International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10s
DOI: https://doi.org/10.17762/ijritcc.v11i10s.7588
***Article Received**: 28 May 2023 **Revised**: 22 July 2023 **Accepted**: 10 August 2023*

Hybrid CNN+LSTM Deep Learning Model for Intrusions Detection Over IoT Environment

Mr. Thamraj Narendra Ghorsad[1], Dr. Amol V. Zade[2]
[1]Phd Scholars, Computer Science & Engineering,
G H Raisoni University Amravati,
Amravati,India
raj.ghorsad@gmail.com
[2]Professor, Computer Science & Engineering,
G H Raisoni University Amravati,
Amravati,India
amol.zade@ghru.edu.in

Abstract— The connectivity of devices through the internet plays a remarkable role in our daily lives. Many network-based applications are utilized in different domains, e.g., health care, smart environments, and businesses. These applications offer a wide range of services and provide services to large groups. Therefore, the safety of network-based applications has always been an area of research interest for academia and industry alike. The evolution of deep learning has enabled us to explore new areas of research. Hackers make use of the vulnerabilities in networks and attempt to gain access to confidential systems and information. This information and access to systems can be very harmful and portray losses beyond comprehension. Therefore, detection of these network intrusions is of the utmost importance. Deep learning-based techniques require minimal inputs while exploring every possible feature set in the network. Thus, in this paper, we present a hybrid CNN+LSTM deep learning model for the detection of network intrusions. In this research, we detect DDOS types of network intrusions, i.e., R2L, R2R, Prob, and which belong to the active attack category, and PortScan, which falls in the passive attack category. For this purpose, we used the benchmark CICIDS2017 dataset for conducting the experiments and achieved an accuracy of 99.82% as demonstrated in the experimental results.

Keywords- Intrusion Detection System, Deep Learning, CNN, Bi-LSTM, CICIDS2017.

I. INTRODUCTION

The Internet of Things (IoT) models are becoming increasingly complex as a result of the growing requirement for the expansion of the IoT automation communication network [1]-[2]. Users are becoming utilized to services that are data-driven, which is driving investment into machine learning-based IoT systems. Today, technologies based on the Internet of Things and artificial intelligence are employed in each aspect of human existence. In healthcare, complicated operations include biomedical signal modelling, symptom identification utilizing X-rays, pattern recognition in genetic information, an automatic pathology system for tumours identification, and ECG interpretation use of artificial intelligence techniques [3]-[4]. The aircraft industry can potentially benefit from the adoption of machine learning techniques. Electrical impedance levels produced by eddy's current examination were subjected to content-based image extraction approaches and artificial intelligence techniques [5]-[6]. Eddy's current examination is a challenging procedure used in the aerospace industry to identify flaws. IoT solutions are used in several fields in addition to artificial learning. Unwanted threats are being introduced into IoT networks as a result of their increasing complexities. Data breaches and anomalies in IoT-enabled devices are increasingly occurring. IoT-enabled devices are an easier threat for assault since they

disseminate information through wireless media [7]-[8]. Normal communication attacks on local networks are only effective against nearby nodes or a smaller localized area, whereas attacks on IoT systems spread across a much wider region and have catastrophic repercussions for IoT systems [9].

From this point on,defines against cyberattacks requires a protected Internet of Things architecture. The adoption of security mechanisms makes them subject to IoT device vulnerabilities. Information is money for various investors and start-up businesses. Some information is sensitive and categorized for use by governmental organizations and certain commercial organizations. An intruder can use a backdoor created by IoT device vulnerabilities to obtain sensitive information from any substantial business [10]-[11]. The content-based approach has the benefit of working more quickly than some other approaches and being able to solve

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10s
DOI: https://doi.org/10.17762/ijritcc.v11i10s.7588
***Article Received**: 28 May 2023 **Revised**: 22 July 2023 **Accepted**: 10 August 2023*

the issue presented with unidentified vulnerabilities [12]-[13]. The main objective of this study is to build an intelligent, safe, and dependable IoT-based network that can recognize its vulnerabilities, provide a strong barrier against any attacks and autonomously recover. Therefore, a machine learning-based approach is suggested that can recognize whenever the network is in an unusual situation and safeguard it. Various machine learning classifiers have been used for such tasks.

The main contribution of the Hybrid CNN+LSTM Deep Learning Model for Intrusions Detection over an IoT Environment is the integration of two powerful deep learning architectures, namely Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM), to enhance the accuracy and efficiency of intrusion detection in IoT environments. The hybrid model takes advantage of CNN's ability to extract meaningful features from raw data and LSTM's capacity to handle sequential information. The model extracts features from raw data using multiple 1D convolutional layers, followed by pooling layers to reduce the dimensionality. The resulting feature maps are then fed into an LSTM layer to capture the temporal dependencies and identify patterns in the data. The proposed model also uses a dropout layer to prevent overfitting and improve the generalization ability of the model. The proposed hybrid CNN+LSTM model has shown promising results in accurately detecting various types of intrusions in IoT environments, making it a valuable contribution to the field of intrusion detection.

The rest of the paper is organized as follows. Section 2 describes related work. The description of the architectural model is presented in Section 3. The results are analyzed, compared, and discussed in Section 4. Section 5 concludes the paper and Section 6 presents the future direction of this work.

II. RELATED WORK

IoT-enabled devices have expanded quickly, and communication among them may provide significant concerns, including network load through IoT environments [8]-[9]. Among the vulnerabilities that may be utilized against the IoT include malicious activities, like DoS attacks and DDoS attacks, etc.

Machine learning and deep learning have been used in several research to enhance IoT protections and security by increasing the effectiveness and reliability of identifying security issues in IoT and preventing it prior they cause any damage [10]. The research that employed IDS based on ML and DL in the IoT is reviewed in this section.

In Liu, X et. al. [11] proposed a Bi-LSTM-based deep learning approach for IoT intrusion detection. The model achieved high accuracy in detecting various types of attacks, including denial of service, user-to-root, and remote-to-local attacks.

In Lee, S. et. al. [12] in proposed a hybrid model combining Bi-LSTM and convolutional neural networks (CNN) for intrusion detection in IoT networks. The model showed improved performance in detecting different types of attacks, including port scanning, brute force, and DDoS attacks.

In Lee, S. [13], a study proposed a Bi-LSTM-based deep learning model for detecting insider threats in IoT environments. The model was trained on network traffic data and showed promising results in detecting abnormal user behavior.

In Lee, W.-H et. al. [14] proposed a Bi-LSTM-based intrusion detection system for wireless sensor networks in IoT environments. The model was able to detect various types of attacks, including sinkhole, selective forwarding, and black hole attacks, with high accuracy.

In [15] developed an intrusion-detecting technique utilizing a deep learning approach for IoT and discovered that as the number of IoT devices grows, so do the potential threat and susceptibility. This investigation made use of the CNN approach over the Bot-IoT datasets, the investigators conducted a comparison of CNN and various machine learning methods, including RF and MLP. The suggested CNN had achieved an accuracy score of 91.27%. The performance of the suggested approach was compared to RF, which achieved outperformed 100% in DDoS and DOS assaults. The suggested strategy of intrusion detection over the IoT network is consequently necessary since it might improve performance and decrease false alerts.

III. PROPOSED METHODOLOGY

The proposed Hybrid CNN+LSTM Deep Learning Model for Intrusion Detection over an IoT Environment is designed to improve the detection of various types of cyber-attacks that may occur in IoT devices. The model combines the strength of both Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) networks. The CNN is utilized to extract the spatial features of network traffic data, while the LSTM is used to capture the temporal dependencies. In this proposed methodology, the preprocessed IoT network traffic data will be fed into the CNN network, which will be followed by the LSTM network. The CNN network will identify the important features from the input traffic data using convolution and pooling operations, while the LSTM network will learn the temporal patterns of the network traffic. The extracted features will be then merged and classified using a fully connected layer. The proposed model will be trained on a large dataset of labeled network traffic data and evaluated using various performance metrics such as accuracy, precision, recall, and F1-score. The ultimate goal is to create an accurate and efficient intrusion detection system for IoT environments.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10s
DOI: https://doi.org/10.17762/ijritcc.v11i10s.7588
***Article Received**: 28 May 2023 **Revised**: 22 July 2023 **Accepted**: 10 August 2023*

IV. DATASET DESCRIPTION

Data selection is required in order to train proposed models and assess their dependability throughout the testing stage. This study used a CICIDS2017 dataset which is publically available on Kaggle that closely reflects actual real-world data, and comprises the most recent and relatively prevalent assaults. It also contains the outcomes of a network traffic analyzer performed via CICFlowMeter, including flows categorized according to the time stamp, source, and destination IP addresses, source and destination ports, protocols, and attack (CSV files). The specification of retrieved characteristics is further usable.

Developing this dataset with authentic background traffic generation as the primary focus. This study profiled the abstract behavior of human interactions using our proposed system, which also creates genuine neutral baseline traffic. On the basis of various protocols like HTTP, HTTPS, FTP, SSH, and email methods.

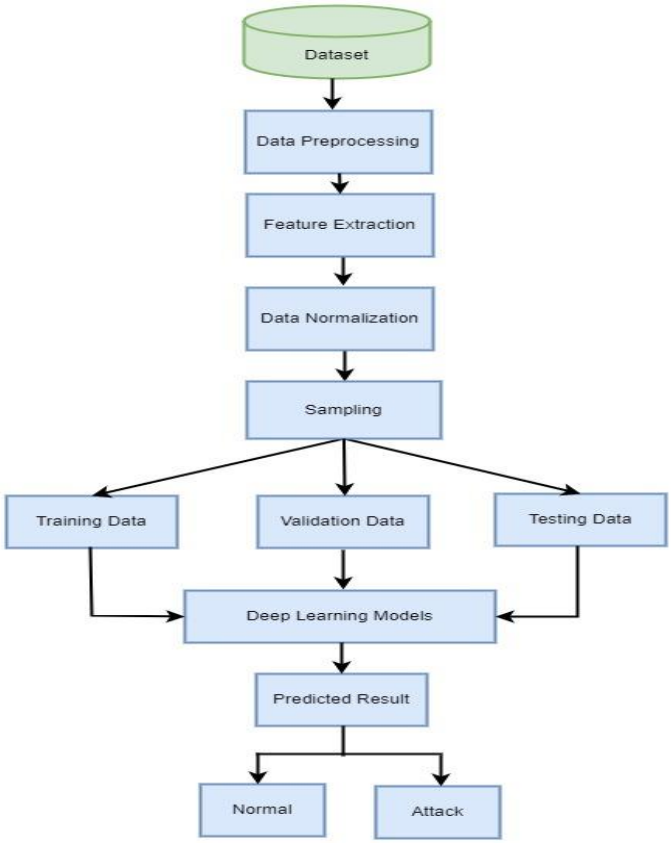

Figure 1: Complete framework for attack detection in IoT Environment

Figure 1 shows the complete framework of the proposed model for attack detection in an IoT environment. Let us understand the working principles behind the research work. This study uses two variants of LSTM in our research, Bidirectional LSTM, and hybrid CNN+LSTM.

A. Data Pre-processing

A dataset that is useful for the detection of intrusion is generated at the pre-processing stage. Pre-processing mostly concentrates on routine tasks like transforming the raw data into a usable form as well as into the other one. Using data from the train data to propose a model or test the data which is input data also eliminates data redundancy in the dataset. The feature extraction process uses this data as input to choose relevant features. To avoid misunderstandings in testing and when learning, certain portions of this information with blank fields will be removed.

B. Feature Extraction

With the careful selection of the most crucial characteristics that rendered the raw data appropriate for processing, this study, was able to minimize the dimensionality of the dataset. Large datasets frequently include a great deal of data redundancy and linked data which can be removed without losing crucial data. In the context of CICIDS 2017, chose the top 13 characteristics as shown below.

TABLE I. TOP 13 CHARACTERISTICS IN CICIDS 2017

Characteristics	Description
Source ip address	IP address of the attacker's System
Source port	Port number of the attacker's System
Destination ip address	IP address of the victim's System
Destination port	Port number of the victim's System
Duration	Log the total duration of the transaction
Source bytes	Number of bytes sent from the source to the destination
Destination bytes	Number of bytes sent from the destination to the source
Source TTL	Source to destination time to live value
Destination TTL	Destination to source time to live value
Source load	The transmission rate in bits per second
Destination load	Reception rate in bits per second
Source packets	Number of packets sent from the source to the destination
Destination packets	Number of packets sent from the destination to the source

The label converted into two Types:
Binary Classification- $(0;\ 1), 0 = Normal, 1 = Attack.$

C. Normalization

The purpose of normalization is to change the data such that their distribution is comparable. Because it gives each parameter the same weight, the proposed model may regard them all as being equally important. Considering each relevant feature subspace contains M rows and N columns. The Z-Scale normalization can be implemented as the given equation.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10s
DOI: https://doi.org/10.17762/ijritcc.v11i10s.7588
***Article Received**: 28 May 2023 **Revised**: 22 July 2023 **Accepted**: 10 August 2023*

$$\mu_m = \frac{\sum_{i=0}^{M-1} x_{im}}{M}$$

$$\sigma_m = \frac{\sum_{i=0}^{M-1}(x_{im} - \mu_m)^2}{M}$$

z-scale normalized feature vector can be obtained as

$$z_m = \frac{X_m - \mu_m}{\sigma_m}$$

Where, $\mu_m$is the mean of the entries of the M is the column of feature vector and σ_m represent standard deviation of the entries of the M column.

V. TRAINING AND VALIDATION PROCESS

The data are changed throughout the data processing step so that the model can interpret them more effectively and produce predictions that are more effective. The form of the data that will be analysed by the proposed hybrid CNN+LSTM layers is then altered. The size of the data is altered and an appropriate Timesteps parameter is chosen (samples, timesteps, features). The number of earlier samples that the proposed LSTM model considers is referred to as a timestep. The proposed model's training and validation phases include feeding the Bi-LSTM layers and hybrid CNN+LSTM with time-series data.

VI. BI-LONG SHORT-TERM MEMORY (BI-LSTM):

Bi-LSTMs represent a unique kind of RNN. They are especially helpful for learning about long-term dependency. Whenever the dataset is vast and the records have a lot of redundancies, RNNs struggle to learn additional information. As a result, LSTM was created to manage data for larger input channels and operate for long epochs. RNNs struggle to recall information for extended periods of time because their layers have such a basic structure, such as only one Tanh layer [19]. In contrast, a typical LSTM model has four interconnected layers that function in a unique manner to help people retain significant data for a significant period.

The uppermost horizontal line within a unit that signals the unit state is horizontal. Including or removing data from the unit is a capability of the LSTM that is carefully controlled via gates. A series-like architecture may be found inside the unit. The initial stage in the proposed LSTM model is to determine which data is not essential and to decide which data will be discarded from the unit state in the model. The "memory gate layer," a sigmoid layer, is responsible for this choice.

$$f_t = \sigma(W_f \,.\left[h_{t-1,}x_t\right] + b_f)$$

Where, $h_{t-1,}$represents the output of the previous time stamp,

x_t represent input

b_f represents the bias

The following step is to decide which new data we will keep in the unit state of the model. This is divided into two sections: The "input gate layer" determines that values will be updated in step one. In step two, the "tanh layer" builds a vector matrix of new candidate data that might be added to the state of the model. The two will be combined in the following phase to provide an update on the state of the model.

$$i_t = \sigma\,(W_i.\left[h_{t-1,}x_t\right] + b_i)$$

$$C_t = \tanh\,(W_c\,.\left[h_{t-1,}x_t\right] + b_c\,)$$

The previous unit state $C(t-1)$ has to be updated to the new unit state at this moment. Something to accomplish was previously chosen in the earlier phases; now all that is left is to carry it through. Secondly, the earlier decisions to forget information are multiplied by the former condition. Next, add to the state of the cell. This represents the updated candidate values, adjusted in accordance with the decision to modify the state value for each.

$$C_t = f_t \times C_t + i_t \times \tilde{C}_t$$

The output stage is the last step. Our unit state can enable this output, but it will be a filtered version. The first step is to choose which portions of the cell state will be output in a sigmoid layer. Next, divide the unit state by the sigmoid gate's output to ensure that just the sections chosen to output after pushing the values to be between -1 and 1.

$$o_t = \sigma(W_0.\left[h_{t-1,}x_t\right] + b_0)$$

$$h_t = o_t \times \tanh\,(C_t)$$

a. 1D-CNN

1D-CNN consists of 1-dimensional convolution layers, pooling layers, dropout layers, and activation functions for handling the 1-dimensional data. 1D-CNN is configured using the following hyper-parameters: number of CNN layers, neurons in each layer, size of the filter, and subsampling factor of each layer.

The convolution layer is the basic application of the filter to an input. Utilizing the filtering operation repeated times creates a feature map, which indicates the particular attributes related to the data points. Convolution is a linear operation that involved containing the multiplication with inputs with a set of weights. For this case, inputs are multiplied with the single-dimensional array weights, known as the kernel. This operation gives a unique value for each pass and executing this operation results in multiple values, known as a feature map.

Once the feature map is computed, each value is passed to the ReLU activation function. ReLU is a linear activation function that transforms the input as zero if it is negative, otherwise, it outputs the same input. The ReLU activation function allows the model to perform better, learn from the training data faster, and overcomes the vanishing gradient problem. It is demonstrated using Equation (1) as follows:

$$R(z) = max(0, z)$$

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10s
DOI: https://doi.org/10.17762/ijritcc.v11i10s.7588
***Article Received**: 28 May 2023 **Revised**: 22 July 2023 **Accepted**: 10 August 2023*

Here, z represents the input being provided to the activation function and R(z) represents a positive output of the activation function. Convolution layers are often followed by another block of CNN pooling layers. Internally, the Sub-Sampling technique is being used to reduce the reliance on precise positioning of the feature maps on the model; however, the feature maps should be independent of the positioning of information to avoid the model from overfitting. The computation of the architecture is dependent on the complexity and the number of parameters; pooling layers operate on the feature maps for the generation of the mapped pooled features. The selection of mapped pooled features varies based on the size of pooling filter applied, stride, and type of poolingmax pooling, average pooling. Max pooling sets the value as the maximum value of the feature for each patch whereas average pooling calculates the average value for each patch. Deep learning neural networks may likely overfit the training data, which could reduce the overall performance of the model when evaluated on the unseen data. So, we employed the use of dropout layers. It is a regularization technique that ignores some neurons that are further processing randomly. It sets the inputs to 0 at each step with the frequency of rate during the training phase for preventing the model from overfitting. Active inputs are then scaled up by a certain factor such that the sum of all the inputs remains the same using Equation (2).

$$z = \frac{1}{1 - rate}$$

Hence, this makes the training process noisy by imposing more responsibility on some nodes. It is only used during the training process. However, dropout increases the weight of the network, and scaling is required up to the chosen dropout rate. They are then followed by the dense layers, fully connected with the previous layer, and an activation function for mapping the results to the output.

b. HYBRID CNN+LSTM

CNN is suitable for extracting data features; LSTM is suitable for processing time series, solving the dependency problem between time-series data, and improving recognition accuracy. This paper combines the advantages of the two algorithms and proposes the CNN+LSTM algorithm. Convolution neural network (CNN) [21] evolved from multilayer perception (MLP) [22]. Compared with traditional feature selection algorithms, this algorithm can learn features better. The more traffic data CNN can learn, the more useful features there are, the better the classification, which is suitable for large-scale network environments. As shown in Figure 1, its structure is divided into a convolution layer, a pooling layer, and a fully connected layer. The role of the convolution layer is to extract features, and the role of the pooling layer is to sample the features. Finally, the fully connected layer is responsible for connecting the extracted features and obtaining the classification results through the classifier.

The long-term memory network (LSTM) is an improved recurrent neural network (RNN) method, which aims to alleviate the explosion gradient problem. Compared with traditional RNN units, LSTM uses a set of gate functions to control feedback so that short-term errors will eventually be deleted while persistent features will be retained. +e data processing flow is shown in Figure 2.

The LSTM is abstracted into four subnets (p-net, g-net, f-net, and q-net), a collection of gate controllers, and a link to the memory component. The figure's input and output are controlled by the vector's size, x (t). The state's (t) contains information about the present learning.

The CNN-LSTM method is capable of expressing both temporal and spatial information. Due to the fact that an intrusion assault occurs in real time, the methods of attack are varied, as is the target or point of attack. To extract features, a CNN is utilized, and high-level features may be retrieved using the convolution kernel operation, which has been successfully used in image processing [23]-[25]. Additionally, LSTM utilizes gate functions to regulate the remembering and forgetting of previous data, making it ideal for processing long-term sequence data and increasing detection accuracy [26]-[27]. As a result, the CNN-LSTM algorithm model is suitable for intrusion detection processing in this study. Figure 3 illustrates the CNN-LSTM algorithm paradigm, and the particular stages are as follows:

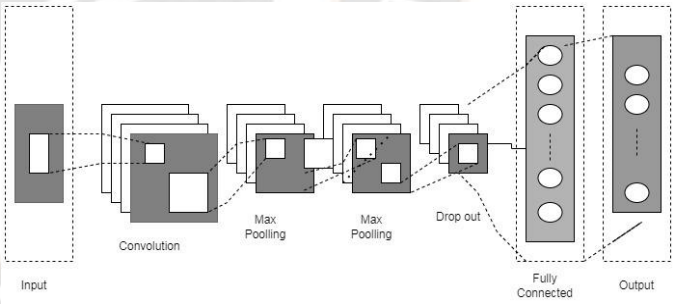

Figure 2: 1D-CNN Model

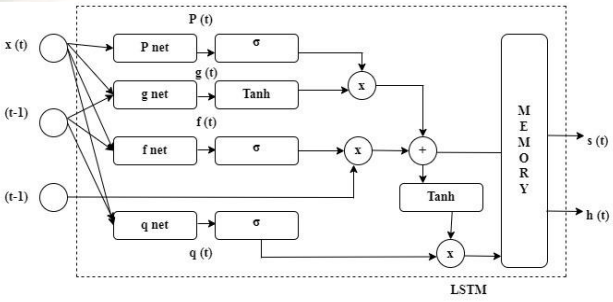

Figure 3: Data Processing Model for LSTM

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10s
DOI: https://doi.org/10.17762/ijritcc.v11i10s.7588
***Article Received**: 28 May 2023 **Revised**: 22 July 2023 **Accepted**: 10 August 2023*

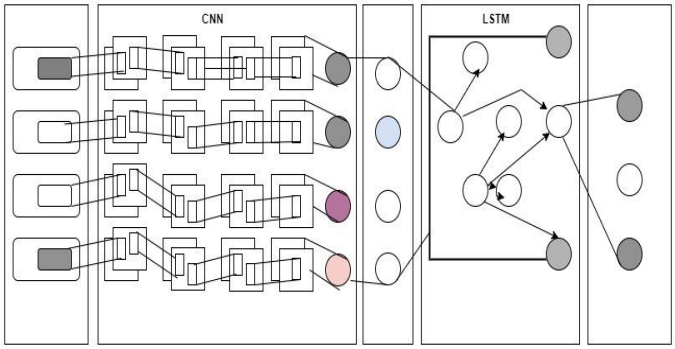

Figure 4: Proposed Hybrid CNN+LSTM Model

(1) The input layer collects CICIDS 2017 data through the flow data collection module. This article uses the dataset to analyze characteristics, including network protocol types, network service types, network connection status, and connection time [28]-[29].
(2) According to the data processing steps, the data are respectively preprocessed, digitized, and normalized. The specific operation steps will be described in detail later.
(3) It sends the processed data to the convolution layer for feature extraction and outputs the features through a one-dimensional convolution operation.Each convolution layer is accompanied by a pooling layer to reduce feature dimensions, accelerate convergence, and remove redundancy features to prevent network overfitting. then all local features are integrated through the fully connected layer to form an overall feature. Finally, the leaky ReLU activation function in the fully connected layer is operated [30]-[32].
(4) Input the features extracted by CNN into LSTM. After the SoftMax function, the classification result of network data is obtained [23]-[33]-[34].
The algorithm steps for intrusion detection and classification over the IoT using a hybrid CNN-LSTM model can be summarized as follows:

Algorithm: Hybrid CNN+LSTM
Input:CICIDS2017 dataset
Output:
Steps
1: Data Collection: Collect CICIDS2017 dataset.
2: Data Preprocessing: Clean the data and remove any irrelevant features. Convert the data into a format suitable for processing by the model.
3: Splitting the data: Split the preprocessed data into training, validation, and testing sets.
4: Training the CNN Model: Train a CNN model on the preprocessed training data to extract important features from the input.
5: Training the LSTM Model: Train an LSTM model on the output of the CNN model to capture the temporal dependencies in the data.
6: Hybrid Model Creation: Combine the trained CNN and LSTM models to create a hybrid CNN-LSTM model.
7: Training the Hybrid Model: Train the hybrid CNN-LSTM model on the preprocessed training data to improve its performance.
8: Testing the Hybrid Model: Test the trained hybrid CNN-LSTM model on the preprocessed testing data to evaluate its accuracy in detecting intrusions.
9: Intrusion Classification: Classify the detected intrusions into different categories based on their type and severity.
10: Performance Evaluation: Evaluate the performance of the hybrid CNN-LSTM model in terms of accuracy, precision, recall, and F1-score.
11: Model Optimization: Optimize the hybrid CNN-LSTM model by fine-tuning the hyperparameters and adjusting the network architecture.
12: Deployment: Deploy the optimized hybrid CNN-LSTM model in the IoT network to monitor and detect intrusions in real time.

VII. EVALUATION CRITERIA

The modeling of objective functions was used to determine the outcome of the Bi-LSTM and Hybrid CNN+LSTM Deep learning algorithms. The scenario used in the study was set up for the performance of each Deep-learning technique tested on the dataset used. Evaluation parameters can be used in assessing the performance of both the deep learning classifiers including accuracy, Precision, Recall, and F1-Score. A performance matrix comes from a confusion matrix. A confusion matrix is a table that visualizes the performance of a classification algorithm tested versus actual classification.
True Positive (TP): The model correctly predicted an instance as positive (intrusion), and the true label is also positive.
False Positive (FP): The model incorrectly predicted an instance as positive (intrusion), but the true label is negative (non-intrusion).
False Negative (FN): The model incorrectly predicted an instance as negative (non-intrusion), but the true label is positive (intrusion).
True Negative (TN): The model correctly predicted an instance as negative (non-intrusion), and the true label is also negative.

$$Accuracy = \frac{T\_P + T\_N}{T\_P + T\_N + F\_P + F\_N}$$

$$Precision = \frac{T\_P}{T\_P + F\_P}$$

$$Recall = \frac{T\_P}{T\_P + F\_N}$$

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

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10s
DOI: https://doi.org/10.17762/ijritcc.v11i10s.7588
***Article Received**: 28 May 2023 **Revised**: 22 July 2023 **Accepted**: 10 August 2023*

VIII. RESULT ANALYSIS

We trained the models with the selected features. In our model, we used 80% of the data for training and the remaining 20% for testing. Therefore, we could train and test the model with only 20% of the dataset. This allowed us to accurately predict the attack.

This section contains experimental results implemented in Collaboratory by Google Research. All experiments were performed using the Python programming language. The datasets were divided into training and testing datasets for the experiments. The CICIDS 2017dataset was divided into training and test sets 80%-20%, respectively. The 50% of data is used for validation from 80% of training data. We used the Kera's library to build our classifiers and we used TensorFlow as the backend of the Kera's library.

TABLE II. PARAMETERS SETTING FOR BOTH THE DEEP LEARNING CLASSIFIERS

Classifiers	Batch Size	Activation Function	Optimizer	Epoch	No of Layers	Dense Value
LSTM	64	Sigmoid	Adam	60	1	128
Proposed Hybrid CNN+LSTM	128	Relu	Adam	200	1+2 CNN layers	16+ 64 Conv.
Best Epoch Proposed Model	128	Relu	Adam	154	1+2 CNN layers	16+ 64 Con.

Table II shows the parameters setting for both the deep learning model. It is clearly show that used hybrid approach and combined CNN and LSTM classifiers with 64 Convolutional layers of CNN and 16 layers of LSTM respectively. While executing this experiment, initially 200 epochs were set for proposed hybrid CNN+LSTM model. At the end of the model, it is observed that model is get converged at 154 epochs. So that best epoch value of this model is 154.

TABLE III. ACCURACY SCORE OF PROPOSED MODELS

Classifiers	Accuracy	Loss
LSTM	98.74	0.0545
Proposed Hybrid CNN+LSTM	99.81	0.0071
Best Epoch Proposed Model	99.82	0.0075

TABLE IV. TRAINING AND TESTING ACCURACY OF BEST EPOCH PROPOSED MODEL

Model	Accuracy on Train Set:	Accuracy on Test Set:
Best Epoch Proposed Model	0.9978	0.9964

TABLE V. RESULT ANALYSIS OF BI-LSTM MODEL FOR BINARY CLASSES

Class	Precision	Recall	F1-Score
0	0.99	0.99	0.99
1	0.99	0.99	0.99
Average	**0.99**	**0.99**	**0.99**

TABLE VI. . RESULT ANALYSIS OF HYBRID CNN+LSTM MODEL FOR BINARY CLASSES

Class	Precision	Recall	F1-Score
0	1.0	1.0	1.0
1	1.0	1.0	1.0
Average	**1.0**	**1.0**	**1.0**

Figure 5: Training and Validation Accuracy of Bi-LSTM Model

The training and validation accuracy graph of a Bi-LSTM model for intrusion detection over an IoT environment can provide insight into how well the model is learning and generalizing to new data. The figure 5 shows how the accuracy of the model changes over time as it is trained on a dataset. the training accuracy (blue line) and validation accuracy (orange line) both increase and plateau at around 99% after approximately 60 epochs. This suggests that the model is effectively learning the patterns in the training data and is able to generalize well to new, unseen data in the validation set. There is also minimal overfitting, as the gap between the training and validation accuracy curves is relatively small.

Figure 6: Training & validation loss of Bi-LSTM Model

The training loss (blue line) and validation loss (orange line) both decrease over time, indicating that the model is becoming

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10s
DOI: https://doi.org/10.17762/ijritcc.v11i10s.7588
***Article Received**: 28 May 2023 **Revised**: 22 July 2023 **Accepted**: 10 August 2023*

better at making predictions on the data. The training loss decreases more rapidly than the validation loss, which suggests that the model is slightly overfitting to the training data.

Overall, this figure 6 shows that the Bi-LSTM model is able to effectively learn the patterns in the data and make accurate predictions on new data. However, to further improve the model's performance, techniques such as early stopping and regularization can be used to prevent overfitting and improve generalization to new data.

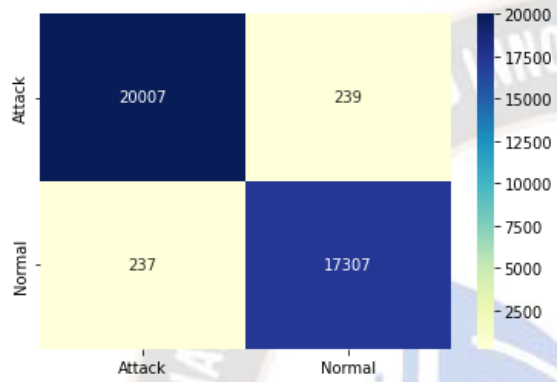

Figure 7: Confusion Matrix for Bi_LSTM Model

Figure 7 shows the confusion matrix is a table that is often used to evaluate the performance of a Bi_LSTM model. It shows the number of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) for a set of predictions compared to the true values.The Bi-LSTM model predicted 20007 out of 20246 attack instances correctly (true positive), and 17307 out of 17544normal instances correctly (true negative). It also incorrectly predicted 237normal instance as intrusive (false positive), and 239attack instances as non-intrusive (false negative).

c. HYBRID CNN-LSTM

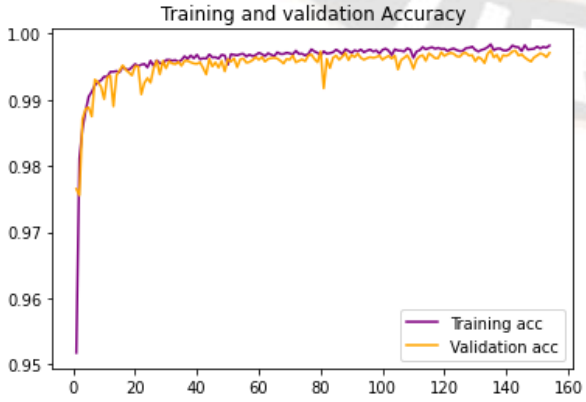

Figure 8: Training and Validation Accuracy of hybrid CNN-LSTM Model

The training accuracy (Purple line) and validation accuracy (orange line) both increase and plateau at around 99% after approximately 60 epochs. This suggests that the model is effectively learning the patterns in the training data and is able to generalize well to new, unseen data in the validation set. There is also minimal overfitting, as the gap between the training and validation accuracy curves is relatively small.

Overall, figure 8 demonstrates that the hybrid CNN-LSTM model is able to effectively detect intrusions in an IoT environment with high accuracy. The inclusion of the convolutional layer allows the model to extract more complex features from the input data, improving its ability to classify intrusions.

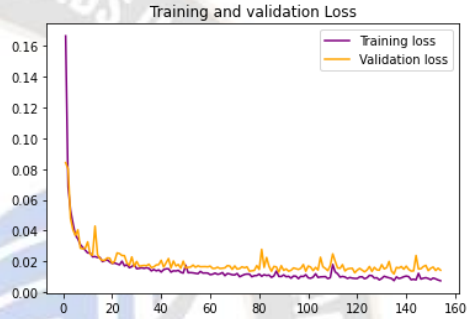

Figure 9: Training and Validation Loss of hybrid CNN-LSTM Model

The training loss (Purple line) and validation loss (orange line) both decrease over time, indicating that the model is becoming better at making predictions on the data. The training loss decreases more rapidly than the validation loss, which suggests that the model is slightly overfitting to the training data.

Overall, figure 9 shows that the hybrid CNN-LSTM model is able to effectively learn the patterns in the data and make accurate predictions on new data. However, to further improve the model's performance, techniques such as early stopping and regularization can be used to prevent overfitting and improve generalization to new data.

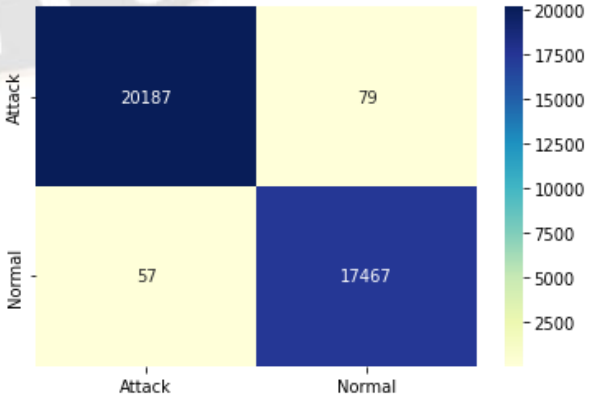

Figure 10: Confusion Matrix for hybrid CNN-LSTM Model

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10s
DOI: https://doi.org/10.17762/ijritcc.v11i10s.7588
***Article Received**: 28 May 2023 **Revised**: 22 July 2023 **Accepted**: 10 August 2023*

Figure 10 shows the confusion matrix is a table that is often used to evaluate the performance of a Hybrid CNN-LSTM model. It shows the number of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) for a set of predictions compared to the true values. The Hybrid CNN-LSTM model predicted 20187 out of 20266 attack instances correctly (true positive), and 17467 out of 17524 normal instances correctly (true negative). It also incorrectly predicted 57 normal instance as intrusive (false positive), and 79 attack instances as non-intrusive (false negative).

IX. COMPARATIVE ANALYSIS

TABLE VII. PCOMPARATIVE ANALYSIS OF PROPOSED METHOD WITH EXISTING STATE-OF-ART METHOD

Ref.	Classifiers Used	Accuracy Score
Chen, L et. al. (2020) [26]	CNN	96.55
Roopak, M. et. al. (2019) [34]	CNN, LSTM	97.16
Sivamohan, S. et. al. (2021) [41]	RNN and CNN	98.46
Qazi, E.U.H. et. al. (2022)	1-D-CNN	96.5
Atefi, K et. al. (2020) [36]	DNN	98.96
Asad, M. et. al. (2020) [39]	DL, Feed-Forward Back-Propagation Network	98
Proposed	Hybrid CNN+LSTM	**99.82**

X. CONCLUSION

Intrusion detection systems (IDSs) are crucial for ensuring the security and integrity of Internet of Things (IoT) networks. With the increasing number of connected devices and the amount of data generated, it is essential to develop efficient and accurate IDSs to detect and prevent security threats.A hybrid CNN-LSTM model has emerged as a promising approach for intrusion detection in IoT networks. This model combines the strengths of convolutional neural networks (CNNs) and long short-term memory (LSTM) networks, enabling it to extract complex features from raw data and capture temporal dependencies in the data. This can lead to improved accuracy and efficiency in detecting intrusions and reducing false positives.Moreover, the hybrid CNN-LSTM model can be trained on large amounts of data generated by IoT devices, which can improve its performance and make it more robust to different types of attacks. This makes it a highly effective IDS for IoT networks, particularly when compared to traditional IDSs. However, the effectiveness of the hybrid CNN-LSTM model is highly dependent on the quality and quantity of the training data. As such, it is important to develop effective data collection and pre-processing techniques to ensure that the model is accurately trained. The hybrid CNN-LSTM model is a promising approach for intrusion detection in IoT networks. Its ability to extract complex features and capture temporal dependencies makes it highly effective in detecting security threats. As IoT networks continue to grow and evolve, the development and implementation of effective IDSs will become increasingly important to ensure the security and integrity of IoT devices and data.

REFERENCES

[1] Mahmudul Hasan, Md. Milon Islam, Ishrak Islam, M.M.A. Hashem, Attack and Anomaly Detection in IoT Sensors in IoT Sites Using Machine Learning Approaches, Internet of Things (2019), doi: https://doi.org/10.1016/j.iot.2019.100059

[2] M.-O. Pahl, F.-X. Aubet, All eyes on you: Distributed Multi-Dimensional IoT microservice anomaly detection, in: 2018 14th International Conference on Network and Service Management (CNSM) (CNSM 2018), Rome, Italy, 2018.

[3] M.-O. Pahl, F.-X. Aubet, S. Liebald, Graph-based IoT microservice security, in: NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium, IEEE, 2018, pp. 1–3.

[4] H. H. Pajouh, R. Javidan, R. Khayami, D. Ali, K.-K. R. Choo, A two-layer dimension reduction and two-tier classification model for anomaly-based intrusion detection in IoT backbone networks, IEEE Transactions on Emerging Topics in Computing

[5] Banaamah, A.M.; Ahmad, I. Intrusion Detection in IoT Using Deep Learning. Sensors 2022, 22, 8417. https://doi.org/10.3390/s22218417

[6] Zhang, J.; Pan, L.; Han, Q.-L.; Chen, C.; Wen, S.; Xiang, Y. Deep learning-based attack detection for cyber-physical system cybersecurity: A survey. IEEE/CAA J. Autom. Sin. 2021, 9, 377-391

[7] Jaffar, S. ., & Baharum, A. . (2023). Gamification Framework for Engagement Design Using Pls-Sem. International Journal of Intelligent Systems and Applications in Engineering, 11(4s), 194–202. Retrieved from https://ijisae.org/index.php/IJISAE/article/view/2646

[8] Lee, I. Internet of Things (IoT) cybersecurity: Literature review and IoT cyber risk management. Futur. Internet 2020, 12, 157.

[9] Almiani, M.; AbuGhazleh, A.; Al-Rahayfeh, A.; Atiewi, S.; Razaque, A. Deep recurrent neural network for IoT intrusion detection system. Simul. Model. Pract. Theory 2020, 101, 102031.

[10] Azumah, S.W.; Elsayed, N.; Adewopo, V.; Zaghloul, Z.S.; Li, C. A deep lstm based approach for intrusion detection iot devices network in smart home. In Proceedings of the 2021 IEEE 7th World Forum on Internet of Things (WF-IoT), New Orleans, LA, USA, 26–31 July 2021.

[11] Bambang Susilo and Riri Fitri Sari (2020). Intrusion Detection in IoT Networks Using Deep Learning Algorithm, Information 2020, 11, 279; doi:10.3390/info11050279

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10s
DOI: https://doi.org/10.17762/ijritcc.v11i10s.7588
***Article Received**: 28 May 2023 **Revised**: 22 July 2023 **Accepted**: 10 August 2023*

[12] Lee, S., Shin, D., Kang, S., Kim, J., & Kim, C. (2021). A Hybrid Deep Learning Model for IoT Intrusion Detection. Sensors, 21(15), 5247. https://doi.org/10.3390/s21155247

[13] Liu, X., Zhang, Y., Xie, Y., Wang, Y., & Zhang, W. (2021). Deep Learning Based Intrusion Detection for IoT Systems Using Bi-LSTM and Convolutional Neural Network. IEEE Access, 9, 84015-84025. https://doi.org/10.1109/ACCESS.2021.3088101

[14] Lee, W.-H., Chen, Y.-C., Lee, Y.-H., & Yang, C.-C. (2021). Intrusion Detection in IoT Networks Using Bi-LSTM and Convolutional Neural Networks. In Proceedings of the International Conference on Network-Based Information Systems (pp. 311-316). Springer. https://doi.org/10.1007/978-3-030-79150-3_33

[15] Lee, S., Park, J., Kim, S., & Yoon, S. (2020). IoT Intrusion Detection Using Bidirectional LSTM. In Proceedings of the 15th International Conference on Availability, Reliability and Security (pp. 1-8). ACM. https://doi.org/10.1145/3407023.3407047

[16] Khan, M. A., Abbas, H., Hussain, M., & Bilal, K. (2020). A Comprehensive Study on Intrusion Detection Systems in IoT Environments. Journal of Ambient Intelligence and Humanized Computing, 11(9), 4173-4200. https://doi.org/10.1007/s12652-019-01471-4

[17] Wei, Y.-C., Hu, Y.-H., & Hsu, S.-S. (2020). Anomaly Detection for IoT Cybersecurity Using Bidirectional Long Short-Term Memory Networks. IEEE Access, 8, 79773-79782. https://doi.org/10.1109/ACCESS.2020.2998123

[18] Zhang, Y.; Chen, X.; Jin, L.; Wang, X.; Guo, D. Network Intrusion Detection: Based on Deep Hierarchical Network and Original Flow Data. IEEE Access 2019, 7, 37004-37016.

[19] Jamil, F.; Kim, D. An Ensemble of Prediction and Learning Mechanism for Improving Accuracy of Anomaly Detection in Network Intrusion Environments. Sustainability 2021, 13, 10057

[20] Asad, M.; Asim, M.; Javed, T.; Beg, M.O.; Mujtaba, H.; Abbas, S. Deepdetect: Detection of distributed denial of service attacks using deep learning. Comput. J. 2020, 63, 983-994

[21] Bhardwaj, A.; Mangat, V.; Vig, R. Hyperband tuned deep neural network with well posed stacked sparse autoencoder for detection of DDoS attacks in cloud. IEEE Access 2020, 8, 181916–181929.

[22] Altunay, Hakan & Albayrak, Zafer. (2023). A hybrid CNN + LSTM-based intrusion detection system for industrial IoT networks. Engineering Science and Technology an International Journal. 38. 10.1016/j.jestch.2022.101322.

[23] Ali Alferaidi, Kusum Yadav, Yasser Alharbi, Navid Razmjooy, Wattana Viriyasitavat, Kamal Gulati, Sandeep Kautish, Gaurav Dhiman, "Distributed Deep CNN-LSTM Model for Intrusion Detection Method in IoT-Based Vehicles", Mathematical Problems in Engineering, vol. 2022, Article ID 3424819, 8 pages, 2022. https://doi.org/10.1155/2022/3424819

[24] W. Cao, "CNN-based intelligent safety surveillance in green IoT applications," China Communications, vol. 18, no. 1, pp. 108–119, 2021.

[25] M. Ganesan and N. Sivakumar, "IoT based heart disease prediction and diagnosis model for healthcare using machine learning models," in Proceedings of the 2019 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN), pp. 1-5, Pondicherry, India, March 2019.

[26] R. Fotohi, S. Firoozi Bari, and M. Yusefi, "Securing wireless sensor networks against denial-of-sleep attacks using RSA cryptography algorithm and interlock protocol," International Journal of Communication Systems, vol. 33, no. 4, Article ID e4234, 2020.

[27] Rajesh Kumar Chaudhary, M. K. C. (2021). The Role of School Management Towards Staff Motivation for Effective Performance in Nepal: During the Covid-19. International Journal of New Practices in Management and Engineering, 10(01), 01–11. https://doi.org/10.17762/ijnpme.v10i01.93

[28] R. Fotohi and S. F. Bari, "A novel countermeasure technique to protect WSN against denial-of-sleep attacks using firefly and Hopfield neural network (HNN) algorithms," The Journal of Supercomputing, vol. 76, pp. 1–27, 2020.

[29] M. Zaminkar and R. Fotohi, "SoS-RPL: securing internet of things against sinkhole attack using RPL protocol-based node rating and ranking mechanism," Wireless Personal Communications, vol. 114, 2020.

[30] G. D. L. T. Parra, P. Rad, K. K. R. Choo, and N. Beebe, "Detecting Internet of Things attacks using distributed deep learning," Journal of Network and Computer Applications, vol. 163, Article ID 102662, 2020.

[31] R. Fotohi, E. Nazemi, and F. S. Aliee, "An agent-based self-protective method to secure communication between UAVs in unmanned aerial vehicle networks," Vehicular Communications, vol. 26, Article ID 100267, 2020.

[32] M. Roopak, G. Y. Tian, and J. Chambers, "An intrusion detection system against ddos attacks in iot networks," in Proceedings of the 2020 10th Annual Computing and Communication Workshop and Conference (CCWC), pp. 0562–0567, IEEE, Las Vegas, NV, USA, January 2020.

[33] T. Alladi, A. Agrawal, B. Gera, V. Chamola, B. Sikdar, and M. Guizani, "Deep neural networks for securing IoT enabled vehicular ad-hoc networks," in Proceedings of the ICC 2021-IEEE International Conference on Communications, pp. 1–6, IEEE, Montreal, QC, Canada, June 2021.

[34] Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai, "Kitsune: an ensemble of autoencoders for online network intrusion detection," 2018, https://arxiv.org/abs/1802.09089.

[35] A. Khraisat, I. Gondal, P. Vamplew, J. Kamruzzaman, and A. Alazab, "Hybrid intrusion detection system based on the stacking ensemble of c5 decision tree classifier and one class support vector machine," Electronics, vol. 9, no. 1, p. 173, 2020.

[36] G. J. M. Ariyathilake, M. H. R. Sandeepanie, and P. L. Rupasinghe, "SQL injection detection and prevention solution for web applications," 2021, http://ir.kdu.ac.lk/handle/345/5253.

[37] Chen, L.; Kuang, X.; Xu, A.; Suo, S.; Yang, Y. A Novel Network Intrusion Detection System Based on CNN. In Proceedings of the 2020 Eighth International Conference on

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10s
DOI: https://doi.org/10.17762/ijritcc.v11i10s.7588
***Article Received**: 28 May 2023 **Revised**: 22 July 2023 **Accepted**: 10 August 2023*

Advanced Cloud and Big Data (CBD), Taiyuan, China, 19–20 September 2020; pp. 243-247.

[38] Mwangi , J., Cohen, D., Silva, C., Min-ji, K., & Suzuki, H. Improving Fraud Detection in Financial Transactions with Machine Learning. Kuwait Journal of Machine Learning, 1(4). Retrieved from http://kuwaitjournals.com/index.php/kjml/article/view/148

[39] Roopak, M.; Tian, G.Y.; Chambers, J. Deep learning models for cyber security in IoT networks. In Proceedings of the 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 7–9 January 2019; pp. 452-457.

[40] Sivamohan, S.; Sridhar, S.S.; Krishnaveni, S. An Effective Recurrent Neural Network (RNN) based Intrusion Detection via Bidirectional Long Short-Term Memory. In Proceedings of the 2021 International Conference on Intelligent Technologies (CONIT), Karnataka, India, 25–27 June 2021; pp. 1-5.

[41] Atefi, K.; Hashim, H.; Khodadadi, T. A hybrid anomaly classification with deep learning (DL) and binary algorithms (BA) as optimizer in the intrusion detection system (IDS). In Proceedings of the 2020 16th IEEE International Colloquium on SIGNAL Processing & Its Applications (CSPA), Langkawi, Malaysia, 28–29 February 2020; pp. 29-34.

[42] Asad, M.; Asim, M.; Javed, T.; Beg, M.O.; Mujtaba, H.; Abbas, S. Deepdetect: Detection of distributed denial of service attacks using deep learning. Comput. J. 2020, 63, 983–994.