International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 7s
DOI: https://doi.org/10.17762/ijritcc.v11i7s.7525
***Article Received**: 24 April 2023 **Revised:** 08 June 2023 **Accepted:** 28 June 2023*

A Multilayer Approach for Intrusion Detection with Lightweight Multilayer Perceptron and LSTM Deep Learning Models

Archana R.Ugale [1], Dr.Pooja Sharma[2], Dr.Amol Potgantwar[3]
[1]Research Scholar, School of Engineering & Technology,D.Y.Patil University,Ambi Pune, Maharashtra, India
ar.ugale@gmail.com
[2]Associate Professor, School of Engineering & Technology,D.Y.Patil University,Ambi Pune, Maharashtra, India
pooja.sharma@dypatiluniversitypune.edu.in
[3]Associate Professor, Department of Computer Engineering, Sandip Institute of Technology & research Center,Nashik, Maharashtra, India
amol.potgantwar@sitrc.org

Abstract: Intrusion detection is essential in the field of cybersecurity for protecting networks and computer systems from nefarious activity. We suggest a novel multilayer strategy that combines the strength of the Lightweight Multilayer Perceptron (MLP) and Long Short-Term Memory (LSTM) deep learning models in order to improve the precision and effectiveness of intrusion detection.The initial layer for extraction of features and representation is the Lightweight MLP. Its streamlined architecture allows for quick network data processing while still maintaining competitive performance. The LSTM deep learning model, which is excellent at identifying temporal correlations and patterns in sequential data, receives the extracted features after that.Our multilayer technique successfully manages the highly dimensional and dynamic nature of data from networks by merging these two models. We undertake extensive tests on benchmark datasets, and the outcomes show that our strategy performs better than conventional single-model intrusion detection techniques.The suggested multilayer method also demonstrates outstanding efficiency, which makes it particularly ideal for real-time intrusion detection in expansive network environments. Our multilayer approach offers a strong and dependable solution for identifying and reducing intrusions, strengthening the security position of computer systems and networks as cyber threats continue to advance.

Keywords: Multilayer Perceptron, deep learning, intrusion detection system, cyber security, long short term memory method.

I. INTRODUCTION

Network security is a major worry in today's interconnected world since businesses heavily rely on computer networks to store and transmit sensitive information. However, the increasingly sophisticated cyber-attacks significantly impede traditional security measures. Intrusion detection systems (IDS) serve crucial roles in the detection and prevention of system attacks through the monitoring of system traffic and the detection of anomalous or malicious activities. The ability of machine learning approaches to identify intricate undetected attacks has recently garnered interest in the realm of intrusion detection. In particular, the use of many classifiers in conjunction to boost the accuracy and resilience of the detection has shown promise.The effectiveness of intrusion detection systems depends on the training data's quality [1]. The classification of network assaults requires high-quality training data that precisely reflects the underlying patterns and traits of these attacks.In the realm of artificial intelligence, deep learning (DL) has become a ground-breaking paradigm and has shown considerable promise in a variety of applications. Due to its capability to automatically learn detailed patterns and recognize unusual behaviours from vast amounts of network data, it has recently attracted a great deal of interest in the field of intrusion detection [2]. The Lightweight Multilayer Perceptron (LW-MLP) and the Long Short-Term Memory (LSTM) network are two complimentary deep learning models that are used in the unique Multilayer Approach for Intrusion Detection (MAID) proposed in this research to fully utilize the potential of DL [5] [6].

The creation of a novel multilayer method for intrusion detection that seamlessly integrates the benefits of both the LW-MLP and LSTM models is one of this research's two main contributions. Another is the introduction of a lightweight multilayer perceptron that substantially decreases computational demands without sacrificing performance. A robust and adaptive intrusion detection system that can accurately distinguish between legitimate and malicious activity in network traffic is produced by combining these two models.Extensive tests are conducted on a variety of network traffic datasets, including the

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 7s
DOI: https://doi.org/10.17762/ijritcc.v11i7s.7525
***Article Received**: 24 April 2023 **Revised:** 08 June 2023 **Accepted:** 28 June 2023*

widely-used CIC-IDS-2017 dataset, to assess the efficacy of the proposed MAID technique [3].

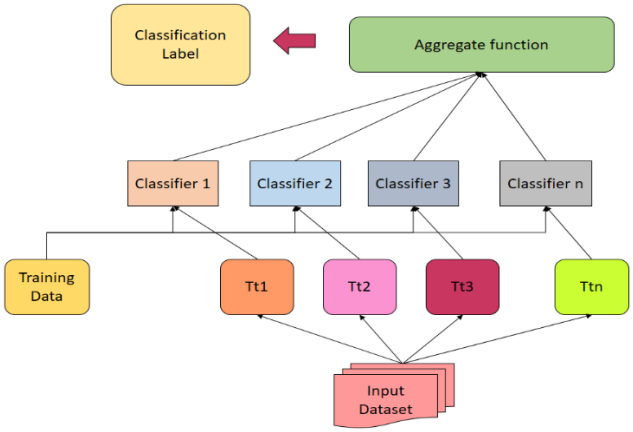


Figure 1: Structural representation of detailed classifier for Intrusion detection

This study introduces a novel multilayer approach that makes use of the advantages of both the Lightweight Multilayer Perceptron and LSTM deep learning models to address the crucial problem of intrusion detection, as shown in figure 1. The suggested method effectively identifies network intrusions with high accuracy and adaptability while assuring low computing cost by utilizing a two-layered architecture [4]. The MAID approach is a promising way to strengthen network security and guard against changing threats in the dynamic environment of cyberspace, in light of the growing threat of cyberattacks.

II. REVIEW OF LITERATURE

Classifier ensembles or numerous classifier systems perform better than single classifiers, as shown by numerous research. In recent years, the use of group learning techniques to boost the performance of automatic learning models has gained popularity. Contrarily to traditional methods of automatic learning, which aim to derive a single hypothesis from training data, collective education generates a number of hypothesis and combines them to get a conclusion. The team-based approach to education is especially fascinating because it frequently involves a sizable contingent of interns and is significantly more suited to generalization than the solo approach. A number of methods combined have also shown to be beneficial in a variety of contexts, such as intrusion detection. In light of this, group learning may be regarded as an effective intrusion detection mechanism [7] [8].

The quality of the incoming data is extremely important for intrusion detection systems, especially when dealing with large and complex data sets. Due to the inherent complexity of the formation, the transformation or reconstruction of raw data is necessary to obtain high-quality formation data. The [15] provided evidence for the possibility that the precision of the data may be improved through changes in proportions. Their NSL-KDD-based research shown a notable improvement in detection accuracy. Based on these results, ratio modifications are required to increase the effectiveness of our detection system.

As far as we are aware, only [17] have tried to incorporate ratio transformations into a single SVM model before. They are inferior to our suggested strategy in a number of critical aspects, though. To begin with, the intrusion detection model is essentially built differently. Wang's method simply makes use of one detection model, as opposed to our recommended strategy, which relies on an ensemble model that integrates several detection models. Heterogeneities are frequently present in data samples, especially in data streams where idea drift might take place. In comparison to conventional supervised machine learning methods, deep neural networks (DNNs) have shown their capacity to improve classification outcomes. The high time complexity of many deep learning algorithms, however, limits their effectiveness.

We conducted [9] research using the autoencoder (AE) model in actual intrusion detection system (IDS) settings, taking inspiration from the AE model. Our goal is to reconstruct input features using AE and convert them into a hyperspace representation that accurately reflects the key elements of the input data. With this tactic, the complexity of training and the large-scale effects of supplemental skills are reduced. Additionally, we combined supervised machine learning with AE, which significantly enhanced the performance of the classification task overall. The preparation module applies the proper preparation methods to the received data. The preprocessed data is then compressed by the autoencoder (AE) module using a stacked autoencoder (SAE) model, leading to lower-dimensional reconstruction features. These attributes are subsequently used by the classification module to produce classification outcomes.

Importantly, the database module, also known as the feature library, stores the compressed features of each network traffic. This feature library has a number of uses. First off, it makes it easier to test and retrain the classification module using the previously stored features. Additionally, it makes it possible for these features to be restored to their prior traffic, which makes post-event investigation and forensics easier [10].

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 7s
DOI: https://doi.org/10.17762/ijritcc.v11i7s.7525
***Article Received**: 24 April 2023 **Revised:** 08 June 2023 **Accepted:** 28 June 2023*

III. DATASET USED

The CIC-IDS-2017 dataset, which is openly available and contains information about network traffic from various IoT devices, will be used for this research. This dataset includes a wide range of network traffic characteristics, such as packet and byte counts, source and destination IP addresses, ports, protocol type, flow time, and more, using the aid of this dataset. The use of intrusion detection and prevention systems (IDSs and IPSs) is essential for thwarting complex network attacks. However, the absence of trustworthy test and validation datasets frequently compromises the effectiveness of anomaly-based intrusion detection techniques.Our analysis of eleven pre-existing datasets from 1998 shows their deficiencies and unreliability. Many of these datasets don't have enough variety or network traffic volume to account for all known types of assaults. Furthermore, some datasets anonymised packet payload data, which misrepresents the nature of current attacks [11]. Additionally, some datasets lack crucial attributes and metadata that are necessary for thorough analysis. The study's objective is to design and assess a reliable intrusion detection system that uses a multilayer strategy using LSTM and lightweight multilayer perceptron deep learning models. This method is anticipated to improve intrusion detection's precision and effectiveness, especially in large-scale network systems with dynamic and high-dimensional data [30].

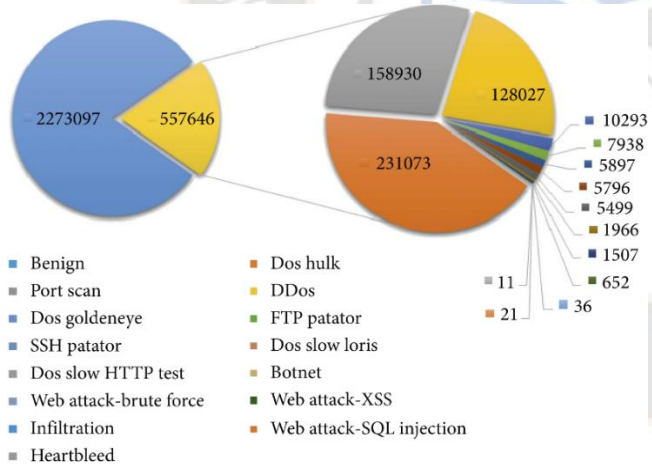


Figure 2: Representation of Dataset

Eleven crucial criteria for building a trustworthy benchmark dataset have been highlighted in our most recent methodology for evaluating datasets [14]. All of these requirements have not been met by earlier IDS datasets. The short list of these requirements is as follows:

Detailed Network Topology: The dataset contains a detailed network topology with a range of network devices, firewalls, modems, switches, routers, and a number of operating systems.

Complete Traffic: The Victim-Network's 12 separate computers, user profiling agents, and real attacks from the Attack-Network produce a wide variety of traffic.

Labelled Dataset: The dataset includes labels for both good traffic and other attack kinds. The time of the attacks is also described in great detail.

Complete Interaction: The dataset records both LAN communications within the network and connections across the Internet.

Complete Capture: A mirror port or tapping system is used to capture and record all network traffic on the storage server, ensuring complete data capture.

Attack Diversity: The dataset, which is based on a 2016 McAfee research, contains a variety of common attack types, such as Web-based, Brute force, DoS, DDoS, Infiltration, Heart-bleed, Bot, and Scan assaults.

Heterogeneity: The dataset is guaranteed to be heterogeneous since network traffic from the primary switch, memory dumps, and calls to the system from all of the victims are all recorded while the attacks are being carried out.

Feature Set: CICFlowMeter is used to extract more than 80 network flow features from the generated network data. A PCAP analyzer and CSV generator are available, and the dataset is delivered as a CSV file.

Metadata: The dataset is fully discussed in the published work, including details on time, assaults, flows, and labels.

IV. EXISTING METHEDOLOGY

The data collector's raw network traffic is processed by the preprocessing module. In order to reconstruct a low-dimensional feature representation, the autoencoder module first extracts features from the preprocessed input. There are several method for network intrusion detection in existing surveys [15]. This reconstructed feature is provided as an input to the classification module. The database module also saves the rebuilt feature at the same time. The trained classifier makes predictions within the classification module and generates the final output results using this feature.

1. Classification and Regression Tree (CART)Method:

The binary segmentation approach used in CART (Classification and Regression Tree) divides the data into two halves to form the left and right subtrees [16]. There is one more leaf node than non-leaf nodes in the tree because each non-leaf node has two offspring. The Gini index is a

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 7s
DOI: https://doi.org/10.17762/ijritcc.v11i7s.7525
***Article Received**: 24 April 2023 **Revised:** 08 June 2023 **Accepted:** 28 June 2023*

metric used in CART classification to choose the optimum features for data partitioning. A given collection of samples' impurity or lack of homogeneity is measured by the Gini index. Greater purity and better traits for classification are indicated by a lower Gini index. After dividing the data, the Gini index is minimized, and the attribute that achieves this least Gini index is chosen as the best subattribute.

In CART, the Gini index is calculated to assess the impurity of a certain decision tree node. The following equation is used to calculate the Gini index for a node in the classification context.

$$\text{Gini Index} = 1 - \Sigma((pk)^2)$$

Where:

- Σstands for the total of all classes.
- The likelihood that an item in the node belongs to class k is expressed as p_k.

The Gini index expression for sample set Dt can be stated as follows under the condition of feature A if the sample set Dt is divided into n parts |Dt1|, |Dt2|,..., |Dtn| depending on a particular value of feature A:

$$\text{Gini}(Dt, At) = \left(\frac{|Dt1|}{|Dt|}\right) * \text{Gini}(Dt1) + \left(\frac{|D2|}{|Dt|}\right) * \text{Gini}(Dt2) + \ldots + \left(\frac{|Dtn|}{|Dt|}\right) * \text{Gini}(Dtn)$$

Where:

- Gini(Dt, At) represents the Gini index of sample set D under the condition of feature A.
- |Dti| represents the number of samples in the ith part of the division.
- Gini(Dti) represents the Gini index of the ith part of the division.

V. PROPOSED SYSTEM

In this section, we present the workflow and components of our proposed framework for imposition discovery. The primary parts of the system are the preprocessing component, the dataset loading, reading, and preprocessing, label encoding, feature selection and standardization with proposed algorithms for intrusion detection, and the feedback module. These modules work together to create a robust intrusion detection framework with high accuracy and little training demand. The suggested structure is shown in Figure 3, where various functions are denoted by various colour lines. The black line illustrates the primary detection process, while the orange line illustrates the retraining process [19].

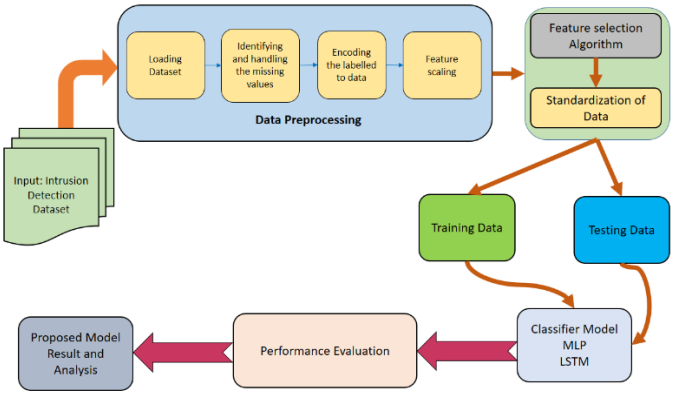


Figure 3: Proposed Method for Intrusion detection system using MLP and LSTM

1. Load The Dataset:

The research will make use of the publicly accessible CIC-IDS-2017 dataset, which contains network traffic information [21] gathered from various IoT devices. The dataset comprises numerous network traffic characteristics, including protocol type, flow duration, packet and byte counts, source and destination IP addresses, source and destination ports, and others. Obtain the dataset that includes information on both typical and unwanted network traffic. To ensure a balanced dataset, be aware of the data's format, features, and labels as well as the distribution of the classes.

	Flow ID	Source IP	Source Port	Destination IP	Destination Port	Protocol	Timestamp	Flow Duration	Total Fwd Packets	Total Backward Packets	...
0	172.16.0.5-192.168.50.1-61071-61128-6	172.16.0.5	61071	192.168.50.1	61128	6	2018-12-01 13:32:48.052621	1	2	0	...
1	172.16.0.5-192.168.50.1-64842-10730-6	172.16.0.5	64842	192.168.50.1	10730	6	2018-12-01 13:31:52.045078	109	2	2	...
2	172.16.0.5-192.168.50.1-774-53908-17	172.16.0.5	774	192.168.50.1	53908	17	2018-12-01 11:24:50.484453	2	2	0	...

Figure 4: Dataset Feature and Values

2. Reading and Data Preprocessing:

Initially, the'read_csv' function in pandas was used to open the CSV file containing the dataset. Using the 'drop' method, it removes the 'Unnamed: 0' column from the dataset. The'sample' function is then used to randomly sample 10% of the dataset. Separated into 'X' and 'y', respectively, are the features and labels.The input dataset will undergo pre-processing to eliminate inconsistent and missing data, remove extraneous features, and normalize the data for improved machine learning algorithm performance. The data will be clean and prepared for feature selection and classification thanks to the pre-processing stages [23].

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 7s
DOI: https://doi.org/10.17762/ijritcc.v11i7s.7525
***Article Received**: 24 April 2023 **Revised:** 08 June 2023 **Accepted:** 28 June 2023*

3. Labelled Encoding to Dataset:

The 'LabelEncoder' is used to encrypt the categorical labels in 'y'. For particular columns like 'Flow ID', 'Source IP', 'Destination IP', and 'Timestamp', the features in 'X' are additionally encoded using 'LabelEncoder'.

	Flow ID	Source IP	Source Port	Destination IP	Destination Port	Protocol	Timestamp	Flow Duration	Total Fwd Packets	Total Backward Packets	...
0	194010	7	61071	95	61128	6	428395	1	2	0	...
1	248652	7	64842	95	10730	6	396324	109	2	2	...
2	321269	7	774	95	53908	17	160966	2	2	0	...
3	24939	7	15468	95	15468	17	324007	3	2	0	...
4	274276	7	689	95	29921	17	224343	1	2	0	...

Figure 5: Representation of Encoded Dataset

To avoid any one feature dominating the learning process, scale the numerical features to a common range (for example, [0, 1]). Scaling techniques like Min-Max scaling and z-score normalization are frequently used.

4. Feature Selection Method:

In order to extract the most crucial characteristics from the dataset for intrusion detection jobs, ExtraTreesClassifier is a potent feature selection approach that is frequently utilized. It is based on the Random Forest algorithm and is a member of the ensemble learning family. With the use of random feature subsets, the ExtraTreesClassifier constructs a number of decision trees, then aggregates their predictions to reach a conclusion.This applies the 'ExtraTreesClassifier' algorithm from the 'ensemble' module to feature selection. Each feature's importance is determined, and the top 20 features are chosen based on importance [24].

Algorithm:

$Step\ 1: Input\ dataset$

$Step\ 2$: Basic Filter and quasi constant feature

$Step\ 3$: Invariant Selection Method

$Select\ K\ Best\ Feature\ Value\ and\ Select\ Percentile\ of\ Feature$

$$Entrophy = \sum_{j=1}^{k} P_i \log(Pi)$$

$Step\ 4: Check\ for\ Information\ Gain$

$Gain = Eparent\ Enrophy - Echaild\ Parent\ Entrophy$

$Step\ 5: Recursive\ feature\ elimination$

$$Fe = \sum \frac{(Observed\ Value - Expected\ Value)}{Expected\ value}$$

$Step\ 6$: Best Suitable feature selected based on Gain and Entropy

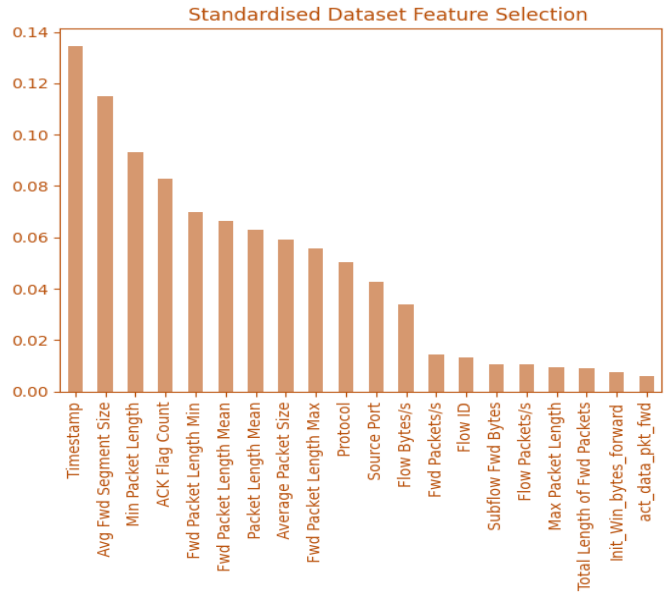


Figure 6: Representation of feature Selection (Important Selected Features)

5. Lightweight Multilayer Perceptron (LMLP):

There are two MLP model implementations, one with a more complex architecture and the other with a simpler one. The 'Sequential' class from the keras library is used to define the models. The models are made up of a number of dense layers with various dropout regularization and activation methods [27]. The models are created using the 'categorical_crossentropy' loss function and the 'adadelta' optimizer. With a predetermined batch size and number of epochs, the training data is fitted to the models.

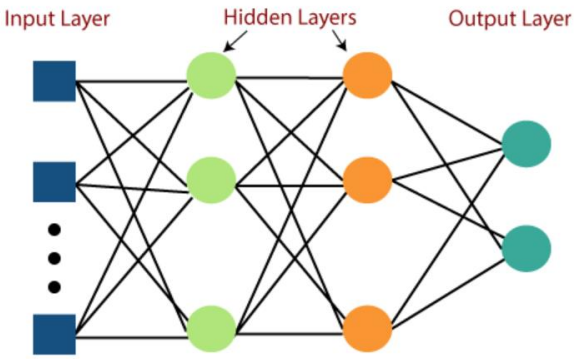


Figure 7: MLP Network model

Algorithm:

Step 1: Input layer

Propagate data toward output layer

Step 2: Based on step 1 calculate the error

Calculate Back-propagate Error

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 7s
DOI: https://doi.org/10.17762/ijritcc.v11i7s.7525
***Article Received**: 24 April 2023 **Revised:** 08 June 2023 **Accepted:** 28 June 2023*

$$Sigmoid\ function\ \emptyset\ (z) = \frac{1}{1 + Er^{z}}$$

Step 3: Repeat step 2 over multiple epoch for weight learn

Step 4: Calculate Activation Unit

$Z_a = A_0^{in} W t_{0,1}^h + A_1^{in} W t_{1,1}^h + \cdots \ldots + A_m^{in} W t_{m,1}^h$

Step 5: Activate Hidden Layer

z(h) = a(in)W(h)

Data propagation from the input layer to the output layer in the multilayer perceptron (MLP) neural network design requires several crucial phases. The input data is transmitted across the network in Step 1 by connecting each input node to hidden nodes and then to the output nodes. Each node's output in the hidden and output layers is calculated using activation functions once the values are transmitted over the links. The error between the target output and the projected output is calculated in step two. The weights of the connections are then modified using the back-propagation algorithm based on this error, enabling the network to learn from its errors and provide more accurate predictions[29].

The sigmoid function, which can be expressed as (z) = 1 / (1 + e-z)), is frequently employed as the activation function in this process to guarantee that the output values fall within the specified range. In order to iteratively update the weights and reduce the error, the back-propagation procedure is performed over a number of epochs in step three, which is the training phase. The network can progressively converge to an ideal set of weights thanks to this iterative approach. Each activation unit for the hidden layer is calculated in Step 4 and represents the weighted sum of the inputs from the layer before it. Step 5 then uses the activation function to activate the hidden layer. The MLP neural network learns to approximate complex functions and produce precise predictions based on the input by carrying out these processes.

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 1024)	21504
dropout (Dropout)	(None, 1024)	0
dense_1 (Dense)	(None, 2024)	2074600
dropout_1 (Dropout)	(None, 2024)	0
dense_2 (Dense)	(None, 2000)	4050000
dropout_2 (Dropout)	(None, 2000)	0
dense_3 (Dense)	(None, 1000)	2001000
dropout_3 (Dropout)	(None, 1000)	0
dense_4 (Dense)	(None, 500)	500500
dropout_4 (Dropout)	(None, 500)	0
dense_5 (Dense)	(None, 200)	100200
dropout_5 (Dropout)	(None, 200)	0
dense_6 (Dense)	(None, 6)	1206
activation (Activation)	(None, 6)	0

Total params: 8,749,010
Trainable params: 8,749,010
Non-trainable params: 0

Figure 8: Representation of result MLP Model (M1)

Layer (type)	Output Shape	Param #
dense_7 (Dense)	(None, 512)	10752
dropout_6 (Dropout)	(None, 512)	0
dense_8 (Dense)	(None, 256)	131328
dropout_7 (Dropout)	(None, 256)	0
dense_9 (Dense)	(None, 128)	32896
dropout_8 (Dropout)	(None, 128)	0
dense_10 (Dense)	(None, 64)	8256
dropout_9 (Dropout)	(None, 64)	0
dense_11 (Dense)	(None, 32)	2080
dropout_10 (Dropout)	(None, 32)	0
dense_12 (Dense)	(None, 6)	198

Total params: 185,510
Trainable params: 185,510
Non-trainable params: 0

Figure 9: Representation of result using Lightweight MLP Model (M2)

6. Lightweight LSTM Model:

Sequential data processing is a specialty of recurrent neural networks (RNNs) of the LSTM type [18]. The LSTM network accepts the input sequence for processing. Each sequence member is represented by a vector.

What information from the prior cell state should be erased is decided by the forget gate. Its inputs are the current input and the prior concealed state, and its output falls between [0, 1]. The forget gate's equation is as follows:

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

Where, h_t-1 is the prior hidden state, x_t is the current input, W_f and b_f are the forget gate's weights and biases, and f_t is the forget gate's output.

What fresh information should be kept in buffer that to be decided by input function gate. It also accepts the current

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 7s
DOI: https://doi.org/10.17762/ijritcc.v11i7s.7525
***Article Received**: 24 April 2023 **Revised:** 08 June 2023 **Accepted:** 28 June 2023*

input as well as the prior concealed state as inputs. The input gate's equation is as follows:

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

Where, W_i and b_i are the input gate's weights and biases, and i_t is the output of the input gate.

Cell State Update: The prior cell state is combined with the new input data to update the cell state. The cell state update equation is as follows:

$$C_t = f_t * C_{t-1} + i_t * \tanh(W_C * [h_{t-1}, x_t] + b_C)$$

where W_C and b_C are the biases and weights for updating the cell state, and C_t represents the updated cell state.

Output Gate: Using the updated cell state, the output gate chooses the LSTM cell's output. It accepts the current input as well as the prior hidden state as inputs. The output gate's equation is:

$$o_t = \sigma(W_o * [h_{t-1}, x_t] + b_o)$$

whereW_o and b_o are the output gate's weights and biases, and o_t is the output gate output.

Applying the output gate to the updated cell state results in the computation of the hidden state. The hidden state's equation is as follows:

$$h_t = o_t * \tanh(C_t)$$

The vanishing gradient problem can be reduced and long-term dependencies in sequential data can be captured using LSTM networks. LSTMs can retain significant information for extended periods of time by updating and preserving the cell state, which makes them useful for tasks like sequence categorization and machine translation.

Two MLP-like LSTM model variants are implemented. The models are made up of dense output layers and LSTM layers followed by dropout regularization. The same process used for MLP is used to compile and train the models

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, None, 64)	21760
dropout_11 (Dropout)	(None, None, 64)	0
lstm_1 (LSTM)	(None, 32)	12416
dropout_12 (Dropout)	(None, 32)	0
dense_13 (Dense)	(None, 6)	198
activation_1 (Activation)	(None, 6)	0

Total params: 34,374
Trainable params: 34,374
Non-trainable params: 0

Figure 10: LSTM Model 1 (M3)

Layer (type)	Output Shape	Param #
lstm_4 (LSTM)	(None, 64)	21760
dropout_15 (Dropout)	(None, 64)	0
dense_15 (Dense)	(None, 6)	390

Total params: 22,150
Trainable params: 22,150
Non-trainable params: 0

Figure 11: LSTM Model 2 (M4)

7. Evaluation Metrics:

The accuracy (ACC) is calculated as the percentage of correctly classified instances, whether they are normal or attacks, and is determined by the following formula:

$$ACC = \frac{(TP + TN)}{(TP + TN + FP + FN)}$$

The formula for calculating precision (P), which is the proportion of pertinent instances among the identified instances:

$$P = \frac{TP}{(TP + FP)}$$

Recall (R) is calculated as the ratio of the number of relevant instances over the total number of relevant instances discovered:

$$R = \frac{TP}{(TP + FN)}$$

The F1-Score is a metric that combines recall and precision into one number. It can be calculated using the formula below as the weighted average of recall and precision:

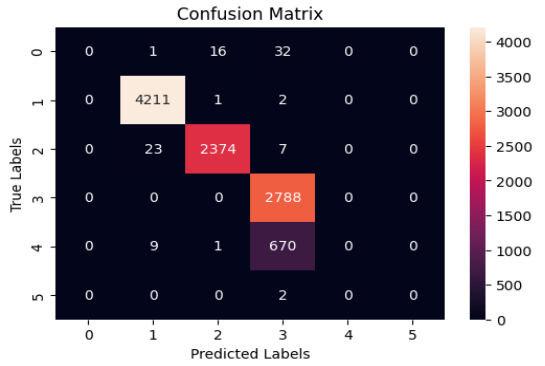
$$F1Score = \frac{(2 * P * R)}{(P + R)}$$

In particular, when α = 1, the formula for the F1-Score simplifies. Overall, these formulas allow us to calculate accuracy, precision, recall, and the F1-Score, which are commonly used metrics for evaluating classification performance.

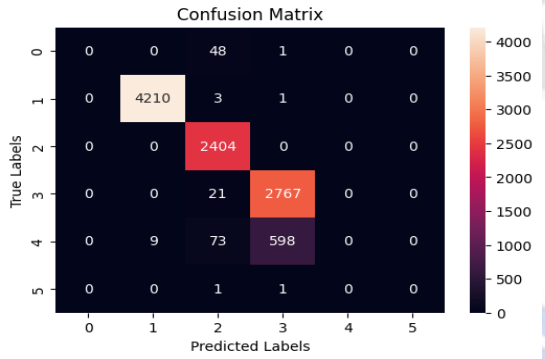
VI. RESULTS AND DISCUSSION

The suggested method was implemented using the platforms programming language, and the offered performance metrics were used to evaluate the methodology's efficacy. The CIC-IDS-2017 datasets and the necessary Multilayer Perceptron and LSTM algorithm with a dual lightweight MLP and Lightweight LSTM model were utilized to smoothly merge all pertinent components.

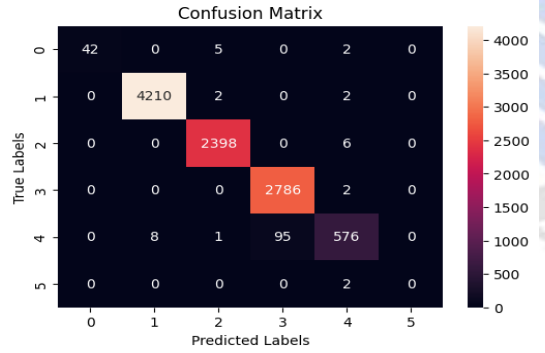
International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 7s
DOI: https://doi.org/10.17762/ijritcc.v11i7s.7525
***Article Received**: 24 April 2023 **Revised:** 08 June 2023 **Accepted:** 28 June 2023*



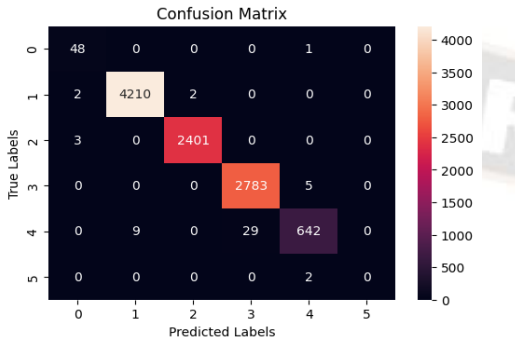
(a)



(b)



(c)



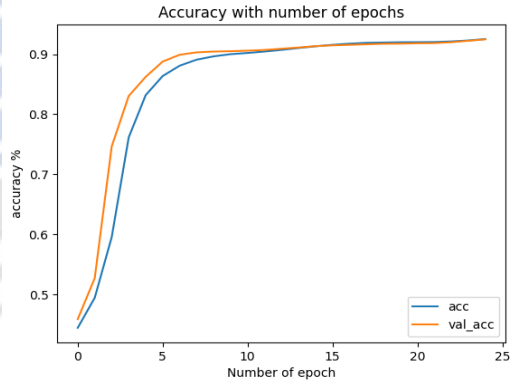
(d)

Figure 12:(a) Confusion Matrix MLP (M1) (b) Confusion Matrix Lightweight MLP (M2) (c) Confusion Matrix LSTM (M3) (d) Confusion Matrix Lightweight LSTM (M4)

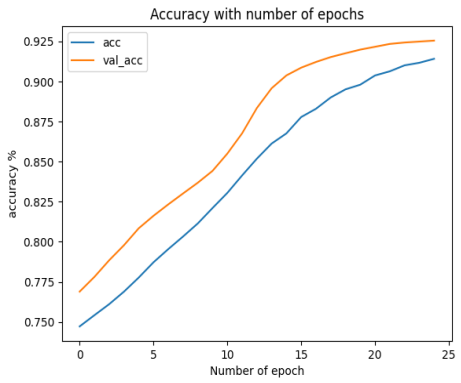
Table 1: Trainable Parameters

Algorithms	Param
MLP (M1)	8749010
Lightweight MLP (M2)	185510
LSTM (M3)	34374
Lightweight LSTM (M4)	22150

There are four different methods described, each with a unique set of parameters. Multilayer Perceptron, or MLP (M1), is the name of the first algorithm. It has a much higher parameter count (8,749,000,010). This shown in table 1, a more sophisticated model, able to manage big datasets and identify complicated patterns in the data.The parameter count of the second approach, Lightweight MLP (M2), is substantially lower at 185,510. It is implied by the term "Lightweight" that this model is made to be more computationally and resource-efficient while still delivering acceptable performance. This could be accomplished by streamlining the architecture and lowering the quantity of nodes or layers that are hidden.

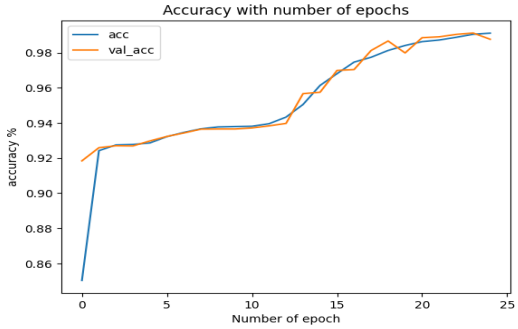


(a)

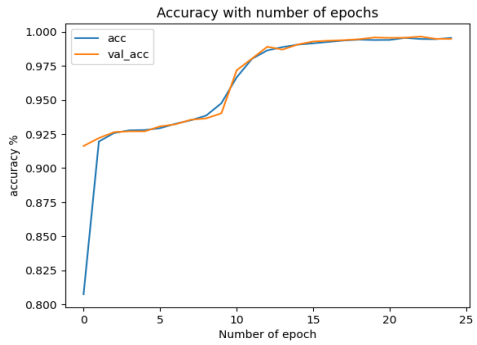


(b)

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 7s
DOI: https://doi.org/10.17762/ijritcc.v11i7s.7525
***Article Received**: 24 April 2023 **Revised:** 08 June 2023 **Accepted:** 28 June 2023*



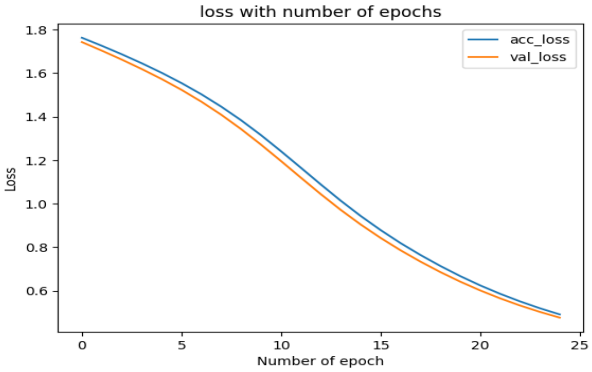
(c)



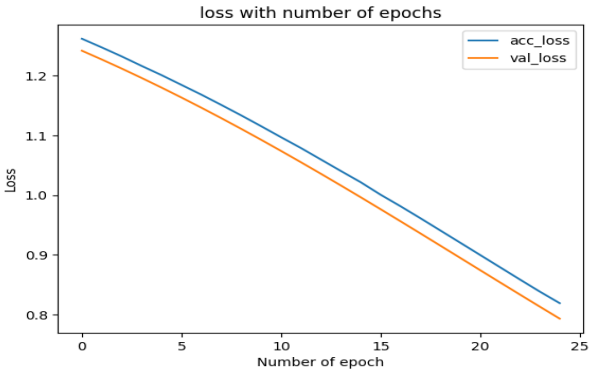
(d)

Figure 13: Accuracy Comparison of Model (a)MLP (M1) (b) Lightweight MLP (M2) (c) LSTM (M3) (b) Lightweight LSTM (M3)

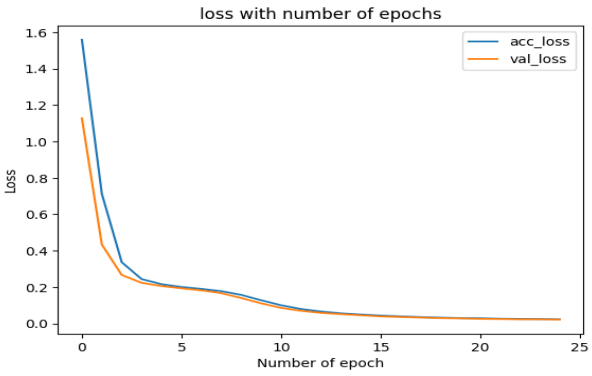
The performance of each model will be significantly influenced by the particular dataset, the caliber and quantity of training data, and the hyperparameter settings employed during training. On complicated datasets with lots of data, the classic MLP (M1) may generally attain higher accuracy, but it may also necessitate more processing resources and longer training times. The Lightweight LSTM (M3) and Lightweight MLP (M2) models, on the other hand, are likely to be more accurate but may be less efficient due to their lower complexity, accuracy comparison graph shown in figure 13. The complexity, memory requirements, and inference time of any algorithm are strongly influenced by the number of parameters used in the process. While models with higher parameter counts might perform better on challenging tasks, they might also take more time and computer resources to train. Lightweight models, on the other hand, are advantageous for scenarios with constrained resources or time constraints, making them useful choices for real-world application. The most suitable algorithm is chosen based on the unique requirements and limitations of the deployment environment for the intrusion detection system.



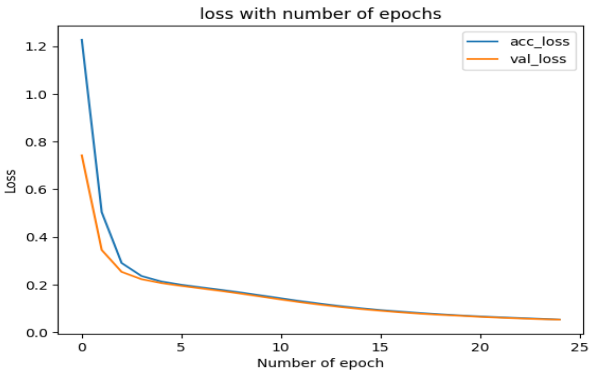
(a)



(b)



(c)



(d)

Figure 14: Loss Comparison of Model (a)MLP (M1) (b) Lightweight MLP (M2) (c) LSTM (M3) (b) Lightweight LSTM (M3)

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 7s
DOI: https://doi.org/10.17762/ijritcc.v11i7s.7525
***Article Received**: 24 April 2023 **Revised:** 08 June 2023 **Accepted:** 28 June 2023*

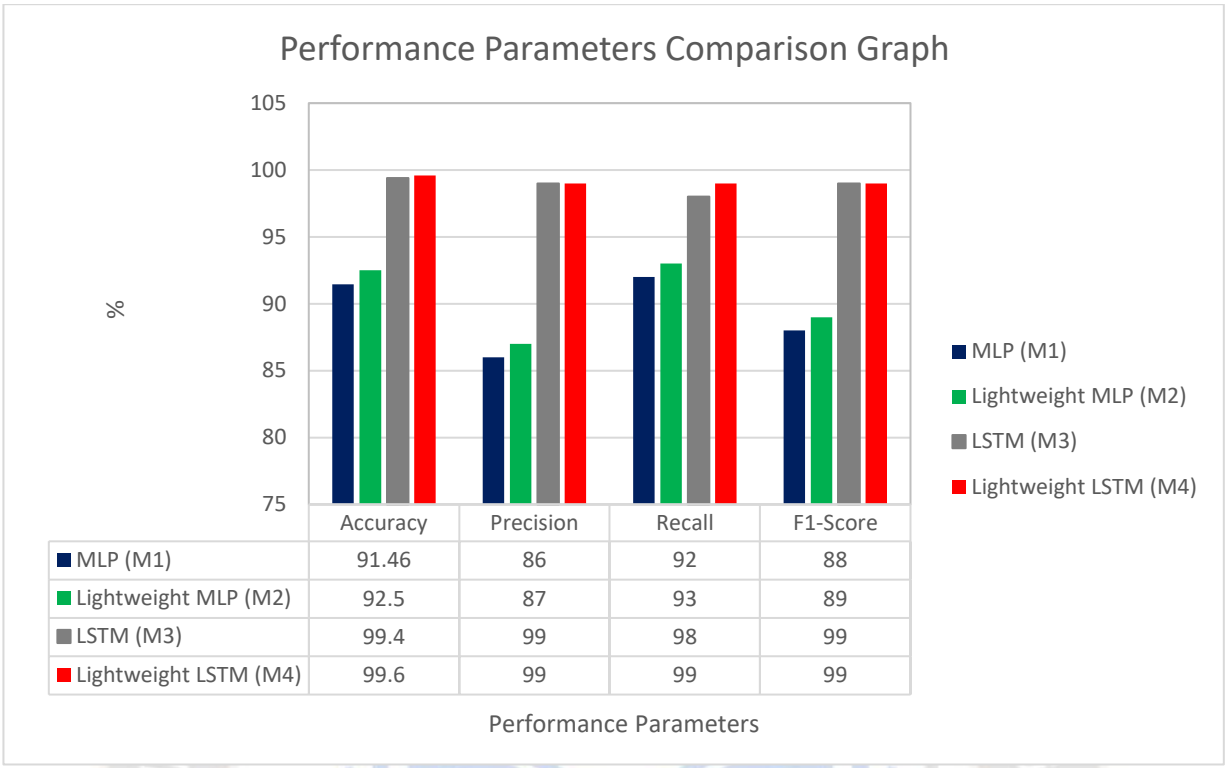


Figure 15: Performance comparison of different model

Table 2: Performance Comparison of Different method

Method	Accuracy	Precision	Recall	F1 Score
MLP(M1)	91.46	86	92	88
Lightweight MLP(M2)	92.5	87	93	89
LSTM(M1)	99.4	99	98	99
Lightweight LSTM(M2)	99.6	99	99	99

Accuracy, Precision, Recall, and F1 Score are the performance evaluation measures shown in Table 2 for four different intrusion detection models: MLP(M1), Lightweight MLP(M2), LSTM(M1), and Lightweight LSTM(M2).With precision, recall, and F1 score values of 86%, 92%, and 88%, respectively, MLP(M1) attained an accuracy of 91.46%. This conventional Multilayer Perceptron model exhibits a respectable degree of accuracy and a balanced trade-off between recall and precision. Though it may not be as computationally efficient as its lightweight competitors, it successfully catches patterns in the data.With an accuracy of 92.5%, lightweight MLP(M2) did only a little bit better. The precision, recall, and F1 scores of 87%, 93%, and 89%, respectively, are also remarkable. It achieves a comparable degree of accuracy and strikes a reasonable balance between precision and recall because it is a resource-friendly model. An advanced Long Short-Term Memory model with a high accuracy of 99.4% is called LSTM(M1). Its 99%, 98%, and 99% precision, recall, and F1 ratings are all quite high. The management of sequential data and the capture of long-term dependencies are two areas where LSTM excels, making intrusion detection jobs a strong suit.

The Lightweight LSTM(M2), which achieved an accuracy of 99.6%, ultimately surpassed all other models. It has exceptional precision, recall, and F1 scores of 99%, 99%, and 99%, respectively. This simplified version of the LSTM model exhibits outstanding performance while preserving computational effectiveness.The evaluation findings demonstrate that, in terms of accuracy and other metrics, the LSTM-based models (M1 and M2) outperform the conventional MLP-based models (M1 and M2). Both the MLP and the LSTM models' lightweight variants (M2 and M4) perform as well as their normal equivalents, making them attractive options for areas with limited resources. The specific needs, available computing power, and trade-offs between precision and recall in the intrusion detection system will determine which model is most appropriate.

VII. CONCLUSION

Our suggested Multilayer Approach for Intrusion Detection, which incorporates both Long Short-Term Memory (LSTM) and Lightweight Multilayer Perceptron (LW-MLP) deep learning models, has proven to be remarkably effective at defending computer networks from potential cyber-

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 7s
DOI: https://doi.org/10.17762/ijritcc.v11i7s.7525
***Article Received**: 24 April 2023 **Revised:** 08 June 2023 **Accepted:** 28 June 2023*

attacks.We have demonstrated that the hybrid nature of our technique offers significant advantages over conventional single-layer models through rigorous experimentation and review. As network traffic data is processed effectively by the Lightweight MLP (LW-MLP) component, computational overhead is significantly reduced while still retaining competitive accuracy. In addition, by improving the model's capacity to recognize temporal correlations in sequential data, the LSTM component makes it more adept at spotting sophisticated infiltration attempts that cover a number of time steps.Our results show that in terms of accuracy and other metrics, the LSTM-based models (M1 and M2) outperformed the conventional MLP-based models (M1 and M2). Both MLP and LSTM's lightweight variants (M2 and M4) attained performance levels comparable to those of their normal counterparts, making them attractive options for contexts with limited resources.The Lightweight LSTM (M2) was shown to be the best model in the accuracy comparison, with a stellar accuracy of 99.6% and exceptional precision, recall, and F1 scores, all at 99%. This shows how effective and efficient Lightweight LSTM can be as an intrusion detection system, especially in real-time and resource-constrained circumstances.LSTM-based customized architectures and lightweight deep learning models will continue to be essential for improving network security and defending against changing cyber threats. Our method offers a solid platform for developing increasingly more complex intrusion detection systems, supporting continuous efforts to guarantee the integrity and security of computer networks around the world.

REFERENCES

[1] P. Ambika, "Machine learning and deep learning algorithms on the Industrial Internet of Things (IIoT)," Advances in Computers, vol. 117, no. 1, pp. 321–338, 2020.

[2] R. Ashima, A. Haleem, S. Bahl, M. Javaid, S. K. Mahla, and S. Singh, "Automation and manufacturing of smart materials in Additive Manufacturing technologies using the Internet of Things towards the adoption of Industry 4.0," Materials Today: Proceedings, vol. 45, pp. 5081–5088, 2021.

[3] L. M. Gladence, V. M. Anu, R. Rathna, and E. Brumancia, "Recommender system for home automation using IoT and artificial intelligence," Journal of Ambient Intelligence and Humanized Computing, pp. 1–9, 2020.

[4] T. Sherasiya, H. Upadhyay, and H. B. Patel, "A survey: intrusion detection system for internet of things," International Journal of Computer Science and Engineering (IJCSE), vol. 5, no. 2, pp. 91–98, 2016.

[5] J. B. Awotunde, R. G. Jimoh, S. O. Folorunso, E. A. Adeniyi, K. M. Abiodun, and O. O. Banjo, "Privacy and security concerns in IoT-based healthcare systems," Internet of Things, pp. 105–134, 2021.

[6] E. A. Adeniyi, R. O. Ogundokun, and J. B. Awotunde, "IoMT-based wearable body sensors network healthcare monitoring system," in IoT in Healthcare and Ambient Assisted Living, pp. 103–121, Springer, Singapore, 2021.

[7] K. Amit and C. Chinmay, "Artificial intelligence and Internet of Things based healthcare 4.0 monitoring system," Wireless Personal Communications, pp. 1–14, 2021.

[8] F. E. Ayo, S. O. Folorunso, A. A. Abayomi-Alli, A. O. Adekunle, and J. B. Awotunde, "Network intrusion detection based on deep learning model optimized with rule-based hybrid feature selection," Information Security Journal: A Global Perspective, vol. 29, no. 6, pp. 267–283, 2020.

[9] Jayadurga, R. ., Veeramakali, T. ., Ali Sohail, M. ., Alangudi Balaji, N. ., Kumar C., K. ., & L. P., S. (2023). Deep Learning Based Detection and Classification of Anomaly Texts in Social Media. International Journal of Intelligent Systems and Applications in Engineering, 11(4s), 78–89. Retrieved from https://ijisae.org/index.php/IJISAE/article/view/2574

[10] S. N. Ajani and S. Y. Amdani, "Probabilistic path planning using current obstacle position in static environment," 2nd International Conference on Data, Engineering and Applications (IDEA), 2020, pp. 1-6, doi: 10.1109/IDEA49133.2020.9170727.

[11] M. Abdulraheem, J. B. Awotunde, R. G. Jimoh, and I. D. Oladipo, "An efficient lightweight cryptographic algorithm for IoT security," in Communications in Computer and Information Science, pp. 444–456, Springer, 2021.

[12] A. Bakhtawar, R. J. Abdul, C. Chinmay, N. Jamel, R. Saira, and R. Muhammad, "Blockchain and ANFIS empowered IoMT application for privacy preserved contact tracing in COVID-19 pandemic," Personal and Ubiquitous Computing, 2021.

[13] A. H. Muna, N. Moustafa, and E. Sitnikova, "Identification of malicious activities in industrial internet of things based on deep learning models," Journal of information security and applications, vol. 41, pp. 1–11, 2018.

[14] E. Sitnikova, E. Foo, and R. B. Vaughn, "The power of hands-on exercises in SCADA cybersecurity education," in Information Assurance and Security Education and Training, pp. 83–94, Springer, Berlin, Heidelberg, 2013.

[15] S. Dash, C. Chakraborty, S. K. Giri, S. K. Pani, and J. Frnda, "BIFM: big-data driven intelligent forecasting model for COVID-19," IEEE Access, vol. 9, pp. 97505–97517, 2021.

[16] G. Tzokatziou, L. A. Maglaras, H. Janicke, and Y. He, "Exploiting SCADA vulnerabilities using a human interface device," International Journal of Advanced Computer Science and Applications, vol. 6, no. 7, pp. 234–241, 2015.

[17] D. Kushner, "The real story of stuxnet," IEEE Spectrum, vol. 50, no. 3, pp. 48–53, 2013.

[18] P. W. Khan and Y. Byun, "A blockchain-based secure image encryption scheme for the industrial Internet of Things," Entropy, vol. 22, no. 2, p. 175, 2020.

[19] Q. Yan and F. R. Yu, "Distributed denial of service attacks in software-defined networking with cloud computing," IEEE Communications Magazine, vol. 53, no. 4, pp. 52–59, 2015.

[20] A. C. Enache and V. Sgârciu, "Anomaly intrusions detection based on support vector machines with an improved bat algorithm," in 2015 20th International Conference on Control Systems and Computer Science, pp. 317–321, Bucharest, Romania, May 2015.

[21] Prof. Romi Morzelona. (2019). Histogram Based Data Cryptographic Technique with High Level Security. International Journal of New Practices in Management and Engineering, 8(04), 08 - 14. https://doi.org/10.17762/ijnpme.v8i04.80

[22] O. Folorunso, F. E. Ayo, and Y. E. Babalola, "Ca-NIDS: a network intrusion detection system using combinatorial algorithm approach," Journal of Information Privacy and Security, vol. 12, no. 4, pp. 181–196, 2016.

[23] H. Zhang, D. D. Yao, N. Ramakrishnan, and Z. Zhang, "Causality reasoning about network events for detecting stealthy malware activities," Computers & Security, vol. 58, pp. 180–198, 2016.

[24] M. R. Kabir, A. R. Onik, and T. Samad, "A network intrusion detection framework based on Bayesian network using a wrapper approach," International Journal of Computer Applications, vol. 166, no. 4, pp. 13–17, 2017.

[25] Y. Hu, A. Yang, H. Li, Y. Sun, and L. Sun, "A survey of intrusion detection on industrial control systems," International Journal of Distributed Sensor Networks, vol. 14, no. 8, 2018.

[26] T. Cruz, L. Rosa, J. Proenca et al., "A cybersecurity detection framework for supervisory control and data acquisition systems," IEEE Transactions on Industrial Informatics, vol. 12, no. 6, pp. 2236–2246, 2016.

[27] Botha, D., Dimitrov, D., Popović, N., Pereira, P., & López, M. Deep Reinforcement Learning for Autonomous Robot Navigation. Kuwait Journal of Machine Learning, 1(3). Retrieved from http://kuwaitjournals.com/index.php/kjml/article/view/140

[28] J. Camacho, A. Pérez-Villegas, P. García-Teodoro, and G. Maciá-Fernández, "PCA-based multivariate statistical network monitoring for anomaly detection," Computers & Security, vol. 59, pp. 118–137, 2016.

[29] M. Grill, T. Pevný, and M. Rehak, "Reducing false positives of network anomaly detection by local adaptive multivariate smoothing," Journal of Computer and System Sciences, vol. 83, no. 1, pp. 43–57, 2017.

[30] L. A. Maglaras, J. Jiang, and T. J. Cruz, "Combining ensemble methods and social network metrics for improving accuracy of OCSVM on intrusion detection in SCADA systems," Journal of Information Security and Applications, vol. 30, pp. 15–26, 2016.

[31] R. O. Ogundokun, J. B. Awotunde, E. A. Adeniyi, and F. E. Ayo, "Crypto-Stegno based model for securing medical information on IOMT platform," Multimedia tools and applications, pp. 1–23, 2021.

[32] J. Soto and M. Nogueira, "A framework for resilient and secure spectrum sensing on cognitive radio networks," Computer Networks, vol. 115, pp. 130–138, 2017.

[33] Khetani, V. ., Gandhi, Y. ., Bhattacharya, S. ., Ajani, S. N. ., & Limkar, S. . (2023). Cross-Domain Analysis of ML and DL: Evaluating their Impact in Diverse Domains. International Journal of Intelligent Systems and Applications in Engineering, 11(7s), 253–262.