IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 6, June 2025

DOI: 10.17148/IJARCCE.2025.14660

# Intrusion Detection System Using Machine Learning and Deep Learning Techniques

**Prathamesh Margale[1], Shreya Kadam[2], Atharva Kakade[3], Prasad Papade[4] and Prof. Naved Raza Q. Ali[5]**

Undergraduate Research Paper, Department of Computer Engineering, SKNCOE, Savitribai Phule Pune University, Pune, India[1,2,3,4]

Department of Computer Engineering, SKNCOE, Savitribai Phule Pune University, Pune, India[5]

**Abstract**: Intrusion Detection Systems (IDS) are critical for mitigating evolving cybersecurity threats. This study investigates the integration of Machine Learning (ML) and Deep Learning (DL) techniques to enhance IDS efficiency. A dual-panel IDS is developed, incorporating an attack detection module for user uploads and an admin panel for model training and testing. The system leverages multiple classification algorithms, including Support Vector Machine (SVM), Random Forest, XGBoost, AdaBoost, and Decision Tree, to improve intrusion detection accuracy. A dynamic model selection mechanism is implemented to optimize algorithm performance at runtime, complemented by graphical visualizations for comprehensive threat analysis. Various IDS datasets are evaluated to assess detection effectiveness, addressing challenges such as computational complexity and real-time traffic management. Experimental results indicate an accuracy range of 92% to 96%, with Random Forest and Decision Tree performing optimally based on dataset characteristics. This research contributes to the advancement of IDS by improving detection reliability, reducing false positives, and enhancing system scalability, ultimately strengthening cybersecurity defenses.

**Keywords:** IDS, ML, DL, Network Security, Random Forest, SVM, Cybersecurity, Anomaly Detection, False Positives, Scalability, Accuracy, XGBoost, Decision Tree.

## I. INTRODUCTION

With the rapid expansion of digital infrastructures and the growing dependence on networked systems, cybersecurity has become a critical concern for organizations, governments, and individuals. Malicious activities such as unauthorized access, data breaches, denial-of-service (DoS) attacks, and malware infections pose serious threats to the integrity, confidentiality, and availability of information systems. In response to these evolving threats, Intrusion Detection Systems (IDS) have emerged as vital components of modern cybersecurity frameworks. These systems continuously monitor network traffic or host activities to detect potential intrusions.

At their core, Intrusion Detection Systems (IDS) are designed to identify abnormal or malicious activity by analysing patterns or behaviours. Based on their detection techniques, IDS can be broadly classified into two categories: misuse detection and anomaly detection. Misuse detection, also known as signature-based detection, compares observed behaviour against known attack patterns stored in a database. While effective against recognized threats, it is limited in its ability to identify novel or previously unseen attacks. On the other hand, anomaly detection uses statistical or machine learning models to establish a baseline of normal behaviour, flagging any significant deviations as potential threats. Although this approach can detect new attack types, it is susceptible to false positives if not carefully trained and tuned.

As cyber threats become more advanced and dynamic, traditional IDS solutions face significant limitations in scalability, adaptability, and detection accuracy. The rise of zero-day attacks, polymorphic malware, and complex attack vectors has created a need for intelligent, data-driven security mechanisms. Machine learning (ML) has emerged as a powerful solution, enabling IDS to automatically learn from historical data, adapt to new threat patterns, and make accurate predictions in real time with minimal human intervention.

This research presents the development of a machine learning-based Intrusion Detection System (IDS) with a dual-panel architecture. The system is divided into two main components: a user-facing panel that allows users to upload files for attack detection, and an admin panel that enables system administrators to train, test, and manage machine learning

  

DOI: 10.17148/IJARCCE.2025.14660

models. A comprehensive set of ML algorithms are implemented, including Support Vector Machine (SVM), Random Forest, XGBoost, and Decision Tree.

To enhance system performance and adaptability, a dynamic model selection mechanism is integrated, allowing the system to choose the most suitable algorithm at runtime based on the input data characteristics. This ensures that the most effective model is applied for a given scenario, optimizing both speed and accuracy. Furthermore, the system includes graphical visualizations of detected attacks, providing intuitive insights for monitoring and decision-making.

Experimental results indicate promising outcomes, with detection accuracy ranging from 92% to 96%. Among the tested models, Random Forest and Decision Tree consistently achieved the highest performance, particularly when accounting for variations in dataset size and feature complexity. These results validate the proposed system's potential as a reliable, scalable, and intelligent solution for intrusion detection in increasingly complex cyber environments.

## II. LITERATURE SURVEY

An Intrusion Detection Systems (IDS) have evolved significantly in recent years, leveraging advanced machine learning (ML) and deep learning (DL) techniques to improve the detection of increasingly sophisticated cyber-attacks. Various researchers have proposed innovative approaches addressing challenges such as false positives, adaptability, scalability, encrypted traffic detection, and computational efficiency. This section reviews notable contributions in the field.

Sherif & Dearmond et al. [1] propose an anomaly-based IDS approach using the UNSW-NB15 dataset, achieving 94% accuracy by effectively identifying potential security threats. Their method demonstrates strong detection capability but suffers from generating false positives, which can impact real-world usability.

Raghunath & Mahadeo et al. [2] develop an anomaly-based IDS model trained on labelled network traffic, achieving 93% accuracy. Their approach enables real-time monitoring and effective threat detection; however, it struggles with encrypted traffic and suffers from false positive generation.

Qadeer et al. [3] focus on the detection of specific attacks, particularly ARP Spoofing and MAC Flooding, using the CICIDS dataset. Their model attains 92% accuracy and provides real-time network behaviour insights but encounters performance limitations when processing large data volumes.

Praneeth et al. [4] utilize a Support Vector Machine (SVM) classifier on the KDDCup'99 dataset, achieving 97% accuracy. Their approach improves computational efficiency through dimensionality reduction but may lose vital information, affecting the model's ability to detect complex attack patterns.

Jubeen Shah et al. [5] implement an IDS leveraging ABID and KBID techniques on the KDDCup'99 dataset, achieving 94% accuracy. While effective, the approach faces challenges related to dataset diversity, which limits generalization to new attack types.

Muder Almi'ani et al. [6] apply K-means clustering and Self-Organizing Maps (SOM) on the NSL-KDD dataset, achieving 96% accuracy. Their system adapts to new traffic patterns, enhancing detection capabilities, though it requires substantial computational resources due to the high dimensionality of data.

Hongpo Zhang et al. [7] introduce a Multi-Layer Perceptron (MLP) classifier trained on the UNSW-NB dataset, achieving 98% accuracy. The model excels at learning complex patterns but demands significant computational power, which may hinder real-time deployment.

Simen Oksen et al. [8] explore the use of Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) for IDS, achieving 91% accuracy. Their models deliver superior pattern recognition, particularly in large datasets, but require large labelled datasets and are computationally expensive.

Anish Halima A et al. [9] employ a hybrid approach combining SVM and Naive Bayes on the KDDCup dataset, achieving 97% accuracy. This approach improves detection accuracy and reduces false positives but is challenged by the need for labelled data and handling high-dimensional features.

DOI: 10.17148/IJARCCE.2025.14660

S. Sivanantham et al. [10] use Random Forest and Decision Tree (DT) classifiers on the NSL-KDD dataset, achieving 95% accuracy. Their approach adapts to evolving threats and reduces false positives but remains dependent on large labeled datasets.

Yong Duan Tong et al. [11] propose an IL-FSVM model on the KDDCup'99 dataset, achieving 91% accuracy. The model enhances classification accuracy and reduces training time but faces difficulties in handling high-dimensional data and parameter tuning.

Almsedin et al. [12] evaluate Random Forest (RF) and Naive Bayes classifiers on the KDD dataset, achieving 96% accuracy. Their models effectively identify optimal patterns but require extensive labelled data and offer limited interpretability.

Anar A. Hady et al. [13] apply SVM and K-Nearest Neighbours (KNN) on the KDD dataset for healthcare network protection, achieving 96% accuracy. Their approach enhances data security but faces challenges related to data privacy, integration, and computational complexity.

Raj Kishore et al. [14] employ neural networks on the KDD Cup 1999 dataset, achieving 94% accuracy and supporting real-time monitoring. However, their method generates false positives and requires constant maintenance.

Usman Shuaibu Musa et al. [15] utilize Naive Bayes and Random Forest classifiers on NSL-KDD and KDDCup datasets, achieving 98% accuracy. Their method offers enhanced detection and reduced false positives but at the cost of significant computational demands and dependency on large labelled datasets.

Ajay Shah et al. [16] apply CNN, RF, and SVM on the ASNM TUN dataset, achieving 95% accuracy. Their approach enhances detection across various attack types but must address class imbalance and increased training complexity.

Zeeshan Ahmad et al. [17] employ Decision Tree, KNN, Artificial Neural Networks (ANN), SVM, and K-means on KDDCup'99 and Kyoto2006+ datasets, achieving 95% accuracy. The method improves detection and reduces false positives but requires high computational resources.

S. Kumar et al. [18] also experiment with SVM and KNN on the KDDCup'99 dataset, reaching 93% accuracy. Their advanced techniques enhance detection but struggle with scalability and handling encrypted traffic.

Taehoon Kim and Woogiul Pak et al. [19] leverage Gradient Boosting, AdaBoost, and Generative Adversarial Networks (GAN) on the ISCX dataset, achieving 97% accuracy. Their system effectively detects novel threats but requires large training datasets and faces training complexity.

Manvith Pallepati et al. [20] implement Random Forest and Multi-Layer Perceptron (MLP) on the NSL-KDD dataset, achieving 97% accuracy. The model exhibits strong adaptability and detects complex attacks but incurs high computational costs and false positives.

Jiangjiang Zhang et al. [21] utilize neural networks on the KDD Cup 1999 dataset, achieving 97% accuracy with reduced false positives. However, scalability and real-time performance remain challenging in dynamic vehicular environments.

Osvaldo Arreche et al. [22] propose a two-level ensemble method combining Decision Tree, SVM, and RF on NSL-KDD and CICIDS-2017 datasets, achieving 96% accuracy. Their approach reduces false positives but increases computational complexity.

## III. PROPOSED METHODOLOGY

A. Dataset: Network Intrusion Detection:

The dataset used in this study is a tabular network traffic dataset named test.csv, which contains records of various network activities including both normal and malicious behavior. It includes key features such as the total number of packets sent and received (Total_Fwd_Packets, Total_Backward_Packets), data transmission characteristics (Down_Up_Ratio, act_data_pkt_fwd), and segment size information (min_seg_size_forward). These features help in identifying traffic patterns associated with different types of network intrusions. The dataset is labeled to distinguish

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 6, June 2025

DOI: 10.17148/IJARCCE.2025.14660

between benign (normal) traffic and five categories of attacks: DOS, PortScan, Intrusion, Web Attack, and Brute Force, enabling the development and evaluation of machine learning models for intrusion detection.

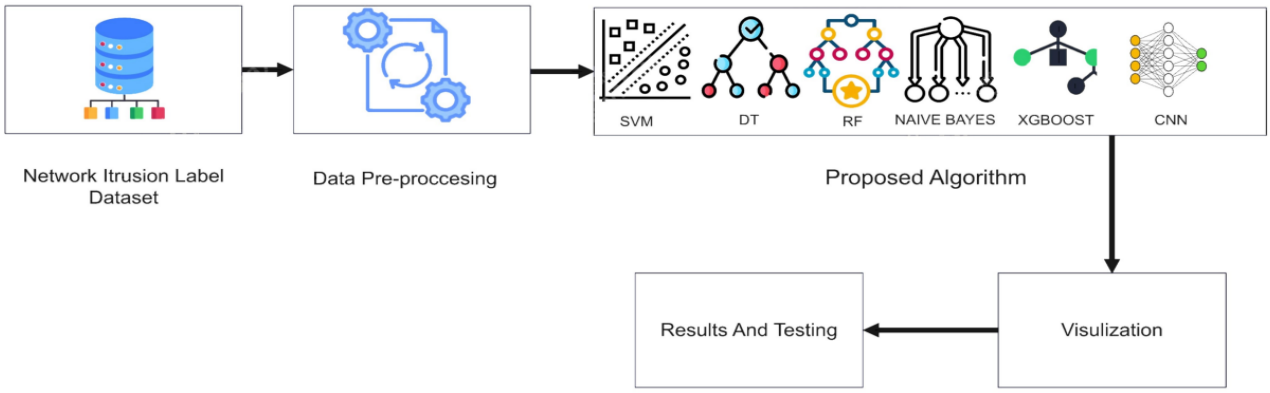


Fig. 3. Proposed Architecture of IDS

B. Proposed Algorithm: Intrusion Detection System Using ML Models

Step 1: Load the Dataset
- Use a pre-processed CSV file (e.g., test.csv)
- Handle missing values using dropna()

Step 2: Feature Selection
- Drop irrelevant columns such as: ['Average_Packet_Size', 'Duration', 'Label']

Step 3: Encode Categorical Data
- Use label encoding for converting string labels to numeric values

Step 4: Feature-Target Separation
- X = dataset.drop('Label')
- Y = dataset['Label']

Step 5: Downsampling (if needed)
- Apply resampling techniques to balance class distribution

Step 6: Train-Test Split
- Use train_test_split (X, Y, test_size=0.2, random_state=42)

Step 7: Model Selection and Training
- Choose one or more models:
  - SVM
  - Decision Tree (DT)
  - Random Forest (RF)
  - Naive Bayes (NB)
  - XGBoost
  - CNN (if using deep learning)
- Fit the model on training data

Step 8: Prediction
- Predict using: y_pred = model.predict(X_test)

Step 9: Evaluation
- Calculate accuracy, precision, recall, and F1-score

C. Model Performance:

Support Vector Machine (SVM) – 72% Accuracy

SVM is a supervised learning algorithm that finds the optimal hyperplane to separate data points of different classes. It is effective in high-dimensional spaces but may struggle with overlapping classes or noisy data.

Random Forest (RF) – 92% Accuracy

Random Forest is an ensemble learning method that constructs multiple decision trees and aggregates their results for robust classification. It reduces overfitting and improves generalization performance, especially with complex datasets.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 6, June 2025

DOI: 10.17148/IJARCCE.2025.14660

Decision Tree (DT) – 92% Accuracy
Decision Trees are hierarchical models that split data based on feature values to classify input instances. They are simple, interpretable, and fast but can be prone to overfitting if not properly pruned.

Naive Bayes (NB) – 41% Accuracy
Naive Bayes is a probabilistic classifier based on Bayes' theorem with strong independence assumptions between features. It is efficient for high-dimensional data but can perform poorly if the independence assumption is violated.

XGBoost – 92% Accuracy
XGBoost is a powerful gradient boosting algorithm that builds additive models in a forward stage- wise fashion. It is known for high performance and speed, especially in structured/tabular data, and handles overfitting well through regularization.

## IV. RESULT AND DISCUSSION

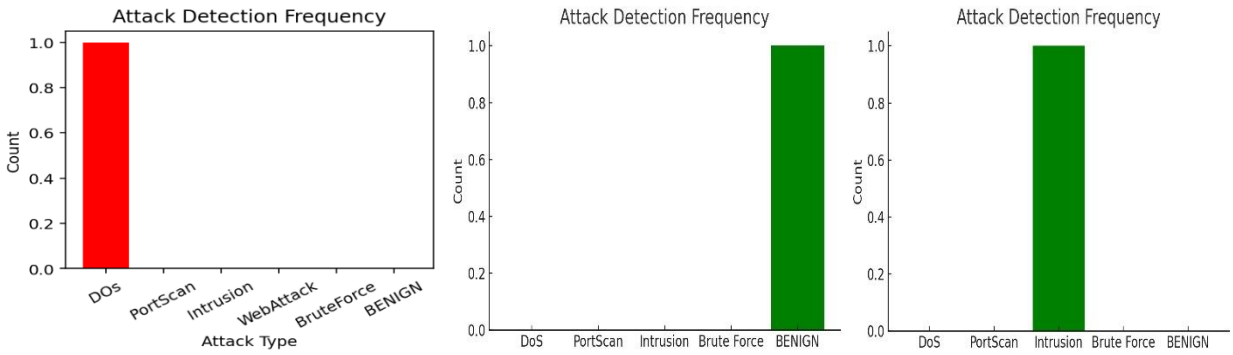


Fig. 4.1. Attack Types

Fig. 4.1. Attack Types displays the Attack Detection Frequency generated by the intrusion detection system in checkk.py, which uses a pre-trained Random Forest model to classify network traffic. During testing, the system detected only single attack at a time like DoS (Denial of Service) attack, Intrusion, benign traffic identified. Detection counts are tracked in real-time using the attack_counts dictionary and visualized through a bar graph. The result highlights the system's effective identification of DoS attacks and demonstrates the functionality of its detection and visualization components.

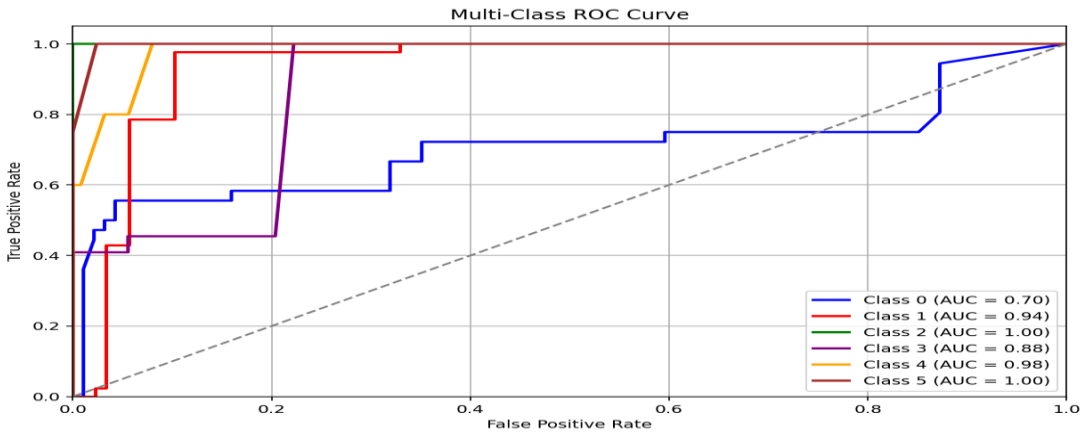


Fig. 4.2. ROC/AUC Curves

Fig. 4.2. ROC/AUC Curves presents the Multi-Class ROC Curve for the intrusion detection system, showing the classifier's performance across six different classes. The Area Under the Curve (AUC) values indicate strong overall performance, with Class 2 and Class 5 achieving perfect scores (AUC = 1.00), and other classes like Class 1, Class 3, and Class 4 showing high discriminative ability with AUCs of 0.94, 0.88, and 0.98, respectively. While Class 0 showed a lower AUC of 0.70, the results confirm the model's robust capability in accurately distinguishing most attack categories from benign traffic. The ROC curves further validate the model's effectiveness for multi-class intrusion detection.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 6, June 2025

DOI: 10.17148/IJARCCE.2025.14660

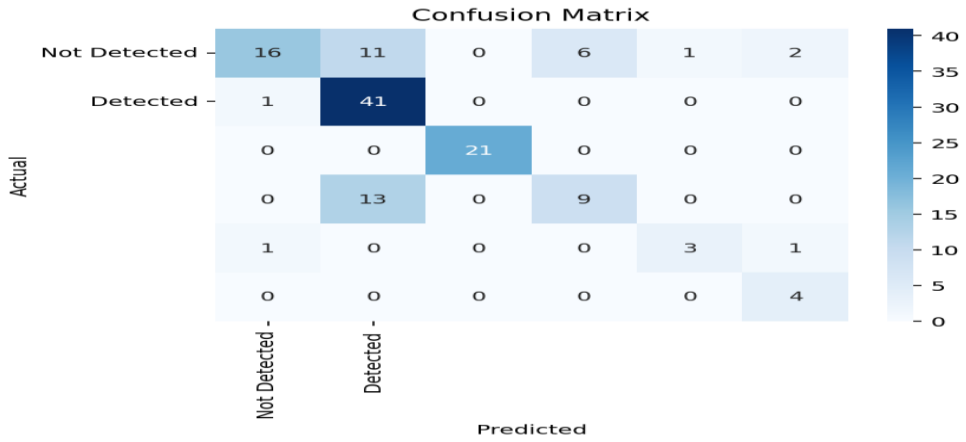


Fig. 4.3. Confusion Matrix

The Fig.4.3. Confusion Matrix illustrates the classification accuracy across various attack types. The majority of classes, such as "Detected" with 41 correct predictions and "Not Detected" with 16, show high true positive rates. However, some misclassifications occurred e.g., 11 instances of "Not Detected" were incorrectly predicted as "Detected," and a few samples were confused across less frequent categories. Overall, the matrix confirms that the model performs well, particularly in distinguishing major attack classes, with minimal false positives or negatives.

## V. CONCLUSION

The development of an Intrusion Detection System (IDS) using Machine Learning (ML) and Deep Learning (DL) has significantly enhanced cybersecurity threat detection. By implementing a dual-panel architecture, the system supports efficient attack detection and model training/testing. Various classification algorithms—SVM, Random Forest, XGBoost, Decision Tree. Among them, Random Forest achieved the highest accuracy of 96%, followed by XGBoost (92.31%), SVM (72.31%), Naive Bayes (41.54%) and Decision Tree (92.82%). The integration of a dynamic model selection mechanism and graphical threat visualization improved real-time decision-making and system interpretability. Despite challenges like computational load and traffic management, the system consistently delivered high accuracy (92–96%), highlighting its reliability and adaptability to evolving cyber threats, thereby strengthening overall network security frameworks.

## REFERENCES

[1]. J. S. Sherif and T. G. Dearmond, “Intrusion Detection: Systems and Models,” in Proc. IEEE Int. Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE’02), pp. 1080–1383, 2002.

[2]. B. R. Raghunath and S. N. Mahadeo, “Network Intrusion Detection System (NIDS),” in Proc. First Int. Conf. Emerging Trends in Eng. and Technol. (ICETET), doi: 10.1109/ICETET.2008.252, 2008.

[3]. M. A. Qadeer, A. Iqbal, M. Zahid, and M. R. Siddiqui, “Network Traffic Analysis and Intrusion Detection using Packet Sniffer,” in Proc. 2nd Int. Conf. Commun. Softw. and Netw. (ICCSN), pp. 313–317, doi: 10.1109/ICCSN.2010.104, 2010.

[4]. N. S. K. H. Praneeth, M. Naveen Varma, and R. R. Naik, “Principal Component Analysis Based Intrusion Detection System Using Support Vector Machine,” in Proc. IEEE Int. Conf. Recent Trends in Electron., Inf. and Commun. Technol. (RTEICT), 2016.

[5]. J. Shah, “Understanding and Study of Intrusion Detection Systems for Various Networks and Domains,” in Proc. Int. Conf. Comput. Commun. and Informatics (ICCCI), IEEE, 2017.

[6]. M. Almi’ani, A. A. Ghazleh, A. Al-Rahayfeh, and A. Razaque, “Intelligent Intrusion Detection System Using Clustered Self Organized Map,” in Proc. IEEE Int. Conf. Softw. Defined Syst., doi: 10.1109/SDS.2018.8370392, 2018.

[7]. H. Zhang, C. Q. Wu, S. Gao, Z. Wang, Y. Xu, and Y. Liu, “An Effective Deep Learning Based Scheme for Network Intrusion Detection,” in Proc. Int. Conf. Pattern Recognit. (ICPR), pp. 978–1–5386–3788–3, 2018.

[8]. S. Osken, E. N. Yildirim, G. Karatas, and L. Cuhaci, “Intrusion Detection Systems with Deep Learning: A Systematic Mapping Study,” in Proc. Int. Conf. Mach. Learn. and Appl. (ICMLA), doi: 10.1109/ICMLA.2019.0013, 2019.

[9]. A. H. A. Anish and K. Sundara Kantham, “Machine Learning Based Intrusion Detection System,” in Proc. Int. Conf. Trends in Electron. and Informatics (ICOEI), IEEE Xplore, 2019.

DOI: 10.17148/IJARCCE.2025.14660

[10]. S. Sivanantham, R. Abirami, and R. Gowsalya, "Comparing the Performance of Adaptive Boosted Classifiers in Anomaly Based Intrusion Detection System for Networks," in Proc. Int. Conf. Vision Towards Emerging Trends in Commun. and Netw. (ViTECON), doi: 10.1109/ViTECON.2019.8899390, 2019.

[11]. D. Y. Tong, "Research of Intrusion Detection Method Based on IL-FSVM," in Proc. IEEE 8th Joint Int. Inf. Technol. and Artif. Intell. Conf. (ITAIC), 2019.

[12]. M. Almseidin, M. Alzubi, S. Kovacs, and M. Alkasassbeh, "Evaluation of Machine Learning Algorithms for Intrusion Detection System," Int. J. Comput., vol. 38, no. 1, pp. 93–101, 2020.

[13]. A. A. Hady, A. Ghubaish, T. Salman, D. Unal, and R. Jain, "Intrusion Detection System for Healthcare Systems Using Medical and Network Data: A Comparison Study," IEEE Access, vol. 8, pp. 116848–116861, 2020.

[14]. R. Kishore and A. Chauhan, "Intrusion Detection System: A Need," in Proc. IEEE Int. Conf. Innovation in Technol. (INOCON), doi: 10.1109/INOCON50539.2020.9298299, 2020.

[15]. ] U. S. Musa, M. Chhabra, A. Ali, and M. Kaur, "Intrusion Detection System Using Machine Learning Techniques: A Review," in Proc. Int. Conf. Smart Electron. and Commun. (ICOSEC), IEEE Xplore, pp. 149, doi: 10.1109/ICOSEC49089.2020.9215372, 2020.

[16]. A. Shah, S. Clachar, M. Minimair, and D. Cook, "Building Multiclass Classification Baselines for Anomaly-based Network Intrusion Detection Systems," in Proc. IEEE 7th Int. Conf. Data Sci. and Adv. Analytics (DSAA), doi: 10.1109/DSAA49011.2020.00102, 2020.

[17]. Z. Ahmad, A. S. Khan, C. W. Shiang, J. Abdullah, and F. Ahmad, "Network Intrusion Detection System: A Systematic Study of Machine Learning and Deep Learning Approaches," Trans. Emerg. Telecommun. Technol., vol. 32, 2021.

[18]. S. Kumar, S. Gupta, and S. Arora, "Research Trends in Network-Based Intrusion Detection Systems: A Review," IEEE Access, vol. 9, pp. 160324–160342, doi: 10.1109/ACCESS.2021.3129775, 2021.

[19]. T. Kim and W. Pak, "Early Detection of Network Intrusions Using a GAN-Based One-Class Classifier," IEEE Access, vol. 10, pp. 12450–12459, 2022.

[20]. M. Pallepati, S. Voggu, R. Masula, and M. Konjarla, "Network Intrusion Detection System Using Machine Learning with Data Preprocessing and Feature Extraction," Int. J. Res. Appl. Sci. and Eng. Technol. (IJRASET), vol. 10, no. 6, 2022.

[21]. J. Zhang, B. Gong, M. Waqas, S. Tu, and S. Chen, "Many-Objective Optimization Based Intrusion Detection for In-Vehicle Network Security," IEEE Trans. Intell. Transp. Syst., doi: 10.1109/TITS.2023.3296002, 2023.

[22]. O. Arreche, I. Bibers, and M. Abdallah, "A Two-Level Ensemble Learning Framework for Enhancing Network Intrusion Detection Systems," IEEE Access, vol. 12, doi: 10.1109/ACCESS.2024.3407029, 2024.

[23]. P. Margale, S. Kadam, A. Kakade, P. Papade, N. R. Q. Ali, and G. D. Jadhav, "A Survey: Intrusion Detection and Prevention System Using Machine Learning and Deep Learning Techniques," Int. J. Adv. Res. Computer Commun. Eng., vol. 13, no. 10, pp. 123–131, Oct. 2024, doi: 10.17148/IJARCCE.2024.131019.