KEC Conference 2026
ISSN: 3021-940X (Print)
DOI: 10.3126/injet.v3i2.95492

Comparative Analysis of Ensemble Machine Learning and Hybrid Deep Learning Models for Intrusion Detection

Sujan Dharel[1], Prabhat K.C.[2], Ram Krishna Maharjan[3*]

[1]*Department of Electronics and Computer Engineering, Pulchowk Campus, IOE, Tribhuvan University, Lalitpur, Nepal, 080msice019.sujan@pcampus.edu.np*
[2] *Department of Electronics and Computer Engineering, Pulchowk Campus, IOE, Tribhuvan University, Lalitpur, Nepal, itsmeprabhat10@gmail.com*
[3] *Department of Electronics and Computer Engineering, Pulchowk Campus, IOE, Tribhuvan University, Lalitpur, Nepal, rkmahajn@ioe.edu.np*

Abstract

The emergence of cloud computing and corporate data centers has increased traffic on the Internet, making them more vulnerable to cyberattacks. Signature-based IDSs are not capable of recognizing emerging threats due to the dependence on known patterns of attacks. Therefore, machine learning and deep learning can be used as tools for intrusion detection in the modern network environment. This paper examines three popular machine learning and hybrid IDS models: Random Forest (RF), XGBoost, and CNN-LSTM. Their effectiveness was evaluated based on the use of the CSE-CIC-IDS2018 dataset containing real network traffic and such attacks as brute force attacks, DDoS, botnets, and others. Such techniques as data cleaning, feature encoding, normalization, and feature selection were used in the preprocessing step. The models were compared using several quality metrics such as accuracy, precision, recall, F1-score, false positive rate, and receiver operating characteristic. Experimental results prove that XGBoost shows the best performance with 97.97% accuracy, 0.9797 F1-score, and 0.0051 false positive rate, followed by Random Forest with the accuracy of 97.90%. At the same time, CNN-LSTM has shown an excellent result in recognizing temporal patterns in network traffic with the accuracy of 97.40%. These findings illustrate the efficacy of ensemble-based approaches on tabular datasets and the power of hybrid deep learning algorithms for sequential data. In general, this research stresses the significance of choosing the right IDS models by considering accuracy, speed, and deployability aspects.

Keywords: Intrusion Detection System (IDS), Network Intrusion Detection, CSE-CIC-IDS2018, Ensemble Learning, Deep Learning, XGBoost, CNN–LSTM, Cybersecurity, Network Traffic Analysis

1. Introduction

The expansion of digital infrastructure, cloud computing and corporate networks has led to a rise in network traffic and complexity, which, in turn, has increased the risk of advanced cyber threats like Distributed Denial of Service (DDoS), brute-force and botnet attacks [1]. Such attacks can lead to service disruption, data leaks, and significant financial damage, which highlights the importance of intrusion detection as a key element in network security strategies [2]. Conventional intrusion detection systems (IDS) primarily use signature-based approaches that can detect known attack patterns but struggle to keep pace with new and unknown attacks [3]. Furthermore, rule-based intrusion detection systems are ineffective at dealing with high-volume traffic and multiple attack patterns, resulting in increased false positives.

Machine learning and deep learning approaches have proven to be effective in overcoming these challenges. Ensemble techniques like Random Forest and XGBoost can capture intricate patterns in network flows and offer good generalization across various attack scenarios [4]. Likewise, deep learning models such as CNN-LSTM capture both spatial and temporal features in network flows, enhancing their accuracy in enterprise and cloud networks [5]. In this research, the effectiveness of these techniques is compared using the CSE-CIC-IDS2018 dataset [6] [7].

* *Corresponding author*

2. Research Questions

Motivated by the above considerations, this study addresses the following research questions:

- How effectively can ensemble machine learning models such as Random Forest and XGBoost detect network intrusions using flow-based features?
- Does a hybrid CNN–LSTM deep learning model provide improved intrusion detection performance compared to ensemble machine learning models when trained on the same dataset and preprocessing pipeline?

3. Literature Review

Early intrusion detection research relied heavily on rule- and signature-based systems [8], which perform well for known attack patterns but are limited against zero-day and polymorphic attacks. To overcome these limitations, researchers increasingly adopted machine learning approaches for more adaptive intrusion detection [9]. Supervised methods such as decision trees and Random Forest have shown strong performance on labeled datasets, although they may still experience false positives and limited generalization in complex network environments [10].

With the availability of large-scale benchmark datasets such as CIC-IDS2017 and CSE-CIC-IDS2018, deep learning approaches have gained prominence in intrusion detection. CNNs and RNNs can automatically learn hierarchical and sequential traffic features, improving detection accuracy, while hybrid CNN–LSTM architectures combine spatial feature extraction with temporal modeling, making them effective for detecting evolving attack patterns in cloud and enterprise networks [11]. Recent studies have also explored transformer-based and attention-driven IDS models, which show strong capability in learning complex sequential traffic patterns, although their higher computational cost and training complexity may limit practical deployment [12].

Hybrid and ensemble approaches have further improved IDS performance. RF-based frameworks combined with autoencoders have reduced false alarms and enhanced detection accuracy, while gradient boosting methods such as XGBoost have demonstrated high accuracy and near-perfect AUC scores on CSE-CIC-IDS2018 [13]. Despite these advances, many studies focus on individual algorithms or limited comparisons, making consistent evaluation difficult. In particular, comparative studies between ensemble machine learning models and hybrid deep learning architectures under a unified preprocessing and evaluation framework remain limited. To address this gap, this study compares Random Forest, XGBoost, and CNN–LSTM on CSE-CIC-IDS2018 using a consistent experimental pipeline to better understand their relative strengths for modern intrusion detection.

4. Methodology

In this research work, an intrusion detection system framework was designed based on machine learning, which uses the CSE-CIC-IDS2018 dataset to compare ensemble learning approaches (Random forest, XGBoost) against a CNN-LSTM hybrid method. The process entails data preprocessing, feature extraction, training of the model, and evaluation of its performance. Data were preprocessed by cleaning, encoding, normalization, and dimensionality reduction using correlation and mutual information techniques. 80% of the dataset was used for training, while the remaining 20% was used for testing.

4.1 Proposed Intrusion Detection System Architecture

The framework comprises four stages: data preprocessing, feature selection, model training, and performance evaluation (Figure 1). Preprocessing handles inconsistencies and encoding; feature selection uses correlation and mutual information. After the dataset was split, models train and compare using standard metrics, confusion matrices, and ROC curves.

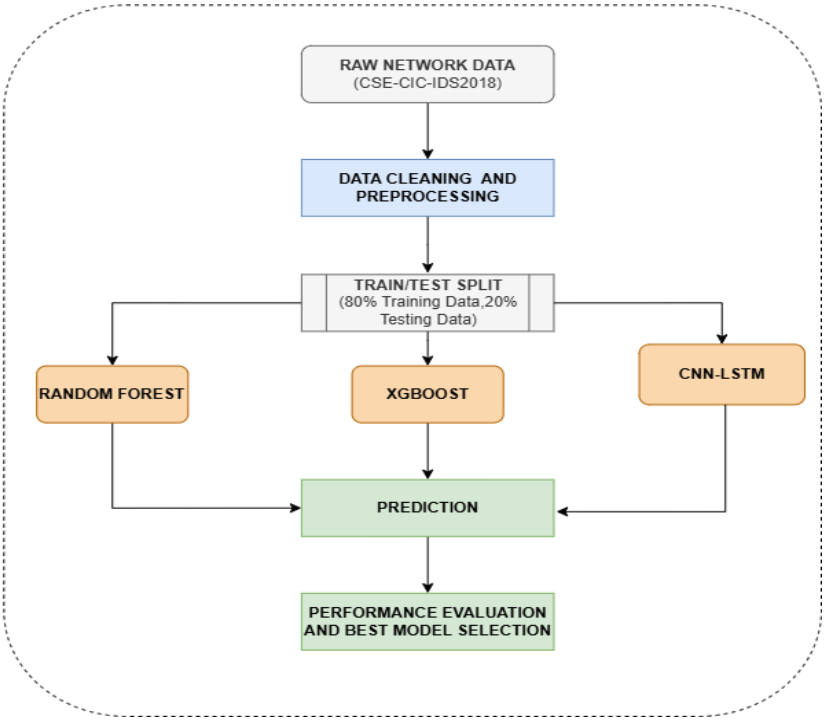

Figure 1. Proposed Intrusion Detection System Architecture

4.2 Dataset Description

The CSE-CIC-IDS2018 dataset, a typical benchmark dataset developed by the Canadian Institute for Cybersecurity (CIC) and Communications Security Establishment (CSE) was used in this study. This dataset contains real network traffic from an enterprise environment, both benign traffic and attacks: brute force, DoS/DDoS, botnet and web attacks. To provide balanced performance comparison, Randomly Sampled from the CSE-CIC-IDS2018 Dataset nearly 30,000 records from the five traffic classes were selected as shown in Table 1, resulting in a dataset of 150,000 flow records. While the balanced dataset allows for fair comparison, it is not a realistic representation of network traffic, where the number of normal flows is much greater than attacks, and network traffic is more diverse. Therefore, while CSE-CIC-IDS2018 dataset is a good dataset to compare models, the results should be expected as benchmark results, not as real-world results.

Table 1. Network traffic class descriptions

Class	Count	Description
Benign	29,873	Normal network traffic
Bot	29,672	Botnet infected host communication
DoS Hulk	30,045	DoS attack targeting web servers
FTP-BruteForce	30,230	Password brute-force attack on FTP services
SSH-Bruteforce	30,180	SSH password brute-force attack

4.3 Data Preprocessing

The preprocessing workflow starts with data cleaning, to eliminate missing data, duplicate rows and invalid values from the dataset. Next, categorical variables, such as protocol names, are transformed into numerical values via label encoding. Numerical features are normalized to the range [0,1], facilitating model training. Features are selected via correlation analysis and mutual information to retain the most informative and discriminative features and remove redundancy. The resulting dataset is split into 80% training and 20% testing sets, where the Label attribute is used to represent the five different traffic classes for classification.

4.4 Intrusion Detection Models

Three classification models are implemented to evaluate intrusion detection performance.

4.4.1 Random Forest

RF is an ensemble of decision trees built on random subsets of data and features. The final prediction is obtained by majority voting:

$$\boldsymbol{P}_{\mathrm{RF}}(\boldsymbol{x}) = \mathrm{majority}(\boldsymbol{h_i}(\boldsymbol{x})), \boldsymbol{i} = \boldsymbol{1}, \ldots, \boldsymbol{T} \quad \text{(Equation 1)}$$

where $\boldsymbol{h_i}(\boldsymbol{x})$ is the prediction of the $\boldsymbol{i}$-th decision tree and $\boldsymbol{T}$ is the number of trees.

4.4.2 XGBoost

XGBoost is a gradient-boosting algorithm that builds decision trees sequentially to minimize a regularized loss function:

$$\boldsymbol{\mathcal{L}} = \sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} \boldsymbol{l}(\boldsymbol{y_i}, \boldsymbol{\hat{y}_i}) + \boldsymbol{\lambda\Omega}(\boldsymbol{f}) \quad \text{(Equation 2)}$$

where $\boldsymbol{l}(\cdot)$ is the loss, $\boldsymbol{\hat{y}_i}$ is the prediction, and $\boldsymbol{\Omega}(\boldsymbol{f})$ penalizes model complexity to improve generalization.

4.4.3 CNN-LSTM Hybrid Model

The CNN–LSTM architecture first applies convolutional layers to capture spatial feature relationships, followed by LSTM layers to model temporal dependencies in flow sequences. The convolution operation is expressed as:

$$z = \sigma(W * x + b) \quad \text{(Equation 3)}$$

where W is the filter kernel, x is the input feature vector, b is the bias, and $\sigma(\cdot)$ is the activation function (e.g., ReLU).

The CNN–LSTM model captures both spatial and temporal patterns across five traffic classes: Benign, Bot, DoS-Hulk, FTP-Brute Force and SSH-Brute force. Flow features are reshaped into 1D sequences and processed through Conv1D and pooling layers for feature extraction and dimensionality reduction. LSTM layers then model temporal dependencies, followed by dropout-regularized dense and SoftMax layers for multi-class classification. This hybrid architecture is effective in detecting complex and time-dependent cyberattacks.

4.5 Experimental Setup

We used Python's scikit-learn and XGBoost libraries to implement Random Forest and gradient boosting, and we used TensorFlow/Keras to make CNN-LSTM. The data was split into two groups: 80% for training and 20% for testing. The CNN-LSTM model was trained for 50 epochs with a learning rate of 0.01 and a batch size of 256. There were 100 trees and a maximum depth of TensorFlow/Kera20 Random Forest. The learning rate for XGBoost was 0.1, 200 estimators and the maximum depth was 6, and it used subsampling to make it work better and not overfit. These settings were picked because they worked well in the past and were quick.

5. Evaluation Metrics

Performance is evaluated using accuracy, precision, recall, F1-score, and false positive rate. High values of accuracy, precision, recall, and F1-score indicate effective detection of attacks, while a low false positive rate reflects minimal misclassification of benign traffic. ROC curves and AUC further demonstrate the model’s strong ability to distinguish between attack and normal traffic across different thresholds.

6. Experimental Results

6.1 Training and Validation Behavior

This comparison is illustrated through the loss curves shown in Figures 2 and 3, providing insights into the convergence behavior and learning stability of ensemble and deep learning approaches. Model validation was performed using 15% of the dataset reserved as testing/validation data.

6.1.1 Training and Validation Loss Analysis

The loss curves provide insight into the convergence behavior and generalization capability of the models. The CNN–LSTM model exhibits a gradual reduction in training loss with a consistently stable validation loss, indicating smooth convergence and effective learning of underlying traffic patterns. The close alignment between the curves confirms minimal overfitting and strong generalization. In contrast, the XGBoost model demonstrates rapid stabilization with minimal variation in loss, reflecting efficient learning inherent to ensemble methods. The consistency between training and validation performance further indicates robust generalization without instability.

Since Random Forest is a non-iterative ensemble method, it does not produce comparable epoch-wise loss curves; however, its performance is considered in the overall model comparison.

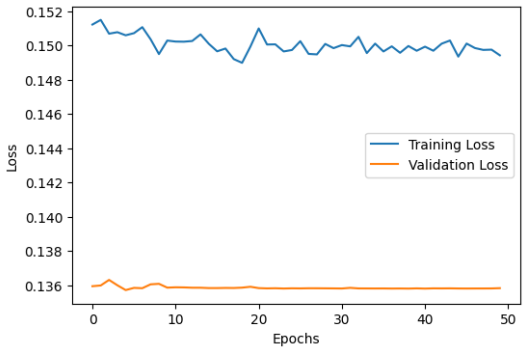

Figure 2. CNN-LSTM model indicating convergence behavior

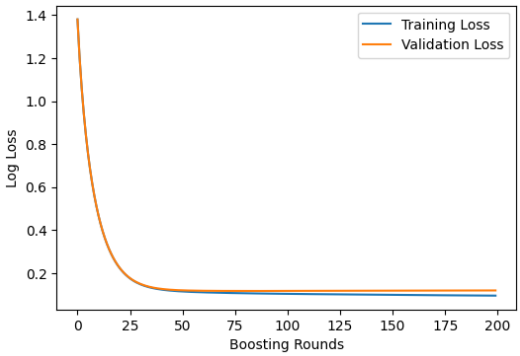

Figure 3. XGBoost model indicating stable learning

6.1.2 Convergence and Stability

Both models achieve stable convergence with no significant divergence between training and validation loss. However, CNN–LSTM benefits from progressive learning of temporal features, while XGBoost attains faster convergence with slightly more stable behavior.

6.2 Model Performance Comparison

Table 2. Comparative performance of IDS models based on accuracy, precision, recall, F1-score, and false positive rate.

Model	Accuracy	Precision	Recall	F1 Score	FPR
XGBoost	97.97%	97.97%	97.97%	97.97%	0.0051
Random Forest	97.90%	97.90%	97.90%	97.90%	0.0052
CNN–LSTM	97.40%	97.40%	97.41%	97.41%	0.0065

Overall performance of the three models is summarized in Table 2 indicates that XGBoost achieves the highest overall accuracy and the lowest false positive rate, highlighting its effectiveness for tabular network traffic data. Random Forest follows closely, offering comparable performance with the advantage of better interpretability. Meanwhile, CNN–LSTM ranks slightly lower but still demonstrates strong capability in modeling sequential patterns within the data.

6.3 Confusion Matrix Analysis

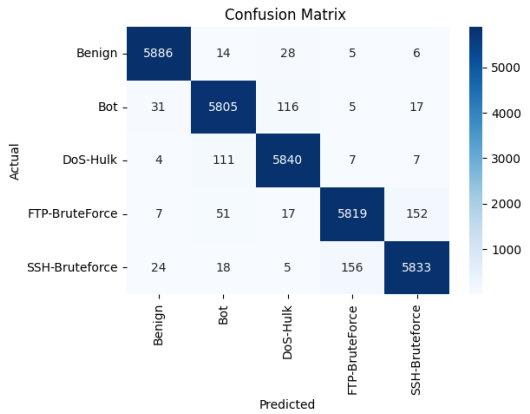

Figure 4. Confusion matrix of the CNN–LSTM model.

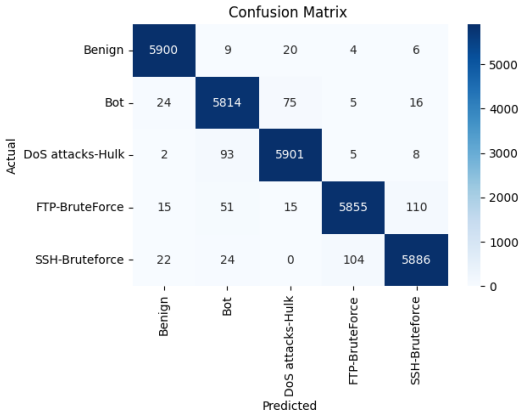

Figure 5. Confusion matrix of the XGBoost model.

Since both Random Forest and XGBoost are ensemble learning models, and XGBoost demonstrated superior performance, further analysis focuses on comparing it with the deep learning model (CNN–LSTM). In this context, the confusion matrices provide detailed insight into classification performance across different traffic classes, where strong diagonal values indicate that most samples are correctly classified, confirming high detection accuracy.

For the XGBoost model (Figure 5), correct predictions are highest across all classes, with Benign (5900), Bot (5814), DoS-Hulk (5901), FTP-Brute Force (5855), and SSH-Brute force (5886). Misclassifications are minimal, primarily occurring between FTP-Brute Force and SSH-Brute force, indicating slight confusion between similar attack patterns.

For the CNN–LSTM model (Figure 4), correct classifications remain high, including Benign (5888), Bot (5805), DoS-Hulk (5840), FTP-Brute Force (5819), and SSH-Brute force (5833). However, slightly higher misclassification is observed compared to XGBoost, due to overlapping behavioral characteristics.

6.3.1 Per-Class Error Analysis

- **Benign:** Very high correct classification
- **Bot & DoS-Hulk:** Consistently well detected with minimal errors
- **FTP & SSH-Brute force:** Most confusion occurs between these classes due to similar attack behavior

Overall, both models achieve high true positive rates with low misclassification. XGBoost shows slightly better class separation, while CNN–LSTM maintains strong performance with minor errors in complex traffic patterns.

6.4 ROC Curve Analysis

The Receiver Operating Characteristic (ROC) curve illustrates the trade-off between the true positive rate (TPR) and false positive rate (FPR) across varying classification thresholds, providing a comprehensive evaluation of model performance. The Area Under the Curve (AUC) serves as a key indicator of a model's ability to distinguish between attack and benign traffic, where values closer to 1 indicate better discrimination. All models achieve high AUC values (≈0.99), indicating strong classification capability. XGBoost achieves the highest AUC (0.9920), followed by Random Forest (0.9916) and CNN–LSTM (0.9908), showing only minor differences in performance.

The ROC curves for XGBoost and CNN–LSTM are illustrated in Figures 6 and 7, respectively, demonstrating their strong discriminative ability across varying thresholds.

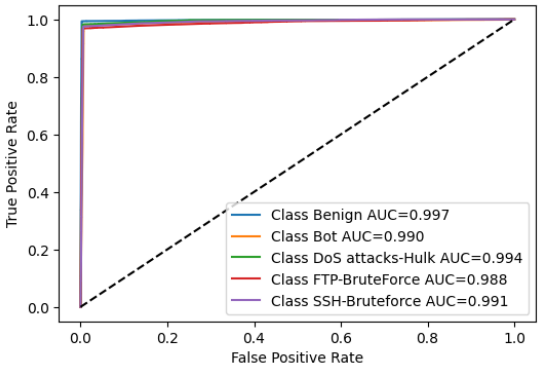

Figure 6. ROC curve for XGBoost model

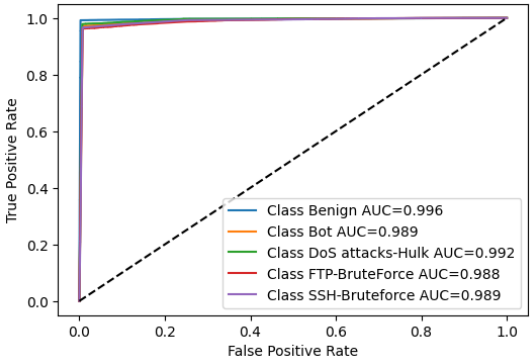

Figure 7. ROC curve for CNN-LSTM model

6.5 Model Complexity and Training Dynamics

The results as shown in Table 3 indicate that XGBoost achieves the fastest training and inference, while CNN–LSTM incurs significantly higher computational cost due to its deep architecture and larger parameter space. Random Forest shows moderate computational requirements. Additionally, the training and validation curves demonstrate stable convergence without overfitting, confirming good generalization across models.

Table 3. Computational performance comparison of IDS models

Model	Training Time (s)	Inference Time (s)	Complexity / Parameters
Random Forest	53.67	2.28	100 trees
XGBoost	36.80	0.90	200 estimators
CNN–LSTM	982.60	8.70	215,685 parameters

7. Conclusion and Discussion

This research evaluated three models, Random Forest, XGBoost and CNN-LSTM, for intrusion detection on the CSE-CIC-IDS2018 dataset, with the same preprocessing and evaluation procedures. The findings reveal that ensemble models are more effective than deep learning models for this task, with XGBoost performing best (97.97% accuracy, 0.9797 F1-score and the lowest false positive rate of 0.0051). Random Forest achieved a comparable accuracy of 97.90% accuracy, whereas CNN-LSTM performed at 97.40% accuracy but with added flexibility to capture temporal relations, albeit at the expense of increased computational overhead and less efficiency than tree-based models.

In summary, all three models had detection rates of more than 97%, indicating their effectiveness in intrusion detection systems. However, XGBoost was found the best model in terms of accuracy, efficiency and speed for tabular network traffic, making it more practical. CNN-LSTM is still useful for sequential data. The next steps are to implement a real-time IDS, multiclass classification, and investigate the use of transformer models for enhancing IDS performance.

Acknowledgements

This work was supported by the Department of Electronics and Computer Engineering, Pulchowk Campus, IOE, Tribhuvan University, Lalitpur, Nepal. The authors also acknowledge the Canadian Institute for Cybersecurity for providing the CSE-CIC-IDS2018 dataset, which facilitated the experimental evaluation in this study.

References

[1] R. Sommer and V. Paxson, "Outside the Closed World: On Using Machine Learning for Network Intrusion Detection," *2010 IEEE Symposium on Security and Privacy*, pp. 305-316, 2010.

[2] R. Holdbrook, O. Odeyomi, S. Yi and K. Roy, "Network-Based Intrusion Detection for Industrial and Robotics Systems: A Comprehensive Survey," *Electronics*, vol. 13, no. 2079-9292, 2024.

[3] O. Alnasser, J. Muhtadi, K. Saleem and S. Shrestha, "Signature and anomaly based intrusion detection system for secure IoTs and V2G communication," *Alexandria Engineering Journal*, vol. 125, no. 1110-0168, pp. 424-440, 2025.

[4] F. Wicaksana and C. Umam, "Comparative Analysis of Random Forest and Xgboost Performance for Network Flow-Based Malware Classification," *INOVTEK Polbeng - Seri Informatika*, vol. 11, pp. 108-115, 2026.

[5] S. S. Bamber, A. V. R. Katkuri, S. Sharma and M. Angurala, "A hybrid CNN-LSTM approach for intelligent cyber intrusion detection system," *Computers & Security*, vol. 148, no. 0167-4048, p. 104146, 2025.

[6] I. Sharafaldin, A. H. Lashkari and A. A.Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization (CSE-CIC-IDS2018)," Available:https://registry.opendata.aws/cse-cic-ids2018, 2018.

[7] M. A. Gharib, I. Sharafaldin, A. H. Lashkari and A. A. Ghorbani, CSE-CIC-IDS2018: A Realistic Cyber Defense Dataset, Canadian Institute for Cybersecurity, University of New Brunswick, and Communications Security Establishment, 2018.

[8] P. R. Kothamali and S. Banik, "Limitations of Signature-Based Threat Detection," *REVISTA DE INTELIGENCIA ARTIFICIAL EN MEDICINA* , 03 2022.

[9] D. Denning, "An Intrusion-Detection Model," *IEEE Transactions on Software Engineering*, Vols. SE-13, p. 2, 1987.

[10] M. Imani, A. Beikmohammadi and H. R. Arabnia, "Comprehensive Analysis of Random Forest and XGBoost Performance with SMOTE, ADASYN, and GNUS Under Varying Imbalance Levels," *Technologies*, vol. 3, no. 2227-7080, p. 13, 2025.

[11] Sowmya, A. Ta and E. Mary, "A comprehensive review of AI based intrusion detection system," in *Measurement: Sensors*, 2023.

[12] J. Santoso, B. Hartono, F. Silalahi and M. Muthohir, "Transformers in Cybersecurity: Advancing Threat Detection and Response through Machine Learning Architectures," *Journal of Technology Informatics and Engineering*, vol. 3, pp. 382-396, 2024.

[13] M. F. Mohammad, W. Elmedany and M. S. Sharif, "Hybrid AI-Driven Intrusion Detection Systems," in *The 6th Joint International Conference on AI, Big Data and Blockchain (AIBB 2025)*, Springer Nature Switzerland, 2025, pp. 178--194.