National College of Ireland

Enhanced the intrusion detection accuracy rate and performance using deep CNN-LSTM.

MSc Research Project
Msc. Cyber Security

Sweety Kaushik
Student ID: X19205783

School of Computing
National College of Ireland

Supervisor: Ross Spelman

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Sweety Kaushik
Student ID:	X19205783
Programme:	Msc. Cyber Security **Year:** 2020-2021
Module:	Internship
Supervisor:	Ross Spelman
Submission Due Date:	16-08-2021
Project Title:	**Enhanced the intrusion detection accuracy rate and performance using deep CNN-LSTM.**
Word Count:	5428 **Page Count: 21**

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: ……Sweety Kaushik……

Date: ……16-08-2021……

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):	

Enhanced the intrusion detection accuracy rate and performance using deep CNN-LSTM.

Sweety Kaushik
X19205783

Abstract

The Internet of Things (IoT) is a fast-increasing worldwide network that facilitates the connection and sharing of data amongst all intelligent devices. The Internet of Things units is handled continuously. Attackers may be able to take advantage of these gadgets very fast in the next generation, making the Internet of Things a major concern. With an intrusion detection system, the Internet of Things detects IoT threats and network problems. When an intrusion into devices is involved, it is quite easy for IoT devices to be lost or damaged. The Internet of Things devices, which serve as a defence for the entire network, is sometimes neglected as being critical to the security of the network and the protection of the data collected. The purpose of this study is to propose a novel model for improving the precision and efficiency of intruder detection systems for the Internet of Things devices. The model is based on the 2017 CICIDS data set and a deep CNN-LSTM neural network.

Keywords: Cybersecurity, Intrusion Detection, CNN-LSTM. 2017 CICIDS Dataset.

1 Introduction

The intrusion detection system protects IoT devices from harmful attacks. It protects the network from unwanted activity. Data security is crucial to preventing data breaches. Attackers also prevent users from regulating current cyber threats like malware or phishing. Intrusion Detection Systems need to be secure (IDS). IoT devices continue to grow. The risks and attraction of IoT devices grow. We need data from IoT devices over traditional equipment. The IoT devices include several sensors to collect data. Power is the main issue with IoT devices. Normal safety procedures fail. It's hard to investigate IoT devices. Humans just control these gadgets. The IDS scans network traffic for threats. Intrusion detection categorises binary or international data. An IDS cannot stop malicious traffic. It can only detect network intrusions. IoT devices employ intrusion detection systems. The IDS enhances classification [1].

Deep learning is a relatively recent field of research in IDS, with a range of proposed approaches. For over a decade, the intrusion detection system has evolved. It relies on machine learning and in-depth learning and is employed in many networks and devices. As in previous forms of neural networks, the modern deep CNN-LSTM [2] neural network benefits from visual imaging. This approach uses deep CNN-LSTM layers to eliminate feedback and produce results with high accuracy. This study presents a strategy to identify intrusions with a deep CNN-LSTM neural network. The above-mentioned deep learning technology is combined with the data set CICIDS-2017. We developed in this part an id model using the latest data set from CICIDS 2017, which contains the most recent serious attacks [3]. This study looks at the IDS design technique on IoT devices.

The IDS cyber security architecture builds accurate deep learning models. In order to attain this goal, each technique favours particular design decisions. A few IoT IDSs, for example, don't comprehend the issue, or use a contradicting data set to evaluate their models. In addition, some IDS rely on old IoT network traffic

or unreliable information. This understanding prompted scientists to create a model that can tackle many problems.

Users today are more worried about data security. For users who may leak classified data, we focus on IDS put in Internet of Things devices. To increase the precision of intrusion detection systems, the Bagging Ensemble and other machine learning algorithms were used [4]. The CICID Set 2017 dataset now has a deep CNN-LSTM neural network to improve data duplication detection. We need IoT devices for intrusion detection [5].
This research subject was chosen to improve performance by applying AdaBoost-based algorithms to detect intrusions in the CICIDS 2017 data set. Using deep neural CNN-LSTM networks in CICIDS 2017. The new CICIDS 2017 dataset can be used to train a CNN-LSTM classification with an estimated 95% accuracy. Several repeating entries for both training and testing result from the data set's imbalance. Using the CICIDS 2017 dataset, we developed an in-depth neural network/LSTM approach.

How can the detection rate of intrusion detection systems be effectively increased using novel techniques such as Convolutional Neural Network-LSTM layers and the updated dataset CICIDS-2017?

This report is comprised of three sections. The study's structure is summarized as follows:

Section 1 summarizes the Introduction, study's background, purpose, research question, objectives, and reason for doing the study. The second section explores the literature that was chosen for the review. Additionally, a review of the literature is undertaken in the field of earlier academics' research. Section 3 & 4 examines and briefly describes the methods & Design specification employed to finish the report's study. Sections 5,6 and 7 will review the finished work, present an overview, and provide replies in the form of results and recommendations for additional research.

2 Related Work

Through visual and behavioral analysis, machine learning algorithms (ML algorithms) have gained significance in recent years in computer science (Anomaly detection) [6]. Additionally, machine learning is quite promising in specific domains [7][8], but deep learning yields novel insights. According to Parampottupadam and Moldovann[9], the amount of data necessary for training is not always more than for standard machine learning models. Algorithms for machine learning use statistical models to produce predictions without human input. They include supervised, unchecked, semi-supervised algorithms and machine enhancements. These are supervised machine learning and profound learning algorithms. Efficacy algorithms of previous trials have been assessed. Algorithms like logistic regression, Gaussian Naive Bays, K-nearest neighbors, treaties, adaptive stimulation and random woods have been utilized (RF). The CNN, Convolutional Neural Networks, Deep Networks and CNN-LSTM are our favorite profound learning algorithms (DNN).
Recent research has concentrated on the use of machine learning to identify potentially malicious cyber security activity. Vinayakumar et alDNN 's architecture was built using KDDCup-99, NSL-KDD, UNSW-NB15, WSNDS, and CICIDS2017[10]. The binary precision of DNN's five veiled layers is 92.7, 78.1, 98.2, and 93.1 percent. [11] Zhang et al. used deep hierarchical networks to detect network intrusions using current flow data from the CICIDS2017 and CTU datasets. The accuracy of the CNN + LSTM

classification algorithm was 99.8% for CICIDS 2017 and 98.7% for CTU data. Using ML classification methods such as decision-making tree, SVM, RF, and Naive, the UNSW-NB15 data set achieved 97.49 percent accuracy. Beluch, et al. [12] Faker et al. [13] examined three data subsets from UNSW-NB15 and CICICIDS2017 (DNN, RF, and GBT). Using UNSW-NB15 data and CICIDS 2017 data, the researchers concluded with maximum performance in binary and multitasks threat categorization.

Based on their research, Liu & al [14] revealed the greatest detection and accuracy rate compared to other IDS classifications of intrusional detection models based on convolutionary neural networks (NNC). CNN was demonstrated to be a suitable solution to the problem for significantly intruded detection. The authors argue that CNN-based algorithms' performance classification approaches are an alternative. Wang et al. [15] developed an intrusion detection system based on a CNN that can train autonomously, significantly reducing the rate of false alarms (FAR). This study demonstrates how in-depth learning techniques can be used to accurately extract network traffic.

The recommendations of Yin et al [16] include that RNN-IDS be compared to ANN, RF, SVM etc. A very accurate classification model exceeding classical binary and multi classification approaches is generated by RNN-IDS. Yin and others. Shone et al. [17] unveiled a non-symmetrical functional training unattended deep self-encoder (NDAE). By comparing the auto encoder to a deep encoder, this study increases the performance of KDD99 and NSL KDD99 (NDAE). Wu et al. [18] developed CNN and RNN to detect attacks, however their model differed from the model in our study since CNN and RNN were developed independently.

Naseer et al. [19] studied the adequacy of intrusion detection systems based on deep learning techniques. Ding and Zhai [20] compared model performance to that of traditional machine learning techniques for multi-class classification. DL was created by Otoum for IDS in wireless sensor systems (WSNs) and also contrasted Boltzmann's hybrid IDS, adaptively supervised and clustered, with the clustered RBCIDS and adaptive IDS (ASCH-IDS).

In order to detect a network interference and a residual architecture of learning, Chouhan et al. [21] established the CBR-CNN channel. The task uses uncontrolled, stacked car encoders (SAE) and evaluates the performance with an NSL-KDD data set of the CBR-CNN approach. A Vinayakumar et al. [22] development of an IDS was developed to recognise and classify unforeseen and unexpected DNN cyber threats. The NSL-KDD, UNSW-NB15, Kyoto, WSN-DS and CICIDS2017 model tests were carried out using DNN.

Chiba et al. [23] suggested a DNN model with enhanced genetic algorithms (IGAAs) and cloud cloud-based IDS anomaly networks that included latest CICIDS2017, NSL-KDD2015 and CIDDS-001 data sets (SAA). Zhang et al. [24] used the SQL network identification model for the DBN assault attack. Faker and Dogdu[25] have been using data from CICIS2017 and UNSW NB15 to develop better intrusion detection systems, both with DNN, RF and gradient enhancement tree (GBT). In prior work, new concepts or tactics were presented to improve deep learning algorithms.

To address security concerns, Aloqaily et al. [26] presented an autonomous cloud-based smart accessible automobile intrusion detection system. However, in many cases, intrusion detection in real-world environments is difficult to employ as modelling is pre-processed mostly in metadata forms in experimental contexts. Few studies have shown how they can be implemented in real time.

2.1.1 **2.2 Literature Review Gap**

Ring et al. [27] conducted a comparison of the features of data sets utilised in prior investigations. These findings reveal that many existing data sets depict repeated inefficient attacks, such as service denial (DoS), UDP inundation and brutal strength that are different from current trends in web offensive attack. Types of assaults and data patterns are always changing and therefore it is necessary to establish a generic objective

model which is not connected to specific current developments. Moreover, the majority of the data sets supplied are typically over-adapted as a result of doubled or flow-based information, which improves the model performance greatly under experimental conditions. True positive alarms are a big worry when the paradigm in real-world services is adopted. In addition, a study by Sabhnani and Serpen[28] has shown that pattern classification or machine learning algorithms are not successfully taught when utilising a DDD99 dataset.

On the other hand, most earlier studies assessed the performance of the model in trial utilising deep learning or machine learning approaches for KDD99 data sets. Yin et al. [29] employed KDDTest +- for the performance test using the RNN, in addition to the ML-, NB-, RF-, and DT techniques to learning, Vinayakumar et al. [12] have recently experimented with the DNN model, employing freely available information such as KDD 99, NSL-KDD and UNSW-NB 15. Gu et al. [30] revealed in their study that validated training data can considerably enhance the capability of detective research, which is vital for successful research. Moustafa et al. [31] also evaluated the features of many public data sets and concluded that data sets for non-reality could mislead researchers.

A. Convolutional Neural Network (CNN)

Dense connections between DNN layers were introduced to CNNs. This technique serves to train different layers of high dimensional data classification into the class of output layers using nonlinear CNN maps to train input data. The cornerstone of a CNN that is added to layers when necessary is concentration and pooling layers. Filters are used in the convolutional layer and have reduced input dimensions. The filters converge into mappings of functions for the entire input signal. A sub-set of this is sampled by the pooling layer on the function maps, which decreases the overall maps size [31].

The deep convolution layers directly gain multi-dimensional inputs, removing the requirement for usual systems to duplicate data. CNN consequently works well with multidimensional data types such as photographs and audio signals. In addition, CNN requires fewer variables that minimise complexity and enhance learning to reach the same network depth as other deep networks [32]. Based on their ability to process complicated data, CNNs have been lately explored as extractors and classifying intrusion detection functions.

1 Intrusion Detection System (IDS)

A series of internal or external malicious measures to harm the target system [33] are included in the intrusions. Intrusion detection comprises identifying suspected computer intruders by OS surveillance, network traffic, and operational analysis. A system is consequently utilised for intrusion detection (IDS) [34]. It contains a number of mechanisms and tools.

Most IDSs provide equivalent network security features. Most IDSs. Most IDSs. Most IDSs. Most IDSs. An IDS begins with the data collection of the events tested. It watches all event data closely and compares events from a variety of sources. For an IDS, the detector is crucial and uses various ways and comparable strategies based on the scenario. Capacity for prevention is also feasible. The technology was developed as a framework to identify and prevent intrusions [34] in this setting.

1 CNN-LSTM

The CNN layers are used to extract input data, while the LSTM layer aids with architectural prediction. CNN-LSTMs were designed for the digital time series in order to predict and apply the text in the image sequences. The approach proposed by CNNLSTM identifies the principal sequence as blocks, separates the features of every block and enables the LSTM to comprehend the functioning of each block [35].

3 Research Methodology

A. Proposed Plan: -

As a benchmark for meaning, the 2017 CICIDS data set was used. This section proposes an integrated strategy for increasing the rate of network attack detection and response (ADR) and decreasing false alarms (FAR). As illustrated in Fig. 1, the proposed work is separated into the following categories. The IDS is proposed and begins with an analysis of the CICIDS 2017 data. Pre-processing, train data, classification, and modelling are all components of the IDS model.

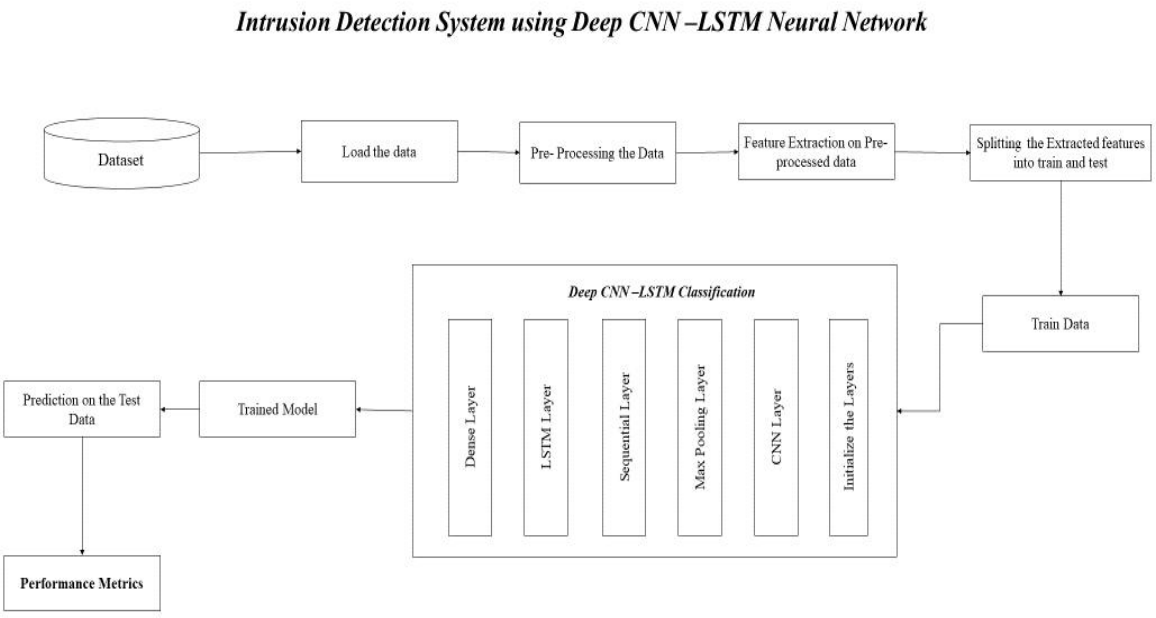


Figure 1. Flow Diagram of Deep CNN-LSTM Neural Network

1. **Dataset CICIDS Loading**: - The CICIDS-2017 dataset is loaded in google colaboratory environment.
2. **Data Pre-Processing**: - At this stage, all the missing values that exist in dataset will be removed.
3. **Feature Selection**: - Anova Feature was used to select the best features from datasets.
4. **Dataset Split in Train and Test:** - We split the CICIDS dataset into train and test at ratio of 20:80, we chose this ratio to reduce the possible losses and enhance the accuracy.
5. **Classification Steps:** - Here, we used a Deep CNN-LSTM model to classify the trained data set.
6. **Trained data**: - The trained data will be saved.

7. **Prediction of the test data**: - Here we will specify the trained model file to get the prediction result.

8. **Performance metric**: - At this stage, we will implement the performance metric to get the results such as confusion matrix, F1 Score, Re-call and accuracy.

B. **Research Methodology: -**

There are two types of research assessments: qualitative and quantitative. It is qualitative research. Data are gathered by observations and the researchers' judgement in qualitative analysis. The studies investigated by the deep CNN –LSTM Network researchers and the intrusion detection system are compared. Their function, accuracy, reminder, precision, FAR and FI values are compared to our data sets. We suggest design and approach following comparative investigations and studies.

Explorative, explanatory and descriptive research have 3 different types of study aim. In this paper we preferred to utilize the descriptive as well as explanatory method of research.. This research method was chosen to focus on the past studies conducted by other authors and on the outcomes of IDs with a different dataset to be investigated to carry out explanative and descriptive investigations in this study. In terms of efficiency and performance it is illogical to compare the different techniques to deeper learning. The reasons for this include (1. the data set utilized and (2) the data set used, (3) pre-processing, (4) deep network configuration and (5) hardware platforms. The following areas are as follows: To accomplish a fair comparison result, several comparative research investigations using a common calculation framework and general parameters affecting various profound study models are necessary.

C. **Basic Principles**

The term "deep learning" refers to multiple levels of concatenation. The first layer is the input layer, while the final layer is the output layer. Additionally, hidden layers are placed between the input and output layers. Each layer is composed of several units known as neurons. The input layer's size is determined by the size of the input data, whereas the output layer is formed of C units corresponding to the category C.

The convolutional neural network with several layers is depicted in Figure 2. (CNN). The three primary layer divisions are as follows:

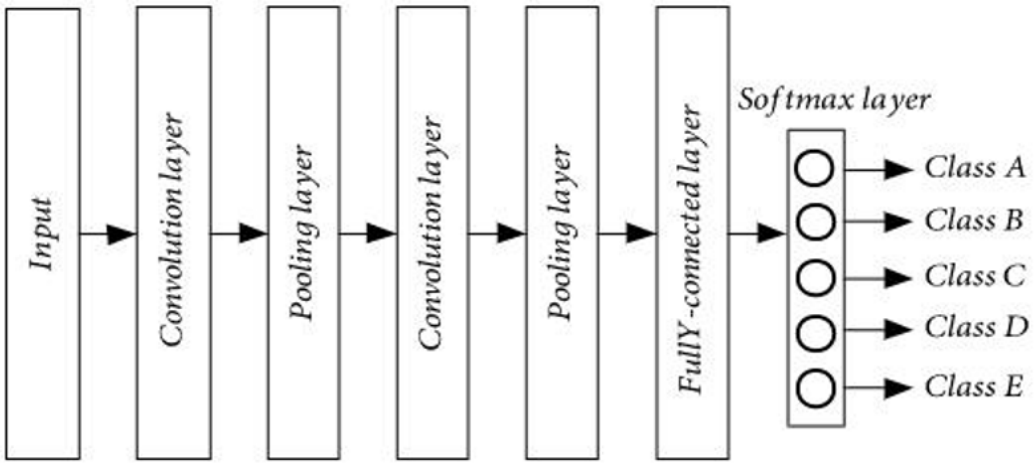


Figure 2. CNN Architecture

1. **Convolutional layer**: For input data, a collection of filters, sometimes referred to as convolutional kernels, is used. Each filter slides across the input data while designing a map function. By combining all feature vectors, the convolution layer's maximum performance is achieved.
2. **Pooling layer:** its subsamples on the image minimise the map function's dimensionality. Pooling is most frequently used in two ways: average pooling and maximal pooling.
3. **Max pooling layer**: The output of the prior layer is combined into a single input for the following layer.

D. Dataset description: -

CICIDS-2017: This data set was gathered in a small, simulated network with normal traffic. Six distinct sorts of current attacks are carried out by a separate network. There are options for raw grabs and NetFlow with 80 features. This set of data is part of a small number of the research that have been analysed. In our investigations, we use this data set to simulate current network traffic [36].

E. Pre-processing stage: -

The missing data and information will be eliminated at this step. The following sub-categories separate this processing:

Features Extraction: -

The implementation of a lightweight IDS for IoT applications is a primary focus of this research. Thus, the performance of detection models must be enhanced by reducing the volume, noise, memory, and processing complexity of the data set. All operations consume a total of 12 GB of RAM. Extraction of features reduces processing time and speeds up training and detection. In comparison to the reference set, the required steps reduced memory consumption by 70% to 512MB.

Split Dataset into Train & Test set: -

In the stage of dataset splitting into train and test, I split dataset by using 'train_test_split' function' from library sklearn.model_selection. here we divided our dataset into 80% for training and 20% for testing, and the way & reason for splitting the dataset in ratio of 20:80 is that it will probably minimize the losses and enhanced the accuracy.

Training and Optimization of CNN Framework

With two 1D causal convolution layers, two dense layers, and a softmax layer, softmax is applied to multiclass tasks. The proposed method of training and optimization is as follows: Throughout the universe, over construction is combated using maximal pooling, batch normalisation, and dropout. We utilise Adam Optimizer to adjust the weights and optimise the cross-entropy loss function. Two algorithms have the following advantages: Algorithm for adjusting gradients (AdaGrad) (RMSP).

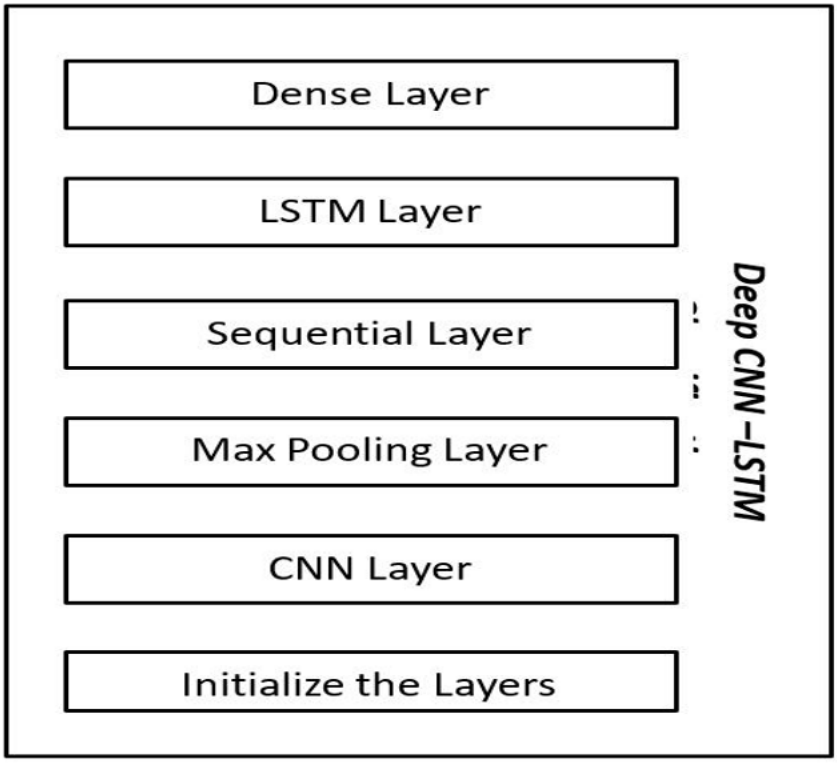


Figure 2: Deep CNN-LSTM

1. The 1-dimensional causal convolution layer is a first level with 64 filters with a three-filter size on input vectors.

2. The first layer to collect 64 filters of a 3-filter size across the entrance vectors is one-dimensional causal convolution layer.

3. It substitutes the data covered by the filter with the maximum value in a 1-dimensional global pooling layer. It prevents the learnt features from overlapping with consecutive layers by taking entire value.

4. The convolutional layer norms the input from the previous layer into the LSTM layer before going to the next layer.

5. 128 cached units and a dropout percentage of 30 percent are completely linked to the dense layer.

6. Max pooling thick layer with “softmax” activation feature: it creates five multi class units that match five categories of traffic.

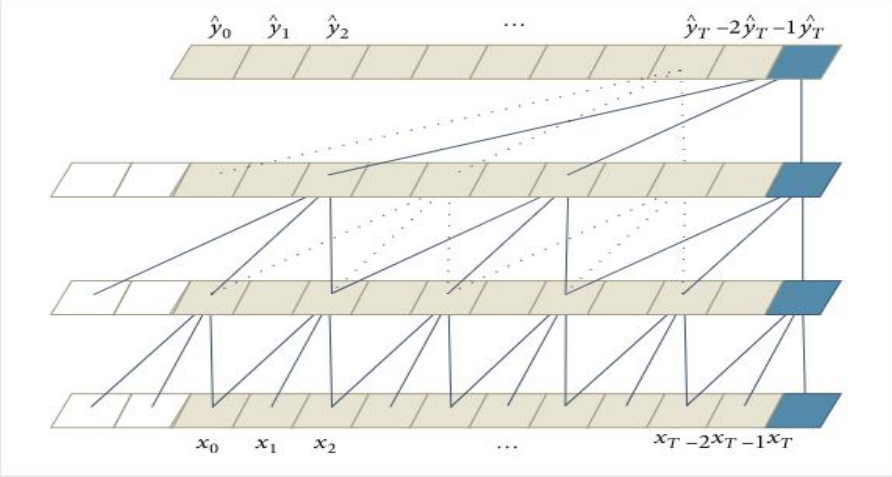


Figure3. ID Causal Convolution

F. Architecture of the proposed CNN system: -

Figure 3 shows the cumulative CNN project architecture and Section 3 details its design. The architecture suggested is the following steps:

1. Dataset: An unstable data collection could lead to misunderstanding, as described above. To solve this problem, produce synthetic samples and use categorical and continuous datasets from the minority classes**.**

2. Feature Extraction: With this strategy, we can clean the data collection by deleting unnecessary features and moving memory usage to a reduced datatype. We reduce the space for the feature.

3. Data division: the dataset is divided into: training, validation, and testing subsets in the fight against overfitting.

4. Trained and predictive model: throughout this procedure we apply the functional change in the training subset. Log transformation and regular scaler are implemented using continuous numerical properties. Label encoding generally covers categories that simply substitute for each categorical column with a specific integer. This procedure is then used for test sub-sets and validation.

5. Training and optimization: the CNN LSTM model is now being developed, as indicated in section 3. Adam optimizer and the validation subset are used in the training subset.

6. Classification: TCNN models generated in the test subset shall be utilized to apply a standard or some type of attack to each test record of its class label.

4 Design Specification

The design flow the intrusion detection system with dataset CICIDS-2017 is categorized into three steps as follow.

Data pre-processing: The dataset for deep CNN-LSTM is derived from the obtained CSV file, as is the pre-processing of the data. The jupyter notebook and numerical python framework are used to process the dataset's features. This operation will eliminate all null values and reductant records from the dataset. Following that, the arrays are converted into matrices, which serve as the primary input for CNN.

Modelling: This is the primary step of implementation; it is during this step that the deep learning algorithm CNN is built, and the various CNN are fused with LSTM layers. This is accomplished with the help of the tensorflow and keras frameworks. To maximise efficiency and minimise processing time, a graphics driver is employed in conjunction with the notebook's juypter framework. The total procedure is executed on both GPU and CPU, which fully utilises the system's resources and generates the evaluation metrics. Precision, accuracy, FI-score, and recall.

Visualization: The metrics generated in the second stage are converted into graphic representation in the form of graphs and confusion matrices to facilitate reading understanding.

Software And Hardware Pre-requisite –

Dataset	CICIDS 2017
Computer	High-performance computer (HPC) technology
RAM	12 GB DDR4
Software	Python Ver 3.8.1, Excel, Google Colabs.
Function	Relu & Softmax activation function is used
Training Set	Keras Tensorflow, scikit learn

5 Implementation

A. Implementation

This section explains the datasets that were used, the implementation of the proposed system, the experimental environment, and the additional results that were obtained, all of which are provided and analyzed in further depth in this section.

B. Performance Matrices

This section discusses the most frequently used methods for evaluating ML and DL performance in IDS. All measurement measures & it includes information on the classes in concern.

1. **True Positive (TP):** The analysis comprised data sets precisely forecast as the classifier's Attack.
2. **False Negative (FN):** Data occurrences wrongly anticipated as normal cases
3. **False Positive (FP):** Instances of information incorrectly labelled an Attack.
4. **True Negative (TN):** Events classified as Normal instances appropriately.

The diagonal confusion matrix represents the accurate forecast, whereas nondiagonal parts are the wrong forecast for a given classification device. This confusion matrix characteristic is shown in Table 1. Furthermore, the many evaluation tools used in recent studies are:

Precision: It is the exact number of attacks for all samples predicted as attacks.

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

1. **Recall:** It is the percentage of samples appropriately classified as Attacks to all samples classified as Attacks. Additionally, the term "Detection Rate" is used.

$$\text{Recall} = \text{Detection Rate} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \quad (2)$$

2. **False alarm rate:** Also known as the false positive rate, it is defined as the ratio of wrongly predicted Attack samples to all Normal samples.

$$\text{False Alarm Rate} = \frac{\text{FP}}{\text{FP} + \text{TN}}. \quad (3)$$

3. **True negative rate:** It is defined as the ratio of correctly diagnosed Normal samples to all samples classified as Normal samples.

$$\text{True Negative Rate} = \frac{\text{TN}}{\text{TN} + \text{FP}}. \quad (4)$$

4. **Accuracy:** It is defined as the ratio of examples successfully classified into the total number of occurrences. When a dataset is balanced, it is referred to as Detection Accuracy, and it can be used to assess the system's performance.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}. \quad (5)$$

5. **F-Measure:** It can be defined as the harmonic mean of combined variables of precision and recall. In other words, this is a statistical technique for analysis of the accuracy of a system, taking both the accuracy and recall of the system under study into account.

$$\text{F Measure} = 2\left(\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}\right). \quad (6)$$

TABLE 1. Confusion matrix

		Predicted class	
		Attack	**Normal**
Actual Class	**Attack**	True Positive	False Negative
	Normal	False Positive	True Negative

To validate the suggested methodology, these measurements are made using benchmark datasets. The next section explains the popular public NIDS testing dataset.

6 Evaluation

In consequence, 15 attributes have been used for the following phase in our proposed technique. This figure also compares results with those achieved using other machine-learning algorithms such as support vector machine, ransom forest and Bayes, frequently used in cyber-attack detection. Figure 4 illustrates the accurateness of a proposed process and its comparison to the MLP, another widely used deep - learning model.

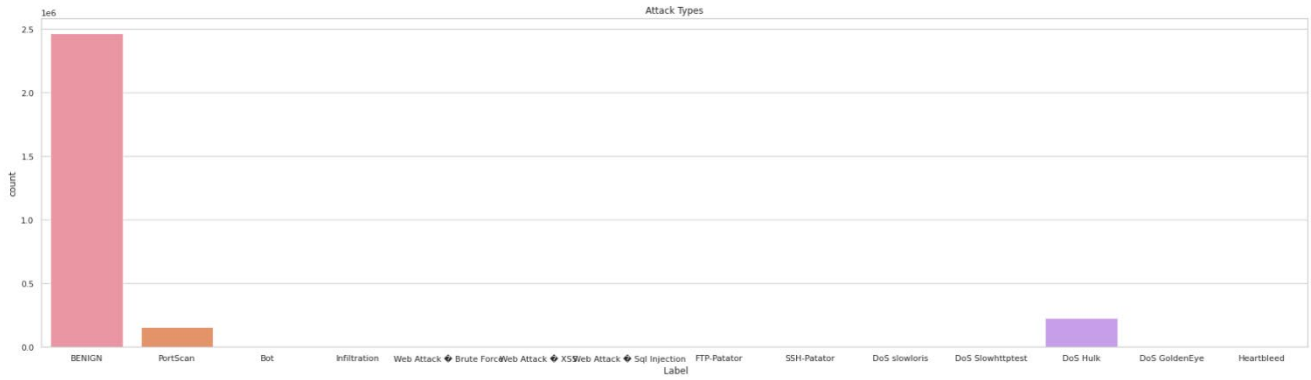


Fig.4. Types of Attackers

6.1 Experiment Study 1: -

The findings of the CICIDS2017 data set, which has classes ranging from 0.1 to 0.5, are depicted in the illustration 5. Approximately 0.97 percent of the time is accurate, 0.97 percent of the time is DR, and 2494 of the time is a false positive.

6.2 Experiment Study 2: -

The findings of the CICIDS2017 data set, which has classes ranging from 0.1 to 0.5, are depicted in the illustration 5. Approximately 0.97 percent of the time is accurate, 0.97 percent of the time is DR, and 2506 of the time is a false positive.

6.3 Experiment Study 3: -

The findings of the CICIDS2017 data set, which has classes ranging from 0.1 to 0.5, are depicted in the illustration 5. Approximately 0.97 percent of the time is accurate, 0.97 percent of the time is DR, and 5000 of the time is a false positive.

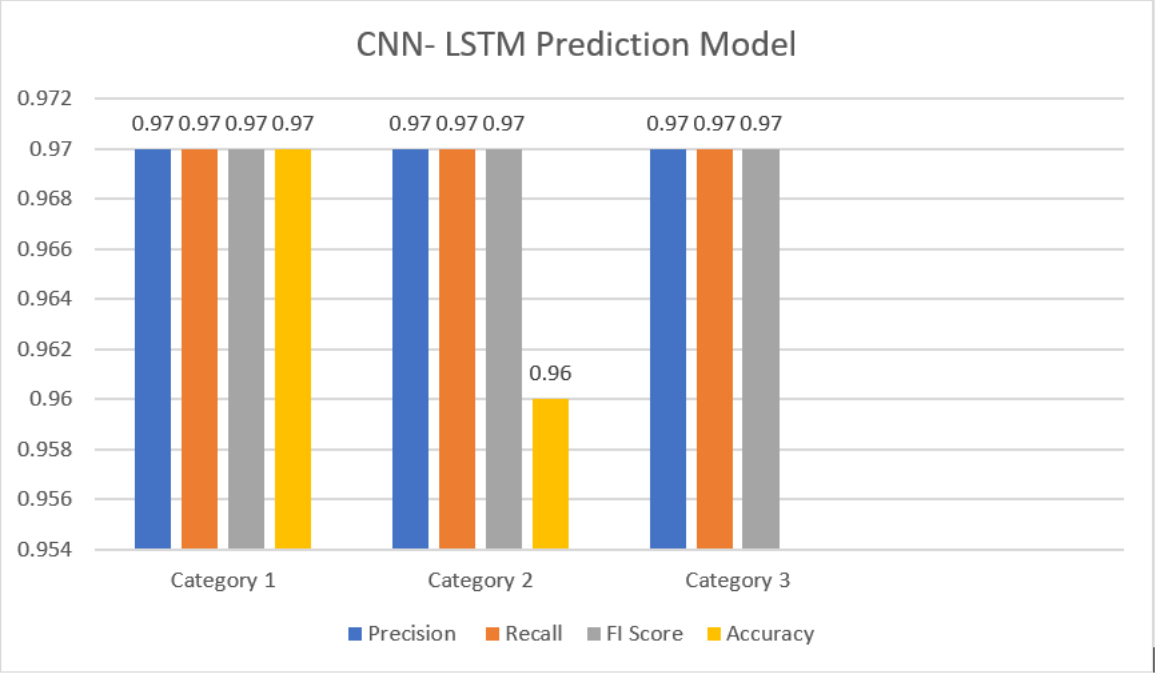


Figure 5: Comparison Precisions, Recalls, FI Scores, and Accuracy

The accuracy of our proposed DDoS attack detection approach shows that the accuracy of competitive solutions is above that of our findings. Fig. 5 classifies the proposed approach, MLP, SVM, Bayes, and Random Forest Methodology as well as other ways.

In view of similar precision, the Fl-Score is an important assessment metric to analyze the overall performance of the approach employed. Fig. 5 shows the Fl-Score values that have been obtained during the entire experiment. It should be noted that, despite the fact that the exact values of SVM have been almost identical, the Fl-Score is greater than that of other machine learning methods.

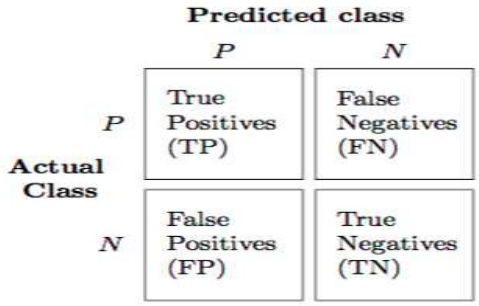
Figure 5: a)Model Accuracy Prediction. b) Model loss Prediction

6.5 Discussion

The suggested method is compared to a thorough study approach comprising of CNN and LSTM, which uses CNN and LSTM, for raw data without any functional selection. Training time has declined dramatically to 2.5 seconds, over double the time needed to study more deeply. The training time has been reduced by a factor of five, which proves our proposed technique in the real-time loT cyber threat detection is productive and successful. Table I compares the method presented to existing advanced DDoS attack detection algorithms. The authors of [37] have devised a method for the identification of unforeseen and unforeseen cyber-attacks, using deep neural network approaches. The authors have tested their proposed data set technique: CICIDS 2017, The experiment was conducted for 50 epochs and each dataset had a number of learning rates.

According to the results, this approach has the best accuracy of 0.97 percent on the CICIDS data set to detect a denial of service (DoS) assault. [38] offers a solution for Software-Defined Networking to identify and prevent DDoS attacks. This approach took advantage of the combination of entropy with a background neural network. A performance assessment was performed using a Java-based Floodlight and Mininet simulation software and determined to be exact at 0.02 per cent of the goal. The authors of [39] present another way to identify DDoS attacks based on a multi-level autoencoder and kernel learning method (MKL). The authors have indicated that they have compared their proposed methodologies to machine learning. \ . A different, effective approach based on a multilayer perceptron referred known as the CS DDoS [40] is used to identify DDoS attacks in the cloud environment. The arrival packs are scanned and categorized as normal or attack data packs as part of the first stage in the proposed method. The cloud is not accessible by malicious transmissions. The authors compared their method with the methodology of machine

learning and discovered that their method was 97.00 percent accurate. We may conclude that our proposed approach is effective enough to exceed DDoS Attack Detection's proposed work.



Predicted Classes	Actual Classes	
	Positive	Negative
Normal	2416	78
Anomaly	75	2431

F

ig.8. Confusion Matric

A. Outcomes Performance: -

A different LSTM model for the evaluation was used to process the individual data sets, and the findings showed that the LSTM model Bi-Directional is more effective and effective. For the data set, the results for the Bi-directional LSTM model were 97%. This clearly reveals that Bi is an efficient LSTM directional model than other intrusion detection approaches. The confusion matrix obtained during the bi-directionality fusion testing phase is depicted in Figure 6, where True Positive (TP) equals 2416 and False Negative (FN) equals78, False Positive (FP) equals 75 and True Negative (TN) equals 2431.

7 Conclusion and Future Work

For the Internet of Things, an intrusion detection system with a novel method to DDoS attack detection has been developed. The first stage of our feature-selection method is multi-target optimization, which is based on six fundamental data reduction objectives. For attack classification, a deep learning model known as the Convolutional Neural Network was utilized in conjunction with LSTM. A huge number of experiments on high-performance GPU-equipped machines were carried out using the current CICIDS2017 dataset. The dataset is pre-processed and standardized before using the proposed approach to be compatible. We were able to cut the amount of training by five since we carried out a function selection before the data assault classification. Our proposed method was used to attain a fantastic

accuracy of 99.03% and a Fl-score of 99.36%. We noticed that our method has surpassed earlier studies, which support the effectiveness of the strategy that we proposed, in the most recent approaches to our analysis. As part of the fog-to-node internet architecture of things, our work for the future on distributed intrusion detection systems (IDS). In addition to identifying cyber threats on IoT networks, the recommended technique can also be utilized to detect various types of threats.

References

1. A. Sami Ali and M. Abdulmunem. (2020). Image classification with Deep Convolutional Neural Network Using Tensorflow and Transfer of Learning. *Journal of College of Education for Women*, Jun., doi: 10.36231/coedw/vol31no2.9.
2. M.U. Chowdhury, F. Hammond, G. Konowicz, C. Xin, H. Wu, J. Li. (2017). A Few-shot deep learning approach for improved intrusion detection, IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conf (UEMCON), 456–462. doi: 10.1109/UEMCON.2017.8249084.
3. W. Lin, H. Lin, P. Wang, B. Wu, J. Tsai. (2018). Using convolutional neural networks to network intrusion detection for cyber threats, IEEE Int. Conf. on Applied Syst. Invention (ICASI) ,1107–1110. doi: 10.1109/ICASI.2018.8394474.
4. Al-Garadi, M.A., Mohamed, A., Al-Ali, A., Du, X., Guizani, M., 2018. A Survey of Machine and Deep Learning Methods for Internet of Things (IoT) Security. arXiv:1807.11023 [cs] http://arxiv.org/abs/1807.11023.
5. A. Yulianto, P. Sukarno, and N. A. Suwastik, (2019). Improving AdaBoost-based Intrusion Detection System (IDS) Performance on CIC IDS 2017 Dataset. *Journal of Physics: Conference Series*, vol. 1192, p. 01, doi: 10.1088/1742-6596/1192/1/012018.
6. Z. Jiang, D. Crookes, B. D. Green, Y. Zhao, H. Ma, L. Li, S. Zhang, D. Tao, and H. Zhou. (2019). Context-Aware Mouse Behavior Recognition Using Hidden Markov Models. IEEE Transactions on Image Processing, vol. 28, no. 3, pp. 1133–1148.
7. R. Abdulhammed, H. Musafer, A. Alessa, M. Faezipour, and A. Abuzneid. (2019). Features dimensionality reduction approaches for machine learning based network intrusion detection. Electronics (Switzerland), vol. 8, no. 3.
8. I. Portugal, P. Alencar, and D. Cowan. (2018). The use of machine learning algorithms in recommender systems: A systematic review. Expert Systems with Applications, vol. 97, pp. 205–227.
9. S. Parampottupadam and A. N. Moldovann. (2018). Cloud-based Real-time Network Intrusion Detection Using Deep Learning. International Conference on Cyber Security and Protection of Digital Services, Cyber Security 2018, pp. 1–8,
10. R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. AlNemrat, and S. Venkatraman. (2019). “Deep Learning Approach for Intelligent Intrusion Detection System,” IEEE Access, vol. 7, pp. 41 525–41 550,
11. X. Zhang, J. Chen, Y. Zhou, L. Han, and J. Lin. (2019). A Multiple-Layer Representation Learning Model for Network-Based Attack Detection,” IEEE Access, vol. 7, pp. 91 992–92 008.

12. M. Belouch, S. El Hadaj, and M. Idlianmiad. (2018). Performance evaluation of intrusion detection based on machine learning using apache spark. Procedia Computer Science, vol. 127, pp. 1–6.
13. O. Faker and E. Dogdu. (2019). Intrusion Detection Using Big Data and Deep Learning Techniques. in Proceedings of the 2019 ACM Southeast Conference. New York, USA: ACM Press, pp. 86–93.
14. Y. Liu, S. Liu, and X. Zhao. (2017). Intrusion detection algorithm based on convolutional neural network. Beijing Ligong Daxue Xuebao/Trans. Beijing Inst. Technol., vol. 37, no. 12, pp. 1271–1275.
15. W. Wang, Y. Sheng, J. Wang, X. Zeng, X. Ye, Y. Huang, and M. Zhu. (2018). HAST-IDS: Learning hierarchical spatial-temporal features using deep neural networks to improve intrusion detection. IEEE Access, vol. 6, pp. 1792–1806.
16. C. Yin, Y. Zhu, J. Fei, and X. He. (2017). A deep learning approach for intrusion detection using recurrent neural networks. IEEE Access, vol. 5, pp. 21954–21961.
17. N. Shone, T. Nguyen Ngoc, V. Dinh Phai, and Q. Shi. (2018). A deep learning approach to network intrusion detection,'' IEEE Trans. Emerg. Topics Comput. Intell., vol. 2, no. 1, pp. 41–50.
18. K. Wu, Z. Chen, and W. Li. (2018). A novel intrusion detection model for a massive network using convolutional neural networks. IEEE Access, vol. 6, pp. 50850–50859.
19. S. Naseer, Y. Saleem, S. Khalid, M. K. Bashir, J. Han, M. M. Iqbal, and K. Han. (2018). Enhanced network anomaly detection based on deep neural networks. IEEE Access, vol. 6, pp. 48231–48246.
20. Y. Ding and Y. Zhai. (2018) Intrusion detection system for NSL-KDD dataset using convolutional neural networks,'' in Proc. 2nd Int. Conf. Comput. Sci. Artif. Intell. (CSAI), pp. 81–85.
21. N. Chouhan, A. Khan, and H.-U.-R. Khan. (2019). Network anomaly detection using channel boosted and residual learning based deep convolutional neural network. Appl. Soft Comput., vol. 83, Art. no. 105612.
22. R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman. (2019). Deep learning approach for intelligent intrusion detection system,'' IEEE Access, vol. 7, pp. 41525–41550.
23. Z. Chiba, N. Abghour, K. Moussaid, A. El Omri, and M. Rida. (2019). Intelligent approach to build a deep neural network-based IDS for cloud environment using combination of machine learning algorithms,'' Comput. Secur., vol. 86, pp. 291–317.
24. H. Zhang, B. Zhao, H. Yuan, J. Zhao, X. Yan, and F. Li. (2019). SQL injection detection based on deep belief network.' in Proc. 3rd Int. Conf. Comput. Sci. Appl. Eng. (CSAE), p. 20.
25. O. Faker and E. Dogdu. (2019). Intrusion detection using big data and deep learning techniques. in Proc. ACM Southeast Conf. ZZZ - ACM SE, pp. 86–93.
26. M. Aloqaily, S. Otoum, I. A. Ridhawi, and Y. Jararweh. (2019). An intrusion detection system for connected vehicles in smart cities. Ad Hoc Netw., vol. 90, Art. no. 101842.
27. M. Ring, S. Wunderlich, D. Scheuring, D. Landes, and A. Hotho. (2019). A survey of network-based intrusion detection data sets. Comput. Secur., vol. 86, pp. 147–167.
28. M. Sabhnani and G. Serpen. (2004). Why machine learning algorithms fail in misuse detection on KDD intrusion detection data set,'' Intell. Data Anal., vol. 8, no. 4, pp. 403–415.
29. Yin, Y. Zhu, J. Fei, and X. He. (2017). A deep learning approach for intrusion detection using recurrent neural networks. IEEE Access, vol. 5, pp. 21954–21961.
30. J. Gu, L. Wang, H. Wang, and S. Wang. (2019). A novel approach to intrusion detection using SVM ensemble with feature augmentation. Comput. Secur., vol. 86, pp. 53–62.
31. N. Moustafa, J. Hu, and J. Slay. (2019). A holistic review of network anomaly detection systems. A comprehensive survey. J. Netw. Comput. Appl., vol. 128, pp. 33–55.
32. Adam Gibson, Josh Patterson. (2017). Deep Learning practitioner's approach, O'REILLY.
33. A. Krizhevsky, I. Sutskever, G. Hinton. (2012). ImageNet classification with deep convolutional neural networks, Adv. Neural Inf. Process Syst, 25 1097–1105

34. K. Scarfone, P. Mell, P. Mell. (2007). Guide to intrusion detection and prevention systems (IDPS), NIST special publication,
35. Babak Moradi, Mohammad Aghapour, Afshin Shirbandi. (2019). Compare of Machine Learning and Deep Learning Approaches for Human Activity Recognition.
36. Sharafaldin, I., Habibi Lashkari, A., Ghorbani, A.A. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. In: Proceedings of the 4th International Conference on Information Systems Security and Privacy. SCITEPRESS - Science and Technology Publications, Funchal, Madeira, Portugal, pp. 108–116.
37. R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al- Nemrat, and S. Venkatraman. (2019). Deep Learning Approach for Intelligent Intrusion Detection System. IEEE Access, vol. 7, pp. 41525-41550.
38. Z. Liu, Y. He, W. Wang, and B. Zhang. (2019). DDoS attack detection scheme based on entrop y and PSO-BP neural network in SDN'. China Commun., vol. 16, no. 7, pp. 144- 155
39. S. Ali and Y. Li. (2019). Learning Multilevel Auto-Encoders for DDoS Attack Detection in Smart Grid Network', iEEE Access, vol. 7, pp. 108647-108659.
40. A. Sahi, D. Lai, Y. Li, and M. Diykh. (2017). An efficient DDoS TCP flood attack detection and prevention system in a cloud environment. IEEE Access, vol. 5, pp. 6036-6048.