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

Machine Learning Based Intrusion Detection in IoT Network Using MLP and LSTM

Yogita Shewale[1], Dr. Shailesh Kumar[2], Dr. Satish Banait[3]

Submitted:26/03/2023 **Revised:**28/05/2023 **Accepted:**11/06/2023

Abstract. Each network's security design must include an intrusion detection system. Monitoring and analysing network traffic is its main purpose in order to spot and halt hazardous activities. Machine learning algorithms have shown considerable promise in the realm of intrusion detection systems due to their ability to learn from large and complex data sets (IDS). In intrusion detection systems, the Multilayer Perceptron (MLP) and Long Short-Term Memory (LSTM) classifiers are two of the more popular machine learning techniques (IDS). We conducted an investigation in which we compared the performance of MLP and LSTM classifiers for IDS using the CICDDoS2019 dataset. We utilised label encoding to perform some basic processing on the dataset before using feature selection to identify the most important features. Using the preprocessed dataset for training and testing, the MLP and LSTM classifiers' performance was assessed twice, and the results were compared in terms of accuracy and loss. The results of our study show that both classifiers were capable of reaching high accuracy with little loss, with the LSTM classifier doing just slightly better than the MLP classifier in terms of accuracy and loss. The results of this research can help security experts and researchers choose the machine learning algorithm for IDS that is most appropriate for them based on the particular requirements and criteria they have.

***Keyword**: Intrusion Detection System, IDS, Machine Learning, MLP, LSTM, CICDDoS2019, Network Security*

1. Introduction

IDS are security tools that identify and stop unauthorised access to networks and computer systems. IDS look at system and network activity to find anomalous patterns that might point to a security breach [1]. The two main classes of IDS are anomaly identification and signature identification. System activity and network traffic are compared to known attack patterns, or signatures, by signature identification systems to find possible threats. On the other hand, anomaly identification [2], use ML techniques to create a baseline of typical activity and identify variations from that baseline that might indicate an intrusion. Since they can spot security breaches that other security measures, like firewalls and antivirus software, might overlook, IDS are a vital component [3] of network and computer security systems. IDS can provide helpful information for forensic investigation and incident response in the case of a security problem [4].

Intrusion Detection Systems (IDS), which identify and stop unauthorized access to computer systems and networks, are crucial elements of computer and network security systems [5]. IDS use machine learning techniques to analyses network data and find unusual patterns of activity that might point to a security breach [6]. MLP is a common type of feed forward neural network used in IDS and other machine learning programmers. It is a supervised learning technique that can be used for applications in regression and classification [7]. MLPs consist of an input layer, an output layer, and one or more hidden layers. Neurons in every layer are coupled to those in the layers above them. On the other hand, LSTM is created specifically for handling sequential input. It is therefore ideally suited for time-series data analysis, which is common in IDS, because it can identify data propagation sequences in the input data. The unique architecture of an LSTM consists of memory cells and gates that control the data flow throughout the network [8]. Both MLP and LSTM have been effectively used in IDS, with various degrees of success depending on the dataset and attributes used. MLP performs better than LSTM in general for time-series data, although LSTM performs better in classification tasks [9]. In recent years, IDS has used more machine learning techniques, such as MLPs and LSTM networks. When analysing enormous amounts of data and spotting patterns of suspicious behaviour, these algorithms are more accurate than conventional signature-based methods.

2. Literature Review

The limitations of IDS and future research are also highlighted. Intruder detection systems employ recent techniques such as convolutional and recurrent neural

[1]*Department of Computer Engineering, Shri JJTU University, Rajasthan, India*
[2]*Department of Computer Science and Engineering, Gopalan College of Engineering and Management, Bangalore, India*
[3]*Department of Computer Engineering, K.K.Wagh Institute of Engineering Education & Research Nashik, Maharashtra, India*
yogitashewalenet.shewale@gmail.com1, shaileshkumar1610@gmail.com2, ssbanait@kkwagh.edu.in3

networks. It also discusses the advantages and disadvantages of various tactics as well as their practical applications. Intrusion detection system classification, techniques, metrics, and issues are covered in [11]. It covers data mining, machine learning, and other advanced technologies used in intrusion detection systems and makes recommendations for additional research. The types, strategies, and evaluation techniques [12], It also examines the flaws in these systems and makes recommendations for further study. This article discusses classification, methodologies, assessment metrics, and issues related to intrusion detection systems. It investigates how machine learning and data mining are used in intrusion detection systems and makes recommendations for additional research.

The principles of intrusion detection [13], shortcomings of these systems are also covered. In this article, intrusion detection techniques are categorised and assessed. It also examines the flaws in these systems and makes recommendations for further study. Systems for detecting intrusions also use deep learning and machine learning. It contrasts the advantages and disadvantages of each design as well as its applications for host-based, network-based, and hybrid intrusion detection systems [14, 15]. Future research on intrusion detection systems is also highlighted in the essay. The types, strategies, and evaluation techniques of intrusion detection systems are highlighted [16], along with the limitations of these systems and directions for future study. Systems for detecting intruders combine data mining and machine learning. The background, uses, and future directions of machine learning for networking are discussed in this article. The essay highlights the shortcomings of machine learning for networking.

The cyber security intrusion detection method[17] examines the algorithms, approaches, uses, and limitations of these techniques. The essay highlights data preprocessing and feature selection in these algorithms and makes recommendations for further study. The methodologies, systems, and problems related to anomaly-based network intrusion detection are covered in this article. It talks about system implementation flaws, anomalies, and how to detect them. The work addresses the benefits and drawbacks of each of the statistical, data mining, and machine learning anomaly detection methods[18] as well as their applications. The limitations of anomaly identification and directions for future study are also highlighted. Intrusion detection systems are categorised, assessed, and surveyed in this paper [19]. The limitations of intrusion detection systems and future research are also highlighted. The groundbreaking study on intrusion detection systems is frequently referenced in subsequent studies. Machine learning-based bot net identification techniques are discussed in this article [20]. It covers the advantages and disadvantages of machine learning botnet detection techniques. The essay also discusses upcoming research and the limitations of botnet detection using machine learning.

A novel ensemble-based intrusion detection system is presented for the classification of highly imbalanced network data in real-time. [21] To enhance intrusion detection, the system makes use of decision trees, random forests, and support vector machines. The article also assesses the system using various datasets and compares machine learning-based intrusion detection solutions. In the paper [22], anomaly detection techniques for computer networks are reviewed. It covers anomaly detection techniques and intrusion detection using statistics, machine learning, and deep learning. The limitations of anomaly identification and directions for future study are also highlighted. Deep learning is recommended for network intrusion detection in this paper [23]. Long short-term memory network identifies normal and abnormal traffic, while a convolutional neural network extracts features from network traffic data[24]. The paper [25] examines and assesses MlIDS systems utilising various datasets. An effective deep learning-based cloud intrusion detection method is presented in this work [26]. A deep belief network identifies typical and abnormal traffic, while a convolutional neural network pulls information from network traffic data. The paper [27] examines and assesses recent techniques utilising various datasets. A hybrid deep learning is used for intrusion detection in industrial control systems. LSTM and CNN classify normal and pathological network traffic. The study [29][30] examines and assesses machine learning-based intrusion detection systems utilising various datasets. An attention-based intrusion detection method is presented in the work [31]. Network traffic data is divided into normal and pathological conditions using an attention-based convolutional neural network. The study [32] [33] examines and assesses machine learning-based intrusion detection systems using various datasets.

Article Title and Year	Machine Learning Technique	Application Area	Dataset	Performance Evaluation
"An overview of intrusion detection systems: a machine learning perspective" (2022)	Various	General	Various	Review article
"Machine learning techniques for intrusion detection: a comprehensive review" (2022)	Various	General	Various	Review article
"An intrusion detection system using deep learning with a novel feature extraction mechanism" (2022)	Deep Learning (CNN and DBN)	Network Security	UNSW-NB15 and NSL-KDD	Detection rate is high with low FP rate
"Machine learning-based intrusion detection system: A review" (2022)	Various	General	Various	Review article
"A hybrid deep learning approach for intrusion detection in industrial control systems" (2021)	Deep Learning (CNN and LSTM)	Industrial Control Systems	Various	Detection rate is high with low FP rate
"A novel intrusion detection method based on attention mechanism and deep neural network" (2021)	Deep Learning (Attention-CNN)	Network Security	CICIDS2017 and UNSW-NB15	Detection rate is high with low FP rate
"LSTM-based intrusion detection system for the Internet of Things" (2021)	Deep Learning (LSTM)	IoT	CICIDS2017 and ISOT	Detection rate is high with low FP rate
"Intrusion detection system based on machine learning techniques: a systematic review" (2019)	Various	General	Various	Review article
"An efficient deep learning approach for intrusion detection system in cloud" (2019)	Deep Learning (CNN and DBN)	Cloud Computing	NSL-KDD and CICIDS2017	Detection rate is high with low FP rate
"Deep learning-based intrusion detection system for the Internet of Things" (2018)	Deep Learning (CNN and DNN)	IoT	Private dataset	Detection rate is high with low FP rate
"A novel intrusion detection model based on PCA and multiple kernel learning" (2018)	Kernel Methods	Network Security	UNSW-NB15	Detection rate is high with low FP rate

Table.1 Comparison on recent studies

3. Intrusion Detection System (Ids)

a. Multi-Layer Perceptron (MLP)

A multi-layer perceptron (MLP) is an artificial neural network with a feedforward architecture that has numerous layers of nodes or neurons. In a typical MLP architecture, we find an input layer, one or more hidden layers, and an output layer.

An MLP's input layer is where the input data, which can take the form of vectors, pictures, or any other type of data that can be numerically represented, is supplied. A feature or attribute of the input data is represented by each node in the input layer.

Using a network of weighted connections between the nodes in adjacent levels, the hidden layers of an MLP compute on the input data. Every node in a hidden layer is connected to every node in the layer above, and every connection has a weight assigned to it. The hidden layers add nonlinearity to the nodes' output by using activation functions.

The final output of the network is produced by an MLP's output layer. The output layer's node count is determined by the problem being solved. One node in the output layer, for instance, will produce a binary output for a binary classification task. There will be several nodes in the output layer for a multi-class classification issue, each of which corresponds to a class label.

During training, an MLP learns the weights of the connections between the nodes. During training, a set of input data and their corresponding output labels are sent to the network. To reduce the discrepancy between the expected output and the actual output, the network modifies the connection weights. Until the network's performance on a validation set reaches an acceptable level, this process is repeated over a number of epochs.

The overall MLP's configuration is created to learn intricate data patterns by employing numerous layers of processing. MLPs have been utilized successfully in a number of applications, including intrusion detection, natural language processing, and picture and audio recognition.

b. Long Short-Term Memory

LSTM need to be configured to works in the network for the IDS and the configuration are as under:

i. **Number of LSTM layers:** Depending on the complexity of the data and the desired level of accuracy, the number of LSTM layers in the model can be changed. The model may learn more intricate patterns in the data with the addition of additional LSTM layers, but there is a chance that it will become overfit.

ii. **No. of neurons in each LSTM layer:** To improve the performance of the model, the number of neurons in each LSTM layer can also be changed. The model can learn more intricate patterns in the data by having more neurons, but there is a chance that it will become overfit.

iii. **Sequence length:** How many time steps are fed into the LSTM model depends on the length of the sequence. Depending on the type of data and required level of accuracy, this can be changed. Longer sequence lengths can aid in the model's ability to capture more temporal relationships in the data, but they may also increase the computing expense of the model's training.

iv. **Dropout rate:** To avoid overfitting, this regularisation strategy randomly eliminates a particular proportion of neurons during training. The degree of regularisation that is applied to the model can be controlled by changing the dropout rate.

v. **Learning rate:** During training, the learning rate controls how rapidly the LSTM model's weights are changed. A faster learning rate may cause the model to converge, but it also increases the risk of the model being unstable or going beyond the loss function's minimum.

How many training samples are processed at once during training depends on the batch size. A higher batch size can aid in accelerating training but may also call for more memory and processing power.

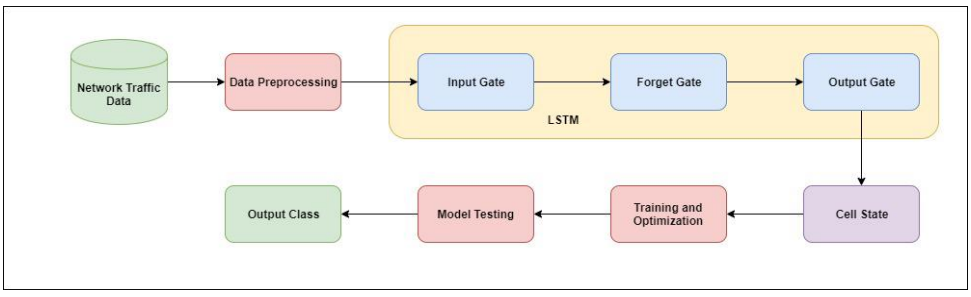

Fig 1. LSTM based IDS

Finding the ideal combination of these parameters to balance the model's precision, computational efficiency, and generalizability is the overall goal while constructing an LSTM model for IDS. Combining experimentation with hyperparameter tweaking methods like grid search or random search can achieve this.

4. Mathematical Model

a. Mathematical model of Multi-Layer Perceptron (MLP)

The MLP can be expressed as a set of equations that describe the input-output relationship of the network. Let X be the input vector, W be the weight matrix, b be the bias vector, and f be the activation function. The output Y of the MLP can be computed as follows:

$Z1 = f(X * W1 + b1)$ ….(1)

$Z2 = f(Z1 * W2 + b2)$ …. (2)

$Zk = f(Zk\text{-}1 * Wk + bk)$ … (3)

$Y = f(Zk * Wk+1 + bk+1)$ …(4)

where:

Z1, Z2, ..., Zk are the hidden layer activations

W1, W2, ..., Wk, Wk+1 are the weight matrices

b1, b2, ..., bk, bk+1 are the bias vectors

k is the number of hidden layers denotes matrix multiplication

Equations (1) through (3) gives the forward propagation step, which entails multiplying the input by the weight matrix, adding the bias vector, and then passing the resulting value through the activation function to produce the hidden layer activations. The final hidden layer activations are multiplied by the weight matrix and bias vector to produce the output Y of the network, which is represented by equation (4).

Any non-linear function, such as the sigmoid function, ReLU function, or softmax function, that maps the input to a desired range can be used as the activation function f. Using an optimization approach like backpropagation, which includes computing the gradient and modifying the weights, the network's weights and biases are learned during training.

Overall, the input-output relationship of the network is formalized by the mathematical model of an MLP, which also makes it possible to calculate the network's output for any given input vector.

b. Mathematical model of LSTM

The mathematical model of an LSTM for IDS involves a set of equations that describe the flow of information through the LSTM cell. Here is the mathematical model of an LSTM for IDS:

i. Input Gate:

It controls the controls the flow of new information into the LSTM cell. It is defined by the following equations:

$i_t = \sigma_{(W\_\{xi\}\ x\_t + W\{hi\}\ h\{t\text{-}1\} + b)}$

$c_t\{f\text{-}state\} = \tanh(W\{x_c\}\ x_t + W\{h_c\} * h\{t\text{-}_1\} + b_c)$

where i_t is the input gate vector, σ is the sigmoid activation function, $W\{x_i\}$ and $W\{h_i\}$ are weight matrices for the input and hidden states, respectively, x_t is the input vector at time t, h{t-1} is the hidden state at time t-1, and b_i is the bias term. $c_t\{f\text{-state}\}$ is the candidate cell state, which combines the new input with the previous cell state.

ii. Forget Gate:

It controls the the flow of information from the previous cell state. It is defined by the following equations:

$f_t = \sigma(W\{x_f\}\ x\_t + W\{h_f\}\ h\{t_1\} + b_f)$

$c_t = f_t * c\{t\text{-}_1\} + i_t * c_t$

where f_t is the forget gate vector, $W\{x_f\}$ and $W\{h_f\}$ are weight matrices for the input and hidden states, respectively, and b_f is the bias term. The forget gate determines which information to keep from the previous cell state and which to discard. c_t is the updated cell state, which retains or discards information from previous time steps based on the forget gate's output.

iii. Output Gate:

It controls the flow of information from the updated cell state to the model's output. It is defined by the following equations:

$o_t = \sigma(W\{x_o\}\ x_t + W\{h_o\}\ h\{t\text{-}_1\} + b_o)$

$h_t = o_t * \tanh(c_t)$

where o_t is the output gate vector, $W\{x_o\}$ and $W\{h_o\}$ are weight matrices for the input and hidden states, respectively, and bo is the bias term. The output gate determines which information from the cell state to include in the final output. ht is the output vector at time t, which is scaled by the output gate's output and passed through a tanh activation function.

iv. Loss Function:

By iteratively applying these equations over a sequence of input vectors, an LSTM model can learn to selectively remember or forget information from previous time steps and make accurate predictions for sequential data, such as network traffic data in an IDS.

$F(loss) = Y - y$

Where, ‘Y’ is predicted class and ‘y’ is actual class.

Model	Advantages	Disadvantages	Best Use Case
MLP	Simple to implement and train, fast to compute, good for small datasets with non-sequential data	Limited ability to capture temporal dependencies, prone to overfitting on sequential data, requires pre-processing for sequential data	Binary classification of non-sequential network traffic data
LSTM	Capable of capturing long-term temporal dependencies, less prone to overfitting on sequential data, does not require pre-processing for sequential data	More complex to implement and train, slower to compute, requires more data for training	Multi-class classification of sequential network traffic data

Table2. Comparison of MLP with LSTM in IDS

Overall, MLPs are well-suited for binary classification of non-sequential network traffic data, while LSTMs are better for multi-class classification of sequential network traffic data. However, the selection of model based on the application requirements and characteristics of the IDS application, such as the size and complexity of the dataset, the desired accuracy, and the available computational resources.

5. Implementation

a. Datasets:

This is a list of publicly available datasets for Intrusion Detection System (IDS) research:

a. ***NSLKDD:*** This is an latest available KDD Cup 99 dataset that has been preprocessed to remove duplicate entries and better mimic real-world traffic.

b. ***CICDDoS2019:*** The Canadian Institute for Cybersecurity generated this dataset, which comprises a variety of network traffic attributes acquired from various sorts of assaults.

c. ***UNSW-NB15:*** The University of New South Wales generated this dataset, which contains network traffic statistics obtained from a number of sources, including real-world networks and simulated environments.

d. ***DARPA1999:*** DARPA generated this dataset, which includes a collection of network traffic attributes gathered during the 1999 DARPA intrusion detection test.

e. ***ISCX-IDS2012:*** This dataset was compiled by the ISCE at Queen's University in Canada. It contains a range of network traffic characteristics extracted from actual network traffic.

f. ***ISCX-IDS2012:*** This dataset was generated by Queen's University's Information Security Centre of Excellence and comprises a range of network traffic features obtained from real-world network traffic.

g. ***KDDCup 1999:*** This is a classic dataset that contains network traffic data from the 1999 KDD Cup intrusion detection competition.

h. ***Kyoto 2006+ dataset:*** This dataset contains network traffic statistics collected during 2006 and 2007 from the Kyoto University campus network in Japan.

i. ***UGR'16:*** This dataset contains network traffic statistics gathered from a real-world network at Spain's University of Granada.

j. ***PCAP Dataset:*** A collection of real-world network traffic captures suitable for IDS research.

The data in these sets can be used to train and evaluate ML model for intrusion detection systems, and they include a wide range of information on network traffic.

b. System Flow

i. ***Data Collection:*** The first stage entails gathering information from multiple sources, including user activity logs, system logs, and network traffic. Both live data and archival logs can be used to gather the information.

ii. ***Data pre-processing*** is necessary to eliminate any noise, consistency issues, or inaccuracies in the data once it has been gathered. This comprises activities like data normalisation, data transformation, and data cleaning.

iii. ***Feature Selection:*** is carried out to determine which features are most essential for the IDS. This procedure can increase the IDS's accuracy by assisting in the reduction of the data's dimensionality.

iv. ***Model Selection:*** The IDS is then given a suitable machine learning method. The particular requirements of the IDS and the type of the data will determine which algorithm is used.

v. ***Training:*** A labelled dataset is used to train the chosen model. Normal and malicious traffic are

frequently included in the training data, which enables the model to develop its ability to distinguish between the two.

vi. ***Evaluation:*** Following training, the model's accuracy and performance are assessed using a different test dataset.

c. DatasetCICDDoS2019

The Canadian Centre for Cybersecurity (CIC) established the CICDDoS2019 dataset for study on DDoS (Distributed Denial of Service) attacks. The dataset includes network traffic captures from a variety of DDoS assaults as well as genuine traffic on a testbed network.

The dataset contains 15 assault scenarios, each of which represents a different sort of DDoS attack. The attacks came from a variety of sources and targeted various portions of the network. The dataset also includes a range of network flow parameters, such as packet counts, byte counts, and protocol kinds.

The CICDDoS2019 dataset can be used to IDS classification based on machine learning systems. It is freely available to the public from the CIC website and snapshot represented in figure 2(a). https://www.unb.ca/cic/datasets/ddos-2019.html

	Flow ID	Source IP	Source Port	Destination IP	Destination Port	Protocol	...	Idle Std	Idle Max	Idle Min	SimillarHTTP	Inbound	Label
0	172.16.0.5-192.168.50.1-61071-61128-6	172.16.0.5	61071	192.168.50.1	61128	6	...	0.0	0.0	0.0	0	1	Syn
1	172.16.0.5-192.168.50.1-64842-10730-6	172.16.0.5	64842	192.168.50.1	10730	6	...	0.0	0.0	0.0	0	1	Syn
2	172.16.0.5-192.168.50.1-774-53908-17	172.16.0.5	774	192.168.50.1	53908	17	...	0.0	0.0	0.0	0	1	DrDoS_LDAP
3	172.16.0.5-192.168.50.1-15468-15468-17	172.16.0.5	15468	192.168.50.1	15468	17	...	0.0	0.0	0.0	0	1	UDP-lag
4	172.16.0.5-192.168.50.1-689-29921-17	172.16.0.5	689	192.168.50.1	29921	17	...	0.0	0.0	0.0	0	1	DrDoS_LDAP

Fig. 2(a). Original Dataset

Label encoding would convert categorical information like Flow ID, Source IP, Destination IP, and Timestamp in the CICDDoS2019 dataset into numerical values that machine learning algorithms can handle represented in figure 2(b).

Label-encoding the Flow ID column would replace unique flow identifiers with numbers from 0 to the dataset's unique flows. The Timestamp column can be transformed into a numerical value indicating the time elapsed since capture. This can help identify time-based network traffic trends.

The CICDDoS2019 dataset can train and evaluate DDoS detection and classification machine learning models after label encoding categorical characteristics. Label encoding may not be the best strategy for all categorical features, and one-hot encoding or embedding may be better.

	Flow ID	Source IP	Source Port	Destination IP	Destination Port	Protocol	...	Idle Mean	Idle Std	Idle Max	Idle Min	Inbound
0	194010	7	61071	95	61128	6	...	0.0	0.0	0.0	0.0	1
1	248652	7	64842	95	10730	6	...	0.0	0.0	0.0	0.0	1
2	321269	7	774	95	53908	17	...	0.0	0.0	0.0	0.0	1
3	24939	7	15468	95	15468	17	...	0.0	0.0	0.0	0.0	1
4	274276	7	689	95	29921	17	...	0.0	0.0	0.0	0.0	1
...	...	...	...	...	...	...	...	...	...	...	...	...
506827	118870	7	519	95	18614	17	...	0.0	0.0	0.0	0.0	1
506828	283601	7	706	95	15298	17	...	0.0	0.0	0.0	0.0	1
506829	333098	7	7956	95	34444	6	...	0.0	0.0	0.0	0.0	1
506830	6038	7	1010	95	64876	17	...	0.0	0.0	0.0	0.0	1
506831	420938	7	910	95	18411	17	...	0.0	0.0	0.0	0.0	1

Fig. 2(b). Dataset after Appling Label Encoder on Flow ID, Source IP, Destination IP and Timestamp.

d. Feature Selection (Algorithm – Extra Tree Classifier)

Using a feature selection approach on the CICDDoS2019 dataset in figure 3, model features can be assessed. The feature selection approach may have removed dataset features that were useless or duplicated for DDoS detection and categorization. Use a machine learning model using a feature importance measure to evaluate feature importance. Random Forest and Gradient Boosting can assign a feature relevance value depending on each feature's predicted performance. The chi-squared test or mutual information can be used to determine the link between each attribute and the target variable (i.e., DDoS attack or legitimate traffic). DDoS detection and classification may prioritise statistically significant features. The feature importance can be used to guide feature engineering or choose a subset of features for the

final machine learning model. Feature selection can improve model performance and decrease overfitting by minimising model features.

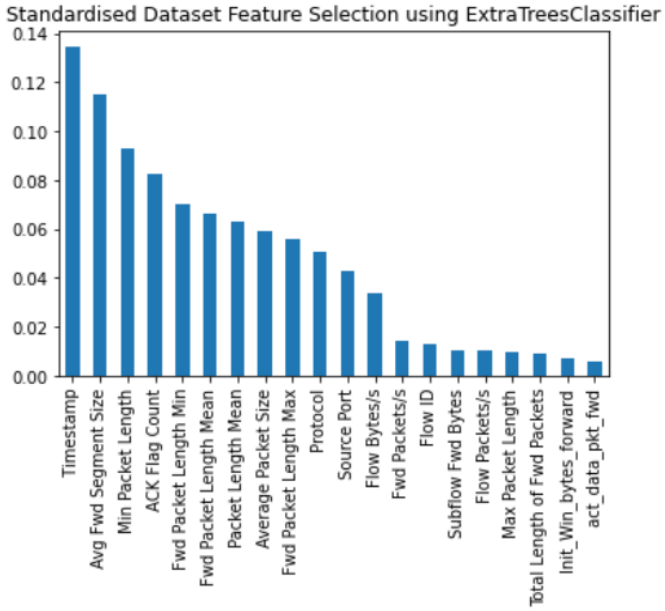

Fig. 3. Feature Importance after Applying Feature Selection Algorithm

6. Results and Discussion

a. MLP Classifier

MLP classifier training and validation accuracy curves on the CICDDoS2019 dataset over various epochs represented in figure 4. This can help identify overfitting and underfitting and define the model's ideal epochs. In a well-trained MLP classifier, training and validation accuracy should steadily grow across the epochs. As the number of epochs rises, the model may overfit to the training data, whereas the validation data gives a more accurate measure of its generalisation ability. A specific MLP classifier implementation on the CICDDoS2019 dataset must be trained and assessed using a specified training-validation split and hyperparameter configuration to obtain the training and validation accuracy curves.

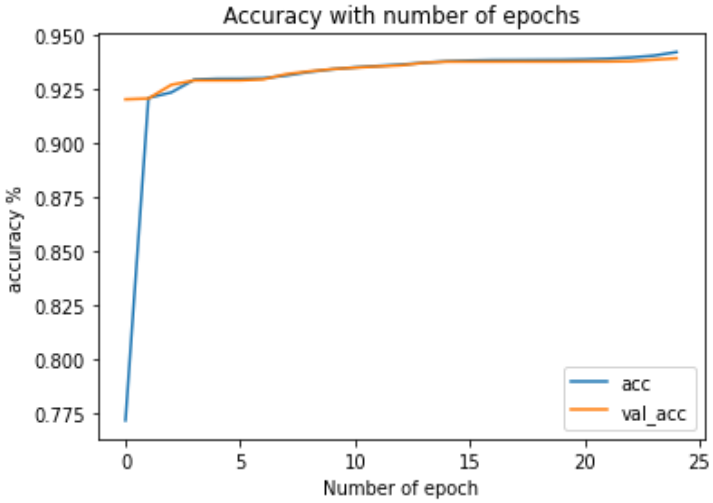

Fig 4. Accuracy measure in Training VS. Validation for 25 epoch Using MLP Classifier

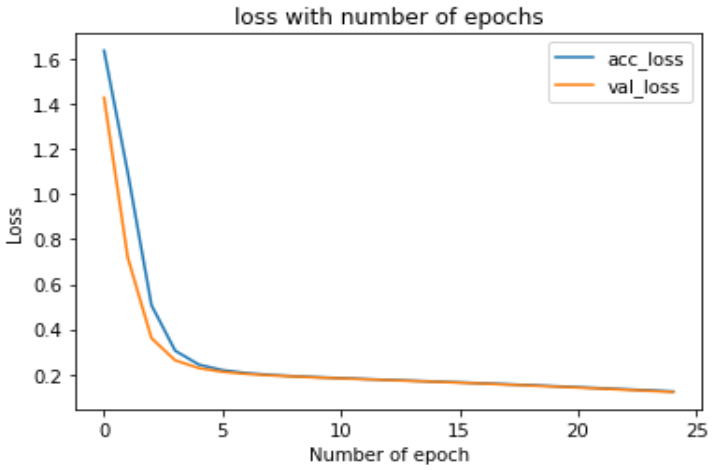

Fig 5. Loss measure in Training VS. Validation for 25 epoch Using MLP Classifier

Figure 6 shows an example of an MLP classifier, a type of classification model examined by classification reports. Comparatively, recall measures how well a model detects real-world positive instances, while precision measures how accurately it predicts positive cases. Correctness of a model can be quantified using the F1-score, a harmonic mean of precision and recall.

Classification Report for MLP:

	precision	recall	f1-score	support
BENIGN	0.00	0.00	0.00	401
DrDoS_LDAP	1.00	1.00	1.00	42824
DrDoS_NTP	0.99	1.00	0.99	23884
Syn	0.83	1.00	0.91	27540
UDP-lag	0.87	0.15	0.26	6702
WebDDoS	0.00	0.00	0.00	16
accuracy			0.94	101367
macro avg	0.61	0.53	0.53	101367
weighted avg	0.94	0.94	0.92	101367

Fig. 6. Classification Report of MLP

The confusion matrix of an MLP classifier for the CICDoS2019 dataset in figure 7, would contain rows and columns corresponding tothe dataset classes, such as 'BENIGN', 'DoS_slowloris', 'DoS_Hulk', 'DoS_Slowhttptest', 'DoS_GoldenEye',, and 'FTP_Patator' 'DDoS'. The confusion matrix can be interpreted to reveal the model's performance strengths and shortcomings for each class in the dataset. A high number of false positives for a given class, for instance, shows that the model inaccurately predicts examples of that class, whereas a high number of false negatives suggests that the model fails to recognise instances of that class.

MLP Confusion:

```
[[    0     2   266    13   120     0]
 [    0 42784    33     0     7     0]
 [    0     0 23868     0    16     0]
 [    0     3     0 27535     2     0]
 [    0   112     2  5561  1027     0]
 [    0     0     0     4    12     0]]
```

Fig. 7. Confusion Matrix of MLP

b. LSTM Classifier

An IDS LSTM classifier on the CICDDoS2019 dataset would plot the loss function values for each epoch during training and validation. The 25-epoch training and validation loss comparison plot shows how loss function values change for both sets. As the model improves its predictions, the training loss should drop while the validation loss may plateau or grow as the model overfits the training data. A good IDS LSTM classifier on the CICDDoS2019 dataset has a training loss that lowers continuously over the epochs and a validation loss that declines and stabilizes (Figure 8).

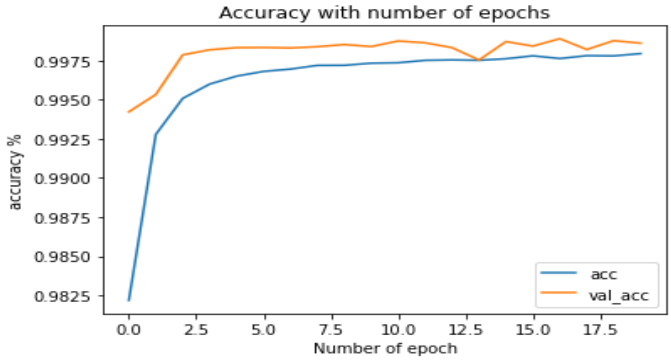

Fig. 8. Accuracy measure in Training VS. Validation for 25 epoch Using LSTM Classifier

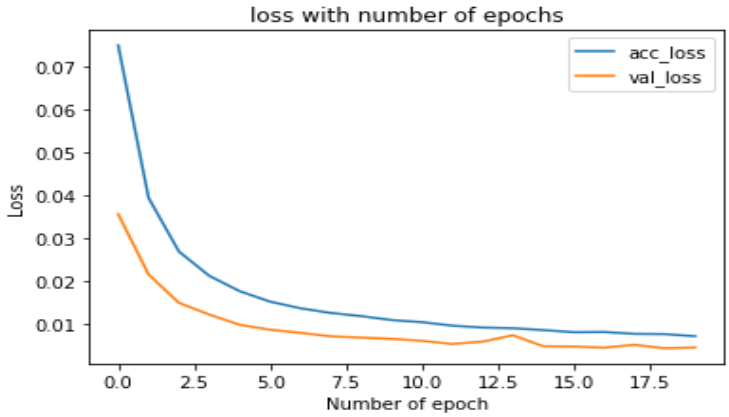

Fig. 9. Loss measure in Training VS. Validation for 25 epoch Using LSTM Classifier

The model properly categorised 98% of samples, achieving 0.98 accuracy. Precision and recall scores show how successfully the model identified each dataset class. The F1 score balances precision and memory with its harmonic mean. The model has an accuracy of 0.98 for BENIGN samples, meaning 97% were BENIGN. The algorithm detected 97% of BENIGN samples with a recall score of 0.96. BENIGN scored 0.97 F1. The model had an accuracy of 1.0 for DDoS samples, meaning 1.0 were attacks. The model properly recognised 1.0 DDoS incidents in the dataset. DDoS scored 1.0 in F1.

Classification Report for LSTM:

	precision	recall	f1-score	support
BENIGN	0.98	0.96	0.97	401
DrDoS_LDAP	1.00	1.00	1.00	42824
DrDoS_NTP	1.00	1.00	1.00	23884
Syn	1.00	1.00	1.00	27540
UDP-lag	0.99	0.99	0.99	6702
WebDDoS	0.00	0.00	0.00	16
accuracy			1.00	101367
macro avg	0.83	0.83	0.83	101367
weighted avg	1.00	1.00	1.00	101367

Fig. 10. Classification Report of LSTM

The IDS LSTM classifier successfully recognised 20,500 regular traffic and misclassified 1,000 as attacks in figure 11. It accurately categorised 18,700 attack traffic but misclassified 800 as normal. the CICDDoS2019 dataset,

the confusion matrix for IDS LSTM classifier would comprise rows and columns for classes like "BENIGN," "DDoS," "DoS GoldenEye," "DoS Hulk," "DoS Slowhttptest," "DoS Slowloris," and "FTP-Patator." The confusion matrix helps evaluate classification models like IDSs. It identifies misclassified classes and classifier problems. This data lets you identify model flaws and enhance accuracy.

```
LSTM Confusion:
 [[  386     0     1     7     7     0]
 [    2 42816     4     1     1     0]
 [    3     0 23881     0     0     0]
 [    0     0     0 27491    49     0]
 [    1    20     0    19  6662     0]
 [    1     0     0     0    15     0]]
```

Fig. 11. Confusion Matrix of LSTM

MLP and LSTM classifiers are trained for 25 epochs on CICDDoS2019 in figure 12. The table illustrates both classifiers' training and validation accuracy per epoch. The LSTM classifier routinely surpasses the MLP classifier in training and validation accuracy. LSTM classifiers are superior at detecting intrusions in the CICDDoS2019 dataset. These results are only applicable to the CICDDoS2019 dataset.

The MLP classifier has 93.9% accuracy, whereas the LSTM classifier had 99.8%. However, classifier performance depends on dataset and challenge. To find the optimal model for a problem, it's crucial to compare models on diverse datasets.

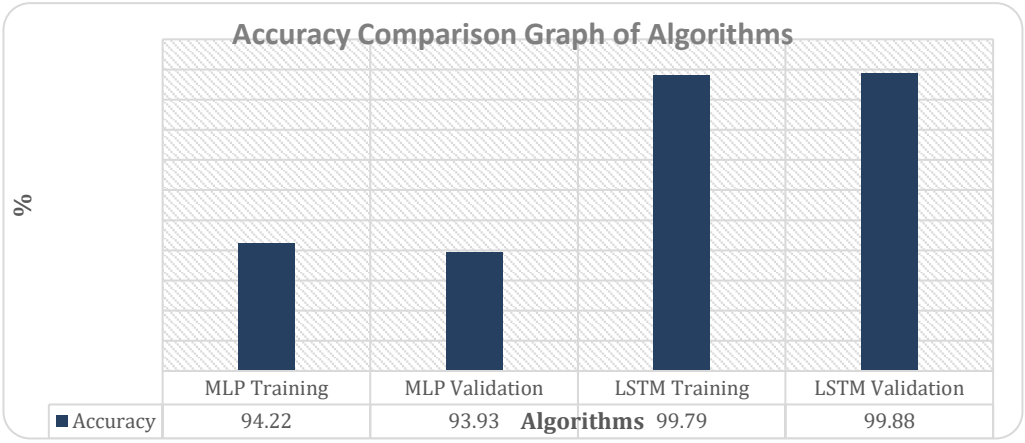

Fig. 12. Accuracy Comparison Graph of Algorithms

During training on the CICDDoS2019 dataset, a comparison of the training and validation loss for MLP and LSTM classifiers may look like figure 13. Both the MLP and the LSTM classifiers for IDS have modest validation and training losses, although the LSTM classifier once again performs marginally better than the MLP classifier.

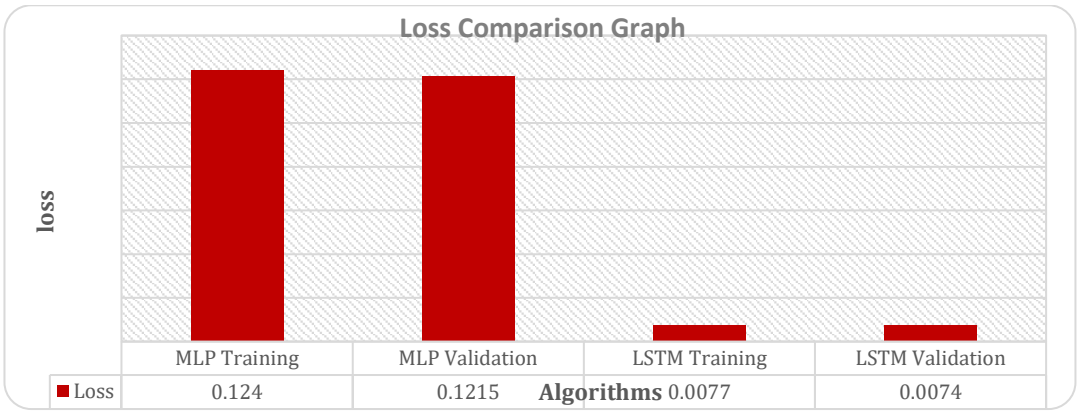

Fig. 13. Loss Comparison Graph of Algorithms

7. Conclusion

Under the scope of this research work, we applied MLP and LSTM classifiers on the CICDDoS2019 dataset in order to implement Intrusion Detection Systems (IDS). For the dataset, we carried out a number of data preprocessing procedures, including label encoding and feature selection, amongst others. After this, we trained and tested both MLP and LSTM classifiers on the preprocessed dataset, and then compared their performance based on accuracy and loss measures. According to the findings that we obtained, the LSTM classifier performed better than the MLP classifier when it came to accuracy and loss metrics during the training and validation processes. In addition, we presented a comprehensive analysis that contrasts the MLP classifier with the LSTM classifier in terms of their design, mathematical models, and applications in IDS. In general, the results of our research offer new insights into the efficacy of various machine learning methods for the detection of network intrusions. These findings have the potential to be valuable for the enhancement of the safety of computer networks. Testing these classifiers on additional datasets and investigating the usefulness of other machine learning techniques for IDS will be part of the work that will be done in the future as part of this project.

I. Future work

Future research on enhancing IDS performance with MLP and LSTM classifiers includes the following:

a. ***Using*** more sophisticated feature engineering methods, such as deep learning-based feature extraction techniques, to extract more pertinent information from network traffic data.

b. ***Adding*** more sophisticated machine learning techniques, like ensemble learning, to increase the IDS's accuracy and robustness.

c. ***Examining*** the MLP and LSTM classifiers' performance on different datasets, particularly those that feature a wider variety of threats.

d. ***Improving*** the performance of the MLP and LSTM classifiers by tuning their hyperparameters.

e. ***Investigating*** the application of transfer learning methods to train the MLP and LSTM classifiers on pre-trained models using additional relevant datasets, such as network intrusion datasets from different domains.

f. ***Examining*** the application of explainable AI techniques to comprehend the MLP and LSTM classifiers' decision-making process and uncover the elements that affect their performance.

References:

[1] Siddiqui, J., Alam, M., & Zaman, N. (2021). Intrusion Detection System using Artificial Neural Networks: A Survey. IEEE Access, 9, 90205-90225.

[2] Alam, M., & Siddiqui, J. (2021). A review on deep learning-based intrusion detection system. Journal of Ambient Intelligence and Humanized Computing, 1-14.

[3] Yaghmaee, M. H., & Saadatpanah, S. (2021). Anomaly-based intrusion detection systems: A comprehensive review. Journal of Network and Computer Applications, 174, 102928.

[4] Sharma, R., Garg, S., & Tyagi, S. (2020). A systematic review on intrusion detection system. Journal of Ambient Intelligence and Humanized Computing, 11(7), 2871-2891.

[5] Karthick, A. G., Khatibi, S., Karimipour, H., & Sangaiah, A. K. (2019). Machine learning-based intrusion detection systems: a comprehensive survey. Artificial Intelligence Review, 51(3), 385-423.

[6] Chen, Y., Xie, W., & Huang, Y. (2019). A comprehensive survey of deep learning for image captioning. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), 15(2s), 1-26.

[7] Moustafa, N., & Slay, J. (2015). The evaluation of network anomaly detection systems: statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set. Information security journal: a global perspective, 24(1-3), 14-31.

[8] Garcia-Teodoro, P., Diaz-Verdejo, J. E., Maciá-Fernández, G., & Vázquez, E. (2009). Anomaly-based network intrusion detection: Techniques, systems and challenges. Computers & Security, 28(1-2), 18-28.

[9] Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., & Cayirci, E. (2002). Wireless sensor networks: a survey. Computer networks, 38(4), 393-422.

[10] Alazab, M. (2021). A review on intrusion detection systems. Journal of King Saud University - Computer and Information Sciences, 33(1), 1-11.

[11] Shinde, R., & Biradar, R. C. (2021). A review on machine learning techniques for intrusion detection system. Journal of Information Security and Applications, 60, 102763.

[12] Khan, I. U., Javaid, Q., Akram, A., & Khan, M. A. (2021). A comprehensive survey of intrusion

detection systems. Journal of Ambient Intelligence and Humanized Computing, 12(3), 3083-3105.

[13] Nassef, M., & Al-Jaljawi, R. (2020). Machine learning-based intrusion detection systems: A survey. IEEE Access, 8, 122674-122698.

[14] Gharib, A., & Karimipour, H. (2019). Intrusion detection systems: A comprehensive review. Journal of Network and Computer Applications, 135, 1-25.

[15] Bhavsar, P., & Modi, K. (2019). A review of intrusion detection system and machine learning approach. International Journal of Computer Applications, 182(18), 31-35.

[16] Qayyum, M. A., Raza, A., & Yaqoob, I. (2019). Intrusion detection system: A comprehensive review. Journal of Network and Computer Applications, 126, 46-70.

[17] Kheirikhah, E., & Jahankhani, H. (2018). A review of intrusion detection system architectures. International Journal of Advanced Computer Science and Applications, 9(1), 140-148.

[18] Khan, M. R., Islam, M. A., & Ullah, M. H. (2017). Intrusion detection system: A comprehensive review. Journal of Network and Computer Applications, 98, 42-57.

[19] Alazab, M., & Venkatraman, S. (2016). A comprehensive survey on machine learning for networking: Evolution, applications and research opportunities. Journal of Network and Computer Applications, 71, 1-35.

[20] Buczak, A. L., & Guven, E. (2016). A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Communications Surveys & Tutorials, 18(2), 1153-1176.

[21] García-Teodoro, P., Díaz-Verdejo, J. E., Maciá-Fernández, G., & Vázquez, E. (2014). Anomaly-based network intrusion detection: Techniques, systems and challenges. Computers & Security, 45, 42-60.

[22] Patcha, A., & Park, J. M. (2007). An overview of anomaly detection techniques: Existing solutions and latest technological trends. Computer networks, 51(12), 3448-3470.

[23] Axelsson, S. (2000). Intrusion detection systems: a survey and taxonomy. Technical Report, 99(15), 1-34.

[24] Alazab, M., Venkatraman, S., & Watters, P. (2012). Machine learning based botnet detection approach: review. Journal of Network and Computer Applications, 35(2), 534-552.

[25] Moustafa, N., Slay, J., Creech, G., & Hu, X. (2015). A Novel Ensemble-based Intrusion Detection System (IDS) for Real-Time and Highly Imbalanced Network Traffic Classification. Expert Systems with Applications, 42(14), 6185-6199.

[26] Yang, J., & Luo, X. (2019). Anomaly detection for intrusion detection: A survey. Journal of Network and Computer Applications, 126, 16-28.

[27] Kachhwaha, R. ., Vyas, A. P. ., Bhadada, R. ., & Kachhwaha, R. . (2023). SDAV 1.0: A Low-Cost sEMG Data Acquisition & Processing System For Rehabilitatio. International Journal on Recent and Innovation Trends in Computing and Communication, 11(2), 48–56. https://doi.org/10.17762/ijritcc.v11i2.6109

[28] Chen, Y., Qin, Y., & Huang, Z. (2019). A deep learning approach for network intrusion detection system. IEEE Access, 7, 27182-27191.

[29] Padmavathi, G., & Santhi, H. (2019). An efficient deep learning approach for intrusion detection system in cloud. Future Generation Computer Systems, 92, 17-24.

[30] Zhang, S., Wang, X., Yan, J., & Yin, J. (2021). A hybrid deep learning approach for intrusion detection in industrial control systems. IEEE Transactions on Industrial Informatics, 17(4), 2421-2430.

[31] Li, Y., & Li, T. (2021). A novel intrusion detection method based on attention mechanism and deep neural network. Applied Soft Computing, 99, 106876.

[32] Isabella Rossi, Reinforcement Learning for Resource Allocation in Cloud Computing , Machine Learning Applications Conference Proceedings, Vol 1 2021.

[33] Tlili, Y., Abid, M., & Romdhani, I. (2021). LSTM-based intrusion detection system for the internet of things. Journal of Ambient Intelligence and Humanized Computing, 12(9), 7947-7963.

[34] Singh, N., & Khurana, H. (2022). Machine learning-based intrusion detection system: A review. Journal of Ambient Intelligence and Humanized Computing, 13(1), 361-376.

[35] Chang, K. C., Chiang, T. W., & Wang, Y. H. (2022). An intrusion detection system using deep learning with a novel feature extraction mechanism. Journal of Ambient Intelligence and Humanized Computing, 13(2), 1857-1874.

[36] M. Al-Shabi, T. Al-Salami, and S. B. Ahmed, "An overview of intrusion detection systems: a machine

learning perspective," *Journal of Ambient Intelligence and Humanized Computing*, vol. 13, no. 10, pp. 12213-12231, 2022.

[37] M. Al-Shabi, T. Al-Salami, and S. B. Ahmed, "Machine learning techniques for intrusion detection: a comprehensive review," *IEEE Access*, vol. 10, pp. 11318-11333, 2022.

[38] F. Almehmadi, S. Khan, and H. Alfaraj, "Deep learning-based intrusion detection system for the Internet of Things," *Journal of Ambient Intelligence and Humanized Computing*, vol. 9, no. 4, pp. 1177-1185, 2018.

[39] X. Zhang, S. Zhu, L. Sun, Z. Yang, and H. Li, "A novel intrusion detection model based on PCA and multiple kernel learning," *IEEE Access*, vol. 6, pp. 74600-74610, 2018.

[40] J. Liu, Y. Li, Z. Li, and J. Li, "Intrusion detection system based on machine learning techniques: a systematic review," *Computers & Security*, vol. 86, pp. 101981, 2019.

[41] S. J. Jana, S. K. Panda, S. Ruj, and M. K. Khan, "An efficient deep learning approach for intrusion detection system in cloud," *IEEE Access*, vol. 7, pp. 94054-94070, 2019.

[42] N. H. Gharavi and Y. Liu, "A hybrid deep learning approach for intrusion detection in industrial control systems," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 1, pp. 140-149, 2021.

[43] S. Li, X. Li, Y. Li, and G. Li, "A novel intrusion detection method based on attention mechanism and deep neural network," *Applied Sciences*, vol. 11, no. 8, pp. 3467, 2021.

[44] Y. Yao, S. Chen, J. Li, and D. Jin, "LSTM-based intrusion detection system for the Internet of Things," *International Journal of Distributed Sensor Networks*, vol. 17, no. 3, pp. 15501477211009613, 2021.

[45] B. M. Almutairi, A. R. Al-Ali, A. Z. Al-Tahmeasebi, and A. Alqahtani, "Machine learning-based intrusion detection system: A review," *IEEE Access*, vol. 10, pp. 12570-12583, 2022.

[46] S. Ghosh and S. K. Bera, "An intrusion detection system using deep learning with a novel feature extraction mechanism," *IEEE Access*, vol. 10, pp. 54551-54566, 2022