1.2: The process of DDoS attack.

has steadily improved. Deep Learning Techniques such as DNN, Auto Encoders, CNN and Tab Net can be used for DDoS detection. However, time correlations of network traffic often generate time-series data [4]. Training the simplest form of Deep Learning algorithms with sequential traffic can lead to losing some information about the data. Therefore, LSTM, BiLSTM and TCN can be used to find the temporal correlations of network traffic data. Train the model with that method can retain all data information with minimal loss.

1.2 Objective

We propose a deep learning technique based on TCN to detect DDoS attacks on the SDN. The proposed model has the best performance compared to different traditional techniques. The contribution of this work includes the following:

- We leverage and propose a deep learning approach based on TCN for detection of DDoS attacks on the SDN. The model can accurately classify the network traffic into malicious or normal.
- We evaluate our model using the newly released dataset DDoS attack SDN dataset, which contains a comprehensive variety of DDoS attacks and addresses the gaps in the existing datasets.
- We compare several state-of-the-art Machine Learning and Deep Learning models that are well known for detection of DDoS attacks and we evaluate our proposed model in terms of accuracy, precision, recall and F1 score.

The rest of the work is structured as follows: Chapter 2 discusses related works on SDN based DDoS detection. The dataset and the proposed works are presented in chapter 3. An exhaustive discussion on the experimental results is summarized in chapter 4. Finally, chapter 5 concludes the work.

Chapter 2

RELATED WORKS

In recent years, many studies have been done to secure SDN using AI techniques. In this section, we briefly discuss the most recent and popular Machine Learning and Deep Learning techniques used to detect DDoS attacks in SDN environments.

Tonkal et.al [1] proposed a Machine Learning Approach using Neighboring Component Analysis (NCA) for DDoS detection in Software-Defined Network. The objective is to classify the SDN traffic as normal or attack traffic. The dataset used is "DDoS attack SDN Dataset". It is a publicly available dataset including a total of 23 features. NCA algorithm is used to identify the most important features. The dataset was classified using k-Nearest Neighbor, Decision Tree, Artificial Neural Network, and Support Vector Machine algorithms. The obtained results are promising out that the proposed model can obtain decent results compared to traditional Machine Learning models.

Aamir et.al [2] proposed a semi-supervised Machine Learning methods for DDoS detection. The network traffic flow contains both normal and DDoS attack traffic. The clustering methods used are agglomerative and K-means clustering. A voting method is used to label the data and classify the attacks from normal traffic. K Nearest Neighbor, Support Vector Machine and Random Forest algorithms are applied to the models for network traffic classification. The dataset used is generated in OPNET Modeler 14.5 simulator. The experimental results shows that 95 percentage, 92 percentage and 96.66 percentage accuracy scores are obtained with kNN, SVM and RF models respectively with optimized parameter tuning in given sets of values. The method is also validated for labelling accuracy using CICIDS2017 dataset.

Cil et al [3] proposed a Deep Learning model for Detection of DDoS attacks with feed forward based Deep Neural Network model. Deep Learning models are more effective for the detection of DDoS attacks on network traffic. Since it has feature extraction and classification algorithms in its structure, as well as layers can update themselves when the model is trained, the DNN model can work quickly and accurately. The CICDDoS2019 dataset is used. The experimental results shows that the DNN model are effective for the detection and classification of DDoS attacks on network traffic.

Haider et.al [4] proposed an ensemble model using CNN for DDoS detection in SDN environments. Two similar Deep Learning models are combined to build an ensemble model and two complimentary models (RNN+LSTM) are used for creating a hybrid model. The CICIDS2017 dataset is used for the experiment. The experimental results shows that ensemble CNN model outperforms all other models.

Elsayed et.al [5] proposed a model called DDoSNet, an intrusion detection system against DDoS attacks in SDN environments. The framework combines RNN and Autoencoder. RNN can focus on the temporal dependencies. Autoencoder can increase the anomaly detection accuracy. The features can be automatically extracted from the input data. Every layer of the Autoencoder is created using simple RNN layer. The CICDDoS2019 dataset is used. When comparing the performance of the DDOSNet to that of other classical ML algorithms, it shows that DDoSNet excels the others.

Cui et al [6] proposed an SDN-enabled Gated Recurrent Unit Recurrent Neural Network (GRU-RNN) intrusion detection system. The model was tested on the NSL-KDD and CICIDS2017 datasets, with results of 89 percentage and 99 percentage accuracy, respectively. When compared to other state-of-the-art algorithms, the GRU-RNN strategy achieves an 89 percentage detection rate in the NSL-KDD dataset with the smallest number of features. In the CICIDS2017 dataset, the GRU-DNN obtains an impressive detection rate of 99 percentage in case of DDoS detection.

The advantages and limitations of recent related works are summarized in Table 2.1.

Sl.No	Title	Technique used	Advantages	Disadvantages
1	Machine Learning Approach Equipped with Neighborhood Component Analysis for DDoS Attack Detection in Software-Defined Networking [1] (2021)	NCA, KNN, Decision Tree, ANN, SVM	NCA gives most relevant features by feature selection, UpToDate dataset	Does not give optimal number of features to be selected, The performance depends on the selected features
2	Clustering based semi-supervised machine learning for DDoS attack classification [2] (2019)	Agglomerative clustering, K means clustering, KNN, SVM, Random Forest	Optimizing and validating the model improves the performance	Clustering approach leads to high false positive values
3	Detection of DDoS attacks with feed forward based deep neural network model [3] (2021)	DNN	High accuracy	Failed to detect adversarial attack
4	A Deep CNN Ensemble Framework for Efficient DDoS Attack Detection in Software Defined Networks [4] (2020)	RNN, LSTM, CNN, RNN+LSTM	Improved accuracy, Minimal computational complexity	Failed to detect adversarial DDoS attacks
5	DDoSNet: A Deep-Learning Model for Detecting Network Attacks [5] (2020)	RNN, AutoEncoder	Temporal correlation of data is considered	Vanishing Gradient problem
6	Intrusion Detection in SDN-Based Networks: Deep Recurrent Neural Network Approach [6] (2019)	GRU-RNN, GRU-DNN	The method offers a lot of potential for real-time detection	The model is not optimized for improving accuracy and reduce controller overhead

Table 2.1: Analysis of recent related works.

Chapter 3

METHODOLOGY

The Deep Learning methods generally used for temporal sequence modelling are RNN, LSTM and GRU. These models have some issues with training and convergance. TCN is a variant of a convolutional neural network that combines the aspects of RNN and CNN for sequence modelling tasks. TCN overcomes the problems of RNN family architectures.The experiments have been conducted in [8], shows that TCN has better performance and more efficient in modelling different sequences than RNN models. Therefore, we choose the TCN model for DDoS detection.
The framework of the proposed model is shown in Fig.3.1 and overall experimentation is shown in Fig 3.2. This work concentrates on the implementation and testing of detecting DDoS attacks in an SDN environment by three approaches. This work concentrates on implementation and testing of detecting DDoS attack in SDN environment by three approaches.

1. Supervised approaches using traditional ML algorithms
2. Unsupervised approach of clustering and dimensionality reduction
3. Supervised approaches using Neural Networks

The two feature selection techniques used are:

1. Neighboring component Analysis
2. XGBoost Feature selection

The feature extraction is done by using Auto Encoder. The supervised Machine Learning techniques are Logistic Regression, Gaussian Naïve Bayes, SVM, Gradient Boosting and XGBoost. We have used LIME library to understand the reasoning behind the Machine Learning models. The K means clustering with PCA and TSNE dimensionality reduction techniques are implemented. In the case of Deep Learning techniques DNN, CNN, LSTM, BiLSTM, Tab Net and TCN are implemented and evaluate the performance of these models for DDoS detection in SDN environments in terms of accuracy, precision, recall and f1-score.

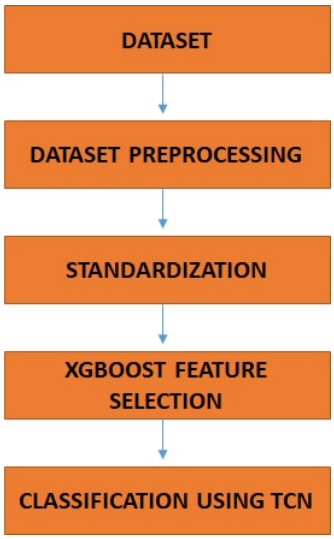

Figure 3.1: Framework of the proposed methodology

3.1 Datasets Used

3.1.1 DDoS Attack SDN Dataset

It is an SDN-specific dataset generated using the Mininet emulator and used for network traffic classification [7]. The network simulation runs for both normal and malicious traffic, including a TCP Syn attack, UDP Flood attack, and ICMP attack. The network simulation runs for 250 minutes duration and collects 1,04345 lines of data. The dataset contains a total of 23 attributes, some of which are retrieved from the switches and others computed. The class label indicates the the type of traffic normal or attack. Normal traffic is labeled 0 and attack traffic is labeled 1.

3.1.2 DDoS Attack SDN Dataset Pre-processing

Before applying Machine Learning and Deep Learning techniques, preprocessing is applied in the dataset. The first step is to convert the Categorical variables, not having numeric values such as source IP, destination IP and protocol . The rx-kbps and tot-kbps features having missing values in the dataset and that is filled with the mean value. The IP address is grouped based on 'dt' feature and count the number of request coming from the IP. Find the correlation between features by using heat map graph. Then standardize the data using standard scalar.

3.1.3 INSDN Dataset

It is a Comprehensive SDN dataset for the validation of intrusion detection systems. Total 84 features and 136743 network flows are available in the data set. The INSDN dataset consist of different types of attacks that can affect the data plane, control layer and application layers. The types of attacks are DDoS, Dos, Probe, BFA and U2R. The dataset's

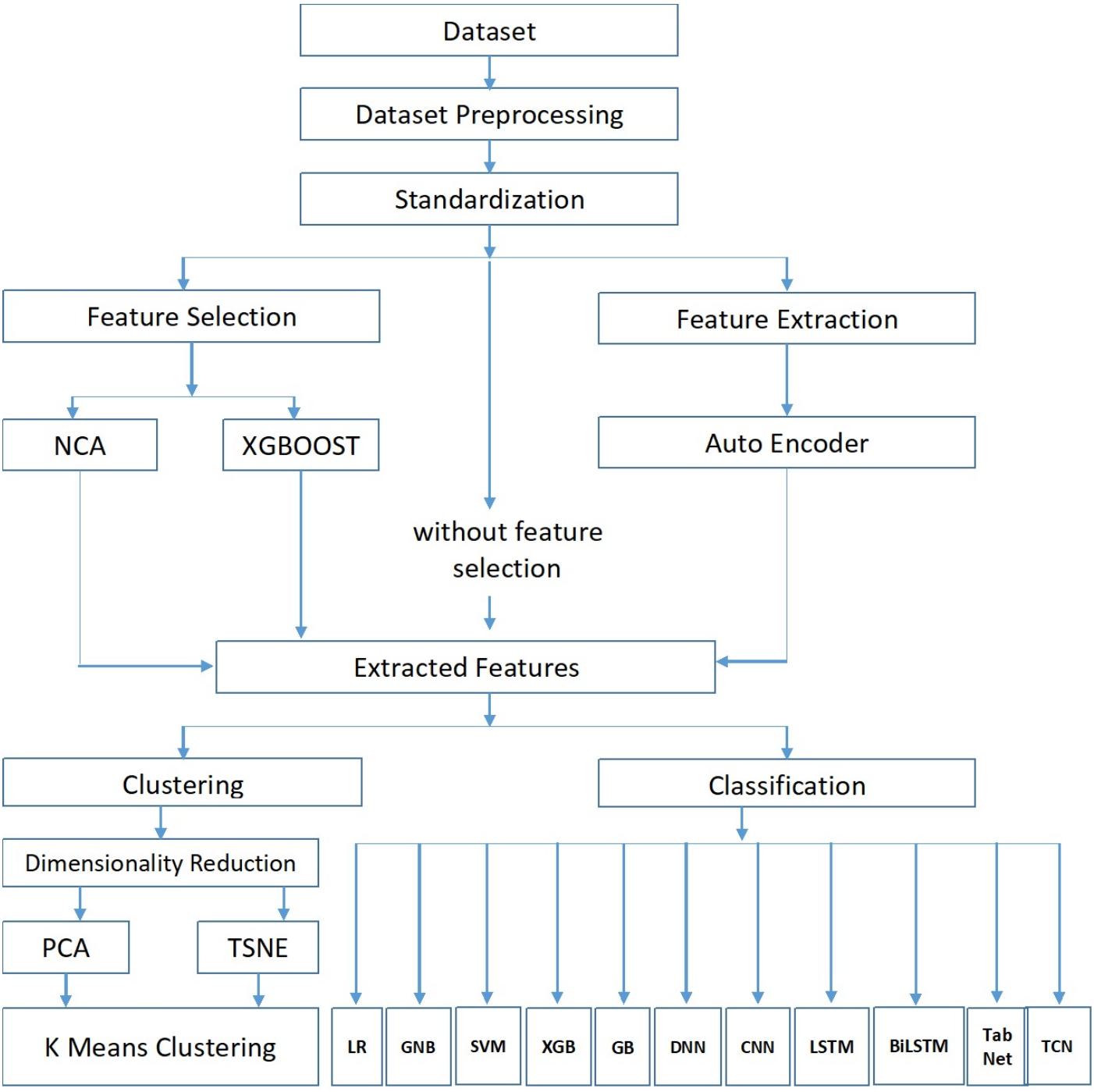

Figure 3.2: Overall Experimentation

Feature Name	Description
dt	Transmission moment of packets over the network device
switch	Switch ID
src	IP address of the sender of the packets
dst	IP address to which the packets was sent
pktcount	Number of packets
bytecount	Number of bytes
dur	Duration
dur-nsec	Duration in nano seconds
tot-dur	Total duration of network flow
flows	Number of flow packets
packetins	Total flow entries in the switch
pktperflow	Packet count during a single flow
byteperflow	Byte count during a single flow
pktrate	Number of packets per sec
Pairflow	Number of flow packets per second
Protocol	Types of communications internet protocols
port-no	Port number of the sender of the packets
tx-bytes	Number of bytes transferred from the switch port
rx-bytes	Number of bytes received on the switch port
tx-kbps	Data transfer rate
rx-kbps	Data Receiving rate
tot-kbps	Sum of tx-kbps and rx-kbps
label	Class label

Table 3.1: Features in Dataset.

attack sources are divided into two categories; Internal and External. Internal users with full access to the SDN are the source of the internal attacks. internal attacks are rare in production networks, but become more severe over time they can cause malicious behavior in network components. Before launching new attacks on target servers, the attacker attempts to exploit vulnerabilities in the individual users of the network system. Compromised hosts in the INSDN dataset are used for initiating various attacks from an internal SDN network. External attacks often use external networks to launch these attacks. The attacker mainly modifies the SDN network with malicious activities such as DoS and malware.

3.1.4 INSDN Dataset Pre-processing

In the pre-processing stage the distribution of five category of attack is plotted. Removed the duplicate values from the dataset. The categorical variables such as Flow ID, Source and destination IP and Time stamp are converted into numerical values. The source IP is grouped based on 'Timestamp' feature and count the number of request coming from the IP. Find the correlation between features by using heat map graph.
Resampling is performed to reduce the class imbalance problem in INSDN datset. The imbalance in the data can lead to reduced detection accuracy. Oversampling the minority

class by duplicating the instances is a way to deal with imbalanced datasets. One such technique is known as SMOTE (Synthetic Minority Oversampling Technique). SMOTE is data augmentation for the minority population. The algorithm tries to create new data samples from the line obtained by mapping the neighbouring samples. This process does not add any relevant information to the data; instead, new instances are created by synthesizing old instances. Hence, the issue of data overfitting is resolved. Detection accuracy can also be improved as the model will learn both classes of data uniformly. Finally standardize the data using standard scalar.

3.2 Feature selection Techniques

The main goal of feature selection techniques used is selecting a subset feature by reducing calculation costs and reducing unrelated features of the dataset that affects model performance. In this work, NCA and XGBoost are the two feature selection techniques used to select the most appropriate features to perform an effective classification of more than 100 thousand network records, which consists of 22 features of SDN technology.

3.2.1 Neighboring Component Analysis (NCA)

The NCA model developed on the basis of the KNN algorithm is that it lists the features in order of importance and also provides information on the weight of the features [1].

3.2.2 XGBoost feature selection

A trained XGBoost model automatically calculates feature importance on the predictive modelling problem [8]. XGBoost feature selection technique gives the feature imporatnce graph. This can be used to decide which features to use. The model will transform the dataset into a subset with selected features.

3.3 Supervised Machine Learning Techniques Experimented

The supervised Machine Learning algorithms such as Logistic regression, SVM, Gaussian Naïve Bayes, SVM, Decision Tree, Random Forest and Gradient Boosting are used for DDoS detection.

3.3.1 Logistic Regression

Logistic regression is a linear model for binary classification problems. A linear combination of the product of the independent variable (x1, x2, x3,... xn) and its corresponding weight (w1, w2, w3,... wn) is employed in the sigmoid equation to limit the output to a range of 0 to 1 [9]. The result is expressed as the likelihood of an event occurring. Many applications do not just want a class label, they want to figure out the probability of belonging to a category. For this reason, logistic regression models work well. The goal of logistic regression model is to discover the optimum solution where trying to determine if a new sample fits best into a category as aspects of cyber security problems such as attack detection. The dataset is loaded and splitted in to features and labels. Then the dataset is divided in to training and

testing set (20% for testing and 80% for training). The logistic regression model is created, model is trained on the dataset and test the model.

3.3.2 Gaussian Naïve Bayes

A variant of Naïve Bayes algorithm that follows Gaussian normal distribution and supports continuous data. Bayes theorem is used to calculate the conditional probability [9]. The Bayes Theorem is used to create Naive Bayes Classifiers. The strong independence assumptions between the features are one of the assumptions made. These classifiers make the assumption that the value of one feature is unrelated to the value of any other characteristic. Naive Bayes Classifiers are particularly efficient in supervised learning situations. The Gaussian Naïve bayes model is created, model is trained on the dataset and test the model.

3.3.3 Support vector Machines (SVM)

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane [1]. SVM chooses the extreme points that help in creating the hyperplane, these are support vectors. In SVM, the kernel is utilized to divide data non-linearly. SVM searches support vectors, weights, and bias to achieve this goal.

3.3.4 Gradient Boosting

Powerful ensemble machine learning algorithm that uses decision trees. Gradient boosting involves three elements:

- A loss function to be optimized.
- A weak learner to make predictions.
- An additive model to add weak learners to minimize the loss function.

The decision trees are the weak learners, and the loss function has to be differentiable [10]. Decision trees are used as the weak learner in gradient boosting. A gradient descent procedure is used to minimize the loss when adding trees. The objective is to minimize the loss. The gradient boosting model is created, trained on the training set and test on the test set.

3.3.5 Auto Encoder and XGBOOST Classifier

Auto Encoder is a type of neural network that can be used to learn a compressed representation of raw data. Auto Encoder is used for the feature extraction. The encoder compresses the input, and the decoder attempts to recreate the input from the compressed version provided by the encoder. After training, the encoder model is saved, and the decoder is discarded. The encoder can then be used as a data preparation technique to perform feature extraction on raw data that can be used to train XGBoost classifier. XGBoost, which stands for extreme Gradient Boosting, is a scalable, distributed gradient boosted decision

tree. XGBoost is a decision-tree-based ensemble Machine Learning algorithm that uses a gradient boosting [10]. The trees used by XGBoost are called CART trees (classification and regression trees). The CART tree having real scores of an instance's membership in a group rather than containing a single decision in each leaf node. The decision can be made by converting the scores into categories with a certain threshold after the tree has reached it's maximum depth. The Auto Encoder model is defined with encoder model, bottleneck, and decoder model. Fit the auto encoder model to reconstruct input. Define and save the encoder model without decoder. Compress the input data using encoder model Import the XGBoost classifier. Encode the train data. Encode the test data. Fit the XGBoost classifier model on the training set.

3.3.6 Machine Learning Reasoning with LIME

Although Machine Learning models are becoming very popular day by day, some are still considered "black boxes", i.e. uninterpretable. So that We have used LIME (Local Interpretable Model-Agnostic Explanations) to understand the reasoning behind the Machine Learning models [11]. LIME is an explanatory technique applied to Machine Learning classification or regression predictions to explain and interpret a local prediction, a single observation. By using this library, we can respond to the features of the data that most influence the specific decision.

3.4 Clutering Techniques Experimented

clustering is an important aspect of unsupervised learning. Automatically divide the dataset into clusters based on similarities. This will give us an idea about the underlying patterns of the different groups. Identify different groups of network traffic like normal and DDoS attacks. Here K Means clustering is used for DDoS detection.

3.4.1 K Means Clustering

K Means group the data into a predetermined number of clusters. The goal of K Means is to find similar data points and group them together, trying to keep each cluster as far apart as possible. It calculates "similarity" using Euclidean distance or a simple straight line between two points [2]. Since K Means is scale sensitive, all features should be at the same scale. Features with greater variation are given more weight or emphasis by K Means, and these features have a greater impact on the final cluster structure. Clustering techniques such as K Means struggle to accurately cluster data with many dimensions. Here PCA and TSNE dimensionality reduction techniques are used and compared with K Means clustering.

3.5 Dimensionality Reduction Techniques Experimented

3.5.1 PCA

PCA or Principal Component Analysis is a well-known approach for reducing high-dimensional data to a low-dimensional space [2]. The great thing about PCA is that retain most of the variance or information from our original high-dimensional dataset, despite reducing the

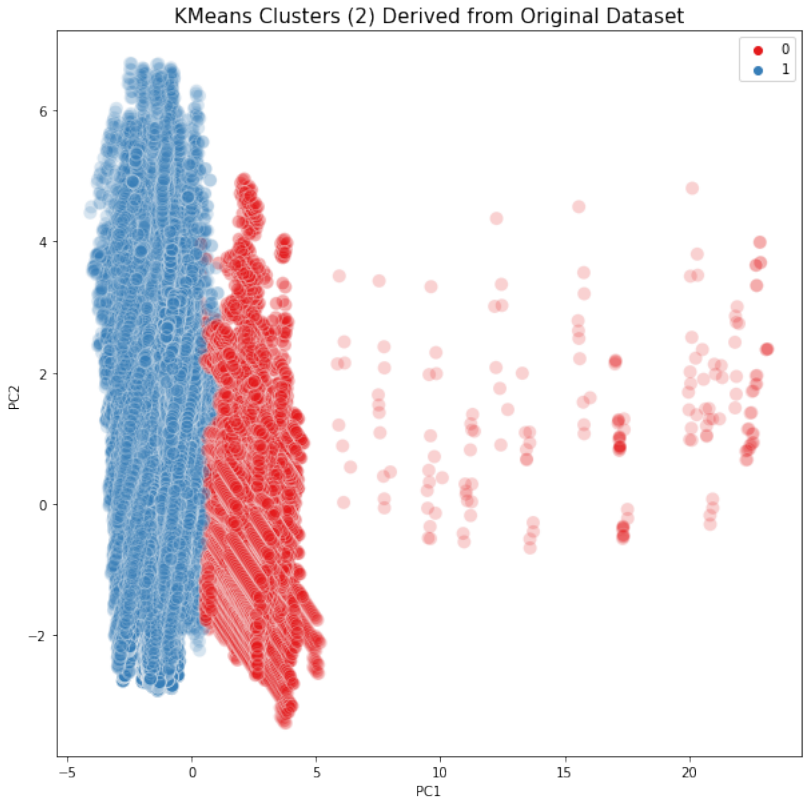

Figure 3.3: K Means clustering on original dataset.

space to the smaller dimension. Data or differences from our original features are "compressed" into principal components(PC). Most of the information from the original features is stored on the first PC. The second PC will contain the second largest amount of data, the third PC will contain the third largest amount of data, and so on. PCs are orthogonal, which means that each PC has its own set of data.

3.5.2 T-SNE

T-SNE is a method for reducing high-dimensional data into a low-dimensional space. t-SNE is primarily a visualization tool and then a dimensionality reduction method. This is also a fantastic strategy for reducing dimensionality. t-SNE can reduce dimensions with nonlinear relationships. T-SNE compares the distance between two local or adjacent points to see how similar they are. Points close to each other are considered similar. The match distance for each pair of points is then converted into a probability for each pair of points via t-SNE. In a highdimensional space, two points close to each other have a high probability value and vice versa [12]. The probability that a group of points will be selected is related to their correspondence in this way. Then, each point is randomly projected into a low-dimensional space.

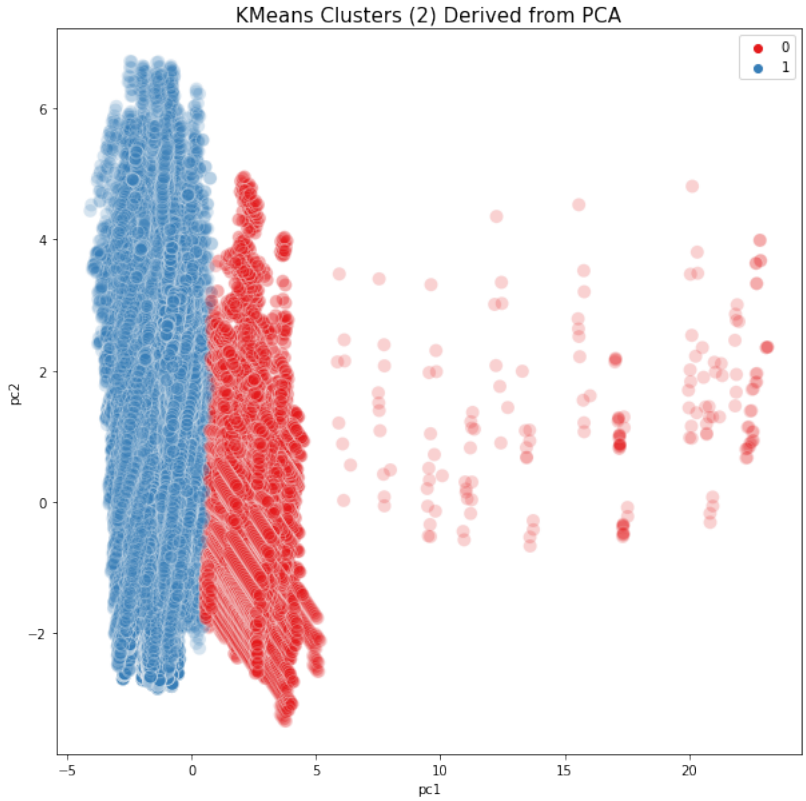

Figure 3.4: K Means clustering on PCA derived data.

3.5.3 Design Steps

The implementation steps are first standardization of data. Standardizing the data will arrange all features on the same scale, the mean is set to to zero and bring the standard deviation to 1. Then apply K Means to the original dataset with k equals two. Then PCA is used for the feature reduction, the dataset is reduced into two main components and plot the clusters derived from PCA. K Means clustering is applied to the main components of PCA. There is a marked improvement in K Means' ability to group the data when we reduce the number of dimensions in to two. Then t-SNE feature reduction is applied and obtain two tSNE component. t-SNE is a computationally heavy algorithm. The computation time can be reduced with the 'n-iter' parameter. Apply K Means clustering in to two components derived from t-SNE. Finally compared the clusters derived from PCA and t-SNE K Means and obtained univariate review of the clusters by comparing the clusters according to each individual feature.

Figure 3.5: K Means clustering on TSNE derived data.

3.6 Deep Learning Techniques Experimented

Deep Learning is capable of automatically finding correlations in raw data, and it has the advantage of supervised and unsupervised learning. Deep Learning techniques can significantly improve the performance of DDoS detection. To classify the network traffic into normal and DDOS attack traffic, DNN, CNN, LSTM. BiLSTM, TabNet and TCN are employed for DDoS detection.

3.6.1 Deep Neural Network (DNN)

The DNN model is comprised of feedforward neural networks that do not have feedback connections[3]. The input and output layers, and the hidden layer (which can be several), are the main components of the FNN. Each layer contains units with weights. The activation procedures of the units from the previous layer are made by these units. Since the DNN model's structure integrates feature extraction and classification procedures, it has the advantages of both supervised and unsupervised learning. The DNN model is created with sequential model having input layer, 3 hidden layer and output layer. The input layer having 22 neurons corresponds to 22 features in the dataset and relu activation function used. The hidden layers consist of equal number of 30 neurons and relu activation function is used. There are also two dropout layers with dropout ratio 0.2. The output layer consists of 2 neurons with softmax activation function because it is a binary classification problem. Compile the model using 'Sparse categorical cross entropy' loss function and 'Adam' optimizer. Fit the model with 20 epochs and batch size selected is 16.

3.6.2 Convolutional Neural Network (CNN)

CNN has proven to be effective in various studies and applications specifically in image classification field[4]. CNN consists of multiples layers: input layer, convolutional layers, pooling layers, fully connected layer and output layer. Deepness of the CNN dependence on the number of layers used. 1D CNN is used for the DDoS detection in SDN environment. The CNN model is created with sequential model having input layer, 2 convolution layer, batch normalization layer, max pooling layer, 2 dropout layer, flatten layer and 2 dense layer. The input layer consist of 22 neurons corresponds to 22 features selected and relu activation function used. The output layer consists of 2 neurons with softmax activation function. CNN model is compiled with Sparse categorical cross entropy and Adam optimizer.

3.6.3 Long Short Term Memory (LSTM)

Long-term and short-term memory model is a special RNN model, which is proposed to solve the problem of vanishing gradient and short term memory of RNN model[5]. LSTM model replaces RNN cells in the hidden layer with LSTM cells to make them have long-term memory ability. The LSTM model is created with sequential model. The LSTM architecture consist of one LSTM layer with 22 neurons and tanh activation function, the kernel regularizer is l2. The Dense layer having relu activation function and l2 regularizer. The output layer having softmax activation function. The model is compiled with sparse categorical cross entropy loss function and Adam optimizer.

3.6.4 Bi Directional Log Short Term Memory (BiLSTM)

BiLSTMs are an extension of traditional LSTMs that can improve model performance on sequence classification problems [6]. In bi directional, we can make the input flow in both directions to preserve the future and the past information. Bidirectional LSTMs train two instead of one LSTMs on the input sequence. The first on the input sequence as-is and the second on a reversed copy of the input sequence. The BiLSTM model is created with sequential model. The BiLSTM architecture consist of one BiLSTM layer with 22 neurons and tanh activation function and the kernel regularizer is l2. The Dense layer having relu activation function and l2 regularizer. The output layer having softmax activation function. The BiLSTM model is compiled with sparse categorical cross entropy loss function and Adam optimizer.

3.6.5 Tab Net

Tab Net is a deep neural network designed for learning from tabular data, developed by the Google Cloud AI research team [13]. Tab Net inputs raw data and trained with gradient descent-based optimization. Tab Net enables flexible integration into end-to-end learning. The sequential attention helps to choose the features and justify at each decision stage to allow better learning. The feature selection and reasoning is performed by single deep learning architecture. Tab Net enables two types of interpretability: local interpretability and global interpretability. Local interpretability visualizes the importance of features and their combination. Global interpretability quantifies the contribution of each feature to the trained model. Tab Net trains each row from a table, selects the relevant features at

each stage using a sparse learnable mask, and aggregates the predictions from each stage to emulate an ensemble effect when making predictions. The Tab Net classifier from Pytorch is used for DDoS detection. The Tab Net classifier model is defined with Adam optimizer, learning rate 0.005, step size =5, gamma=0.2 and mask type=entmax. The model is trained on the dataset and fitted with 30 epochs.

3.6.6 Temporal Convolutional Network (TCN)

TCN is a variant of convolutional neural network for sequence modeling tasks. TCN is a combinarion of CNN and RNN architectures [14]. TCN is capable of processing temporal sequence with any length. The baseline TCN architecture is formed by integrating the Four types of modern convolutional architectures, i.e.,1D convolutional network, causal convolution, dilated convolution, and residual connection.

- 1D Convolution : 1D convolutional network differs from the CNN, 2D Convolutional Network uses a 2D convolution kernel that moves line by line and Columns to extract features from 3D data and a 1D convolutional network uses a 1D kernel for the feature extraction along the sequence direction.
- Causal Convolution: TCN uses causal convolution to maintain a strictly causal temporal relationship, which means that output at time t is convolved only with elements from time t and earlier in the previous layer, that is no information leakage from future to past.
- Dilated convolution: The weakness of the causal convolution is that its receptive field is linearly related to the depth of the network. This will leads to high computational cost for long temporal sequence. To overcome this issue, a dilated convolution is used. A dilated convolution is a convolution where filter is applied over a region larger than its size by skipping input values with a given step. The dilated factor d of the dilated convolution has to be exponentially increse with the depth of the network. This ensures the receptive field covering each input in the history, and enabled to get an extremely large receptive field.
- Residual connection: The receptive field size of the TCN depends on the network depth n, filter size k and dilation factor d. The depth of a TCN should increase in order to obtain large receptive field . Residual connections have proven to be very effective in training deep networks. In a residual network, skip connections are used throughout the network, to speed up training process and avoid vanishing gradient problem in deep models.

Here the Keras TCN model is used for DDoS detection. The model is defined with the compiled TCN from Keras TCN. The hyper parameters used are nb-filters=22, activation=relu, kernel-initializer=he-normal, kernel-size=6, nb-stscks=1, and use-skip-connections=True. The model is trained on the dataset and fitted with 20 epochs.

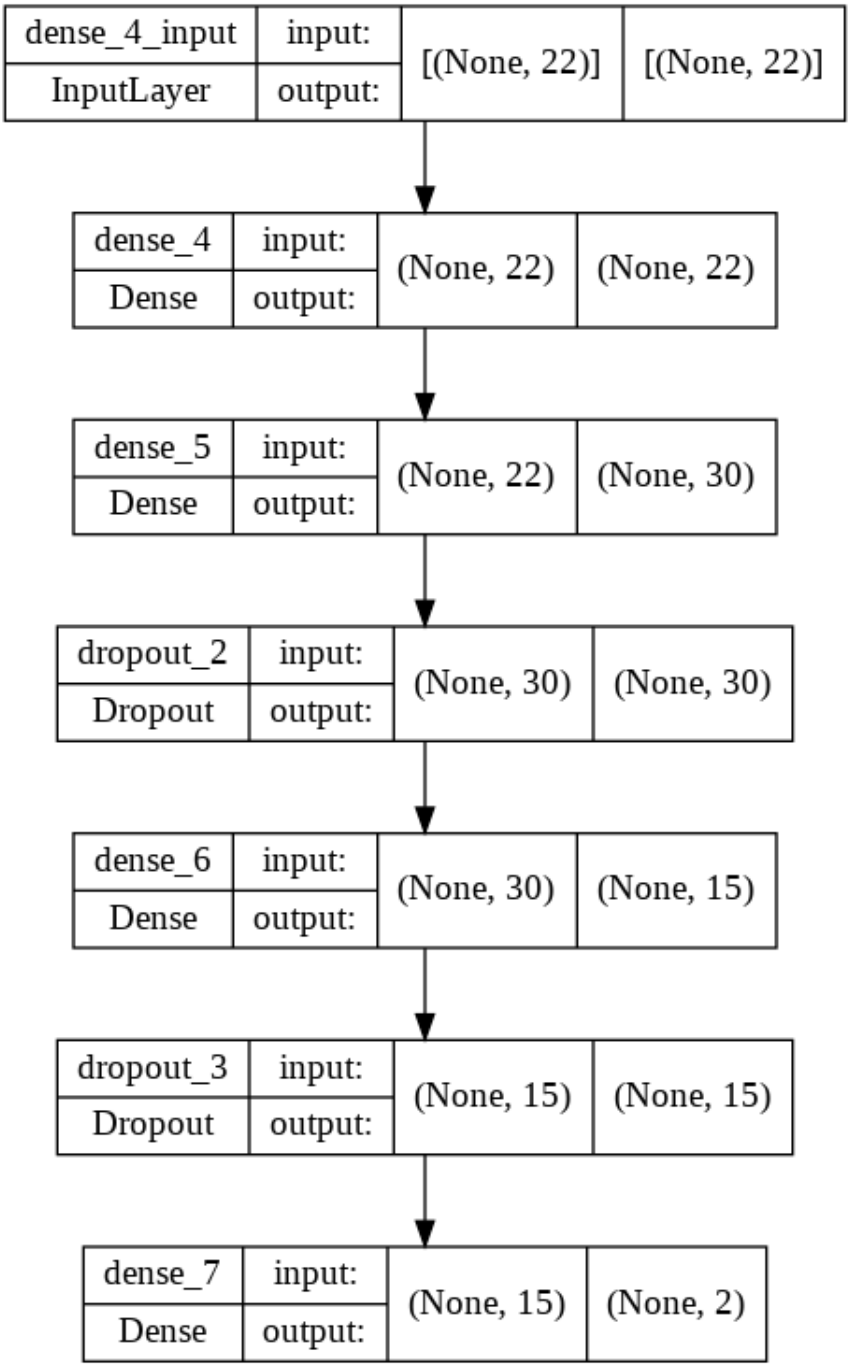

Figure 3.6: DNN architecture used.

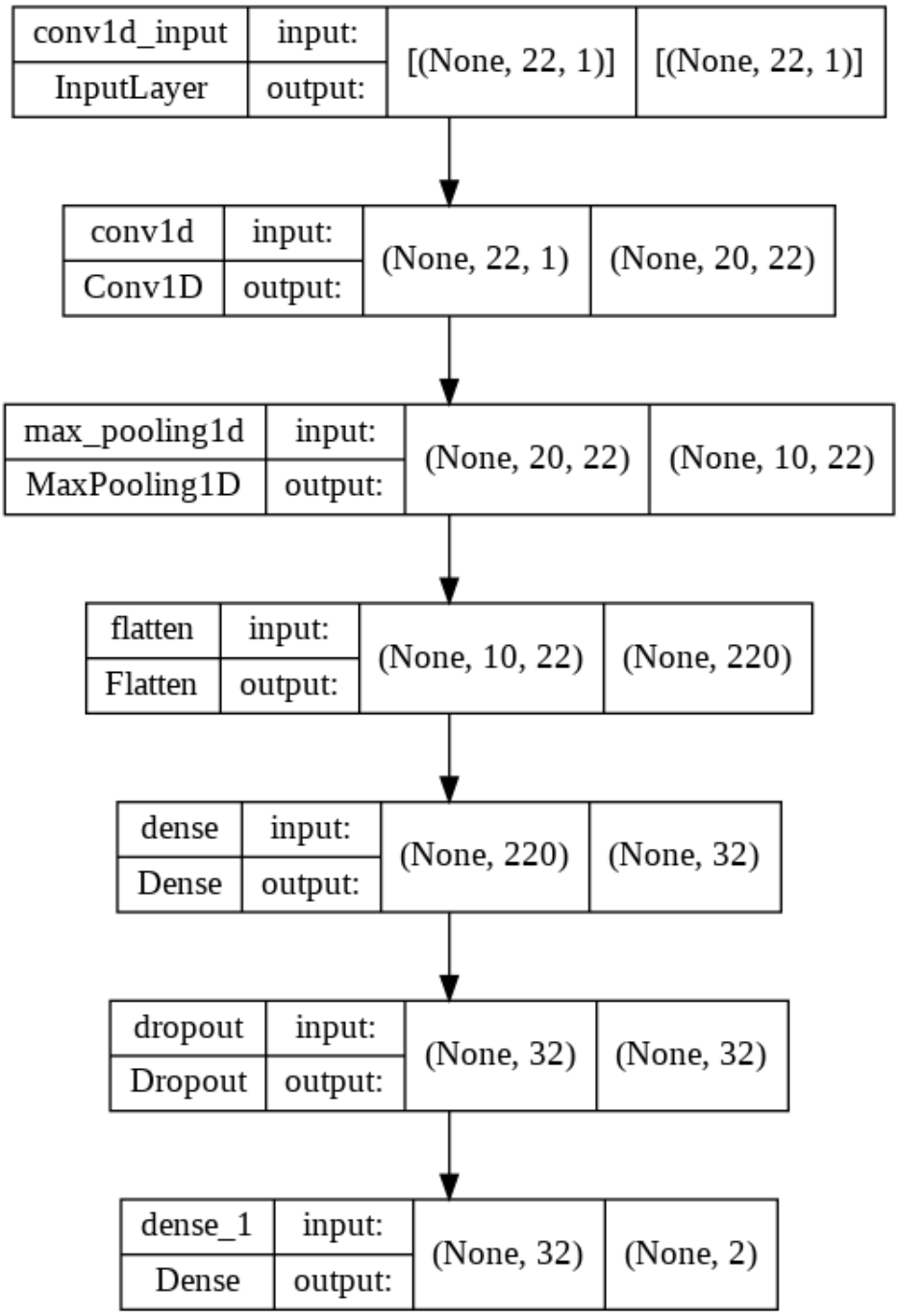

Figure 3.7: CNN architecture used .

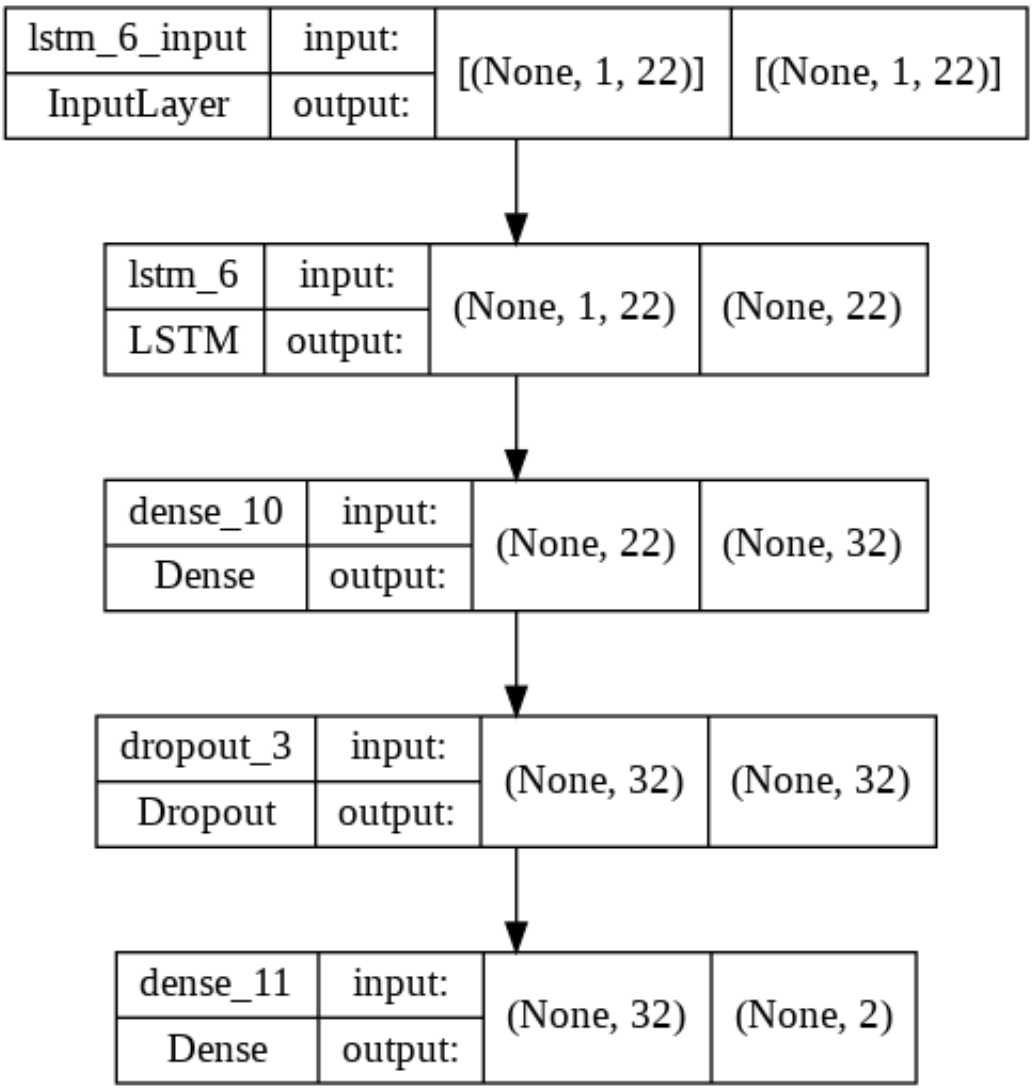

Figure 3.8: LSTM architecture used.

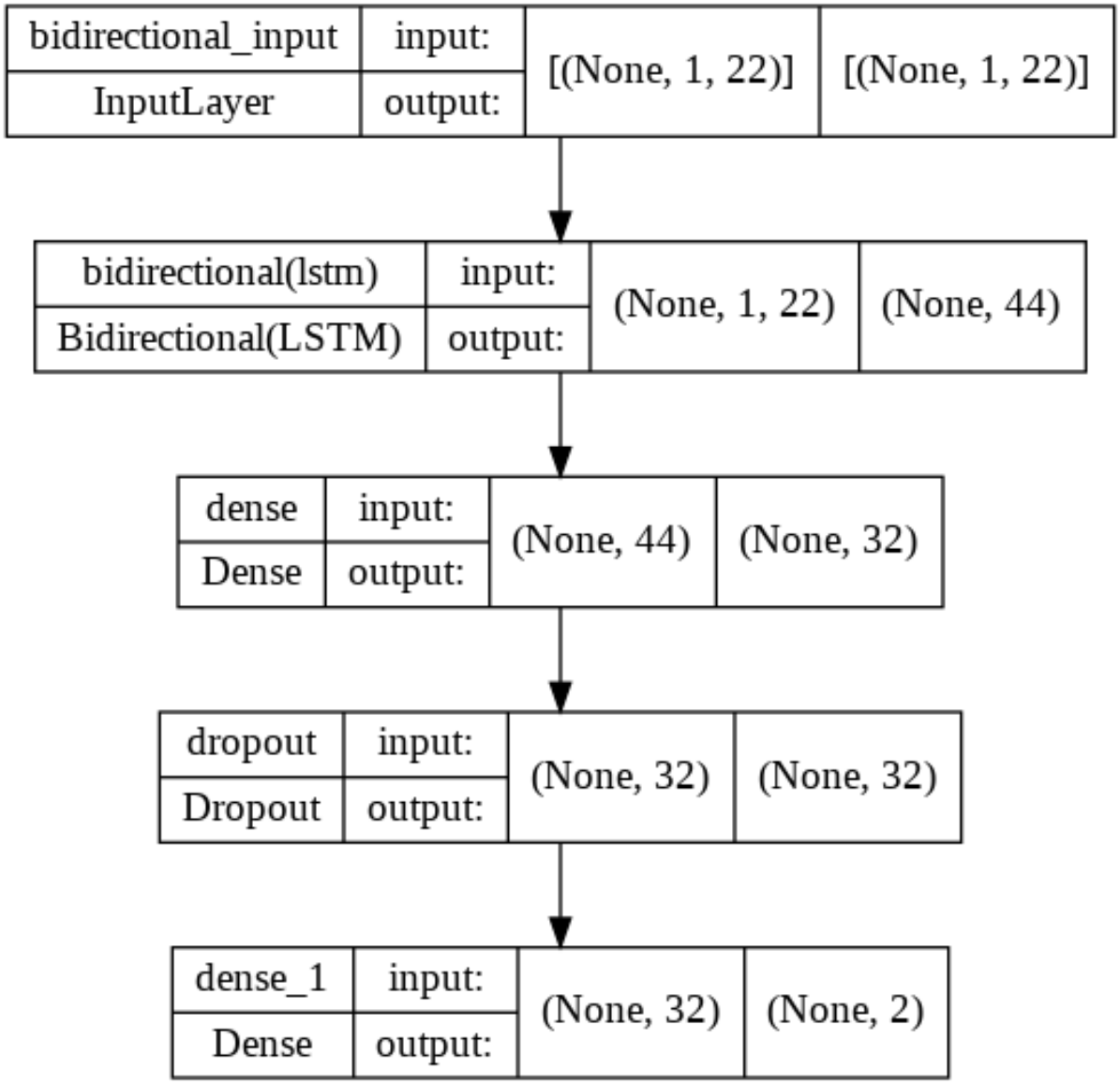

Figure 3.9: BiLSTM architecture used.

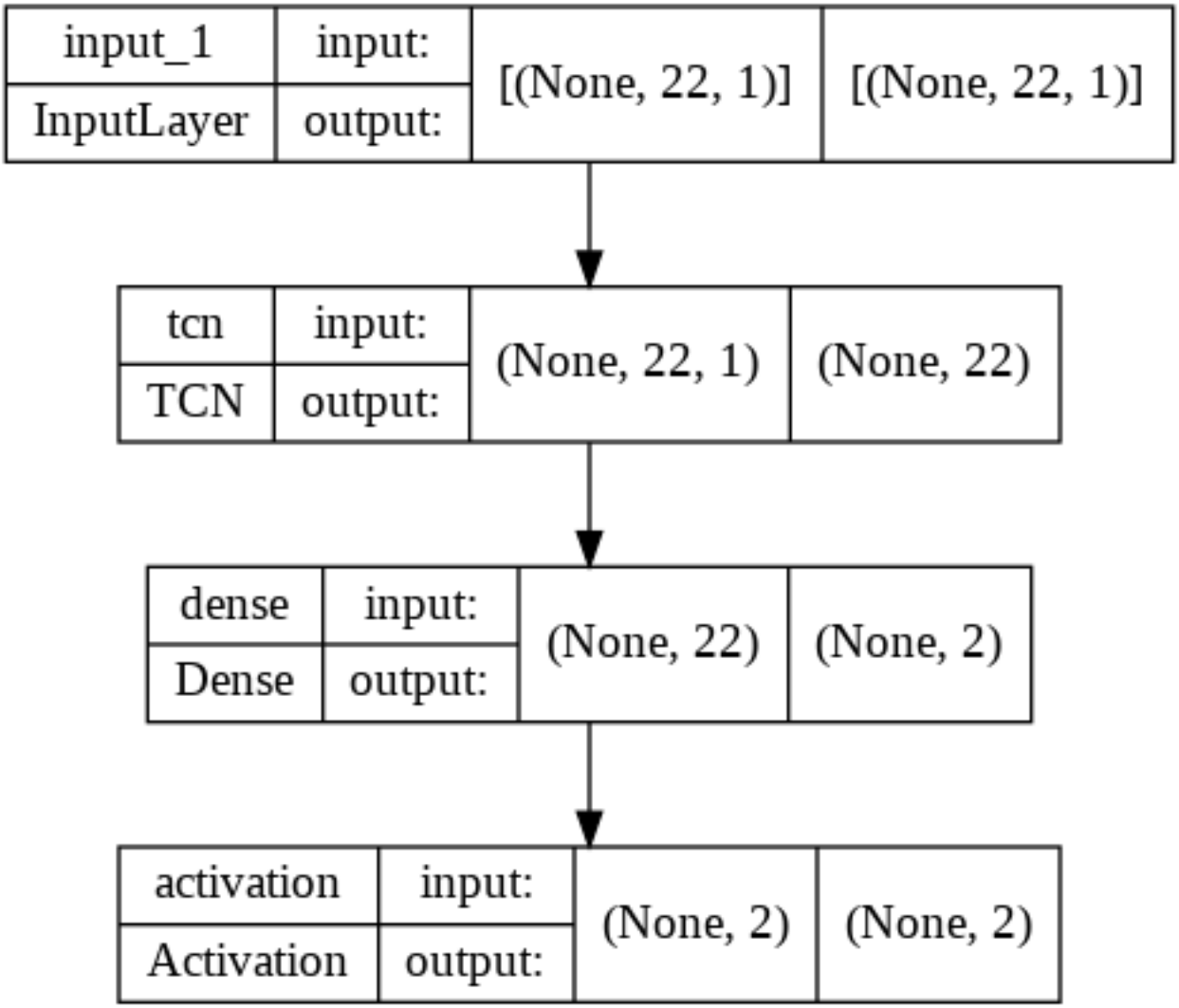

Figure 3.10: TCN architecture used.

Chapter 4

RESULTS AND DISCUSSION

In this section, the experiment environment, performance measurement metrics and experiment results are mentioned.

4.1 Environmental Setup

The hardware used for the experiments includes Windows 10 Pro OS, 64-bit operating system, x64-based processor, Intel(R) Core(TM) i7-1065G7 CPU @ 1.30GHz 1.50 GHz 16 GB RAM. The experimental environment was prepared using Python 3.7 programming language in Jupytor Notebook. The framework used is Keras with TensorFlow as background in the Anaconda environment. Machine learning and deep learning libraries include - NumPy, Pandas, Matplotlib, and Scikit learn.

4.2 Evaluation Metrics

Performance analysis identifies the best model having the highest accuracy. The general evaluation metrics such as Accuracy, Precision, Recall and F1 score are used for evaluation. The Silhouette method is used to evaluate the performance of clustering algorithms.

- Accuracy: The ratio of true identification over the total network traffic is shown.

$$Accuracy = \frac{TruePositive + TrueNegative}{TruePositive + TrueNegative + FalsePositive + FalseNegative} \tag{4.1}$$

- Precision: The ratio of correctly predicted abnormal traffic to the total abnormal traffic.

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \tag{4.2}$$

- Recall: The performance of the model on detecting abnormal network traffic.

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \tag{4.3}$$

- F1 Score : The accuracy of the model in whole dataset. It is the harmonic mean between precision and recall.

$$F1Score = \frac{2 * TruePositive}{2 * TruePositive + FalsePositive + FalseNegative} \tag{4.4}$$

- Silhouette Method: This method calculates the separation between clusters. The silhouette score ranges from -1 to 1. The score closer to 1, indicates that the clusters are well separated.

4.3 Experimental Results

4.3.1 Results of supervised Machine Learning techniques

The experimental results of supervised Machine learning techniques is summarized in the Table 4.1.

Evaluation Metrics	Accuracy (%)	precision	Recall	F1 Score
Logistic Regression	77.11	0.7268	0.6629	0.7554
Gaussian Naive Bayes	67.35	0.5754	0.6241	0.6617
SVM	78.44	0.7616	0.6517	0.7667
Gradient Boosting	98.52	0.9797	0.9904	0.9846
XGBoost	97.58	0.9741	0.9625	0.9743

Table 4.1: Experimental results of Supervised Machine Learning Techniques.

From Table 4.1 it is clearly visible that the Gradient Boosting and XGBoost classifiers are performs well and gives best results in terms of accuracy, precision, recall and F1 score. This indicates that the tree based ensemble classifiers are performing very well compared to other traditional Machine Learning algorithms.

4.3.2 Result of Machine Learning Reasoning

We implemented Machine Learning reasoning with LIME library. From the graphical results obtained, we can see that the model predicts the network traffic is normal or DDoS attack .Lime library helps visualizing the features that help the model to make that decision. After seeing these modeling explanations, we can decide whether to trust the decision or not.

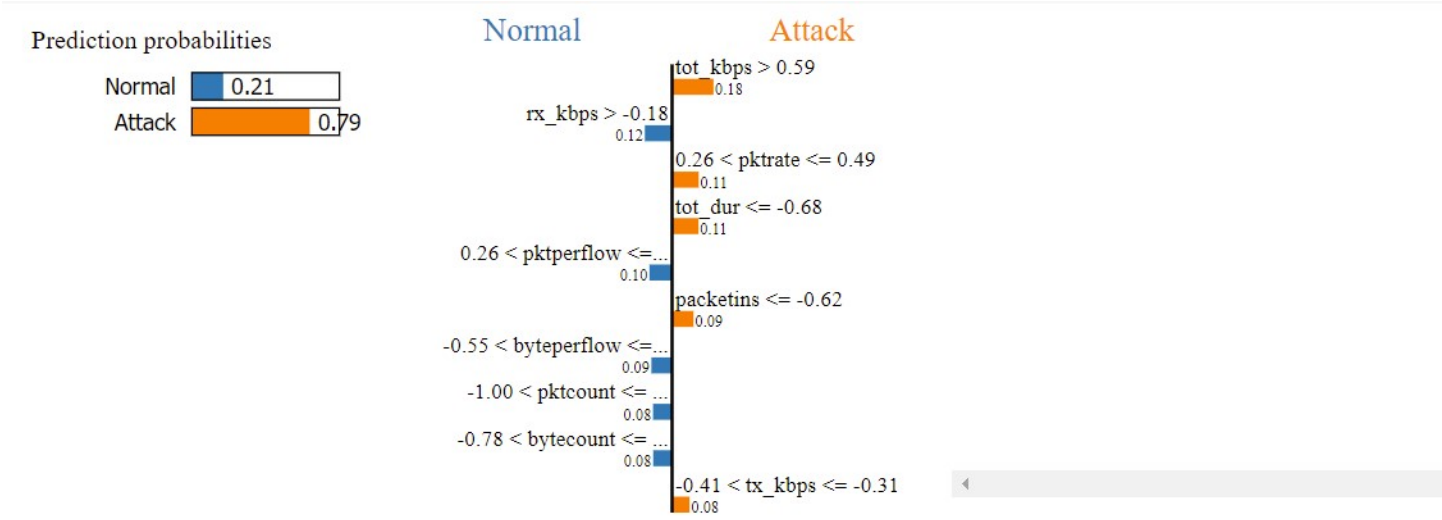

Figure 4.1: Reasoning of Logistic Regression model for DDoS traffic.

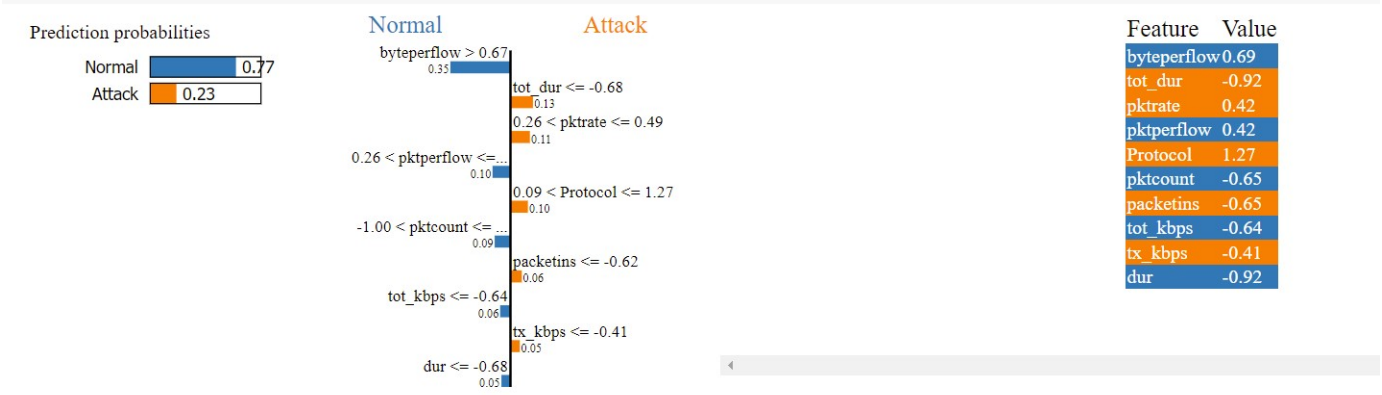

Figure 4.2: Reasoning of Logistic Regression model for Normal traffic .

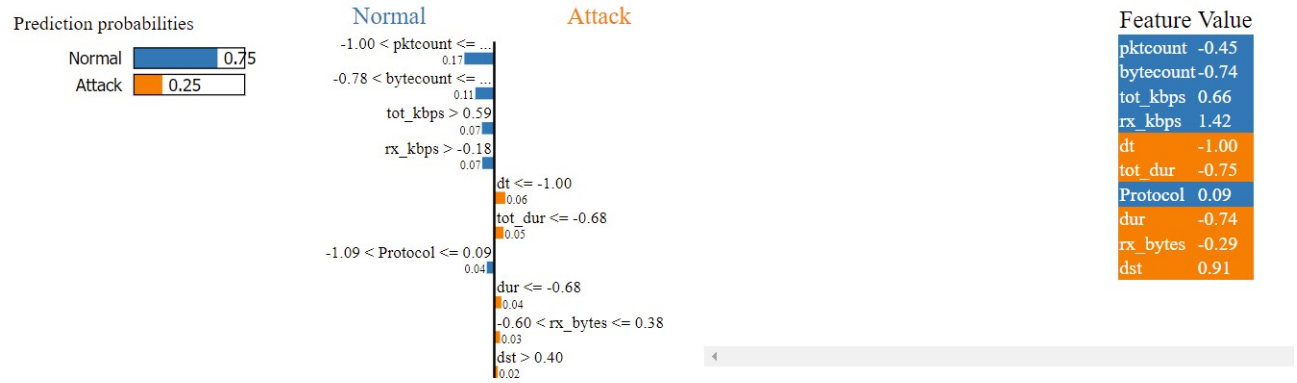

Figure 4.3: Reasoning of Gaussian Naive Bayes model for DDoS traffic .

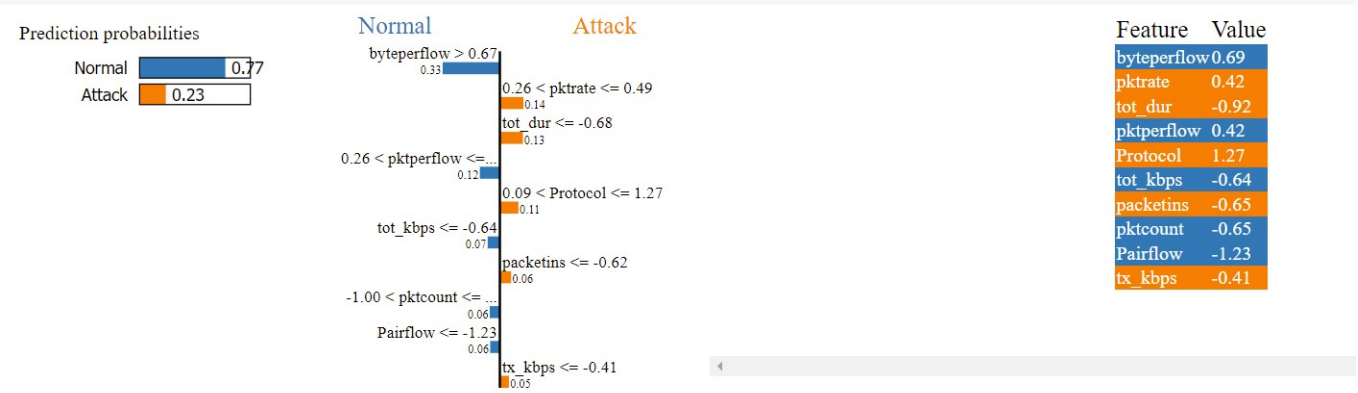

Figure 4.4: Reasoning of Gaussian Naive Bayes model for Normal traffic .

Figure 4.5: Reasoning of Gradient Boosting model for DDoS traffic .

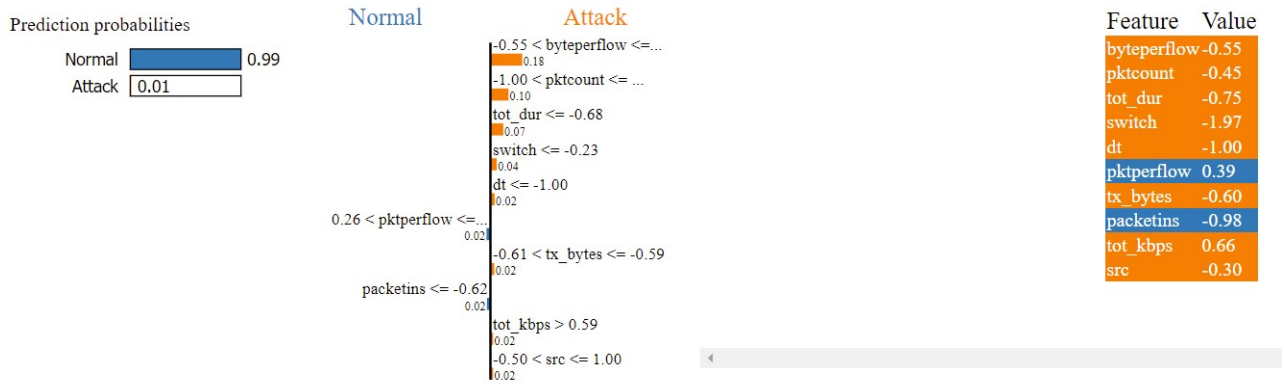

Figure 4.6: Reasoning of Gradient Boosting model for Normal traffic .

4.3.3 Results of clustering techniques

The experimental result of clustering techniques is summarized in the Table 4.2.

Tecchniques	Sillhouette Score
Kmeans on original dataset	0.1961007339513506
Kmeans on PCA	0.4613677492070967
Kmeans on TSNE	0.34130406379699707

Table 4.2: Experimental results of clustering Techniques.

From Table 4.2, We achieved a silhouette score of 0.196 in K-means on original dataset which is on the low end. We can see a definite improvement in K Means ability to cluster our data when we reduce the number of dimensions to 2 principal components. We get a K-means PCA Scaled Silhouette Score of 0.4613677492. Applying K Means to t-SNE derived components we obtained a Silhouette score of 0.3413. Comparing the silhouette score of PCA and TSNE derived K-means clustering indicates that PCA K-means gives the better silhouette score.

4.3.4 Results of feature selection

The feature importance graph of DDoS attack SDN dataset from XGBoost feature selection is shown in Fig.4.1. From this graph we can understand the feature importance of each individual feature and the 14 most effective features are selected. The features selected are 'packetins','byteperflow', 'pktperflow','bytecount','pktcount','dt','Protocol','tot-dur','src','pktrat e','dst', 'dur', 'tx-bytes' and 'rx-bytes'.

The feature weights of each individual feature using NCA feature selection is shown in Fig.3. The 14 most effective features are selected for the training of models are 'src','dst','dt', 'pktcount', 'bytecount', 'dur', 'tot-dur', 'flows', pktperflow','byteperflow','pktrate', 'Protocol', 'rx-kbps' and 'tot-kbps'.

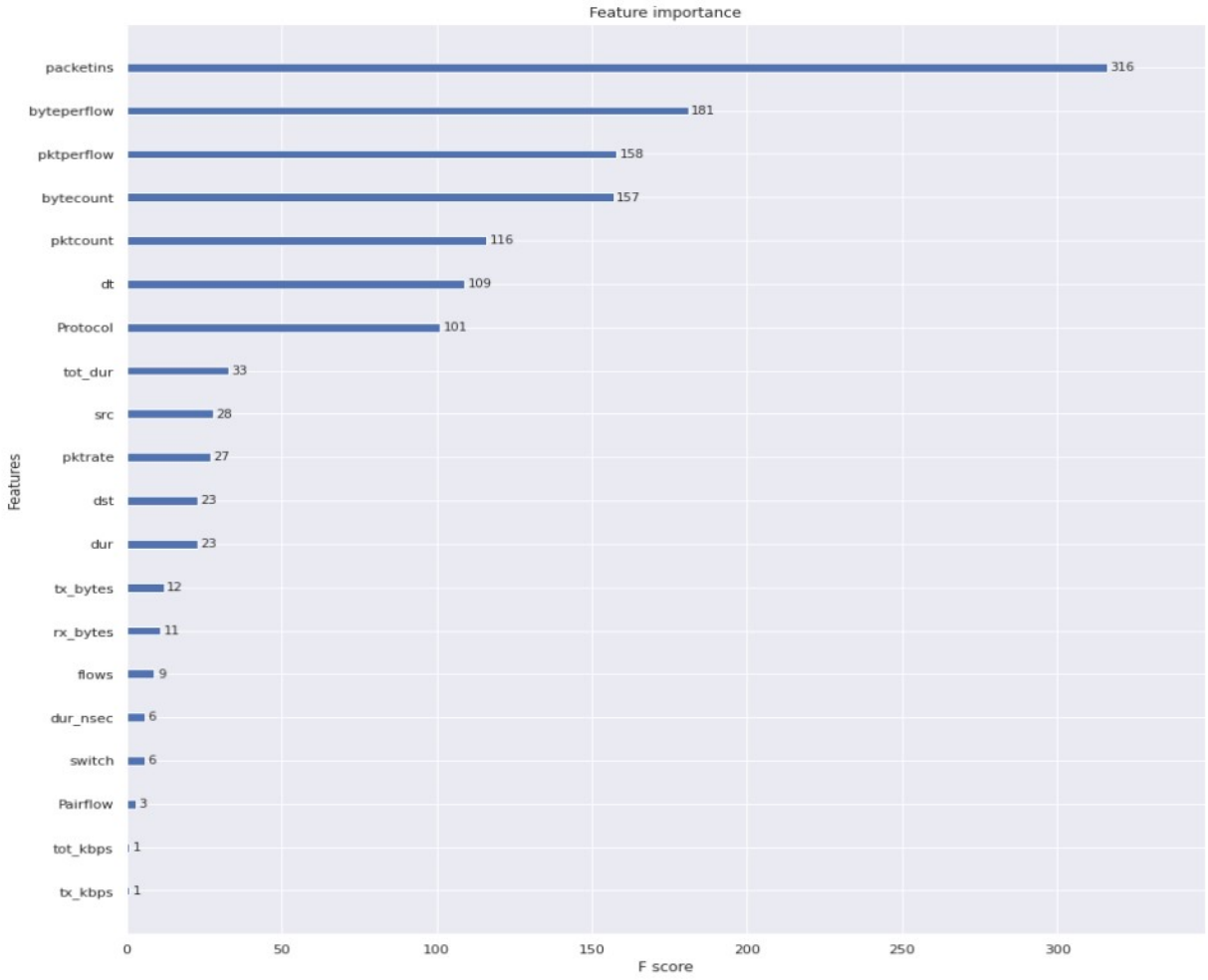

Figure 4.7: XGBoost feature selection- feature importance graph .

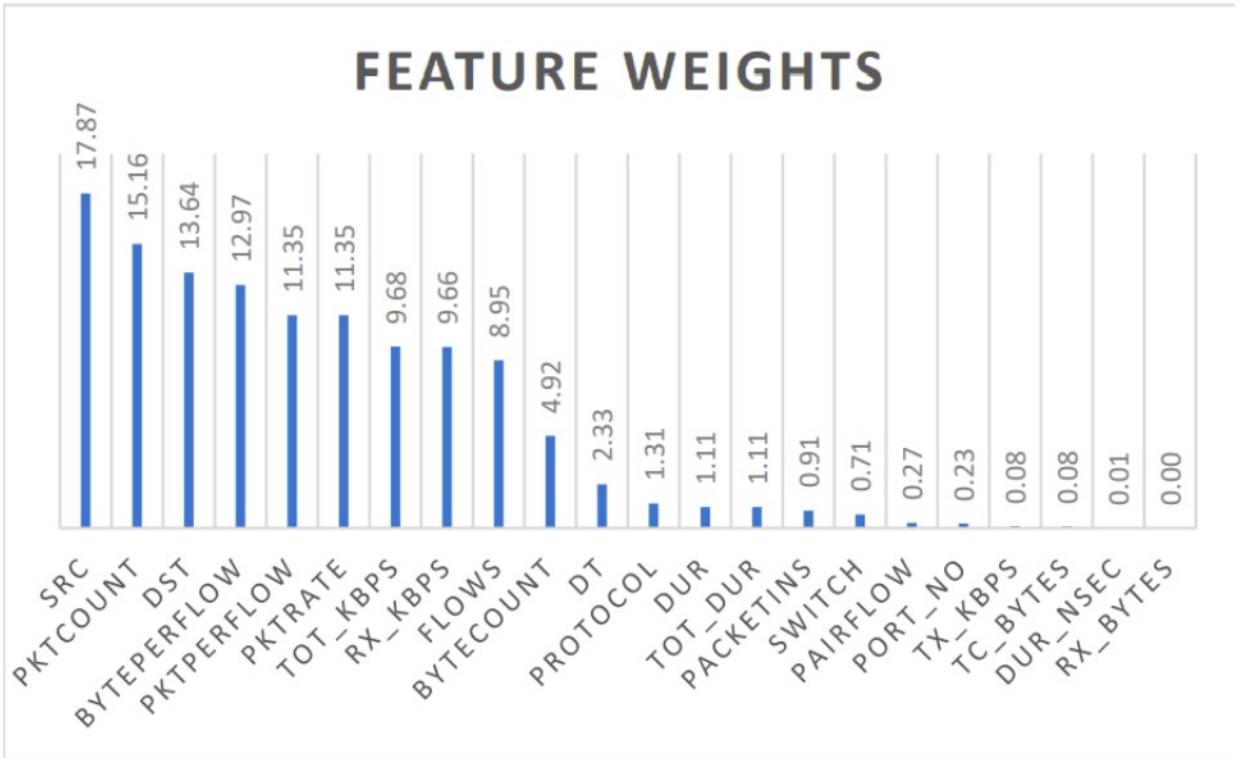

Figure 4.8: NCA feature weights graph .

4.3.5 Results of Deep Learning Techniques

The experimental results of Deep learning techniques are summarized in Table 4.3

Evaluation Metrics	Accuracy (%)	precision	Recall	F1 Score
DNN	77.11	0.7268	0.6629	0.7554
CNN	67.35	0.5754	0.6241	0.6617
LSTM	78.44	0.7616	0.6517	0.7667
BiLSTM	98.52	0.9797	0.9904	0.9846
Tab Net	97.58	0.9741	0.9625	0.9743
TCN	97.58	0.9741	0.9625	0.9743

Table 4.3: Experimental results of Deep Learning Techniques.

The experimental results of Deep Learning techniques with NCA feature selection is summarized in Table 4.4.

Evaluation Metrics	Accuracy (%)	precision	Recall	F1 Score
DNN	77.11	0.7268	0.6629	0.7554
CNN	67.35	0.5754	0.6241	0.6617
LSTM	78.44	0.7616	0.6517	0.7667
BiLSTM	98.52	0.9797	0.9904	0.9846
Tab Net	97.58	0.9741	0.9625	0.9743
TCN	97.58	0.9741	0.9625	0.9743

Table 4.4: Experimental results of Deep Learning Techniques with NCA feature selection.

Evaluation Metrics	Accuracy (%)	precision	Recall	F1 Score
DNN	77.11	0.7268	0.6629	0.7554
CNN	67.35	0.5754	0.6241	0.6617
LSTM	78.44	0.7616	0.6517	0.7667
BiLSTM	98.52	0.9797	0.9904	0.9846
Tab Net	97.58	0.9741	0.9625	0.9743
TCN	97.58	0.9741	0.9625	0.9743

Table 4.5: Experimental result of Deep Learning Techniques with XGBoost feature selection.

4.3.6 Experimental Results in INSDN dataset

The experimental results of DDoS attack SDN dataset implies that our proposed TCN model outperforms all the other models in DDoS detection in SDN environment. In order to generalize the performance of the proposed model, we evaluated the performance of DDoS detection using Tab Net and TCN models with INSDN dataset. The experiments are done without feature selection and with XGBoost feature selection. The XGBoost feature selection gives the feature importance of each individual feature and the 14 most effective feature are selected. The features selected are 'Timestamp','Flow ID','Dst Port','Src Port','Fwd IAT Min','Fwd Pkt Len Mean', 'Fwd Pkt Len Std','B wd Header Len','Init Bwd Win Byts','Fwd Pkt Len Max','TotLen Fwd Pkts','Flow Duration','Tot Bw d Pkts' and 'Fwd Act Data Pkts'. The feature importance graph of XGBoost feature selection in INSDN dataset is shown in Fig 4.3.

4.3.7 Experimental results of INSDN dataset without feature selection

The experimental results of INSDN dataset without feature selection is summarized in Table 4.6 .

4.3.8 Experimental results of INSDN dataset with XGBoost feature selection

The experimental result of INSDN dataset with XGBoost feature selection is summarized in Table 4.7

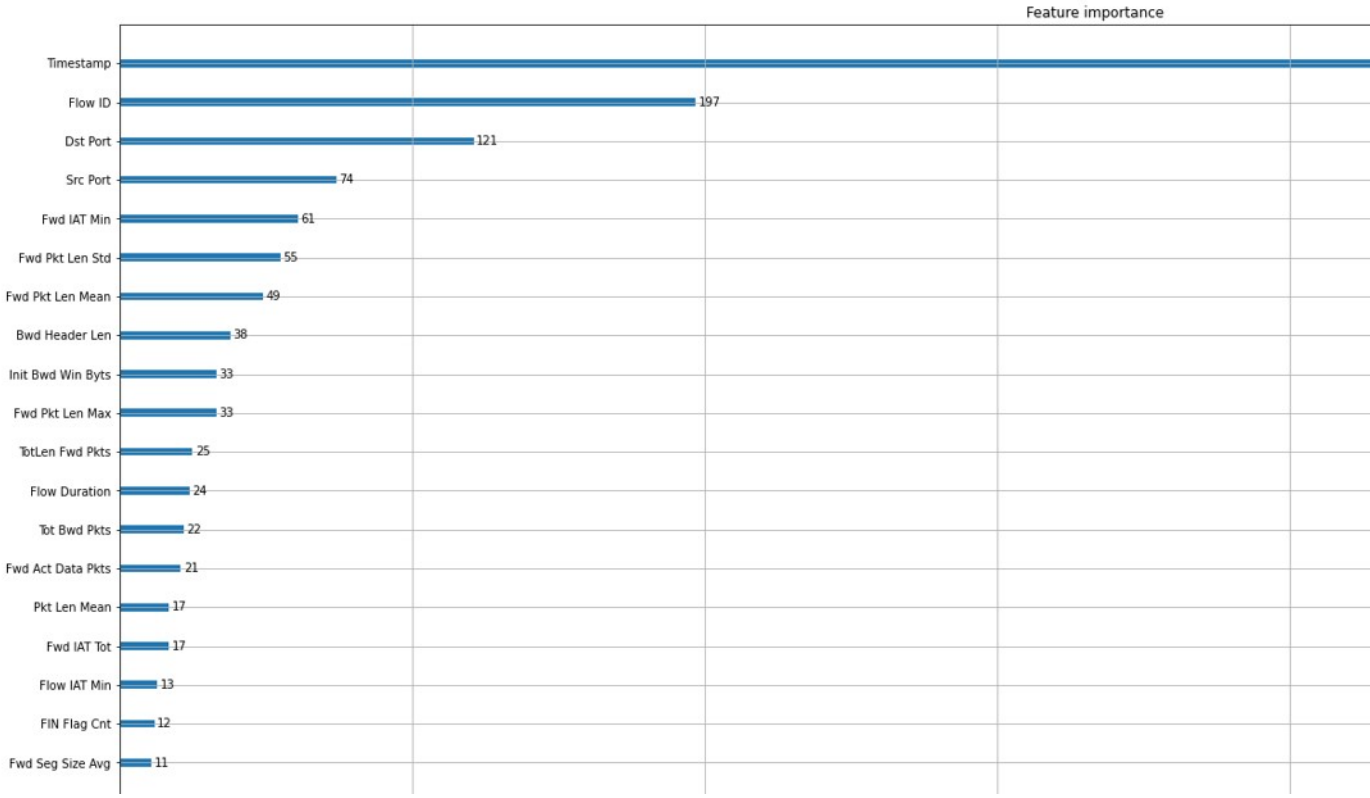

Figure 4.9: XGBoost feature selection in INSDN dataset .

The experiments in INSDN dataset also indicates that the proposed TCN model having the best performance in terms of accuracy, precision, recall and F1 score. The XGBoost feature selection enhances the performance of the model and gives the highest accuracy of 99.98% for DDoS detection in SDN environments with TCN model.

Evaluation Metrics	Accuracy (%)	precision	Recall	F1 Score
Tab Net	99.95	0.9126	0.9083	0.9104
TCN	99.95	0.9997	0.9995	0.7885

Table 4.6: Experimental results of Deep Learning Techniques without feature selection in INSDN dataset.

Evaluation Metrics	Accuracy (%)	precision	Recall	F1 Score
Tab Net	99.98	0.9927	0.9250	0.9501
TCN	99.98	0.9999	0.9998	0.9451

Table 4.7: Experimental results of Deep Learning Techniques with XGBoost feature selection in INSDN dataset.

Chapter 5

CONCLUSION

This work focused on proposing and demonstrate the design, implementation, and testing of detecting DDoS by Temporal Convolutional Network (TCN) that would allow end-users to identify DDoS attacks. We evaluated our model using the DDoS attack SDN dataset, which contains a comprehensive variety of DDoS attacks. We compared several state-of-the-art Machine Learning and Deep Learning models that are well known for detecting DDoS attacks. In this work, we have evaluated the performance of the supervised Machine Learning algorithms, unsupervised Machine learning techniques and supervised neural networks. In the case of supervised Machine Learning algorithms, we have implemented Logistic Regression, Gaussian Naïve Bayes, SVM, Gradient Boosting and XGBoost classifier. We have used the Machine Learning reasoning library LIME to understand the reasoning behind the Machine Learning models. The experimental result implies that Tree-based ensemble algorithms such as Gradient Boosting and XGBoost perform very well compared to other models and gives the best results. The clustering technique is implemented with PCA and TSNE dimensionality reduction technique in K Means clustering. Silhouette score is used to measure the performance of K Means clustering. The K Means clustering with PCA dimensionality reduction technique gives the highest silhouette score. The NCA and XGBoost feature selection methods are used for the feature selection, and Autoencoder is used for the feature extraction. The 14 most effective features are selected according to the feature importance score obtained from NCA and XGBoost feature selection and used for the training. Deep Learning Algorithms such as DNN, CNN, LSTM, BiLSTM and TabNet are employed for DDoS detection and evaluate the performance of the models with and without the feature selection techniques. The XGBoost feature selection gives the best results in all models than the NCA feature selection. The experimental result shows that Deep Learning algorithms can significantly improve the performance of DDoS detection in SDN environments. Compared to other benchmark models, our proposed model TCN performs very well and gives the highest accuracy, precision, recall and F1 score among other models. The XGBoost feature selection enhanced the performance of the TCN model and obtained the highest accuracy of 99.58%. To generalize the performance of the TCN model, we evaluated the performance of the TCN model in the INSDN dataset with XGBoost feature selection and without feature selection. The experimental results indicate that the TCN model performs very well in the INSDN dataset with XGBoost feature selection and gives the best accuracy of 99.98%.

References

[1] Tonkal, Ö., Polat, H., Başaran, E., Cömert, Z. and Kocaoğlu, R., 2021. Machine Learning Approach Equipped with Neighbourhood Component Analysis for DDoS Attack Detection in Software-Defined Networking. Electronics, 10(11), p.1227.

[2] Aamir, M. and Zaidi, S.M.A., 2021. Clustering based semi-supervised machine learning for DDoS attack classification. Journal of King Saud University-Computer and Information Sciences, 33(4), pp.436-446.

[3] Cil, A.E., Yildiz, K. and Buldu, A., 2021. Detection of DDoS attacks with feed forward based deep neural network model. Expert Systems with Applications, 169, p.114520.

[4] Haider, S., Akhunzada, A., Mustafa, I., Patel, T.B., Fernandez, A., Choo, K.K.R. and Iqbal, J., 2020. A deep CNN ensemble framework for efficient DDoS attack detection in software defined networks. Ieee Access, 8, pp.53972-53983. .

[5] Elsayed, M.S., Le-Khac, N.A., Dev, S. and Jurcut, A.D., 2020, August. Ddosnet: A deep-learning model for detecting network attacks. In 2020 IEEE 21st International Symposium on" A World of Wireless, Mobile and Multimedia Networks"(WoWMoM) (pp. 391-396). IEEE.

[6] Cui, Y., Qian, Q., Guo, C., Shen, G., Tian, Y., Xing, H. and Yan, L., 2021. Towards DDoS detection mechanisms in software-defined networking. Journal of Network and Computer Applications, 190, p.103156.

[7] Ahuja, N., Singal, G. and Mukhopadhyay, D., 2020. DDOS attack SDN dataset. Mendeley Data, 1.

[8] Alsahaf, A., Petkov, N., Shenoy, V. and Azzopardi, G., 2022. A framework for feature selection through boosting. Expert Systems with Applications, 187, p.115895.

[9] Kumari, K. and Mrunalini, M., 2022. Detecting Denial of Service attacks using machine learning algorithms. Journal of Big Data, 9(1), pp.1-17.

[10] Alduailij, M., Khan, Q.W., Tahir, M., Sardaraz, M., Alduailij, M. and Malik, F., 2022. Machine-Learning-Based DDoS Attack Detection Using Mutual Information and Random Forest Feature Importance Method. Symmetry, 14(6), p.1095.

[11] Decrypting your Machine Learning model using LIME,towardsdatascience.com

[12] t-distributed stochastic neighbor embedding,https://en.wikipedia.org/wiki/T-distributed-stochastic-neighbor-embedding

[13] Arık, S.O. and Pfister, T., 2021. Tabnet: Attentive interpretable tabular learning. In AAAI (Vol. 35, pp. 6679-6687).

[14] Yue, C., Wang, L., Wang, D., Duo, R. and Yan, H., 2021. Detecting Temporal Attacks: An Intrusion Detection System for Train Communication Ethernet Based on Dynamic Temporal Convolutional Network. Security and Communication Networks, 2021.

LIST OF PUBLICATIONS

Subuhana, N., Aysha Rega, and Sumod Sundar. "Deep learning techniques for breast cancer analysis: A review." In 2021 Fourth International Conference on Microelectronics, Signals Systems (ICMSS), pp. 1-6. IEEE, 2021.