International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Intrusion Detection System using Long Short-Term Memory and Fully Connected Neural Network on Kddcup99 and NSL-KDD Dataset

[1]Ankit Chakrawarti, [2]Dr. Shiv Shakti Shrivastava

Submitted: 24/04/2023 **Revised**: 26/06/2023 **Accepted**: 05/07/2023

Abstract: Implementing intrusion detection models, which include locating and categorizing unauthorized access to a computer network or information system, often uses machine learning techniques. However, numerous difficulties occur as a result of the fact that cybercriminals constantly alter their attack techniques in response to the discovery of new system vulnerabilities. The number of efforts to harm is quickly increasing, and as a consequence, traditional methods cannot analyses massive amounts of data. Therefore, a comprehensive detection strategy that incorporates scalable solutions is necessary to solve the issue. A deep learning model is offered to solve the intrusion classification challenge properly. , Deep Learning (DL) algorithms have produced very accurate outcomes for handling various problems in practically various area. Deep learning methods such as LSTM (Long Short-Term Memory) and FCNN (Fully Connected Neural Network) categorize benign and malicious connections on intrusion datasets. A more precise categorization of multi-class assault patterns is the goal of this endeavor. When applied to five-class issues, the deep learning model that was suggested produces more accurate classifications. When run on the KDDCup99 dataset, it reaches an accuracy of 99.99%, and when run on the NSL-KDD dataset, it reaches 99.95%.In both data set our model secure maximum output.

Keywords: *Intrusion Detection, LSTM, CNN, KDDCup99, KDDCup99.*

1. Introduction

Intrusion detection systems are essential for network management, monitoring, and preventing harmful activity—the explosion in various network traffic results from varying speeds. A wide range of network requests may fool an intrusion detection system (IDS) and indicate an attempt to breach network security [1]. To manage the vast amount of data in terms of its volume, velocity, and diversity, a "big data" architecture is used. It makes available tools that enable the simple collection, injection, pre-processing, storage, analysis, and visualization of high-dimensional data. Several machine-learning approaches have been proposed as potential methods for the scalable architecture created using Apache Spark or Apache Mahout. To identify incursions in the Big Data ecosystem, models based on the Support Vector Machine (SVM), Decision Trees (DT), K-Nearest Neighbor (K-NN), and Neural Networks have been created. When two classes are involved, the models' predictions for the outcomes are sufficient and consistent across all relevant data sets. However, when dealing with difficulties involving many classes, the detection accuracy is insufficient for the minority classes [1]. Because of the imbalanced data used to train the model, it often exhibited a bias toward the underrepresented class. The deep learning approach could be the most effective method for developing a versatile and effective IDS recognizing complicated attacks. This is due to the availability of a lot of training data.

These days, Deep Learning (DL) algorithms have produced amazing outcomes for handling various problems in practically all areas [2]. The three main problems that emerge during the training of neural networks—vanishing gradient, overfitting, and computational burden—can be avoided using deep learning techniques. These difficulties are overcome. Deep learning is an improved form of the neural network. In this version, the learning process and the ability to predict future events based on unseen data are boosted via several changes.

Self-training and optimization of the model learning process are both capabilities of the deep learning technique, also known as hierarchical or deep structural learning. The efficiency of the learning process is enhanced along with the growth in the amount of data collected. In general, a much larger quantity of training data is required in comparison to the requirements of other machine learning algorithms. The concealed layers may be changed, and a broad range of options for increasing the learning process can be chosen based on the needs. Several academic areas, including natural language processing, voice recognition, computer vision, audio recognition, machine translation, social network filtering,

[1]*Department of Computer Science and Engineering, Rabindranath Tagore University, Raisen(M.P.)*
chak03ankit@gmail.com
[2]*Department of Computer Science and Engineering, Rabindranath Tagore University, Raisen(M.P.)*
Shivshakti18@gmail.com

and intrusion detection, are now using Deep Learning (DL) approach [2]. These methods include, among others, Recurrent Neural Networks (RNN), Deep Neural Networks (DNN), and Deep Belief Networks (DBN).

The network or system is not completely secure because of problems in zero-day attack detection, analysis, and response. This is true even though many researchers have used a variety of methodologies to create IDS, as well as the surge in internet users, network traffic, and new security analysis tools.

Our proposed intrusion detection system applies the CNN model for accurate and timely detection of SDN network intrusions. This was motivated by the fact that CNN-based network modelling was previously successfully used to solve various difficult classification issues. In addition, dimensionality reduction was one of the driving forces for our decision to utilize CNN. Anomaly detection has been accomplished using CNN. However, it has not been able to find them with enough precision. The malicious traffic closely resembles the legitimate data, and there is hardly any distinction between the two types of data. The use of CNN would be beneficial in identifying the subtle distinctions that exist between the various borders. To solve this issue, the given work makes use of regularization approaches. These methods assist build a generalized model that fits well on datasets or instances that have not been seen previously, reducing the amount of over fitting. Thus, we can improve CNN's anomaly detection capabilities. To improve the extraction of spatial and temporal information from the input data, we integrated CNN with the LSTM. Neither CNN nor LSTM have been used together in any previous research aimed at detecting network anomalies.

The following are some of the research contributions made by this study:

- Construct a Python-based CNN-LSTM algorithm and then tweak its hyper-parameters using the ADAM optimizer for improved intrusion detection and lower error rates.
- The goal is to properly classify the imbalanced intrusion data into two distributions: one for the majority class and one for the minority class.
- We can better align the two kinds of accuracy measurements by reducing the proportion of false positives and false negatives.
- Create a method of intrusion detection that can tell the difference between "normal," "DoS," "probe," "U2R," and "R2L" cases.

2. Literature Study

The performance of the proposed model is evaluated using the NSL-KDD and CIS-IDS2017 datasets. In addition to a higher TPR for most assault events, quicker data preparation speed and maybe reduced training time are also shown by the experimental results of the proposed model. In particular, the CIC-IDS2017 dataset [11] shows that multi-target classification accuracy may reach 99.91%. The databases revealed both sets of numbers.

We first use a feature selection method based on the concept of "contribution." In the next step, we recreate the multi-feature correlation and extract temporal-spatial data using a combination of a convolutional neural network (CNN) and a long short-term memory (LSTM). Either we investigate how the attention process influences temporal-spatial qualities or combine temporal-spatial variables with correlation features. We build a Deep Neural Network (DNN) to spot outliers in data using the following techniques. HNN improved accuracy by 3.78 percent, 1.31 percent, 0.21 percent, and 1.13 percent on the UNSW-NB15 dataset, the AWID dataset, CICIDS 2017, and CICIDS 2018, accordingly to the results of the experiments [12].

When finding intruders, we use an AS-CNN model incorporating ADASYN and SPC-CNN. Having given it some thought, we've decided to put AS-CNN through its paces on the reliable NSL-KDD dataset. According to the simulation findings, the accuracy is improved by 4.60 percentage points compared to the standard CNN and RNN models, and the DR increased by 10.27 percentage points. The simulation also shows an improvement in detecting 11.34 and 10.27 percentage points. And the FAR dropped by 15.58 and 14.57 percent between the two models, respectively [13].

We analyses the performance of the proposed CNN model using a typical KDD-CUP99 dataset after converting the original traffic vector format into an image format to decrease the amount of work required by the computer. Experiments show that the CNN-IDS model provides superior AC, FAR, and timeliness compared to the reference methods. There is theoretical and practical value in our suggested framework [14].

A deep hierarchical network model is built with the help of a Convolution Neural Network (CNN) to extract spatial information and a Bi-directional Long Short-Term Memory (BiLSTM) to extract temporal features. Examples of such models are the Convolutional Neural Network (CNN) and the Short-Term Long Memory (LSTM). Experiments on the NSL-KDD and UNSW-NB15 datasets confirmed the effectiveness of the proposed network intrusion detection strategy. The

findings suggest that the system's classification accuracy may range from 83.58% to 77.16% [15].

We also convert the raw traffic vector data to a graphical representation to minimize the processing cost. To test the effectiveness of the proposed CNN model, we use the widely used NSL-KDD dataset. The results of the experiments demonstrate that the suggested model achieves better accuracy, FAR, and computing effort than the state-of-the-art methods. It is a technique for detecting intrusions into large networks that is dependable and effective [16].

Using less than one percent of the NSL-KDD KDDTrain+ dataset for training, the proposed Few-Shot Learning (FSL) technique for intrusion detection achieved great accuracy (92.34% for KDD-Test+ and 85.75% for KDD-Test-21). This is in contrast to methods that depend on guesswork, such as J48, Naive Bayes (NB), Random Forest (RF), Support Vector Machine (SVM), and recurrent neural networks (RNN). The experiment utilizing the UNSW-NB15 dataset yielded similar results. In addition, our method improves the sensitivity of detecting Dos, U2R, R2L, and U2R. Particularly, the rates at which U2R and R2L are detected are raised from 13% to 81.50% and from 44.41% to 75.93%, respectively [17]. This is especially true since U2R makes up a small percentage of the dataset.

The Safe Water Treatment (SWaT) dataset is used to assess the accuracy of the models. We computed accuracy, recall, and F1 scores on the SWaT dataset to measure how well this strategy works for spotting outliers. The experimental findings reveal that this approach has a mean accuracy of 0.99, recall of 0.85, and F1 score of 0.91. The table below displays these figures. Based on experimental findings, the suggested strategy can potentially lessen the occurrence of false positives in anomaly detection systems used by industrial control systems [18].

This paper focuses on the application of Long Short-Term Memory (LSTM) based deep learning models for intrusion detection in IoT networks. The authors propose an LSTM-based model that utilizes network traffic data to detect anomalies and potential intrusions in IoT networks. They evaluate the performance of their model using various metrics and demonstrate its effectiveness in detecting intrusion attacks [31].

In this paper, the authors present an intelligent intrusion detection system (IDS) based on deep learning with LSTM. They develop a model that combines LSTM and fully connected neural network architectures to classify network traffic as normal or malicious. The proposed system achieves high accuracy in detecting various types of attacks and outperforms traditional methods. The study provides insights into the effectiveness of LSTM-based models for intrusion detection [32].

This paper presents a comprehensive survey of deep learning approaches for intrusion detection systems (IDS). It provides an overview of different deep learning techniques, including LSTM, and their application in IDS. The authors review and analyze various studies and highlight the advantages and limitations of deep learning-based IDS. The survey serves as a valuable resource for understanding the current state of deep learning in intrusion detection [33].

This paper focuses on the detection of botnet attacks in Internet of Things (IoT) networks using an LSTM-based model. The authors propose a detection model that utilizes LSTM to analyze network traffic and identify botnet activities. The model demonstrates promising results in terms of accuracy and detection rates. The study emphasizes the importance of LSTM-based models in enhancing the security of IoT networks [34].

This paper presents an anomaly-based network intrusion detection system (NIDS) using deep learning techniques. The authors propose a model that combines deep autoencoder and LSTM to detect intrusions by learning normal network behavior. The system is trained on labeled data to distinguish between normal and anomalous network traffic. The study demonstrates the effectiveness of the proposed approach in detecting various types of attacks [35].

In this conference paper, the authors propose an intrusion detection system (IDS) based on a deep learning model using LSTM. They design a network traffic feature extraction method and train an LSTM model to classify network traffic as normal or malicious. The performance of the proposed system is evaluated using real-world network datasets, and the results show its effectiveness in detecting different types of attacks [36].

This paper presents a network intrusion detection system (NIDS) based on deep learning with LSTM-recursive neural network (LSTM-RNN). The authors propose an architecture that integrates LSTM and RNN to classify network traffic into normal or intrusive categories. The model is trained using network traffic data and achieves good accuracy in identifying intrusions. The study highlights the potential of LSTM-RNN models in improving NIDS performance [37].

This paper presents a network intrusion detection system (NIDS) using deep learning techniques. The authors propose a model that combines convolutional neural networks (CNN) and LSTM to analyze network traffic and detect intrusions. The proposed model achieves high accuracy in detecting different types of attacks. The study

demonstrates the effectiveness of deep learning techniques in enhancing the performance [38].

This survey paper provides a comprehensive overview of deep learning techniques for intrusion detection systems (IDS). The authors present a taxonomy of deep learning-based IDS and discuss various models and algorithms, including LSTM. They analyze the strengths and weaknesses of deep learning approaches and highlight future research directions. The survey serves as a valuable resource for understanding the state of the art in deep learning for IDS [39].

This comprehensive review paper focuses on deep learning-based intrusion detection systems (IDS). The authors provide an in-depth analysis of various deep learning techniques used in IDS, including LSTM. They discuss the architecture, training, and evaluation of deep learning models for intrusion detection. The review paper also discusses the challenges and future directions in the field of deep learning-based IDS [40].

On the other hand, early DL approaches required a high number of training parameters as all neighbouring layers are completely coupled. When more parameters are employed, the training process may be slowed down, and the compute cost of the detection model can increase.

3. Background Study

3.1 **Convolutional Neural Networks (CNN)**

CNN is an example of a robustness method that is often used in the field of computer vision. It is part of the .

supervised learning approach. By adding the weight-sharing idea, CNN tackled the parameter explosion issue, which allowed it to accelerate the training process significantly. It has been used in various applications, including the restructuring of faces and pre-processing images. The convolution layer, the pooling layer, and the fully connected layer are the three basic components that make up the architecture of CNN. These layers are the convolution layer, the pooling layer, and the fully connected layer. The CNN organizational structure is laid out understandably in figure 1, which may be seen here. The convolution layer uses the linear operation, which sends the layer's output before it through a filter (kernel) of a certain size. This makes it possible to do the linear operation. The not-in-a-line CNN uses the Relue activation function more than any other. This function may increase the degree of non-linearity, and the feature map's negative values can be reset to their default values of zero. The CNN [2] may have more than one convolution layer. In this architecture, the main convolution layer is used nearly solely to extract the basic characteristics, such as edges and corners.

In contrast, the following layers extract the more complicated features. It is feasible for the pooling layer to minimize the feature dimensions, and as a consequence, it might potentially lessen the amount of necessary processing work. The layer that comes last and has all of its connections completed is the one that will be classified according to what it contains

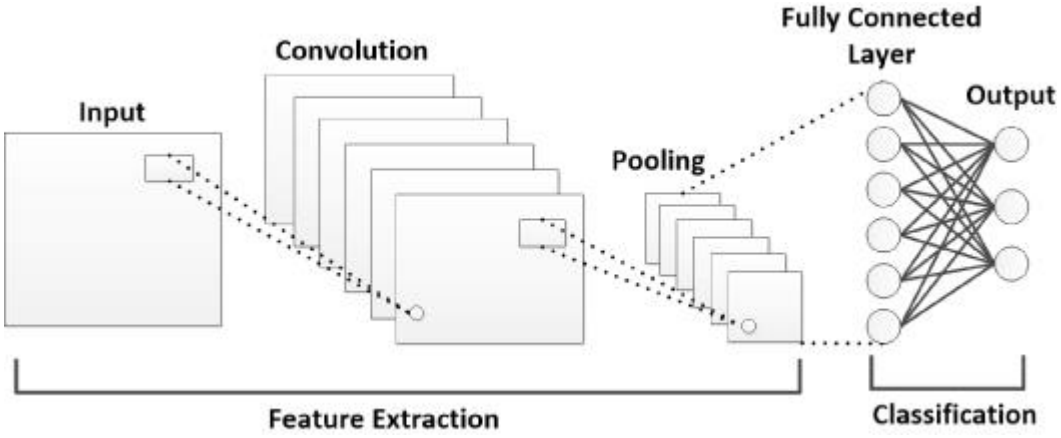

Fig 1: Displays a topology that is fairly typical of the CNN network.

3.2 Long short-term memory (LSTM)

The Long Short-Term Memory (LSTM) algorithm is a subtype of the recurrent neural network (RNN) that is widely used for processing time-series data. The output of every given layer in a typical RNN relies not only on the input it receives at that moment but also on the output of the layer that came before it. Nevertheless, the issue of vanishing gradients is one of the primary obstacles that must be overcome while training a typical RNN. As can be seen in Equation 1, gradients are used in the process of updating the weight values of a neural network. However, when a gradient value becomes exceedingly low as it back propagates over time, the amount of learning it contributes may no longer be considered significant. The RNN is plagued by problems caused by modest gradient

adjustments, particularly in its earliest layers. Because of this, it can't remember what it needs to play back lengthier sequences.

New weight=weight−learning rate∗gradient (1)

Since LSTM [19] was developed to address the problem of a diminishing gradient, this neural network architecture is best suited for long-time step pre-processing. To remember just the most important and useful details, the LSTM algorithm comes up with the concept of internal loops. Each LSTM cell is equipped with three distinct gates—the forget gate, the input gate, and the output gate—to control the flow of information through the cell. If the reader is interested in further information about the computing method of the LSTM, they may look it up in [20].

Specifically, we use the LSTM in this study because temporal correlation in network traffic is a driving factor in the evolution of time-series data [21]. In addition, the capability of CNN to learn spatial information is one factor that has led to the remarkable accomplishments that it has attained in the area of image processing. Integrating the CNN and LSTM can considerably increase the intrusion detection system's accuracy by extracting the raw data's spatial and temporal features. This is because both algorithms are quite effective at recognizing patterns, which is why this is the case.

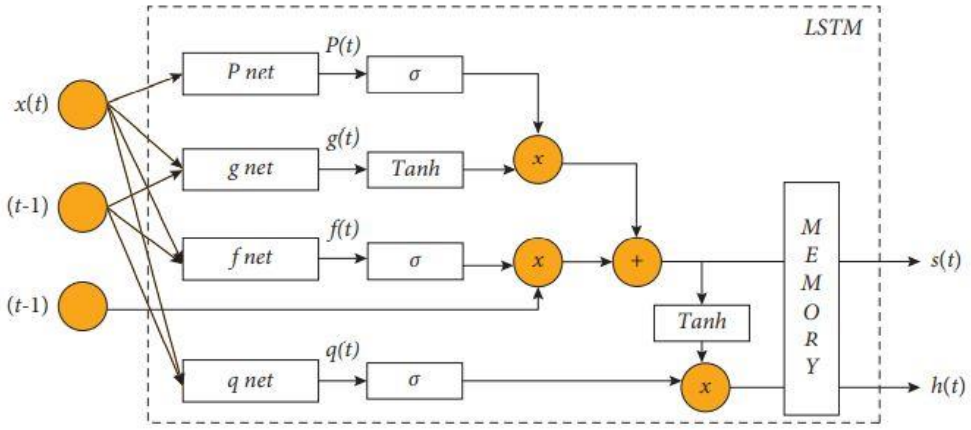

Fig 2: An example of the LSTM's usual construction.

3.3 Overfitting and Regularization

An important problem that may occur in neural networks is referred to as "overfitting" [22]. This problem often manifests itself when the model achieves excellent results on the training data but has extremely poor intuition when it is put to the test. This occurs when the model is trained on the specifics and noise, such as outliers of the training data, but then used to evaluate fresh data points and fails to generalize the situation. A few different approaches may be used to address the issue of overfitting. Increasing the quantity of training data is a frequent strategy for avoiding overfitting. Since an excessive amount of data prevents the model from overfitting any samples, it is forced to generalize to provide reliable outcomes. This approach, on the other hand, is believed to be costly and is limited in terms of the availability of datasets; it might be a substantial obstacle. This limitation is particularly problematic for network traffic. The traffic on the network may include information on customers, and the availability of such datasets may expose personally identifiable information to the general public. Utilizing regularization strategies to implement the additional penalty for the higher weights is an alternate method. As a result, it has the potential to simplify the model. The neural network model may be optimized using many different regularization approaches, such as L1 and L2, to promote lower weights. In addition to the regularization approaches discussed above, dropout is another strategy often used to mitigate the potential for overfitting. The dropout phase involves temporarily disregarding part of the network's neurons at random with probability P, but these neurons are later reinstated for use in the testing phase.

4. Proposed Work

4.1 Building our model

Here we present a high-level overview of the structure of our hybrid model, which analyses and classifies network data in space and time. The model's overarching network structure is seen in Figure 3. The design was developed in two stages.

The first step involves using a convolutional neural network to extract spatial information. Its two convolution layers use 32-bit and 64-bit output dimensions. Both convolution layers employ the same kernel, also with a size of 3x3. Dimensionality reduction is achieved by inserting a Max-pooling layer of size 2x2 after each convolution layer.

The information gathered in the CNN stage is then passed on to the second phase, which consists of three layers: an LSTM layer, a fully connected layer, and an output layer. The output layer is in charge of making everything come together at the end. The fully connected layer and the long short-term memory (LSTM) consist of 128 nodes. In the end, the softmax layer is employed for classification purposes at the output. This layer's purpose is to depict each input stream's probability accurately.

To mitigate overfitting and improve the detection model's performance on unseen data, we used the L2Reg.L2Reg. and dropout techniques. In order to choose the most effective regularization hyper-parameter and the probability values for the dropout procedure, we carried out a number of trials. The dropout is implemented using a probability P of 0.25 after the L2Reg and a probability P of 0.5 after the fully connected layer. L2Reg. is set to 0.1. Because there is a possibility that dropouts may lead to the loss of some information contained within the learning model, we start with a low value for the dropout probability and then gradually boost it to a higher value. This is done because we want to prevent the loss from being passed on to subsequent layers as much as possible. We can only give a binary classification for this inquiry, which indicates that the softmax output falls into either the normal class or the attack class.

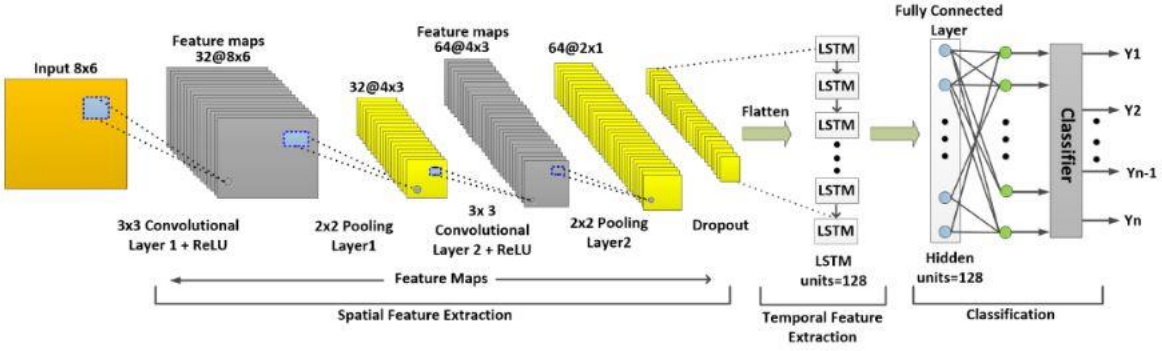

Fig. 3: The proposed architecture of CNN-LSTM

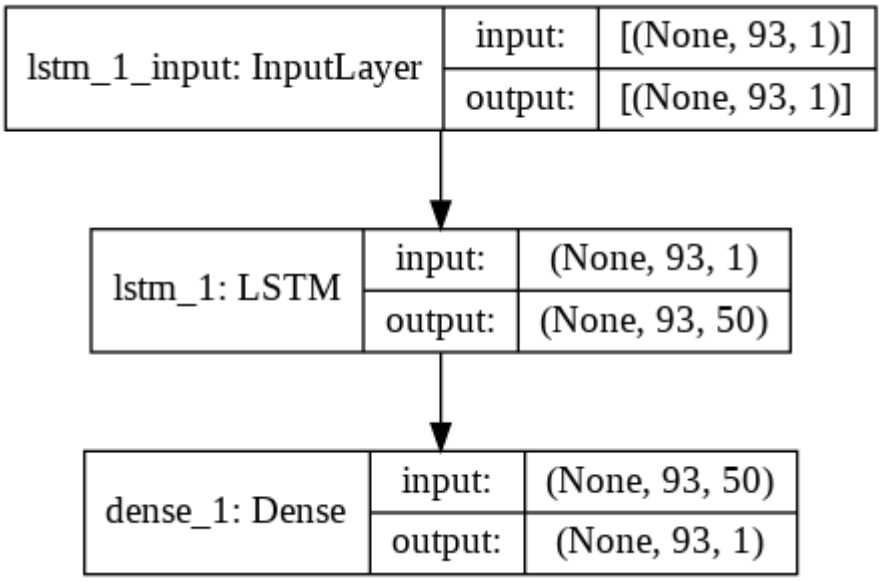

Fig 4: The proposed architecture of model layers

5. Implementation

5.1 Experimental Setup

Table 1: Proposed experimental setup

S.No.	Configuration	Unit
1	Number of CPU cores	16
2	Graphics card	RTX400

3	RAM	64G
4	Disk capacity	1 T
5	Operating System	Linux
6	Keras	2.10
7	Tensorflow	2.10

5.2 Dataset

Table 2: Proposed used dataset

		Dataset			
		KDDCup99 [23]		NSL-KDD [24]	
Attack Category	Description	Train	Test	Train	Test
Normal	Normal connection records	99,548	55321	67343	9710
DoS	The attacker aims to make network resources down	265471	132458	45927	7458
Probe	Obtaining detailed statistics of system and network configuration details	5214	1365	11656	2422
R2L	Illegal access from the remote computer	12365	2684	995	2887
U2R	Obtaining the root or super-user access on a particular computer	2845	658	3058	1028
Total		3,85,443	1,92,486	1,28,979	23,505

There aren't many examples of U2R and R2L attacks in intrusion datasets, but there are a lot of DoS, Probe, and Normal attacks. Because of this, even if the proposed model gets the class distribution of the minority population wrong, the total accuracy is still more than 99%. Most researchers are content with their findings, even though the classification error for unbalanced data is greater.

5.3 Dataset Preparation

All of the incoming information is pre-processed in this manner before being fed into the proposed model:

1. Regularization is a way to plan the values of features between 0 and 1 by the standardization technique (Z-score normalization) and equation 1.

$$x(i) = \frac{x(i) - mean\,(x(i))}{standard\_deviation(x(i))} \qquad (1)$$

Where, i ? [1, 48]

2. Because of the parameter-sharing notion, we decided to employ a 2D-CNN for our technique rather than a 1D-CNN. This allowed us to lower the number of weights included inside a convolution layer. Since network data is not inherently visual, an extra transformation into an image structure is first required. This step involves converting the network data. To satisfy the CNN's input criteria, the 1D-dimensional 48 web-based network traffic characteristics have been converted into an image format with a measurement of 8 by 6.
3. The symbolic qualities are translated into their corresponding numerical values. In addition to the standard class, the label column includes numerous more assault classes. Additionally, it is essential to consider that this research used a binary categorization. As a result, we assigned the value 0 to the normal label and gave the value 1 to each attack label.

5.4 Result

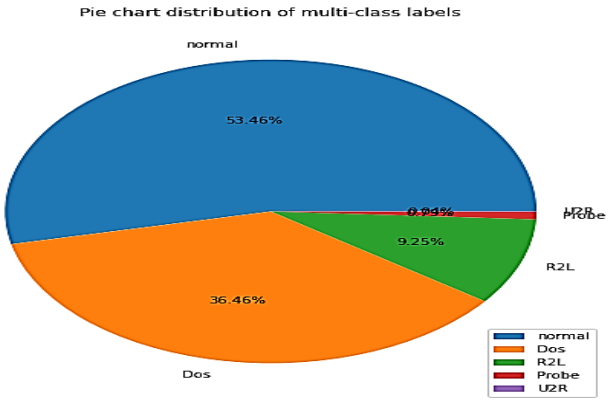

Fig. 5: Pie chart distribution of multi-class labels

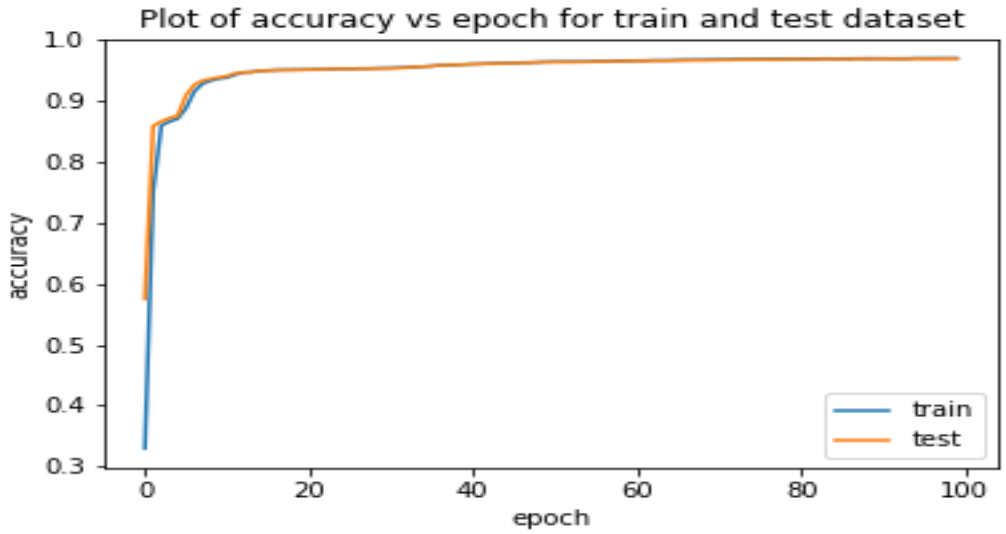

Fig. 6: Plot of accuracy vs epoch for training and test dataset

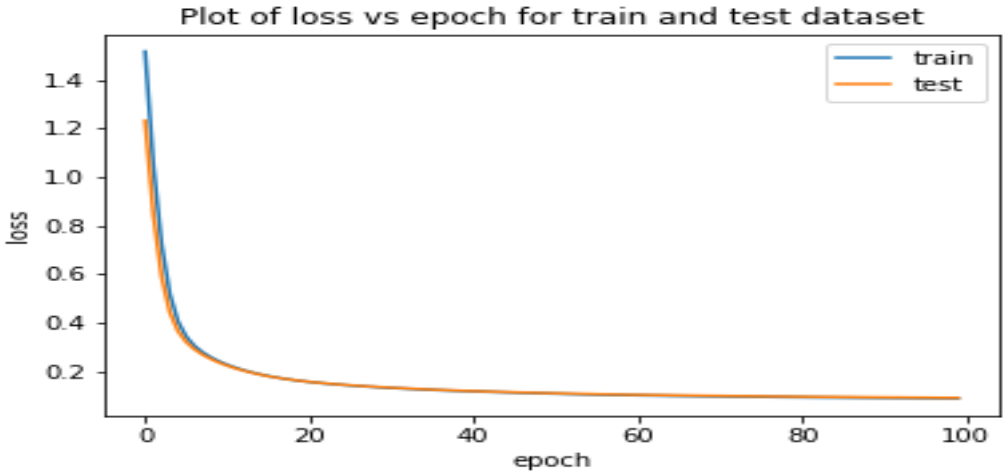

Fig. 7: Plot of loss vs epoch for training and testing dataset

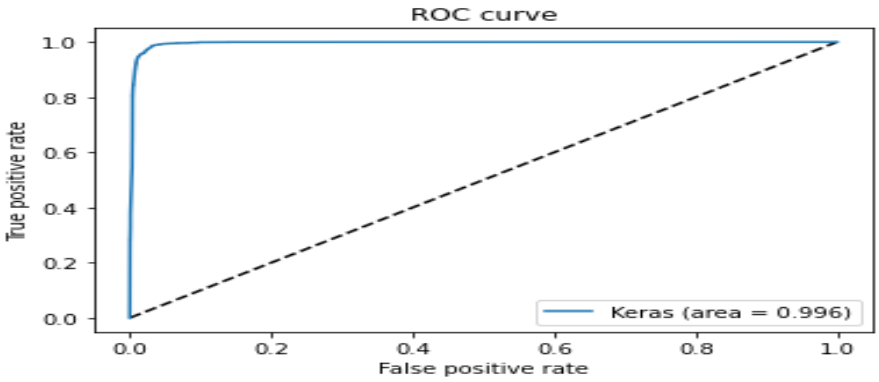

Fig. 8: Plot of ROC curve

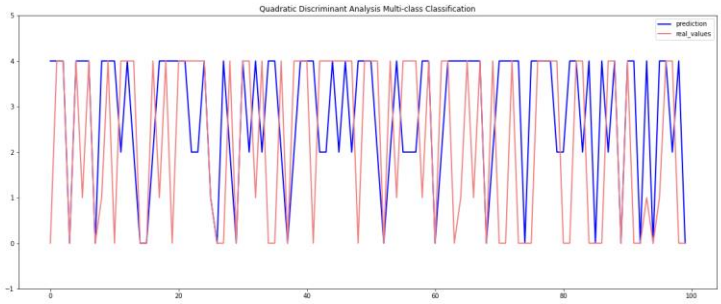

Fig 9: Quadratic discriminant analysis multi-class classification

Table 3. Detailed test results for Multi-class classification- KDDCup 99.

Method	Normal			DoS			Probe		
	TPR (%)	FPR (%)	Acc (%)	TPR (%)	FPR (%)	Acc (%)	TPR (%)	FPR (%)	Acc (%)
LR [25]	98.54	24.81	81.1	96.84	26.17	84.36	0.1	0.7	97.36
NB [25]	52.61	12.25	91.7	62.35	13.62	88.31	0.3	0.5	98.1
KNN [25]	28.31	28.64	65.25	32.74	36.82	62.58	1.5	1.3	97.6
DT [25]	29.17	27.31	65.25	33.95	29.45	63.78	1.8	1.5	97.5
AB [25]	93.74	93.42	26.84	89.74	94.38	27.98	1.5	1.2	97.8
RF [25]	28.65	33.74	66.38	31.95	34.72	63.78	0.7	1.6	97.48
SVM-rbf [25]	32.54	35.84	63.42	36.71	36.98	62.78	78.58	1.9	98.4
DNN1 layer [25]	99.42	8.8	94.35	99.31	4.35	96.78	77.98	0.3	99.78
DNN 2 layer [25]	99.61	8.8	95.28	99.54	6.45	97.36	69.28	0.2	99.34
DNN 3 layer [25]	99.84	6.8	97.84	99.73	5.42	98.25	78.34	0.3	99.74
DNN 4 layer [25]	94.38	6.7	95.87	94.86	8.34	95.78	79.45	0.3	99.58
DNN 5 layer [25]	99.93	8.6	93.65	99.92	8.72	96.48	77.32	0.2	99.37
TFTC[29]	98.15	6.3	94.85	98.95	7.31	98.25	79.85	0.8	98.45
MDPCA-DBNs[27]	98.32	5.8	95.35	98.99	8.32	97.46	81.24	0.5	98.98
Siam-IDS[28]	98.46	4.6	95.18	99.12	9.45	98.34	81.65	0.6	98.36

I-ELM[29]	98.21	5.9	96.84	99.32	11.36	97.64	81.49	0.4	99.12
HFR-MLR[30]	98.47	7.6	95.84	99.14	12.5	94.28	81.95	0.8	99.03
Proposed	99.98	2.3	98.45	99.96	4.5	99.84	84.25	0.1	99.88

Method	R2L			U2R		
	TPR (%)	FPR (%)	Acc (%)	TPR (%)	FPR (%)	Acc (%)
LR [25]	1.6	0.5	98.42	8.7	12.35	98.35
NB [25]	51.63	1.4	98.85	0.4	13.25	98.21
KNN [25]	0.6	0.6	97.26	0.3	14.27	95.26
DT [25]	0.4	0.3	98.46	0.4	12.56	94.27
AB [25]	0.2	0.4	98.37	0.6	13.87	93.85
RF [25]	0.8	0.2	99.34	0.1	24.3	97.25
SVM-rbf [25]	0.7	0.2	99.21	19.27	0.9	97.58
DNN1 layer [25]	27.1	0.1	99.98	11.35	24.68	97.38
DNN 2 layer [25]	27.2	0.1	99.98	0.2	9.24	99.48
DNN 3 layer [25]	0.3	0.1	99.98	0.7	0.1	99.85
DNN 4 layer [25]	0.2	0.1	99.98	0.6	0.1	99.12
DNN 5 layer [25]	5.3	0.1	99.98	14.65	0.3	98.25
TFTC[29]	4.9	0.1	98.62	13.85	0.2	98.14
MDPCA-DBNs[27]	9.7	0.1	98.24	15.36	0.2	98.35
Siam-IDS[28]	8.5	0.1	97.85	14.85	0.3	98.61
I-ELM[29]	6.9	0.1	97.36	14.29	0.2	98.75
HFR-MLR[30]	5.3	0.1	98.25	13.91	0.2	97.06
Proposed	2.3	0.01	99.99	11.02	0.01	99.89

Table 2. Detailed test results for Multi-class classification-NSL-KDD

Method	Normal			DoS			Probe		
	TPR (%)	FPR (%)	Acc (%)	TPR (%)	FPR (%)	Acc (%)	TPR (%)	FPR (%)	Acc (%)
LR [25]	92.34	51.84	69.42	64.85	32.98	79.65	25.36	12.62	91.21
NB [25]	4.68	8.56	59.85	88.34	45.62	42.65	32.54	13.85	87.54
KNN [25]	98.64	42.75	79.25	77.85	89.37	89.84	62.35	13.78	89.32
DT [25]	98.24	38.64	79.14	79.82	5.32	92.84	66.95	12.65	91.24
AB [25]	68.42	28.95	71.36	87.3	5.62	76.39	35.62	12.41	91.35
RF [25]	98.48	42.85	77.58	78.94	32.45	91.25	66.32	13.52	90.25
SVM-rbf [25]	99.87	53.46	74.84	66.87	11.65	89.25	52.36	12.65	90.35
DNN1 layer [25]	97.7	32.84	91.41	79.85	2.6	92.65	63.25	5.21	90.74

DNN 2 layer [25]	97.6	33.45	91.65	78.54	1.8	92.35	74.25	6.95	92.36
DNN 3 layer [25]	97.9	32.41	91.21	77.64	1.4	93.45	66.32	4.25	92.45
DNN 4 layer [25]	97.86	33.56	87.68	77.98	1.6	93.65	65.25	6.32	92.85
DNN 5 layer [25]	97.88	31.25	88.56	77.85	2.5	94.01	66.85	5.62	92.74
TFTC[29]	98.54	36.56	91.45	88.62	6.3	92.52	66.32	6.3	91.36
MDPCA-DBNs[27]	97.85	35.15	90.15	89.41	4.2	94.68	68.45	5.3	92.5
Siam-IDS[28]	96.35	38.45	89.48	84.35	3.5	93.27	69.15	4.5	93.56
I-ELM[29]	98.45	34.52	91.63	82.65	6.5	93.54	68.26	4.8	94.75
HFR-MLR[30]	96.35	36.25	93.45	81.45	4.5	94.35	64.12	5.3	93.65
Proposed	99.56	21.01	98.45	98.45	1.2	99.45	71.25	4.15	96.48

Method	R2L			U2R		
	TPR (%)	FPR (%)	Acc (%)	TPR (%)	FPR (%)	Acc (%)
LR [25]	5.62	8.4	97.25	5.8	9.1	88.24
NB [25]	18.65	6.4	97.14	6.3	4.6	89.62
KNN [25]	14.62	4.3	97.52	4.5	8.7	91.25
DT [25]	45.62	9.5	97.65	9.6	5.6	92.48
AB [25]	46.21	10.5	96.25	21.54	4.6	93.48
RF [25]	56.29	6.1	96.34	12.65	9.5	91.21
SVM-rbf [25]	9.65	50.2	97.15	9.48	11	91.45
DNN1 layer [25]	49.71	0.1	99.45	21.65	1.2	98.45
DNN 2 layer [25]	26.35	0.1	99.25	23.48	2.6	96.48
DNN 3 layer [25]	32.65	0.1	99.62	21.41	0.2	97.85
DNN 4 layer [25]	36.85	0.6	99.84	23.56	0.4	98.35
DNN 5 layer [25]	31.56	0.5	99.31	24.15	0.6	98.45
TFTC[29]	23.54	0.5	98.34	21.14	0.4	98.15
MDPCA-DBNs[27]	26.45	0.3	98.42	19.54	0.7	98.63
Siam-IDS[28]	29.52	0.4	98.64	19.35	0.4	98.77
I-ELM[29]	31.42	0.6	97.68	23.45	0.8	98.25
HFR-MLR[30]	32.55	0.5	98.45	24.63	0.4	98.34
Proposed	36.45	0.1	99.85	89.45	0.2	99.95

6. Conclusion

Intrusion Detection Systems (IDS) that make use of architectures such as Long Short-Term Memory (LSTM) and Fully Connected Neural Network (FCNN) have showed some promise in terms of enhancing the accuracy and efficacy of detecting intrusions in a wide variety of network settings. Deep learning strategies, such as LSTM and FCNN, have been shown to have the potential to improve the capabilities of intrusion detection systems (IDS), according to research studies and surveys. These models take use of the capacity of deep learning algorithms to automatically learn and extract complicated patterns from network traffic data. As a result, they are able to recognize both known and undiscovered intrusion pattern variants. The combination of LSTM and FCNN enables excellent modeling of sequential dependencies in

network traffic data, capturing both short-term and long-term dependencies as well as temporal patterns that may be suggestive of harmful actions. This makes it possible to effectively detect and prevent malicious activity. While FCNN has powerful classification skills, LSTM is best suited for remembering and storing long-term dependencies. LSTM also excels at capturing these relationships. According to the research that has been conducted, intrusion detection systems (IDS) based on LSTM and FCNN architectures have shown better detection accuracy and resilience when compared to conventional rule-based or signature-based techniques. These IDS models that are based on deep learning have shown that they are able to identify a wide variety of attacks, such as network intrusions, botnet activity, and abnormalities in IoT networks.

Intrusion Detection Systems (IDS) with Long Short-Term Memory (LSTM) and Fully Connected Neural Network (FCNN) architectures have a very bright future ahead of them in terms of their potential application breadth. The following are some prospective areas that might benefit from more investigation and development:

- Enhanced Feature Representation: In LSTM-FCNN-based intrusion detection systems, the investigation of innovative strategies for feature representation may lead to improved overall performance. Exploring the various representations of network traffic, such as raw packet-level data, flow-level data, or higher-level protocol-based characteristics, may reveal a wealth of additional information that can be used for intrusion detection.
- Hybrid Models: Integrating LSTM and FCNN architectures with other deep learning models, such as Convolutional Neural Networks (CNN) or Generative Adversarial Networks (GAN), may harness the benefits of each of the models individually and further improve the accuracy and resilience of IDS. It is possible for hybrid models to integrate the benefits of many distinct architectural styles in order to analyze network traffic data in a way that takes into account both geographical and temporal trends.
- Online Education and Adaptive Systems It is essential to develop intrusion detection systems (IDS) that are able to continually adapt to changing network threats and dynamically update their models. Investigating online learning strategies that enable the intrusion detection system (IDS) to adapt in real time to shifting attack patterns and circumstances on the network is one way to greatly increase the efficacy of the system.
- Transfer Learning and Domain Adaptation: Moving information and models from one network environment to another may be useful in many situations, particularly when there is a scarcity of labeled training data or when the data is domain-specific. Investigating transfer learning and domain adaption approaches might be helpful in improving the detection capabilities of IDS models and generalizing them across a variety of network environments.
- Explainability and Interpretability: It is vital to address the difficulty of understanding the decisions produced by LSTM-FCNN models, since their complicated structures might make it difficult to grasp how and why a specific decision was achieved. It is also essential to explain how and why the choice was obtained. Increasing the openness of these models and making it easier to trust the results they provide may be accomplished by developing methods that explain the decision-making process that they use.
- Detection of Adversarial attacks Conducting research on the susceptibility of LSTM-FCNN-based intrusion detection systems (IDS) to adversarial attacks and creating countermeasures will improve these systems' resistance to evasion strategies. Research into adversarial training, robust optimization, and techniques of anomaly detection might help the system become better able to identify complex attacks.
- Integration with Other Security Solutions: Investigating the integration of LSTM-FCNN-based IDS with other security solutions, such as firewall systems, anomaly detection frameworks, or security information and event management (SIEM) systems, can provide a comprehensive security infrastructure and enable proactive defense mechanisms. This can be accomplished by integrating the LSTM-FCNN-based IDS with other security solutions.

References

[1] R. Javed, S. u. Rehman, M. U. Khan, M. Alazab and T. R. G, "CANintelliIDS: Detecting In-Vehicle Intrusion Attacks on a Controller Area Network Using CNN and Attention-Based GRU," in IEEE Transactions on Network Science and Engineering, vol. 8, no. 2, pp. 1456-1466, 1 April-June 2021, doi: 10.1109/TNSE.2021.3059881.

[2] Deore and S. Bhosale, "Hybrid Optimization Enabled Robust CNN-LSTM Technique for Network Intrusion Detection," in IEEE Access, vol. 10, pp. 65611-65622, 2022, doi: 10.1109/ACCESS.2022.3183213.

[3] T. Wisanwanichthan and M. Thammawichai, "A Double-Layered Hybrid Approach for Network Intrusion Detection System Using Combined Naive Bayes and SVM," in IEEE Access, vol. 9, pp. 138432-138450, 2021, doi: 10.1109/ACCESS.2021.3118573.

[4] S. Ho, S. A. Jufout, K. Dajani and M. Mozumdar, "A Novel Intrusion Detection Model for Detecting Known and Innovative Cyberattacks Using Convolutional Neural Network," in IEEE Open

Journal of the Computer Society, vol. 2, pp. 14-25, 2021, doi: 10.1109/OJCS.2021.3050917.

[5] A. Halbouni, T. S. Gunawan, M. H. Habaebi, M. Halbouni, M. Kartiwi and R. Ahmad, "CNN-LSTM: Hybrid Deep Neural Network for Network Intrusion Detection System," in IEEE Access, vol. 10, pp. 99837-99849, 2022, doi: 10.1109/ACCESS.2022.3206425.

[6] Z. Wang, Y. Zeng, Y. Liu and D. Li, "Deep Belief Network Integrating Improved Kernel-Based Extreme Learning Machine for Network Intrusion Detection," in IEEE Access, vol. 9, pp. 16062-16091, 2021, doi: 10.1109/ACCESS.2021.3051074.

[7] J. Yoo, B. Min, S. Kim, D. Shin and D. Shin, "Study on Network Intrusion Detection Method Using Discrete Pre-Processing Method and Convolution Neural Network," in IEEE Access, vol. 9, pp. 142348-142361, 2021, doi: 10.1109/ACCESS.2021.3120839.

[8] J. Alikhanov, R. Jang, M. Abuhamad, D. Mohaisen, D. Nyang and Y. Noh, "Investigating the Effect of Traffic Sampling on Machine Learning-Based Network Intrusion Detection Approaches," in IEEE Access, vol. 10, pp. 5801-5823, 2022, doi: 10.1109/ACCESS.2021.3137318.

[9] M. E. Aminanto, R. S. H. Wicaksono, A. E. Aminanto, H. C. Tanuwidjaja, L. Yola and K. Kim, "Multi-Class Intrusion Detection Using Two-Channel Color Mapping in IEEE 802.11 Wireless Network," in IEEE Access, vol. 10, pp. 36791-36801, 2022, doi: 10.1109/ACCESS.2022.3164104.

[10] H. Sun, M. Chen, J. Weng, Z. Liu and G. Geng, "Anomaly Detection for In-Vehicle Network Using CNN-LSTM With Attention Mechanism," in IEEE Transactions on Vehicular Technology, vol. 70, no. 10, pp. 10880-10893, Oct. 2021, doi: 10.1109/TVT.2021.3106940.

[11] Liu, Z. Gu and J. Wang, "A Hybrid Intrusion Detection System Based on Scalable K-Means+ Random Forest and Deep Learning," in IEEE Access, vol. 9, pp. 75729-75740, 2021, doi: 10.1109/ACCESS.2021.3082147.

[12] S. Lei, C. Xia, Z. Li, X. Li and T. Wang, "HNN: A Novel Model to Study the Intrusion Detection Based on Multi-Feature Correlation and Temporal-Spatial Analysis," in IEEE Transactions on Network Science and Engineering, vol. 8, no. 4, pp. 3257-3274, 1 Oct.-Dec. 2021, doi: 10.1109/TNSE.2021.3109644.

[13] Z. Hu, L. Wang, L. Qi, Y. Li and W. Yang, "A Novel Wireless Network Intrusion Detection Method Based on Adaptive Synthetic Sampling and an Improved Convolutional Neural Network," in IEEE Access, vol. 8, pp. 195741-195751, 2020, doi: 10.1109/ACCESS.2020.3034015.

[14] Y. Xiao, C. Xing, T. Zhang and Z. Zhao, "An Intrusion Detection Model Based on Feature Reduction and Convolutional Neural Networks," in IEEE Access, vol. 7, pp. 42210-42219, 2019, doi: 10.1109/ACCESS.2019.2904620.

[15] K. Jiang, W. Wang, A. Wang and H. Wu, "Network Intrusion Detection Combined Hybrid Sampling With Deep Hierarchical Network," in IEEE Access, vol. 8, pp. 32464-32476, 2020, doi: 10.1109/ACCESS.2020.2973730.

[16] K. Wu, Z. Chen and W. Li, "A Novel Intrusion Detection Model for a Massive Network Using Convolutional Neural Networks," in IEEE Access, vol. 6, pp. 50850-50859, 2018, doi: 10.1109/ACCESS.2018.2868993.

[17] Y. Yu and N. Bian, "An Intrusion Detection Method Using Few-Shot Learning," in IEEE Access, vol. 8, pp. 49730-49740, 2020, doi: 10.1109/ACCESS.2020.2980136.

[18] X. Xie, B. Wang, T. Wan and W. Tang, "Multivariate Abnormal Detection for Industrial Control Systems Using 1D CNN and GRU," in IEEE Access, vol. 8, pp. 88348-88359, 2020, doi: 10.1109/ACCESS.2020.2993335.

[19] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. Neural computation 9, 8 (1997), 1735–1780

[20] Zhengmin Kong, Yande Cui, Zhou Xia, and He Lv. 2019. Convolution and long short-term memory hybrid deep neural networks for remaining useful life prognostics. Applied Sciences 9, 19 (2019), 4156.

[21] Tuan A Tang, Lotfi Mhamdi, Des McLernon, Syed Ali Raza Zaidi, and Mounir Ghogho. 2018. Deep recurrent neural network for intrusion detection in sdn-based networks. In 2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft). IEEE, 202–206.

[22] J. Kolluri, V. K. Kotte, M. S. B. Phridviraj and S. Razia, "Reducing Overfitting Problem in Machine Learning Using Novel L1/4 Regularization Method," 2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184), 2020, pp. 934-938, doi: 10.1109/ICOEI48184.2020.9142992.

[23] https://kdd.org/kdd-cup

[24] https://www.unb.ca/cic/datasets/nsl.html

[25] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," IEEE Access, vol. 7, pp. 41525–41550, Apr. 2019.

[26] H. H. Pajouh, R. Javidan, R. Khayami, A. Dehghantanha, and K.-K.-R. Choo, "A two-layer dimension reduction and two-tier classification model for anomaly-based intrusion detection in IoT backbone networks," IEEE Trans. Emerg. Topics Comput., vol. 7, no. 2, pp. 314–323, Apr. 2019.

[27] Y. Yang, K. Zheng, C. Wu, X. Niu, and Y. Yang, "Building an effective intrusion detection system using the modified density peak clustering algorithm and deep belief networks," Appl. Sci., vol. 9, no. 2, pp. 238–262, Jan. 2019.

[28] P. Bedi, N. Gupta, and V. Jindal, "Siam-IDS: Handling class imbalance problem in intrusion detection systems using siamese neural network," Procedia Comput. Sci., vol. 171, pp. 780–789, Jan. 2020.

[29] H. H. Pajouh, G. Dastghaibyfard, and S. Hashemi, "Two-tier network anomaly detection model: A machine learning approach," J. Intell. Inf. Syst., vol. 48, no. 1, pp. 61–74, Feb. 2017.

[30] Y. N. Kunang, S. Nurmaini, D. Stiawan, and B. Y. Suprapto, "Attack classification of an intrusion detection system using deep learning and hyperparameter optimisation," J. Inf. Secur. Appl., vol. 58, pp. 102804–102818, Mar. 2021.

[31] Bojan Mrazovac, Siniša Ranđić, Dragan Pamučar, and Dušan Vučković. (2020). "Intrusion Detection in IoT Network using LSTM-based Deep Learning Model." Expert Systems with Applications.

[32] Fan Wu, Xinyu Yang, Youwen Yi, Fei Xiao, Xingang Wan, and Xuansong Li. (2019). "An Intelligent Intrusion Detection System Based on Deep Learning with LSTM." IEEE Access.

[33] Hani Alzaid, Basheer Al-Duwairi, and Essam Al-Daoud. (2020). "Deep Learning Approaches for Intrusion Detection Systems: A Survey." Journal of Information Security and Applications.

[34] Shivam Sharma, Samir Garg, Vijay Laxmi, and Manoj Singh Gaur. (2018). "LSTM-Based Botnet Detection Model for Internet of Things." International Journal of Distributed Sensor Networks.

[35] S. Alsulaiman and H. M. Alzaidi. (2018). "Anomaly-based Network Intrusion Detection System using Deep Learning Techniques." International Journal of Advanced Computer Science and Applications.

[36] Li Liu, Xiaoling Huang, Haoyang Lu, and Xingpeng Jiang. (2019). "Intrusion Detection System Based on Deep Learning with LSTM Model." IEEE International Conference on Communication Software and Networks.

[37] Lusia Fitri, Masayu Leylia Khodra, and Husni Thamrin. (2019). "Network Intrusion Detection System Based on Deep Learning with LSTM-RNN." International Journal of Computer Science and Information Security.

[38] Md Rafsan Nahar and Rabindra Kumar Barik. (2019). "Network Intrusion Detection System using Deep Learning Technique." International Journal of Advanced Computer Science and Applications.

[39] Shanmugam, S. P. ., Vadivu, M. S. ., Anitha, D., Varun, M., & Saranya, N. N. . (2023). A Internet of Things Improvng Deep Neural Network Based Particle Swarm Optimization Computation Prediction Approach for Healthcare System. International Journal on Recent and Innovation Trends in Computing and Communication, 11(4s), 92–99. https://doi.org/10.17762/ijritcc.v11i4s.6311

[40] Kshirsagar, P. R., Yadav, R. K., Patil, N. N., & Makarand L, M. (2022). Intrusion Detection System Attack Detection and Classification Model with Feed-Forward LSTM Gate in Conventional Dataset. Machine Learning Applications in Engineering Education and Management, 2(1), 20–29. Retrieved from http://yashikajournals.com/index.php/mlaeem/article/view/21

[41] Verma, M. K., & Dhabliya, M. D. (2015). Design of Hand Motion Assist Robot for Rehabilitation Physiotherapy. International Journal of New Practices in Management and Engineering, 4(04), 07–11.

Mr.Ankit Chakrawarti working as Assistant Professor in Department of Computer Science and Engineering at Chameli Devi group of Institutions, Indore, MP. Prior to that he has more than 7 years of teaching experience. He is pursuing PhD in computer science and engineering. He has completed Master of Technology in Computer Science and Engineering, also done Bachelor of Engineering in Computer Science and Engineering. His research includes Computer Network, Network Security,

and Machine Learning. He is having various research publications in reputed, international and national journal, International-National conferences.

Dr. Shiv Shakti Shrivastava is the Professor in Computer Science department in Rabindranath Tagore University, Bhopal. He has more than 20 years of experience. Under his guidance nine scholars has been awarded Ph.D., Six are also taking guidance and M.Tech.-MCA students also guided by him. He has published more than fourth five International & National papers in difference reputed journals. He has three patents and also trying funded patents. He is connected with many universities like Barkatullah Univ., RGPV, Makhanlal Univ., Bhoj Univ., and many more universities. He has taken many expert lectures, guest lecture, judge in different universities and colleges for technical and other events. He has been attended as guest in different webinars. He has completed three Faculty Development Program and also Successfully Completed training of the IEEE Xplore Digital Library. Ph.D. Thesis reviewed in many other universities. He has taken responsibilities as observers in technical events, examination and cultural activities in others universities.