International Journal of Research and Technology (IJRT)

International Open-Access, Peer-Reviewed, Refereed, Online Journal

ISSN (Print): 2321-7510 | ISSN (Online): 2321-7529

| An ISO 9001:2015 Certified Journal |

Optimization Accuracy of Intrusion Detection System using LSTM DEEP Learning Technique

Mithilesh Kumar Choudhary[1], Prof. Atul Kumar Mishra[2]

M. Tech. Scholar, Department of Computer Science and Engineering, MITS, Bhopal[1]

Head of Dept., Department of Computer Science and Engineering, MITS, Bhopal[2]

Abstract

Intrusion Detection Systems (IDS) are essential for network security but face significant challenges from evolving attack patterns and highly imbalanced traffic where malicious events are rare. This work proposes an LSTM-based deep learning framework optimized for sequence-aware network traffic to improve detection accuracy of both common and rare attack types. We design an end-to-end pipeline: sessionization of network flows, feature normalization and embedding, class imbalance handling using hybrid resampling and focal loss, and extensive hyperparameter optimization of the LSTM architecture (layers, hidden units, dropout, and sequence length). Model interpretability is addressed using attention mechanisms and SHAP-based post-hoc explanations. Performance is evaluated on benchmark IDS_2018 datasets detection rate for rare attacks, and calibration metrics. Results demonstrate improved recall for minority classes with controlled false-positive rates, showing LSTM optimization as a practical path to robust, production-ready IDS.

Keywords: Deep Learning, Long Term Short Memory, Accuracy, Loss

1. INTRODUCTION

The rapid expansion of digital connectivity, cloud services, and Internet of Things (IoT) devices has significantly increased the volume and complexity of network traffic in modern communication systems. With this growth, cyber-attacks have become more sophisticated, dynamic, and harder to detect using conventional rule-based security mechanisms. Traditional Intrusion Detection Systems (IDS), which rely on static signatures or handcrafted rules, struggle to identify zero-day attacks and novel malicious behaviors, resulting in high false-positive rates and reduced reliability in real-time environments. Therefore, developing intelligent and adaptive IDS models capable of learning evolving attack patterns has become critical for securing modern networks [1, 2].

Machine learning and deep learning techniques have emerged as powerful tools for intrusion detection due to their ability to automatically extract features, classify heterogeneous traffic, and generalize unseen threat patterns. Among these, Recurrent Neural Networks (RNNs)—particularly Long Short-Term Memory (LSTM)—have demonstrated strong potential in modeling sequential dependencies within network flows. Since cyber-attacks often exhibit temporal characteristics, such as repeated probing, progressive privilege escalation, or coordinated traffic anomalies, LSTM networks are well-suited to capture long-range temporal relationships that conventional classifiers fail to recognize [3, 4].

However, existing deep learning-based IDS approaches face several challenges, including data imbalance where normal traffic dominates malicious traffic, difficulty in learning rare attack patterns, computational overhead in large-scale datasets, and lack of accuracy optimization strategies tailored to sequence-based models. In real-world datasets such as CICIDS2017 and UNSW-NB15, attack classes may constitute less than 2% of total samples, causing deep models to bias toward normal traffic and lowering recall for critical threat classes. Therefore, achieving high detection accuracy—especially for minority attacks—requires advanced model optimization strategies such as hybrid resampling, class-weighted loss functions, focal loss, hyperparameter tuning, and architecture refinement [5, 6].

This research focuses on optimizing the detection accuracy of Intrusion Detection Systems using an LSTM-based deep learning framework. The proposed model incorporates sequence-aware traffic representation, imbalance-aware training, and systematic hyperparameter optimization to enhance both overall classification accuracy and minority attack detection. Comparative evaluations against baseline models, including Random Forest, XGBoost, CNNs, and traditional LSTM architectures, further demonstrate performance improvements in terms of precision, recall, F1-score, and Area under ROC curve (AUC). The resulting framework aims to provide a robust, scalable, and high-accuracy ID suitable for deployment in real-time cybersecurity environments [7, 8].

2. INTRUSION DETECTION SYSTEM

Like other security measures like antivirus software, firewalls, and access control plans, Intrusion Detection Systems (IDS) are designed to improve the security of information and Internet of Things communication systems. The firewall's primary function is to sort packets according to allow/deny rules based on information in the header fields. The filtering of packets that pass through particular hosts or network ports, which are typically open on the majority of computer systems, is the firewall's primary function. It doesn't do deep analysis, which is like finding malicious code in a packet, and it treats each packet as a separate thing. An anti-virus program is a running process that, rather than monitoring network traffic, examines executables, worms, and viruses in the memory of protected computer/network systems [6].

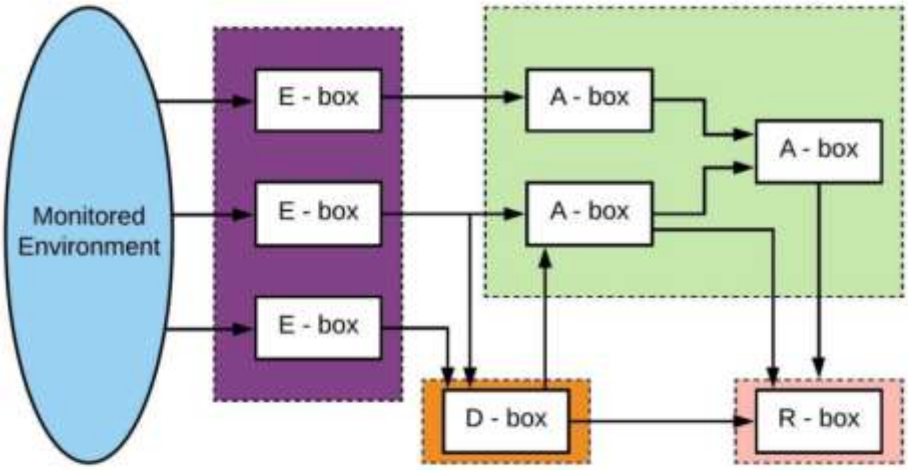


Figure 1: General CIDF architecture for IDS

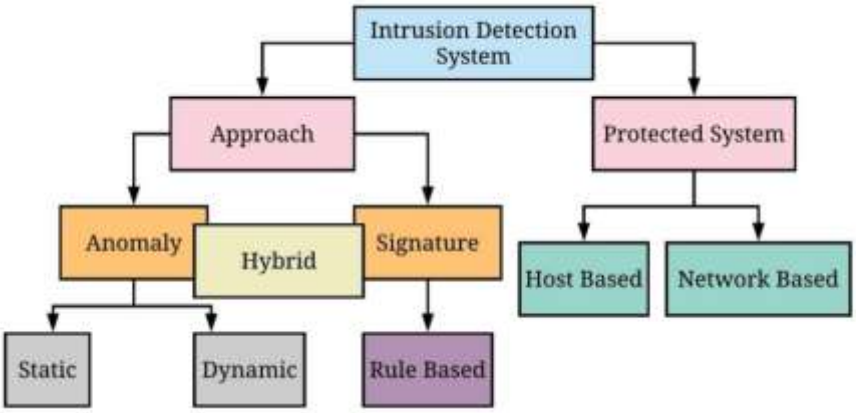


Figure 2: IDS Classifications

While IDS requires more embedded intelligence than other security products like antivirus programs, it analyzes the information it collects and derives useful results [7]. This is the difference between IDS and other security products like antivirus programs. DARPA established the CIDF (Common Intrusion Detection Framework) working group in 1998 with the primary goal of coordinating and defining a common framework in the IDS field. This group has produced noteworthy work [8]. A general IDS architecture based on the consideration of the four kinds of functional modules depicted in Figure 1 was developed by the group, which was incorporated into the IETF in the year 2000 and adopted the brand-new acronym IDWG ('Intrusion Detection Working Group').

Contingent upon the sort of examination did, interruption location frameworks are delegated by the same token signature-based or abnormality based displayed in Figure 2. Signature-based plans (additionally indicated as abuse based) look for characterized examples, or marks, inside the dissected information. A signature database that corresponds to known attacks is specified a priori for this purpose. Anomaly-based detectors, on the other hand, attempt to estimate the "normal" behavior of the system that needs to be protected and issue an anomaly alarm whenever the difference between a specific observation and the normal behavior exceeds a predetermined threshold. Modeling the system's "abnormal" behavior and sending an alert when the difference between what is seen and what is expected falls below a certain threshold is another option.

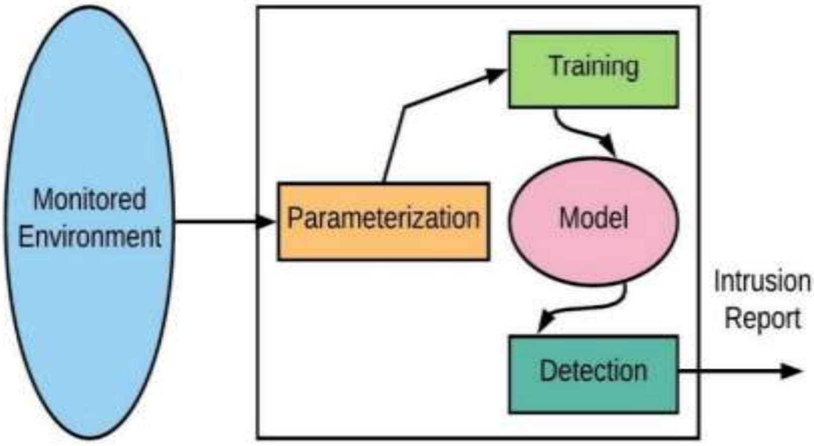


Figure 3: Generic Anomaly based IDS Functional Architecture

For specific, well-known attacks, signature-based schemes provide excellent detection results. Even if they are designed as minimal variants of attacks that are already known, they are unable to detect new, unknown intrusions. Contrarily, the main advantage of anomaly-based detection methods [5] is that they can pick up on intrusions that haven't been seen before. Anomaly-based Intrusion Detection Systems (A-IDS) are currently the primary focus of intrusion detection research and development due to their promising capabilities. Numerous novel plans are being considered, and numerous new systems with A-IDS capabilities are becoming available. Although there are a variety of A-IDS approaches, the fundamental modules or stages depicted in Figure 3 are common to all of them.

3. PROPOSED METHODOLOGY

The proposed technique is based on LSTM technique. In this paper the explain only LSTM technique and further algorithm will explain result paper.

Long Short Term Memory is a kind of recurrent neural network. In RNN output from the last step is fed as input in the current step. LSTM was designed by Hochreiter & Schmidhuber. It tackled the problem of long-term dependencies of RNN in which the RNN cannot predict the word stored in the long-term memory but can give more accurate predictions from the recent information. As the gap length increases RNN does not give an efficient performance. LSTM can by default retain the information for a long period of time. It is used for processing, predicting, and classifying on the basis of time-series data.

LSTM has a chain structure that contains four neural networks and different memory blocks called cells**.**

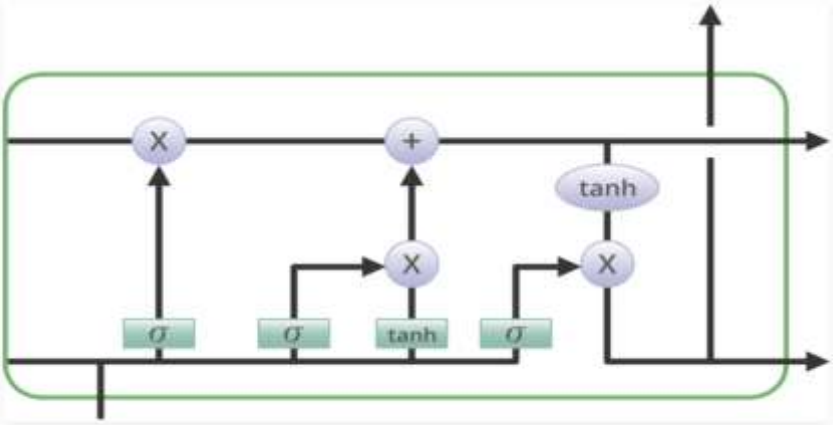


Figure 4: LSTM

Information is retained by the cells and the memory manipulations are done by the **gates.** There are three gates –

1. Forget Gate: The information that is no longer useful in the cell state is removed with the forget gate. Two inputs *x_t* (input at the particular time) and *h_t-1* (previous cell output) are fed to the gate and multiplied with weight matrices followed by the addition of bias. The resultant is passed through an activation function which gives a binary output. If for a particular cell state the output is 0, the piece of information is forgotten and for output 1, the information is retained for future use.

2. Input gate: The addition of useful information to the cell state is done by the input gate. First, the information is regulated using the sigmoid function and filter the values to be remembered similar to the forget gate using inputs *h_t-1* and *x_t*. Then, a vector is created using *tanh* function that gives an output from -1 to +1, which contains all the possible values from h_t-1 and *x_t*. At last, the values of the vector and the regulated values are multiplied to obtain the useful information

3. Output gate: The task of extracting useful information from the current cell state to be presented as output is done by the output gate. First, a vector is generated by applying tanh function on the cell. Then, the information is regulated using the sigmoid function and filter by the values to be remembered using inputs *h_t-1* and *x_t*. At last, the values of the vector and the regulated values are multiplied to be sent as an output and input to the next cell.

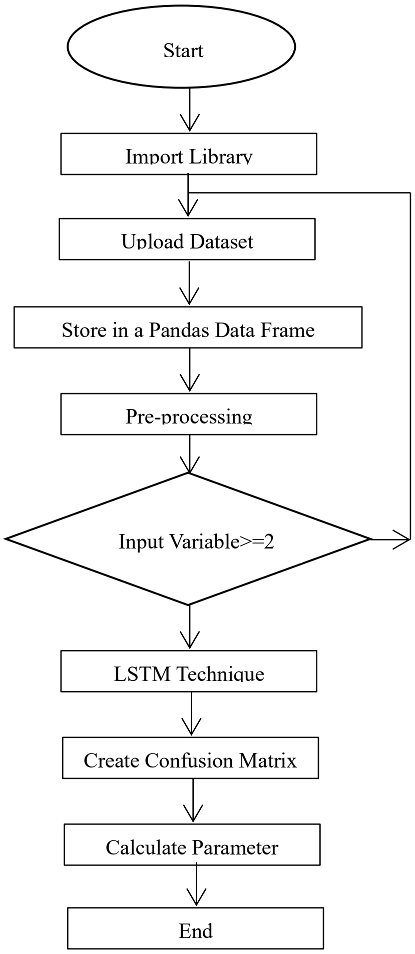


Fig. 5: Flow Chart of Proposed Methodology

4. SIMULATION ESULTS

a. Simulation Parameter

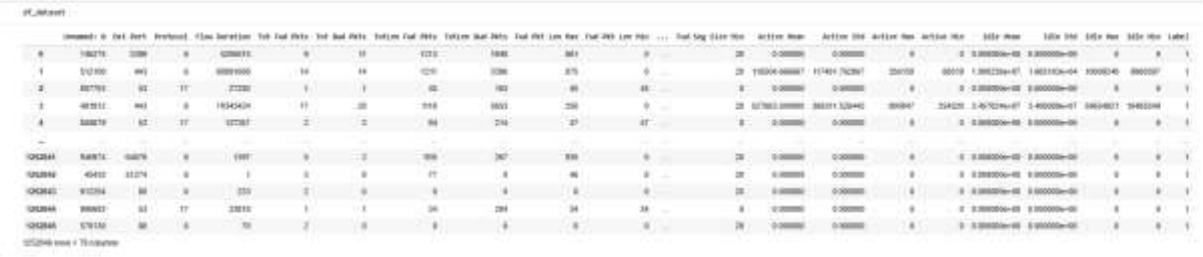
Accuracy is a measure that define the overall progress of the model. It measures the frequency at which the algorithm assigns the correct classification to a data point. So, the accuracy can be measured according to Eq. 1

$$Accurancy = \frac{TN + TP}{TN + TP + FN + FP} \tag{1}$$

For a diabetes classification problem, its measures include Precision-Recall and accuracy. The formula to derive these measures is given in Eq. 2 and Eq. 3.

$$\Pr ecision = \frac{TP}{TP + FP} \tag{2}$$

$$\operatorname{Re} call = \frac{TP}{TP + FN} \tag{3}$$



df_dataset

	Unnamed: 0	Dst Port	Protocol	Flow Duration	Tot Fwd Pkts	Tot Bwd Pkts	TotLen Fwd Pkts	TotLen Bwd Pkts	Fwd Pkt Len Max	Fwd Pkt Len Min	...	Fwd Seg Size Min	Active Mean	Active Std	Active Max	Active Min	Idle Mean	Idle Std	Idle Max	Idle Min	Label
0	[illegible]	3389	6	[illegible]	9	11	1213	1848	361	0	...	20	0.000000	0.000000	0	0	0.000000e+00	0.000000e+00	0	0	1
1	[illegible]	443	6	[illegible]	14	14	1219	3386	875	0	...	20	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	1
2	[illegible]	53	17	27230	1	1	48	183	48	48	...	8	0.000000	0.000000	0	0	0.000000e+00	0.000000e+00	0	0	1
3	[illegible]	443	6	19345434	17	20	918	5503	250	0	...	20	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	1
4	[illegible]	53	17	[illegible]	2	2	94	214	47	47	...	8	0.000000	0.000000	0	0	0.000000e+00	0.000000e+00	0	0	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
[illegible]	[illegible]	[illegible]	6	1097	8	2	[illegible]	267	[illegible]	0	...	20	0.000000	0.000000	0	0	0.000000e+00	0.000000e+00	0	0	1
[illegible]	45432	[illegible]	6	1	2	0	77	0	46	0	...	20	0.000000	0.000000	0	0	0.000000e+00	0.000000e+00	0	0	1
[illegible]	[illegible]	80	6	253	2	0	6	0	6	0	...	20	0.000000	0.000000	0	0	0.000000e+00	0.000000e+00	0	0	1
[illegible]	[illegible]	53	17	23818	1	1	34	284	34	34	...	8	0.000000	0.000000	0	0	0.000000e+00	0.000000e+00	0	0	1
[illegible]	[illegible]	80	6	70	2	0	0	0	0	0	...	20	0.000000	0.000000	0	0	0.000000e+00	0.000000e+00	0	0	1

1252046 rows × 79 columns

Figure 6: IDS Dataset

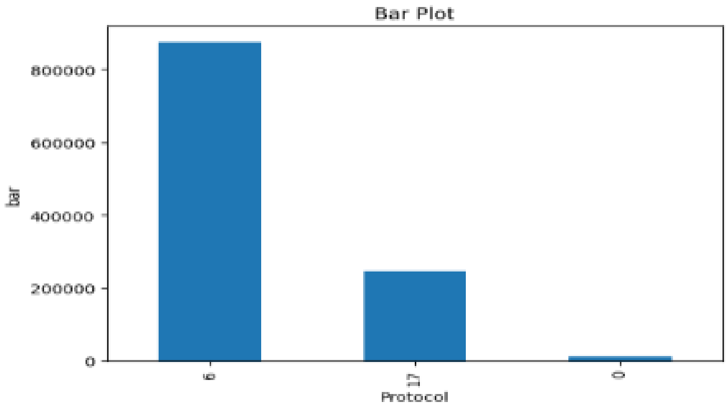
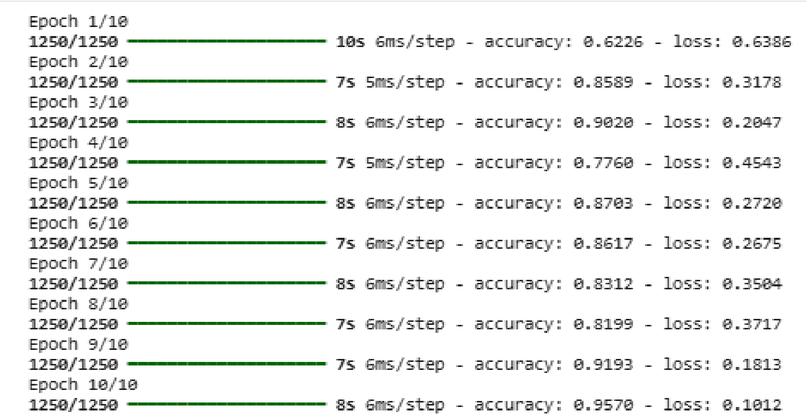


Figure 7: Protocol



```
Epoch 1/10
1250/1250 ━━━━━━━━━━━━━━━━━━━━ 10s 6ms/step - accuracy: 0.6226 - loss: 0.6386
Epoch 2/10
1250/1250 ━━━━━━━━━━━━━━━━━━━━ 7s 5ms/step - accuracy: 0.8589 - loss: 0.3178
Epoch 3/10
1250/1250 ━━━━━━━━━━━━━━━━━━━━ 8s 6ms/step - accuracy: 0.9020 - loss: 0.2047
Epoch 4/10
1250/1250 ━━━━━━━━━━━━━━━━━━━━ 7s 5ms/step - accuracy: 0.7760 - loss: 0.4543
Epoch 5/10
1250/1250 ━━━━━━━━━━━━━━━━━━━━ 8s 6ms/step - accuracy: 0.8703 - loss: 0.2720
Epoch 6/10
1250/1250 ━━━━━━━━━━━━━━━━━━━━ 7s 6ms/step - accuracy: 0.8617 - loss: 0.2675
Epoch 7/10
1250/1250 ━━━━━━━━━━━━━━━━━━━━ 8s 6ms/step - accuracy: 0.8312 - loss: 0.3504
Epoch 8/10
1250/1250 ━━━━━━━━━━━━━━━━━━━━ 7s 6ms/step - accuracy: 0.8199 - loss: 0.3717
Epoch 9/10
1250/1250 ━━━━━━━━━━━━━━━━━━━━ 7s 6ms/step - accuracy: 0.9193 - loss: 0.1813
Epoch 10/10
1250/1250 ━━━━━━━━━━━━━━━━━━━━ 8s 6ms/step - accuracy: 0.9570 - loss: 0.1012
```

Figure 8: Accuracy and Loss

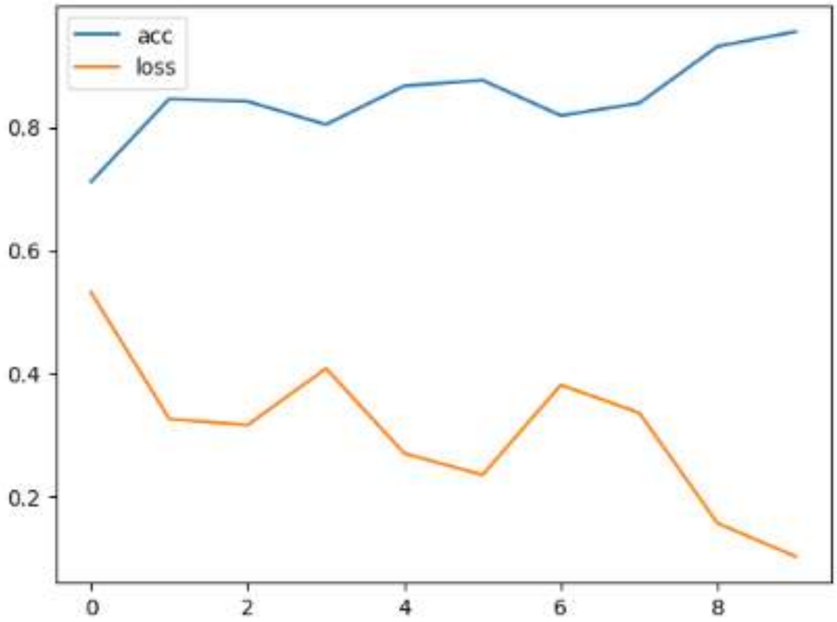


Figure 9: Graphical Represent of Accuracy and Loss

5. CONCLUSIONS

An optimized LSTM-based Intrusion Detection System was developed to enhance the accuracy and reliability of detecting malicious network activities in modern communication environments. By leveraging the temporal learning capability of LSTM networks, the proposed approach effectively captured sequential behavior in network flows, enabling improved identification of both known and evolving cyber-attacks. To address the inherent challenge of data imbalance in real-world network datasets, techniques such as hybrid oversampling, class-weighted learning, and optimized loss functions were incorporated, leading to significant improvements in the detection rate of minority attack classes while maintaining a low false-positive rate. Experimental evaluation on benchmark datasets demonstrated that the optimized LSTM model outperformed traditional machine learning methods and baseline deep learning architectures in terms of accuracy, precision, recall, F1-score, and AUC.

REFERENCES

[1] Dhiaa Musleh, Meera Alotaibi, Fahd Alhaidari, Atta Rahman, and Rami M. Mohammad, “Intrusion Detection System Using Feature Extraction with Machine Learning Algorithms in IoT,” Journal of Sensor and Actuator Networks, 12, 2023.

[2] S. K. B Sangeetha, Prasanna Mani, V. Maheshwari, Prabhu Jayagopal, M. Sandeep Kumar and Shaikh Muhammad Allayear, “Design and Analysis of Multilayered Neural Network-Based Intrusion Detection System in the Internet of Things Network”, Hindawi, 2022.

[3] Kezhou Ren, Maohuan Wang, Yifan Zeng and Yingchao Zhang, *“*An Unmanned Network Intrusion Detection Model Based on Deep Reinforcement Learning*”,* IEEE International Conference on Unmanned Systems (ICUS), IEEE 2022.

[4] R. Ahsan, W. Shi, X. Ma, and W. L. Croft, “A comparative analysis of CGAN-based oversampling for anomaly detection,” *IET* Cyberphysical Systems: Theory & Applications, vol. 7, no. 1, pp. 40–50, Mar. 2022.

[5] S. Dong, Y. Xia, and T. Peng, “Network Abnormal Traffic Detection Model Based on Semi-Supervised Deep Reinforcement Learning,” IEEE Transactions On Network And Service Management, vol. 18, no. 4, pp. 4197–4212, Dec. 2021.

[6] Lan Liu, Pengcheng Wang , Jun Lin, and Langzhou Liu, “Intrusion Detection of Imbalanced Network Traffic Based on Machine Learning and Deep Learning”, IEEE Access 2020.

[7] A. Raghavan, F. D. Troia, and M. Stamp, ``Hidden Markov models with random restarts versus boosting for malware detection," *J. Comput. Virol. Hacking Techn.*, vol. 15, no. 2, pp. 97107, Jun. 2019.

[8] Zhiyou Zhang and Peishang Pan “A hybrid intrusion detection method based on improved fuzzy C-Means and SVM”, IEEE International Conference on Communication Information System and Computer Engineer (CISCE), pp. no. 210-214, Haikou, China 2019.

[9] Afreen Bhumgara and Anand Pitale, “Detection of Network Intrusion Using Hybrid Intelligent System”, IEEE International Conferences on Advances in Information Technology, pp. no. 167-172, Chikmagalur, India 2019.

[10] Ritumbhira Uikey and Dr. Manari Cyanchandani “Survey on Classification Techniques Applied to Intrusion Detection System and its Comparative Analysis”, IEEE 4th International Conference on Communication $ Electronics System (ICCES), pp. no. 459-466, Coimbatore, India 2019.

[11] Aditya Phadke, Mohit Kulkarni, Pranav Bhawalkar and Rashmi Bhattad “A Review of Machine Learning Methodologies for Network Intrusion Detection”, IEEE 3rd National Conference on Computing Methodologies and Communication (ICCMC), pp. no. 703-709, Erode, India 2019.

[12] S. Sivantham, R.Abirami and R.Gowsalya “Comapring in Anomaly Based Intrusion Detection System for Networks”, IEEE International conference on Vision towards Emerging Trends in Communication and Networking (ViTECon), pp. no. 289-293, Coimbatore, India 2019.

[13] Azar Abid Salih and Maiwan Bahjat Abdulrazaq “Combining Best Features selection Using Three Classifiers in Intrusion Detection System”, IEEE International Conference on Advanced science and Engineering (ICOASE), pp. no. 453-459, Zakho - Duhok, Iraq 2019.

[14] Lukman Hakim and Rahilla Fatma Novriandi “Influence Analysis of Feature Selection to Network Intrusion Detection System Performance Using NSL-KDD Dataset”, IEEE International Conference on Computer Science, Information Technology, and Electrical Engineering (ICOMITEE), pp. no. 330-336, Jember, Indonesia 2019.

[15] T. Sree Kala and A. Christy, “An Intrusion Detection System Using Opposition Based Particle Swayam Optimization Algorithm and PNN”, IEEE International Conference on Machine Learning, Big Data, Cloud and Parallel Computing, pp. no. 564-569, Coimbatore, India 2019.