Journal of Intelligent Systems and Internet of Things
Vol. 18, No. 01, PP. 140-149, 2026
DOI: https://doi.org/10.54216/JISIoT.180110

A Distributed İntrusion Detection Using Long Short-Term Memory-Gradient Repeating Unit and Enhanced Density Peak Clustering for Real-Time Cyber Threat Detection

Wisam Ali Hussein Salman[1,*]
[1]Ministry of Education, Karbala, Iraq
Email: wissamali77@gmail.com

Abstract

Due to the huge number of devices that connect to Internet of Things (IoT) networks, these networks have become the main nerve of the organizations that use them due to the large services that the networks provide to companies. In recent years, the number of attacks targeting IoT networks to shut down or violate data privacy has increased, so system developers must build strong protection systems to keep those networks secure. Intrusion detection systems (IDS) and intrusion prevention systems (IPS) are one of the most promising protection systems in securing these networks, but they suffer from several challenges, including high false positive alarms (FPA) and false negative alarms (FNA), in addition to the difficulty of controlling the long-time chains of incoming and outgoing traffic in IoT networks. This paper presents a distributed intrusion detection system (DIDS) based on the use of deep learning algorithms, specifically the enhanced long short-term memory (LSTM) algorithm with the gradient repeating unit (GRU) algorithm, as well as the use of a modern dataset collected from real network data called CICIOT2023. To adjust the threshold and achieve a balanced approach to the detection of anomalies, a hybrid model of the Enhanced Peak Density (DPC) aggregation algorithm with ROC curve analysis was used. The proposed work's main innovation is the combination of top-k feature selection with a hybrid LSTM-GRU architecture optimized for imbalanced datasets using focal loss, SMOTE, and dynamic class weighting. As a result, the intrusion detection pipeline is strong and effective. To evaluate the functioning of the system, standard performance metrics such as AUC-ROC, accuracy, F1-score, and recall were used, as the proposed system proved to be a powerful solution to prevent complex attacks targeting IoT networks as well as the possibility of detecting rare and modern attacks. The proposed model achieved promising results with accurate results reaching (96.0%) and a false negative rate (FNR) of 0.049% and a false positive rate (FPR) of 0.014%.

Received: February 13, 2025 Revised: May 21, 2025 Accepted: July 03, 2025

Keywords: Intrusion detection system; Intrusion prevention system; Machine learning; Deep learning; Artificial intelligent; Long short-term memory; Gradient repeating unit

1. Introduction

The Internet of Things networks are essential partners for the foundations that use them. They are the lifeblood of the continuity of the work of these companies and their development due to the huge number of devices connected to these networks. At the same time, they are a source of worry for these institutions due to the development of attacks targeting them to stop the service or violate privacy [1]. To secure these networks, IDS & IPS systems play an essential role in protecting them by monitoring incoming and outgoing traffic and then detecting and preventing suspicious traffic [2]. However, traditional IDS suffer from their inability to detect advanced threats, and struggle to detect modern attacks [3]. In recent years, IDS systems have occurred that are based on machine learning and deep learning algorithms, leveraging the power of data-driven models to identify outgoing and incoming network traffic as normal or anomalous behavior [4]. Intrusion detection systems suffer from many challenges despite their

capabilities, especially systems that have the following challenges: being based on machine learning, working with high-dimensional data, with the sequential nature of network traffic, and suffering from an imbalance between classes. These challenges can affect the performance of detection models through high FPA & FNA and the inability to detect rare attacks [5]. The proposed work has a distributed intrusion detection system (DIDS) that uses advanced deep learning techniques based on integrated of long short-term memory (LSTM) with gated recurrent un monitor algorithms, to monitoring network traffic data and handling the complexities of modern network environments, ensuring robust and accurate intrusion detection, that making it an ideal system for detecting intrusions for anomalous and rare events in dynamic environments.

In the field of machine learning and artificial intelligence, serial data processing as a very important research field has occurred, especially for tasks that involve the analysis of time series, the handling of natural language, and detect of anomalies [6]. Traditional neural networks, such as feed Forward neural networks, are struggling to deal with time series effectively due to their inability to maintain information about previous inputs [7]. This challenge led to the development of recurrent neural networks (RNNS), which are specially designed to process time series by maintaining the "memory" previous inputs through hidden situations [8]. However, the basic recurrent neural network (RNN) algorithm suffers from many challenges, like a fading gradient problem that makes it difficult for the network to learn long-term dependances [9]. To solve this challenge, Hochreiter & Jurgen Schmidhuber presented LSTM algorithm in 1997 [10]. LSTM is a specific kind of RNN, which has an innovative gates technique to switch the info flow, that allows it to capture long-term outcomes in time sequence with a high efficiently [11]. The core novelty of Long Short-Term Memory algorithm implied in its capability to remember or forget the data selectively over long epochs, making it particularly appropriate for jobs that need considerate the context over the long series [12]. The next figure (1) illustrates the chief structural design of the LSTM algorithm [13].

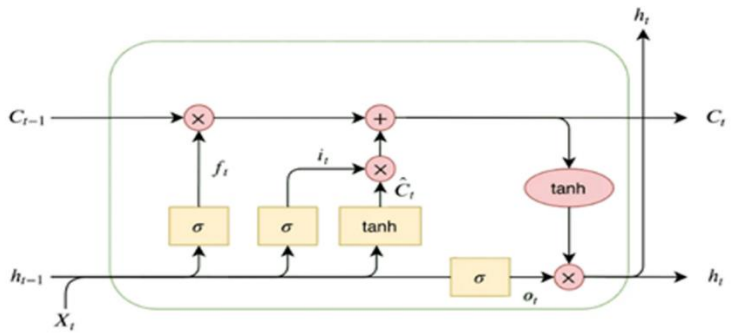

Figure 1. illustrates the chief structural design of the LSTM algorithm

Gated Recurrent Unit (GRU) Algorithm is a deep learning technique evolved from the Recurrent Neural Network (RNN) that is used to process texts and time serials, with the capacity to deal with vanishing or explosion difficulties in the gradients that the classic RNN algorithm suffers from [14]. Gated Recurrent Unit algorithm involves three chief parts [15], first part is the update gate: Its function determines the amount of information to be ignored as well as the amount of information to be kept. The second part is reset gate: Its function is to determine the amount of old information that must be ignored when calculating the new status. The third part is hidden Status: It is controlled based on the update gate and the reset gate to allow the network to retain important and long-term information. The subsequent figure (2) illustrates the focal design of the GRU technique [16].

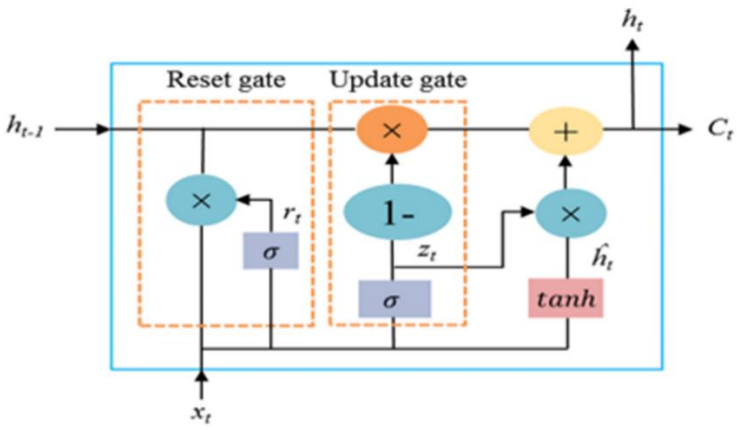

Figure 2. Illustrate the main architecture of the GRU algorithm

Although GRU is a powerful tool for processing long-term serial data, it suffers from some challenges, such as high memory consumption, the need for large quantities of data for training, relatively slow training, the inability to deal with non-serial data, and others [17].

The contribution of this paper lies in the hybridization of LSTM and GRU algorithms, secondly, employing the Custom Focal Loss for imbalanced anomaly detection, lastly, using the Intelligent feature selection and augmentation (SMOTE and SelectKBest).

This paper is organized as follows; section two is related work, methodology of the proposed work is shown in section three, section four shows the results. Lastly, the conclusion is represented in section five.

2. Related Work

The development of distributed intrusion detection systems (DIDS) that are useful from advanced machine learning (ML) and deep learning (DL) techniques has been a central point in cybersecurity research. The use of sequential models such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) for analyzing inbound and outbound network traffic has shown significant promise in detecting intrusion attacks. Below is a review of related work in this field, with reference to key studies that have contributed to the evolution of DIDS and the applied of LSTM/GRU models for intrusion detection and other techniques utilized in this proposed work. (Mateus et al.) presented a mechanism to integrate the gates that regulate the flow of data by combining the LSTM with GRU models to solve the fading gradient problem in the traditional RNN networks, where it has proven its strength to analyze the serial data with a lower calculation complexity when building intrusion detection systems, The proposed model suffers from inefficiency when generalized to different network environments, as well as increased resource consumption and delayed decision making [18].

Nosouhian et al. suggested of an intrusion detection model using the LSTM algorithm to detect network attacks with high accuracy in detecting anomalies, with the use of GRU models in processing serial network data, which has proven effective in preventing fading gradient with its ability to detect known and unknown attack patterns, The proposed model struggles to adapt to new attacks and challenges in parameter setting and optimization and requires large amounts of data [19]. Dealing with unbalanced data classes is a major challenge in the intrusion detection dataset, Sayegh et al. suggested the use of the SMOTE artificial sampling technology to balance the distribution of classes in the NSL-KDD dataset, this has enhanced the rates of detection of attack categories Rare, but this technique suffers from the risk of over-generating samples, does not address temporal data skewing, and increases computational time and complexity [20]. Similarly, Wu et al. has also illustrated that modifying classes during the model-training can enhance the discovery of rare attacks [21].

Moreover, feature selection and normalization are extremely important steps in analyzing the inbound and outbound of network traffic. Shakeela et al. suggested the use of ANOVA F technology for feature selection enhances the performance of the intrusion detection model significantly by removing frequent and unrelated features [22]. Moreover, Siraj et al. also used minimal scale technology and maximum normalization to ensure that all features are on a similar scale, which is very important to converging the deep learning models [23].

The threshold testing process is a critical factor in the efficiency of anomaly detection systems, particularly in the multiple classification stages. To build effective models, the techniques used have been improved with the discovery of new techniques, such as the use of the DPC clustering algorithm to predict density peaks and enhance threshold testing. For example, Carrington et al. a hybrid model utilizing the improved density peak clustering (DPC) algorithm with the ROC curve analysis technique was proposed, the proposed model proved to be more efficient in detecting anomalies accurately and robustly [24]. Furthermore, to accurately evaluate intrusion detection models, several parameters and indicators should be used to know the efficiency of the model, especially in imbalanced datasets. These indicators include accuracy, precision, recall, F1-SCORE, and AUC-ROC; Kumar suggested using a set of indicators and parameters that contribute to the evaluation of the model to ensure its effectiveness in detecting anomalies [25].

3. Proposed Methodology

This section provides a detailed explanation of the methodology used to build the distributed intrusion detection system (DIDS) based on Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) models proposed in this paper. The methodology is divided into several steps, including dataset preprocessing, feature selection, handling class imbalance, model architecture design, model training, and evaluation, it is illustrated below.

3.1. Dataset preprocessing

In proposed work, we used a modern dataset called CICIOT2023, which is a comprehensive dataset of network traffic in the real world; it contains all the scenarios of normal and abnormal traffic. Moreover, it contains a variety of features of approximately (47) features, such as protocol types, IP addresses, and size of the packets, the following table (1) illustrates the Features in dataset.

Table 1: illustrate the Features in dataset

flow_duration	Header_Length	Protocol Type	Duration	Drate
psh_flag_number	ack_flag_number	ece_flag_number	cwr_flag_number	fin_count
rst_count	HTTP	HTTPS	DNS	SSH
fin_flag_number	syn_flag_number	rst_flag_number	IRC	DHCP
ARP	ICMP	IPv	LLC	Max
AVG	Std	Tot size	IAT	Radius
Covariance	Variance	Weight	label	
Rate	TCP	Srate	UDP	
ack_count	Tot sum	syn_count	Min	
Telnet	Number	SMTP	Magnitue	

Moreover, the CICIOT2023 dataset contains (34) different attacks representative of real attacks that the IoT network was exposed to while collecting data in the real world, and it also represents most of the common attacks used by attackers (Yan et al., 2024). Table number (2) illustrates the kinds of attacks and the number of records in each type for a sample of data that included (1,000,000) records.

Table 2: illustrates the kinds of attacks in CICIOT2023 dataset

Class Type	No. of Records	Class Type	No. of Records	Class Type	No. of Records
DDoS-ICMP_Flood	153893	DoS-SYN_Flood	43132	DNS_Spoofing	4691
DDoS-UDP_Flood	115631	BenignTraffic	23320	DoS-HTTP_Flood	4299
DDoS-TCP_Flood	96640	Mirai-greeth_flood	21100	Recon-HostDiscovery	3475
DDoS-PSHACK_Flood	88089	Mirai-udpplain	19222	Recon-OSScan	3192
DDoS-SYN_Flood	87382	Mirai-greip_flood	16165	DictionaryBruteForce	1809
DDoS-RSTFINFlood	85479	DDoS-ICMP_Fragmentation	9744	Recon-PortScan	1775
DDoS-SynonymousIP_Flood	76916	MITM-ArpSpoofing	6664	VulnerabilityScan	959
DoS-UDP_Flood	71192	DDoS-ACK_Fragmentation	6148	DDoS-HTTP_Flood	598
DoS-TCP_Flood	57053	DDoS-UDP_Fragmentation	6137	DDoS-SlowLoris	466
SqlInjection	143	XSS	106	Backdoor_Malware	74

To ensure that the dataset is the network traffic in the real world, IP addresses are simulated with a random generator, as this step guarantees that the dataset represents the real network data.

3.1.1. Handling missing values

All the datasets may have lost some of it value; therefore, we must fill these values by using the deduction technique to be ready for use with machines and deep learning algorithms. This technique is based on using the means to compensate for the lost numerical values and using the appointed to compensate for the lost category values.

3.1.2. Encoding categorical features

To be the dataset suitable for use with machine learning algorithms; categorical values such as protocols (UDP, TCP, and ICMP) have been converted into numeric values by using one-hot encoding technique.

3.1.3. Extracting labels and ip addresses

To determine whether traffic is normal or malicious, target values are extracted from the dataset; moreover, more analysis is made by separating the IP addresses, such as identifying source and destination pairs.

3.1.4. Convert target to binary classification

To simplify the classification task and make it suitable for binary classification models, the target variable is converted into a binary format, as it represents (0) normal traffic and (1) is attack traffic.

3.1.5. Feature selection

To reduce the dimensions of the dataset, Anova F technology was used to choose the most important features depending on their statistical relationship with the target variable and focus on the most related features, thus improving the performance of the model. In proposed work, after we apply Anova F technology, the features that selected were (15) features shown in the following table number (3).

Table 3: illustrates the features selected with their scores

Feature No.	Feature	Score
3	Duration	382181.047738
39	Variance	313665.977278
16	rst_count	308387.041113
18	HTTPS	242732.626658
30	Max	239141.626783
32	Std	227647.472483
37	Radius	226901.739898
11	ack_flag_number	126314.754675
31	AVG	108488.594150
33	Tot size	108344.520252
1	Header_Length	107916.787101
28	Tot sum	100524.310277
36	Magnitue	97589.230740
38	Covariance	75914.802896
15	urg_count	58482.027574

3.1.6. Normalizing feature data using min-max scaling

In this technique, the Min-MAX scaling is used to normalize features and to ensure that all features are on a similar scale through using the following equation.

$$X_{normalized} = X - X_{min} * X_{max} - X_{min}. \quad (1)$$

Where:

$X_{normalized}$: represents the normalized value of (x), scale between (0,1).

X: sample value of the dataset.

$X_{min:}$ the minimum value in the dataset

X_{max}: the maximum value in the dataset.

Applying this technology is extremely important for the convergence of deep learning models and enhancing their performance.

3.1.7. Data splitting

To evaluate the performance of the model, the dataset will be split into two parts: a testing set (40%) and a training set (60%), to ensure a more accurate evaluation, the following table number (4) illustrate the number of records in training and testing dataset.

Table 4: illustrate number of records at training & testing dataset

Dataset records	
Training records	603475
Testing records	402317
Total records	1005792

3.1.8. Handling class imbalance using SMOTE

To address the imbalance in the classes, the method of Synthetic Minority Oversampling Technique (SMOTE) has been used on training data, which creates artificial samples for the synthetic by completing the existing samples, to ensure a balanced dataset.

3.1.9. Adjusted class weights

To enhance the model's ability to detect unusual attacks, the class weights are adjusted during training to ensure that the model pays more attention to the minority class.

3.1.10. Reshaping data for LSTM/GRU models

To ensure that the data collection is suitable for serial modeling using LSTM and GRU networks, training and testing data will be reformed to a 3D format; the form of entry data is (batch size, time steps, and features); moreover, timesteps will represent the sequence length.

3.2. Building the LSTM/GRU model

The proposed model in the research work consists of the following layers:

- **Input layer:** consist of 3D tensors with dimensions (batch size, timesteps, features), determine the expected input shape.
- **LSTM layer:** consist of 128 hidden units with return_sequences=True, this will ensure the hidden state is output for each time step, with apply L2 regularization ($\lambda = 0.001$) to prevent overfitting.
- **Apply batch normalization:** Apply batch normalization: to stabilize and speed up training, it will normalize the output of the LSTM layer.
- **GRU layer:** Contains 128 units for processing the output of the LSTM layer. L2 regularization and batch normalization are used again to improve stability and convergence speed.
- **Output layer:** consist of one dense layer with one unit; moreover, a sigmoid activation function given a probability among (0 and 1).

The improvement and contribution of the proposed model contain utilizing the LSTM/GRU model structural design, with 128 hidden units in each layer that established extraordinary ability in catching sequential dependencies in network traffic. The model's capacity for generalization and training stability was further improved by the application of L2 regularization and batch normalization.

3.3. Model training

The training procedure of the model involves the following phases:

- Adam Optimizer: we will use it for faster convergence.
- Focal Loss Function: we will use it to manipulate the class imbalance through focusing on hard-to-classify patterns.
- Early Stopping: we will use it to prevent the overfitting and monitoring validation loss, we will use the early stopping.
- Model Training: by using the model fit method, moreover, store the training history for analysis.

3.4. Threshold optimization.

- By combining the Enhanced Density Peak Clustering (DPC) algorithm with ROC curve analysis for threshold selection optimization, the second contribution improved the model's capacity to detect anomalies by ensuring a fair trade-off between precision and recall. The following actions will be taken to maximize the threshold:
- **Enhanced DPC algorithm:** This algorithm will be used to identify the local peaks and establish the ideal threshold.
- **ROC curve analysis:** based on the trade-off between the true positive rate (TPR) and false positive rate (FPR), to adjust the threshold.
- **Calculate the combined threshold:** To convert probabilities into binary predictions, the thresholds derived from improved DPC and ROC curve evaluation are merged (0 for normal, 1 for attack).

3.5. Model evaluation

Mutual performance metrics, including accuracy, precision, recall, F1-score, and AUC-ROC, are used to assess the system. Using these metrics would provide a comprehensive evaluation of the proposed model for monitoring network traffic and detecting normal and abnormal behavior.

4. Results

After designing and training the proposed model, a set of standard indicators and coefficients were used to examine the efficiency and effectiveness of the model for monitoring incoming and outgoing traffic in the network and detecting anomalies, including rare and modern attacks. Table no. (5) illustrates the results obtained after implementing the model, which are the basis for its evaluation.

Table 5: Illustrate the Results of Model Execution

Full Training Time	5618.59 seconds	Start Accuracy	0.9855	End Accuracy	0.9919
Average CPU Usage	0.18%	Start Val Loss	0.0233	End Val Loss	0.0119
Memory Usage	2947.72 MB	Start Val Accuracy	0.9766	End Val Accuracy	0.9848
FPR	0.0170	FNR	0.0058		

The following figure (3) illustrates model accuracy curve and model loss curve.

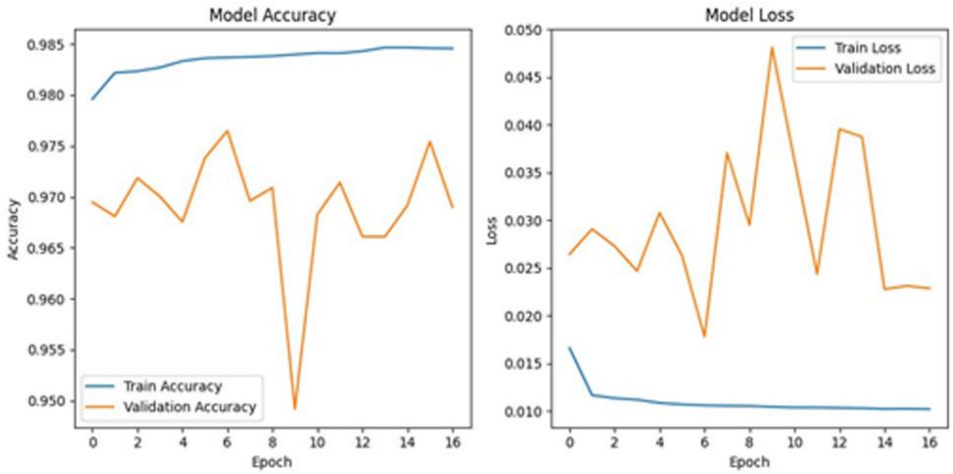

Figure 3. Depicts the model loss curve and accuracy curve

Additionally, the receiver operating characteristic (ROC) curve is shown in figure (4) below.

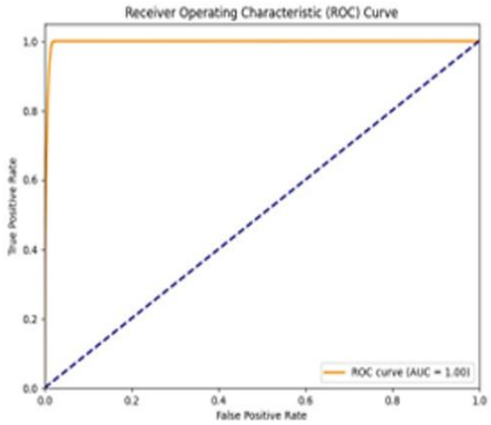

Figure 4. illustrates the receiver operating characteristic (ROC) curve

When we compare the proposed model with previously used techniques like Random Forest & XBoost, we notice that the model is ahead of them in the results we obtained. Table No. (6) Illustrate this.

Table 6: Illustrate compares the proposed model with previously used techniques like Random Forest & XBoost

metrics	**Proposed system**	**Random Forest**	**XBoost**
model type	sequential deep learning	traditional machine learning	advanced machine learning
accuracy	96% - 97%	93% - 95%	95% - 96%
precision	0.97	0.93	0.95
recall	0.95	0.93	0.94
f1-score	0.96	0.91	0.94
fpr	0.014	0.03	0.02
fnr	0.049	0.05	0.04
handle sequential data	excellent	not supported	not supported
sensitivity to imbalance	low	medium	low
resource needs	high	low	medium

5. Conclusion

Cyber threats are one of the most important challenges that IoT networks and associated devices face as they become more complex and sophisticated and, therefore, more harmful. Moreover, to meet these challenges, strong and effective protection systems must be developed that help stop and prevent these attacks completely. In paper, a distributed intrusion detection system (DIDS) based on the use of enhanced short-term long memory (LSTM) algorithms was presented in integration with the gated recurrent unit (GRU) algorithm. The proposed model was evaluated using a CICIOT2023 data set, and to increase the effectiveness and performance of the system, the focus was on testing and improving threshold adjustment using the enhanced DPC algorithm with the ROC curve. A clear methodology was also used to process the dataset before using it, such as data purification, feature selection, and solving the problem of category imbalance. The contribution of the proposed model can be the ability to handle

sequential time series and complex time patterns by hybridizing the two algorithms (LSTM & GRU), which leads to an increase in the intrusion detection accuracy, and the use of the focal loss function and the SMOTE technique was used to solve the data imbalance challenge to improve the detection accuracy of rare attacks. Moreover, the use of advanced data processing techniques like (fixed feature elimination, ANOVA F-Test), and MinMaxScaler was used to reduce noise and improve the model's ability to generalize. After the system was implemented, it showed promising results in detecting intrusion in the network model using approved metrics: accuracy of 96.0%, precision of 97.8%, recall of 96.7%, F1-score of 97.2%, and AUC-ROC of 99.1%. These results show that the system is powerful and efficient in distinguishing among normal traffic and anomalous traffic.

References

[1] S. Yaras and M. Dener, "IoT-Based Intrusion Detection System Using New Hybrid Deep Learning Algorithm," *Advances in Science and Technology Research Journal (ASTRJ)*, vol. 14, pp. 771-779, 2024. doi: 10.3390/electronics.

[2] M. Nuaimi, L. C. Fourati, and B. Ben Hamed, "Intelligent approaches toward intrusion detection systems for Industrial Internet of Things: A systematic comprehensive review," *Journal of Network and Computer Applications*, vol. 215, pp. 103-117, 2023. doi: 10.1016/j.jnca.2023.103637.

[3] S. H. Abbas, W. A. K. Naser, and A. A. Kadhim, "Intrusion Detection System (IDS) and Intrusion Prevention System (IPS)," *Global Journal of Engineering and Technology Advances*, vol. 14, no. 2, pp. 155-158, 2023.

[4] L. Yang and A. Shami, "An open source code for Intrusion Detection System development using Machine Learning," *Software Impacts*, vol. 14, 2022. doi: 10.1016/j.simpa.2022.100446.

[5] R. Khushal and U. Fatima, "Fuzzy machine learning logic utilization on hormonal imbalance dataset," *Computers in Biology and Medicine*, vol. 174, 2024. doi: 10.1016/j.compbiomed.2024.108429.

[6] G. Li and J. J. Jung, "Deep learning for anomaly detection in multivariate time series: Approaches, applications, and challenges," *Information Fusion*, vol. 91, pp. 93-102, 2023. doi: 10.1016/j.inffus.2022.10.008.

[7] W. Wang, Y.-j. Du, K.-w. Chau, C.-T. Cheng, D.-m. Xu, and W.-T. Zhuang, "Evaluating The Performance of Several Data Preprocessing Methods Based On GRU in Forecasting Monthly Runoff Time Series," *Advances in Science and Technology Research Journal (ASTRJ)*, vol. 3, 2021. doi: 10.21203/rs.3.rs-868259/v1.

[8] N. F. Syed, M. Ge, and Z. Baig, "Fog-cloud based intrusion detection system using Recurrent Neural Networks and feature selection for IoT networks," *Computer Networks*, vol. 225, 2023. doi: 10.1016/j.comnet.2023.109662.

[9] S. M. Kasongo, "A deep learning technique for intrusion detection system using a Recurrent Neural Networks based framework," *Computer Communications*, vol. 199, pp. 113-125, 2023. doi: 10.1016/j.comcom.2022.12.010.

[10] S. Hochreiter and J. Schmidhuber, "LONG SHORT-TERM MEMORY," *Neural Computation*, vol. 9, no. 8, pp. 1735-1780, 1997.

[11] M. Moukhafi, M. Tantaoui, I. Chana, and A. Bouazi, "Intelligent intrusion detection through deep autoencoder and stacked long short-term memory," *International Journal of Electrical and Computer Engineering*, vol. 14, no. 3, pp. 2908-2917, 2024. doi: 10.11591/ijece.v14i3.pp2908-2917.

[12] S. E. Vadakkethil, K. Polimetla, S. Velpula, P. K. Pareek, and D. Sontakke, "Improved Whale Optimization Algorithm and Optimized Long Short-Term Memory for DDoS Cyber Security Threat," in *Proceedings of the Third International Conference on Distributed Computing and Electrical Circuits and Electronics*, 2024.

[13] A. Gueriani, H. Kheddar, and A. C. Mazari, "Enhancing IoT Security with CNN and LSTM-Based Intrusion Detection Systems," *IEEE*, vol. 1, 2024. doi: http://arxiv.org/abs/2405.18624.

[14] S. Ma et al., "Improved Seagull Optimization Algorithm to Optimize Neural Networks with Gated Recurrent Units for Network Intrusion Detection," in *Proceedings of the 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, IDAACS, 2021, pp. 100–104. doi: 10.1109/IDAACS53288.2021.9660898.

[15] W. Wang, Y.-j. Du, K.-w. Chau, C.-T. Cheng, D.-m. Xu, and W.-T. Zhuang, "Evaluating The Performance of Several Data Preprocessing Methods Based On GRU in Forecasting Monthly Runoff Time Series," *Research Square*, 2021. doi: 10.21203/rs.3.rs-868259/v1.

[16] S. M., "Recognition of human activity using GRU deep learning algorithm," *Multimedia Tools and Applications*, vol. 82, no. 30, pp. 47733–47749, 2023. doi: 10.1007/s11042-023-15571-y.

[17] S. Nosouhian, F. Nosouhian, and A. K. Khoshouei, "A Review of Recurrent Neural Network Architecture for Sequence Learning: Comparison between LSTM and GRU," *Preprints*, vol. 1, 2021. doi: 10.20944/preprints202107.0252.v1.

[18] B. C. Mateus, M. Mendes, J. T. Farinha, and R. Assis, "Comparing LSTM and GRU models to predict the condition of a pulp paper press," *Energies*, vol. 14, no. 21, 2021. doi: 10.3390/en14216958.

[19] S. Nosouhian, F. Nosouhian, and A. K. Khoshouei, "A Review of Recurrent Neural Network Architecture for Sequence Learning: Comparison between LSTM and GRU," *Preprints*, vol. 23, 2021. doi: 10.20944/preprints202107.0252.v1.

[20] H. R. Sayegh, W. Dong, and A. M. Al-madani, "Enhanced Intrusion Detection with LSTM-Based Model, Feature Selection, and SMOTE for Imbalanced Data," *Applied Sciences (Switzerland)*, vol. 14, no. 2, 2024. doi: 10.3390/app14020479.

[21] W.-c. Wang, Y.-j. Du, K.-w. Chau, C.-T. Cheng, D.-m. Xu, and W.-T. Zhuang, "Evaluating The Performance of Several Data Preprocessing Methods Based On GRU in Forecasting Monthly Runoff Time Series," *Research Square*, 2021. doi: 10.21203/rs.3.rs-868259/v1.

[22] S. Shakeela, N. S. Shankar, P. M. Reddy, T. T. T. Tulasi, and M. M. Koneru, "Optimal ensemble learning based on distinctive feature selection by univariate ANOVA-F statistics for IDS," *International Journal of Electronics and Telecommunications*, vol. 67, no. 2, pp. 267–275, 2021. doi: 10.24425/ijet.2021.135975.

[23] M. J. Siraj, T. Ahmad, and R. M. Ijtihadie, "Analyzing ANOVA F-test and Sequential Feature Selection for Intrusion Detection Systems," *International Journal of Advances in Soft Computing and Its Applications*, vol. 14, no. 2, pp. 185–194, 2022. doi: 10.15849/IJASCA.220720.13.

[24] A. M. Carrington, D. G. Manuel, P. W. Fieguth, T. Ramsay, and V. Osmani, "Deep ROC Analysis and AUC as Balanced Average Accuracy, for Improved Classifier Selection, Audit and Explanation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 1, pp. 329–341, 2023. doi: 10.1109/TPAMI.2022.3145392.

[25] G. Kumar, "Evaluation Metrics for Intrusion Detection Systems-A Study," *International Journal of Computer Science and Mobile Applications*, vol. 2, pp. 11–17, 2014.