An AI-Driven Framework for Network Intrusion Detection Using ANOVA-Based Feature Selection

Salam Allawi Hussein, Sándor Répás
Faculty of Informatics and Electrical Engineering-Department of Electrical Engineering and Infocommunications,
Széchenyi István University of Győr, 9026 Győr, Egyetem tér 1, Hungary

***Abstract*—In the last few years, cyberattacks have become more complex, and it is becoming increasingly necessary to establish secure networks. This study examines enhancements to intrusion detection systems (IDSs) with the implementation of machine learning for the categorization of network traffic attacks. For the current study, we utilize four publicly available datasets: CIC-IDS2017, CIC-DoS2017, CSE-CIC-IDS2018, and CIC-DDoS2019. We examined three machine learning techniques: LightGBM, Random Forest, and XGBoost. Experimental results showed that RandomForest and XGBoost achieved the highest accuracy of 0.99 in both binary and multi-class intrusion detection tasks, maintaining balanced performance with macro F1-scores around 0.86. LightGBM exhibited slightly lower overall performance, but benefited from ANOVA-based feature selection, which improved its recall and model stability. Feature selection also enhanced computational efficiency by reducing feature redundancy while preserving accuracy across models. These results highlight how AI tools could help network security deal with emerging threats and improve the performance of IDS. The study underscores the critical role of feature selection in enhancing model efficiency, hence promoting advancements in automated network security systems that can adapt to evolving cyber threats.**

***Keywords*—Network security; intrusion detection; machine learning; feature selection**

I. Introduction

The rapid growth of networked systems has led to increased vulnerability to cyberattacks, making intrusion detection systems (IDS) a critical component of modern cybersecurity. Traditional IDS often relies on static rule-based approaches that fail to adapt to evolving attack patterns [1]. As public, private, and critical sectors rely more heavily on digital infrastructure, it has become essential to safeguard these systems against unauthorized access, data breaches, and service disruptions. Intrusion detection systems (IDS) remain a key component of network security; however, many still rely on static rules and signature-based detection, which are inadequate against modern, adaptive attack techniques [2], [3], [4]. More recent research has focused on AI's ability to analyze and detect IDS, particularly using machine learning (ML), which appears to fill the lack of limitations [5], [6]. Unlike static systems, ML models for IDS can learn from traffic data, patterns, and adapt to emerging threats [4]. These models increase and enhance detection performance, and respond in near real-time scenarios [7], [1].

The increasing complexity of modern network environments, driven by the proliferation of Internet of Things (IoT) devices, software-defined infrastructures, and large-scale distributed systems, has introduced new security challenges that exceed the capabilities of traditional intrusion detection mechanisms. Recent studies indicate that such environments generate highly heterogeneous and dynamic traffic patterns, which significantly complicate the identification of malicious activities using static detection rules. As a result, intrusion detection systems are required not only to detect known attack signatures but also to adapt continuously to evolving threat behaviors while maintaining acceptable performance in resource-constrained and real-time settings [8], [9]. Another fundamental challenge in contemporary intrusion detection lies in the high dimensionality of network traffic data. Modern datasets often comprise hundreds of features, many of which are redundant or weakly correlated with attack behavior. This characteristic negatively impacts learning efficiency, increases computational overhead, and may lead to overfitting. Recent research emphasizes that effective feature selection is a critical preprocessing step for improving IDS performance, as it enables learning models to focus on the most discriminative attributes while reducing model complexity and enhancing generalization across different attack scenarios [10]. In parallel, anomaly-based intrusion detection has gained renewed attention as a viable strategy for identifying previously unseen and zero-day attacks. Unlike signature-based approaches, anomaly detection models establish a baseline of normal network behavior and identify deviations that may signal malicious activity. Empirical evaluations demonstrate that machine learning–based anomaly detection frameworks can successfully capture subtle behavioral deviations in network traffic, offering improved detection of novel attacks. However, these models remain sensitive to data imbalance and noisy features, reinforcing the necessity of careful feature engineering and selection to ensure stable and reliable detection outcomes [11]. Recent advances in deep learning have further expanded the analytical capabilities of intrusion detection systems by enabling the extraction of complex spatiotemporal patterns from network traffic. Architectures such as convolutional and recurrent neural networks have shown strong potential in modeling temporal dependencies and evolving attack sequences. Nevertheless, existing studies highlight that deep models often suffer from limited interpretability and increased computational cost, which can hinder their deployment in operational environments. These limitations have motivated the integration of feature optimization strategies and explainability mechanisms to balance detection accuracy, efficiency, and transparency [12]. Furthermore, the dynamic nature of contemporary networks introduces the challenge of concept drift, where traffic characteristics and attack patterns change over time. Static models trained on historical data frequently experience performance degradation under such conditions. Adaptive learning strategies, including incremental and ensemble-based frameworks, have demonstrated the ability to sustain detection performance by selectively updating models in response to evolving traffic behaviors. Experimental results in software-defined network environments confirm that adaptive learning mechanisms significantly enhance IDS robustness compared to static detection approaches [13]. Collectively, these observations indicate that effective intrusion detection in modern network environments requires an integrated framework that combines intelligent learning models, efficient feature selection, anomaly-aware detection, and adaptability to evolving threats. Addressing these challenges is essential for improving the robustness, scalability, and practical applicability of IDS across heterogeneous and rapidly changing infrastructures.

A. Research Gap

Although there has been considerable use of machine learning techniques to develop intrusion detection systems (IDS), most research using machine learning approaches have been limited by their reliance upon experiments conducted on a single dataset, or in narrow and constrained experimental designs. Thus, the generalizability of these results are limited. In addition, while feature selection is commonly used as a preprocessing step in developing an IDS, there has been little or no systematic analysis of how this impacts the performance of the model in terms of stability, efficiency, and consistent performance across different types of network environments. Therefore, the practical effectiveness of IDS frameworks that utilize feature selection has remained largely untested, particularly under variable traffic conditions. The above limitations highlight the need for a common testing framework that assess the effects of statistically derived feature selections across multiple datasets and various machine learning models.

B. Contribution

The present research contributes to the area of network intrusion detection through addressing limitations associated with existing IDS (Intrusion Detection System) based on Machine Learning (ML). The first is through a framework of intrusion detection (structured), which incorporates feature selection through ANOVA (Analysis of Variance), as well as multiple supervised learning models for assessing the consistency of the features selected in relation to their importance and redundancy for various types of traffic characteristics. Second, this study represents a comprehensive multi-dataset experiment using many of the most commonly used and adopted benchmark datasets, thereby allowing for a more accurate understanding of how well each model performs relative to its ability to generalize from one dataset to another. Third, the work in this study provides a systematic analysis of the impact of selecting features through statistical means on detection performance and computational efficiency, and thus offers insight into the potential trade-off between the two. Fourth, the comparison of the performance of several machine learning algorithms within an identical experimental environment demonstrates the reliability and utility of simple feature selection methods for intrusion detection in real-world environments.

C. Study Layout

The study is organized as follows: Section II details the related work. Section III present the datasets, methodology and features extraction techniques. Evaluation metric is outlined in Section IV. Test scenarios are detailed in Section V. The results are illustrated in Section VI. Finally, Section VII conclude the summary of the models and future direction.

II. Related Work

The continuous evolution of cyber threats has driven extensive research toward more intelligent, adaptive, and interpretable intrusion detection systems (IDSs). Recent studies have increasingly focused on integrating ensemble learning and deep learning techniques, often combined with optimization and explainability mechanisms, to enhance detection performance across diverse network environments. Hybrid detection pipelines that merge conventional machine learning with deep learning architectures, supported by preprocessing, class balancing, and dimensionality reduction strategies, have demonstrated improved generalization under near real-time constraints [14]. Similarly, automated hyperparameter tuning combined with post-hoc interpretability has been shown to achieve competitive detection accuracy while enhancing operational transparency and trust [15].

Feature engineering and feature selection have also received significant attention as key components in IDS design. Evolutionary and search-based feature selection methods have been reported to effectively reduce high-dimensional feature spaces, leading to improved class balance, lower computational complexity, and faster inference times [10]. Broader empirical analyses indicate that selective feature sets, when paired with ensemble classifiers, can mitigate overfitting and improve adaptability to previously unseen attack patterns [16]. In addition, representation learning approaches under limited supervision have demonstrated the ability to reduce false positive rates while maintaining high detection sensitivity, highlighting practical deployment advantages [17].

Several studies have explored hybrid and protocol-aware detection models to improve intrusion detection performance. Hussein et al. [18] demonstrated that combining ensemble classifiers such as AdaBoost, Random Forest, and XGBoost with Chi-square feature selection significantly enhances detection accuracy on benchmark datasets, particularly for worm-related attacks. Hooshmand and Hosahalli [19] proposed a protocol-aware one-dimensional CNN architecture that integrates Chi-square feature selection and SMOTE to improve the detection of minority-class anomalies, especially in UDP traffic. Furthermore, Qazi et al. [20] introduced HDLNIDS, a hybrid CNN–RNN model capable of capturing both spatial and temporal traffic characteristics, achieving robust performance on the CICIDS-2018 dataset.

The importance of explainable artificial intelligence (XAI) in IDS has also been increasingly recognized. SHAP-based explainability has been incorporated into ensemble learning frameworks to interpret detection outcomes and enhance model transparency [21], while dual-explainability approaches combining SHAP and LIME have been shown to improve detection effectiveness while reducing model complexity [22].

Application-specific IDS frameworks further demonstrate the adaptability of learning-based detection approaches. Online intrusion detection systems tailored for electric vehicle charging infrastructures have employed adaptive ensemble models to address evolving threats in real time [6]. In healthcare and software-defined networking environments, layered detection architectures integrating gradient boosting models and deep neural networks have proven effective against malware and ransomware attacks [23]. Additionally, lightweight IDS solutions for IoT environments have been proposed using statistical feature selection, oversampling techniques, dimensionality reduction, and metaheuristic optimization to balance detection accuracy with resource constraints [24][25][26].

More recently, generative models have been explored in IDS research to support attack simulation, data augmentation, and anomaly detection. Reviews of generative adversarial networks (GANs) and variational autoencoders (VAEs) highlight their potential for enhancing detection robustness, while also emphasizing the need for standardized evaluation protocols [1]. Domain-specific IDS solutions, such as real-time DDoS detection frameworks and smart grid-oriented architectures, further illustrate the necessity of customized detection strategies tailored to specific infrastructure requirements [5], [27].

III. Methodology

A. Workflow Overview

This study follows a four-stage workflow:

- data preprocessing
- feauures extraction and selection
- model training
- model evaluation

The dataset is split into training and test sets using a stratified strategy to preserve class proportions and enable a fair assessment. Three competitive machine learning classifiers XGBoost, Random forest, and LightGBM are trained under a unified pipeline and evaluated using standard classification metrics. The overall workflow is summarized in Fig. 1.

B. Network Traffic Dataset

This study employs the CICIDS Collection compiled by the Canadian Institute for Cybersecurity and made publicly available through Kaggle [28]. The collection integrates several benchmark datasets CIC-IDS2017, CSE-CIC-IDS2018, and CIC-DDoS2019 providing a diverse set of labelled flows that include normal traffic and multiple attack types such as distributed denial of service (DDoS), brute-force login attempts, infiltration, and botnet activity. Each record captures flow-level statistics, including source and destination addresses, ports, protocol type, flow duration, packet length statistics, and other behavioural indicators. The heterogeneous mix of numerical, categorical, and temporal attributes supports advanced analytics and supervised classification. For the purposes of this work, we focus on traffic governed by TCP and UDP, which dominate real-world networks. Prior to modelling, data preprocessing includes normalization, feature encoding, and class balancing to mitigate skewed distributions. The dataset is then split into 80% training and 20% testing subsets using randomized sampling to ensure statistical robustness. Fig. 2 presents the class distribution used in this study.

C. Feature Engineering

Feature selection and engineering are crucial steps in developing effective machine learning models, especially when working with large and complex datasets. These processes refine the input data to ensure that only the most relevant and informative features are used in the model. By focusing on impactful variables, model accuracy is enhanced, computation time is reduced, and interpretability is improved. This subsection outlines the strategies used for feature transformation and selection, which played a key role in improving the performance of the proposed intrusion detection system.

1) Feature Transformation: To prepare the data for supervised learning, categorical class labels are converted into numeric representations to ensure compatibility with the learning algorithms. This transformation preserves class proportions and enables consistent mapping of predicted outputs back to human-readable class names for reporting [29].

2) Feature Selection Using ANOVA F-Test: Feature selection was applied to identify the most discriminative features and reduce the dimensionality of the dataset. The *Analysis of Variance (ANOVA) F-test* was used as a filter-based method to evaluate the statistical significance of each feature with respect to the target classes. This method measures how much the mean of each feature varies across different classes compared to the variance within each class, providing an indication of its discriminative power.

In this study, the `SelectKBest` [30] function from the `scikit-learn` library was employed with the `f_classif` [31] scoring function to rank features based on their F-values. The top 20 features with the highest scores were selected for model training. This approach ensures that only the most relevant features contribute to the classification process, improving model efficiency and reducing overfitting while maintaining predictive performance.

Mathematically, the ANOVA F-statistic for a given feature x_i is defined as:

$$F_i = \frac{MSB_i}{MSW_i} = \frac{\text{variance between class means}}{\text{variance within classes}}$$

where, MSB_i represents the mean square between class means, and MSW_i represents the mean square within each class. A higher F_i value indicates that the feature contributes more to class discrimination.

D. Machine Learning Algorithms

Machine Learning (ML), a core branch of Artificial Intelligence (AI), involves the development of algorithms and statistical models that allow systems to perform specific tasks autonomously without explicit programming. The objective of ML is to enable systems to identify patterns within data, generate predictions, and make informed decisions, with performance improving as the system gains experience from data exposure.

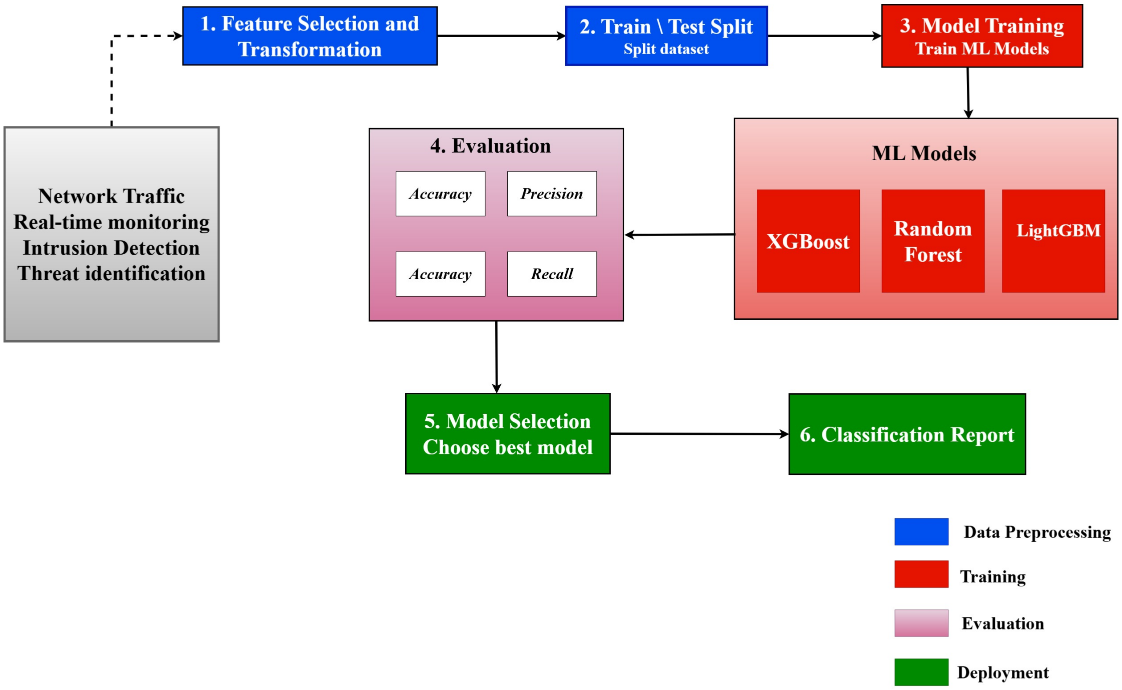


Fig. 1. Intrusion detection system phases.

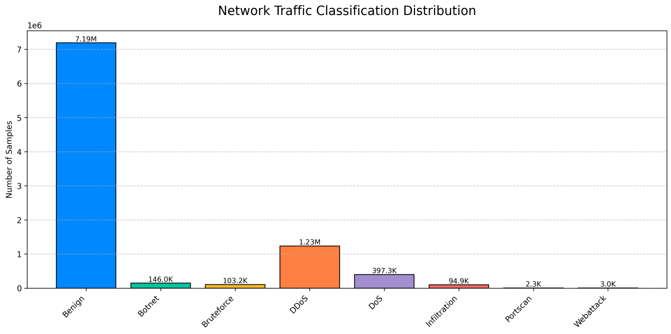


Fig. 2. Network classification dataset distribution.

TABLE I. MACHINE LEARNING MODEL PARAMETERS

Model	Parameter	Value
Random Forest	Number of estimators	100
	Criterion	Gini
	Random state	42
LightGBM	Boosting type	gbdt
	Number of leaves	31
	Learning rate	0.1
	Device	GPU
XGBoost	Booster	gbtree
	Max depth	6
	Learning rate	0.3
	Tree method	gpu_hist
	Predictor	gpu_predictor

1) LightGBM: LightGBM is a gradient boosting framework designed for speed and scalability. Techniques such as Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB) improve efficiency and reduce effective dimensionality [32]. LightGBM is well-suited for near real-time classification on large datasets, making it a practical choice for intrusion detection in networked systems [33], [34].

2) Random Forest (RF): Random Forest is an ensemble of decision trees trained on bootstrapped samples with randomized feature subsets. This approach improves accuracy and reduces variance relative to a single tree. Majority voting yields robust predictions and strong generalization on high-dimensional data [35], [36].

3) XGBoost: XGBoost is a scalable gradient boosting algorithm that builds trees sequentially, correcting residual errors while using a regularized objective to limit over-fitting. In cybersecurity applications, XGBoost has demonstrated strong performance in detecting sophisticated threats across varied environments [23], [3].

Table I explains the hyperparameters of the machine learning models used in the current work.

IV. EVALUATION METRIC

To evaluate the performance of the network intrusion detection models, a set of standard performance indicators was applied, including accuracy, precision, recall, F1-score, and the confusion matrix. These metrics collectively describe how effectively the model distinguishes between normal and attack traffic.

The evaluation metrics are defined as follows:

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

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3}$$

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (4)$$

In these metrics:

- TP (True Positive) represents the number of attack instances correctly identified as attacks.
- TN (True Negative) denotes the number of normal instances correctly classified as normal.
- FP (False Positive) indicates normal traffic mistakenly classified as an attack.
- FN (False Negative) refers to attack traffic incorrectly classified as normal.

The confusion matrix summarizes these outcomes, providing a detailed view of the classifier's performance and highlighting the specific types of misclassification within network traffic.

V. Test Scenarios

Two experimental scenarios were designed to evaluate the proposed framework under different classification objectives: binary and multi-class intrusion detection. Both scenarios used the same datasets, preprocessing steps, and model configurations to ensure consistency.

- Binary Classification Scenario, the task was formulated as a two-class problem: **Benign:** normal network traffic , **Attack:** all malicious traffic categories combined, including DDoS, DoS, PortScan, Botnet, Infiltration, and WebAttack.
- Multi-Class Classification Scenario: The task was extended to classify multiple categories of network traffic.

VI. Results and Discussion

This section presents the performance evaluation of the proposed intrusion detection framework under two classification configurations: binary classification and multi-class classification. Each configuration was tested using RandomForest, LightGBM, and XGBoost models with two feature setups: All Features (AF) and ANOVA-based Feature Selection (FS).

A. Binary Classification Results

Table II summarizes the binary classification performance. All models achieved high accuracy and precision, demonstrating their ability to distinguish between benign and attack traffic, effectively. RandomForest and XGBoost obtained the highest accuracy of 0.9899 and 0.9898, respectively, with perfect precision (1.0000) and F1-scores exceeding 0.97. LightGBM recorded slightly lower performance, but maintained a high AUC of 0.9491 under the AF setup and 0.9578 after feature selection. Feature selection (FS) showed minimal impact on model accuracy but slightly improved LightGBM's Recall and F1-score, indicating that removing less informative features enhanced its generalization capability. Both RandomForest and XGBoost maintained consistent performance between AF and FS, reflecting their robustness to feature dimensionality.

B. Multi-Class Classification Results

Table III presents the multi-class classification results. As expected, the overall accuracy decreased compared to the binary scenario due to the increased complexity of distinguishing between multiple attack categories. RandomForest and XGBoost maintained superior performance, achieving accuracies of 0.99 and macro F1-scores of 0.86 under the feature selection (FS) configuration. After applying ANOVA-based feature selection, RandomForest and XGBoost slightly improved in macro-level Recall and F1-score compared to their All Features (AF) performance, indicating that removing redundant attributes enhanced their generalization capability. Both models also retained high weighted F1-scores of 0.98, confirming consistent classification across major and minor attack classes. LightGBM continued to exhibit lower performance relative to the other models, achieving accuracies of 0.94 (AF) and 0.96 (FS). Although feature selection marginally improved its precision and stability, its macro F1-score remained below 0.60. This suggests that LightGBM is more sensitive to reduced feature dimensionality, while ensemble methods such as RandomForest and XGBoost exploit feature redundancy more effectively to preserve classification robustness across multiple intrusion types.

C. Discussion

The experimental results demonstrate that the proposed intrusion detection framework performs effectively in both binary and multi-class classification settings. Across all models and configurations, high accuracy and precision values indicate strong capability in distinguishing between benign and malicious traffic patterns. In the binary classification experiments, all models achieved near-perfect detection performance. RandomForest and XGBoost recorded the highest accuracies (0.9899 and 0.9898, respectively) with perfect precision and F1-scores exceeding 0.97, confirming their robustness in differentiating between normal and attack traffic. LightGBM showed slightly lower recall but improved marginally under feature selection (FS), indicating that ANOVA-based reduction of redundant features enhanced its generalization without compromising accuracy. In the multi-class classification scenario, overall accuracy decreased compared to the binary setting, as expected from the higher complexity of identifying multiple intrusion categories. RandomForest and XGBoost maintained superior performance, achieving accuracies of 0.99 and macro F1-scores of 0.86 under the FS configuration. The results show that feature selection helped these models improve recall and balance detection across both major and minority classes, while retaining high weighted F1-scores of 0.98. LightGBM exhibited lower macro-level performance (F1-score below 0.60), although its accuracy increased slightly from 0.94 (AF) to 0.96 (FS). This suggests that while feature selection aids in model stability, LightGBM relies more heavily on a full feature space for optimal discrimination across classes. In contrast, ensemble-based models such as RandomForest and XGBoost demonstrate stronger resilience to feature reduction due to their inherent ability to handle feature redundancy. Fig. 3 and Fig. 4 visually compare model performance under AF and FS settings. The figures highlight that feature selection has a minimal, but positive effect on model accuracy and macro F1-score, especially for LightGBM. For RandomForest and XGBoost, performance remained consistently high in both

TABLE II. Binary Classification Performance of Intrusion Detection Models [All Features (AF) and Feature Selection (FS)]

Model	Accuracy	Precision	Recall	F1-Score	AUC
RandomForest (AF)	0.9899	1.0000	0.9543	0.9766	0.9772
RandomForest (FS)	0.9899	1.0000	0.9544	0.9767	0.9772
LightGBM (AF)	0.9760	1.0000	0.8983	0.9464	0.9491
LightGBM (FS)	0.9792	1.0000	0.9156	0.9559	0.9578
XGBoost (AF)	0.9893	1.0000	0.9508	0.9748	0.9754
XGBoost (FS)	0.9898	1.0000	0.9528	0.9758	0.9764

TABLE III. Multi-Class Classification Performance of Intrusion Detection Models [All Features (AF) and Feature Selection (FS)]

Model	Accuracy	Precision (Macro)	Recall (Macro)	F1-Score (Macro)	Weighted F1-Score
RandomForest (AF)	0.98	0.87	0.75	0.78	0.98
RandomForest (FS)	0.99	0.88	0.85	0.86	0.98
LightGBM (AF)	0.94	0.54	0.60	0.57	0.94
LightGBM (FS)	0.96	0.59	0.60	0.59	0.95
XGBoost (AF)	0.99	0.92	0.73	0.77	0.98
XGBoost (FS)	0.99	0.93	0.85	0.86	0.98

TABLE IV. Methodological Comparison with Existing IDS Studies

Study	Dataset	Model Type	Feature Selection	Multi-Dataset	Model Complexity	Accuracy%
Qazi et al. [20]	CICIDS-2018	CNN–RNN	No	No	High	98.90
Proposed Framework	CIC-IDS2017, CIC-DDoS2017, CSE-CIC-IDS2018, CIC-DDoS2019	ML-based Ensemble	**ANOVA**	**Yes**	Lower	99.00

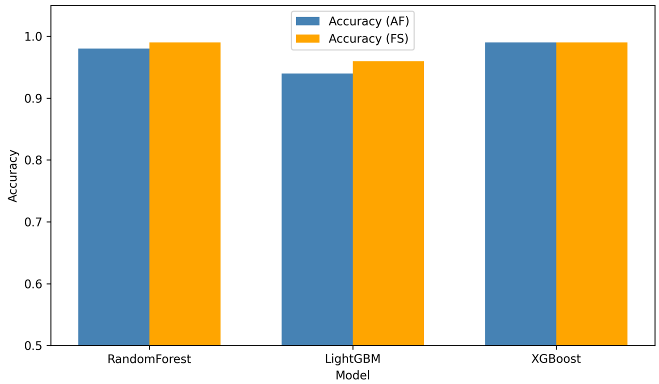


Fig. 3. Accuracy comparison of intrusion detection models using All Features (AF) and Feature Selection (FS).

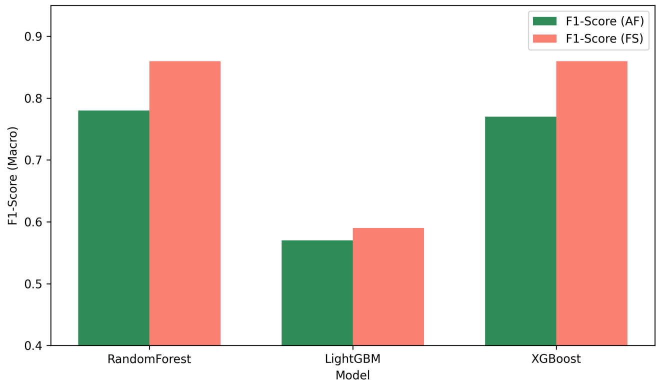
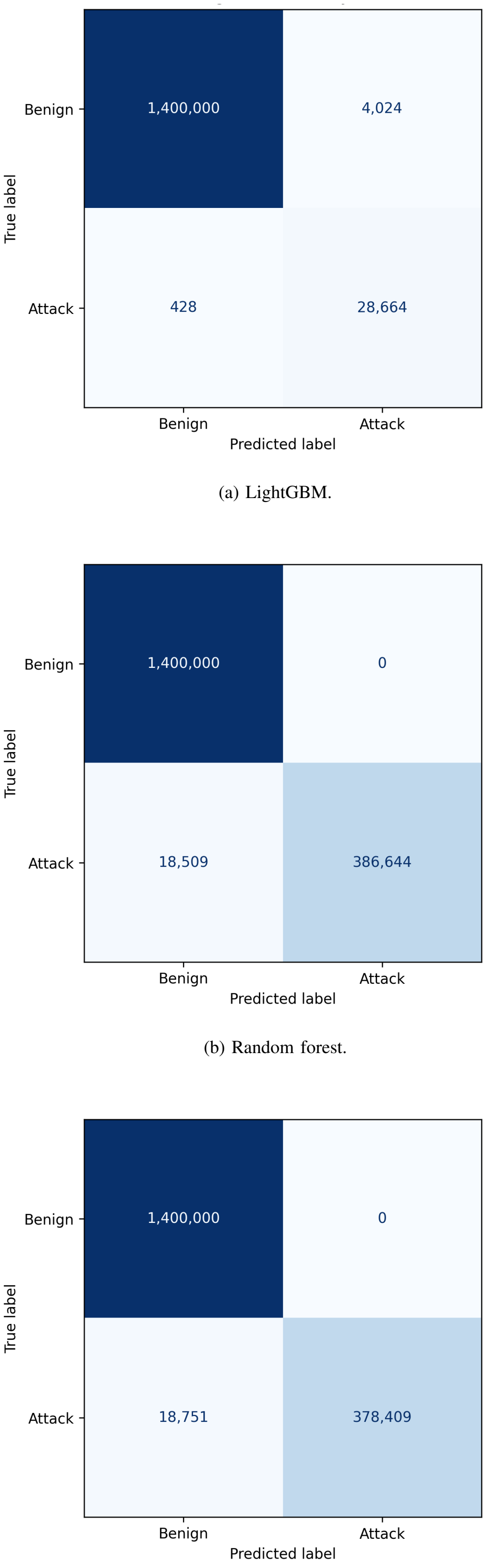


Fig. 4. F1-score comparison of intrusion detection models using All Features (AF) and Feature Selection (FS).

setups, reinforcing their suitability for scalable, high-accuracy intrusion detection across diverse network traffic types.

Fig. 5 shows the confusion matrix of the machine learning models in case of binary classification. In addition, Fig. 6 shows the confusion matrix of XGBoost models in case of feature selection, which illustrates the classification behavior across different attack categories. The model demonstrates strong capability in identifying benign traffic, with nearly all normal flows correctly detected and very few false positives. This behavior is essential for maintaining operational stability and reducing unnecessary alerts in real deployment environments. High detection accuracy is also evident for major attack classes such as DDoS and DoS, confirming that the model effectively captures large-scale malicious activity patterns. Some misclassification occur in less frequent attack types such as Infiltration and Web Attack. These errors likely result from class imbalance and the limited number of representative samples in those categories. Even so, the classifier maintains stable precision and recall across the primary attack types, showing that feature selection and ensemble learning enhance the model's ability to separate overlapping network behaviors. Overall, the confusion matrix supports the quantitative findings reported earlier. The results confirm that the proposed intrusion detection approach achieves high accuracy and robustness across varied network threats, while sustaining a very low false-alarm rate—an essential property for scalable and reliable AI-based intrusion detection in practical applications. On the CICIDS-2018 dataset, the results obtained by the proposed framework are consistent with those reported by Qazi et al. [20], despite the methodological differences between the two approaches. While HDLNIDS relies on a deep CNN–RNN architecture trained and evaluated on a single dataset, the proposed framework adopts a machine learning–based ensemble strategy supported by ANOVA-based feature selection. This design choice enables effective dimensionality reduction while maintaining competitive detection performance.



(a) LightGBM.

(b) Random forest.

(c) XGboost.

Fig. 5. Comparative visualization of intrusion detection performance using Feature Selection (FS) across different machine learning models for binary classification.

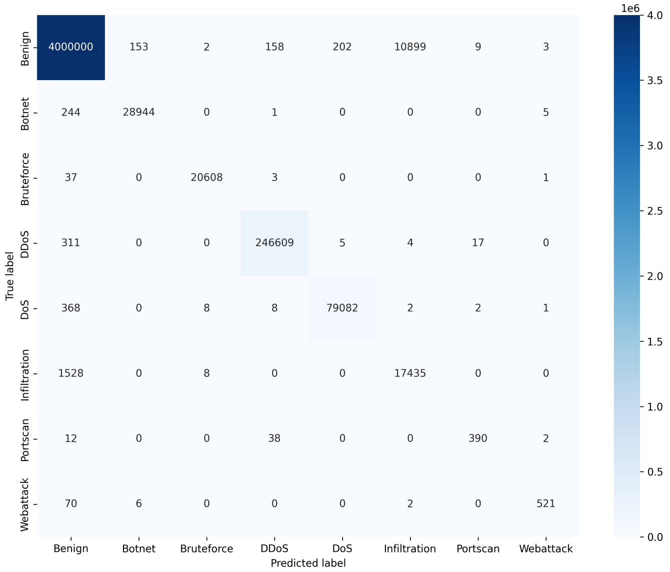


Fig. 6. Confusion matrix of XGBoost model Feature Selection (FS).

Moreover, unlike HDLNIDS, which focuses on a single experimental setting, the proposed approach demonstrates stable behavior across multiple CIC-based datasets, suggesting improved robustness and generalization under heterogeneous traffic conditions. Table IV shows the comparison between our work and one of the latest research studies.

VII. Conclusion

This study presented an AI-driven intrusion detection framework designed to enhance network security through consistent and leakage-free machine learning pipelines. The framework integrated RandomForest, LightGBM, and XGBoost classifiers and was validated under both binary and multi-class settings using multiple datasets to assess robustness and generalization across diverse network conditions. The experimental results demonstrated that all models achieved strong performance in binary intrusion detection, with RandomForest and XGBoost achieving the highest accuracy (0.99) and stable F1-scores above 0.97. In the more challenging multi-class scenario, both models maintained high accuracy (0.99) and balanced detection across attack categories, confirming their reliability in distinguishing complex intrusion patterns. LightGBM showed slightly lower macro-level performance but benefited modestly from ANOVA-based feature selection, which improved its recall and stability. Feature selection proved effective in reducing feature redundancy and improving computational efficiency without compromising classification accuracy. This finding highlights that robust ensemble methods such as RandomForest and XGBoost can maintain high performance even when trained on reduced feature spaces, making them suitable for large-scale and resource-constrained environments. Future work will focus on three directions: First, model scalability and efficiency will be advanced by exploring compact deep architectures such as CNN–RNN hybrids and lightweight transformer variants for real-time deployment. Second, explainability and transparency will be enhanced through SHAP-based feature interpretation to increase analyst trust and interpret model behavior. Third, robustness will be extended through evaluation on adversarial and real-world datasets, integration of continual learning to adapt to evolving threats, and hybridization with anomaly detection mechanisms to identify both known and emerging attacks. Collectively, these efforts aim to develop intrusion detection systems that are accurate, interpretable, and resilient across diverse and dynamic network environments.

Acknowledgment

The authors express their gratitude to Széchenyi István University of Győr for providing institutional support and to the Stipendium Hungaricum Scholarship Programme for its financial assistance.

References

[1] J. Halvorsen, C. Izurieta, H. Cai, and A. Gebremedhin, “Applying generative machine learning to intrusion detection: A systematic mapping study and review,” *ACM Comput. Surv.*, vol. 56, no. 10, Jun. 2024. [Online]. Available: https://doi.org/10.1145/3659575

[2] Y. S. Razooqi and A. Pekar, “Vpn traffic analysis: A survey on detection and application identification,” *IEEE Access*, vol. 13, pp. 132 830–132 848, 2025.

[3] U. Ahmed, M. Nazir, A. Sarwar, T. Ali, E.-H. M. Aggoune, T. Shahzad, and M. A. Khan, “Signature-based intrusion detection using machine learning and deep learning approaches empowered with fuzzy clustering,” vol. 15, no. 1, p. 1726, 2025. [Online]. Available: https://www.nature.com/articles/s41598-025-85866-7

[4] M. A. Talukder, M. Khalid, and N. Sultana, “A hybrid machine learning model for intrusion detection in wireless sensor networks leveraging data balancing and dimensionality reduction,” *Scientific Reports*, vol. 15, no. 1, p. 4617, 2025.

[5] M. M. Abou-elasaad, S. G. Sayed, and M. M. El-dakroury, “Smart grid intrusion detection system based on ai techniques,” vol. 15, no. 2, pp. 195–207, 2025, accessed: 2025-10-19. [Online]. Available: https://www.americaspg.com/article/pdf/3358

[6] F. Makhmudov, D. Kilichev, U. Giyosov, and F. Akhmedov, “Online machine learning for intrusion detection in electric vehicle charging systems,” vol. 13, no. 5, p. 712, 2025. [Online]. Available: https://doi.org/10.3390/math13050712

[7] J. Li, M. S. Othman, H. Chen, and L. M. Yusuf, “Optimizing iot intrusion detection system: feature selection versus feature extraction in machine learning,” *Journal of Big Data*, vol. 11, no. 1, pp. 1–44, 2024. [Online]. Available: https://journalofbigdata.springeropen.com/articles/10.1186/s40537-024-00892-y

[8] M. S. Nawaz, M. A. Raza, B. Raza, M. Ahmad, and F. Syed, “Ai-driven intrusion detection systems for securing iot healthcare networks.” *International Journal of Advanced Computer Science & Applications*, vol. 16, no. 6, 2025.

[9] M. M. Rahman, S. Al Shakil, and M. R. Mustakim, “A survey on intrusion detection system in iot networks,” *Cyber Security and Applications*, vol. 3, p. 100082, 2025.

[10] R. R. Akula and G. Kumar, “Optimizing feature selection in intrusion detection systems using a genetic algorithm with stochastic universal sampling.” *International Journal of Advanced Computer Science & Applications*, vol. 16, no. 1, 2025.

[11] P. Schummer, A. del Rio, J. Serrano, D. Jimenez, G. Sánchez, and Á. Llorente, “Machine learning-based network anomaly detection: Design, implementation, and evaluation,” *AI*, vol. 5, no. 4, pp. 2967–2983, 2024.

[12] Y. Zhang, R. C. Muniyandi, and F. Qamar, “A review of deep learning applications in intrusion detection systems: Overcoming challenges in spatiotemporal feature extraction and data imbalance,” *Applied Sciences*, vol. 15, no. 3, p. 1552, 2025.

[13] R. Basfar, M. Y. Dahab, A. M. Ali, F. Eassa, and K. Bajunaied, “An incremental lstm ensemble for online intrusion detection in software-defined networks.” *International Journal of Advanced Computer Science & Applications*, vol. 16, no. 9, 2025.

[14] S. Çalışır, R. Atay, M. K. Pehlivanoğlu, and N. Duru, “Intrusion detection using machine learning and deep learning techniques,” in *2019 4th International Conference on Computer Science and Engineering (UBMK)*. IEEE, 2019, pp. 656–660.

[15] P. J. MP *et al.*, “An explainable and optimized network intrusion detection model using deep learning.” *International Journal of Advanced Computer Science & Applications*, vol. 15, no. 1, 2024.

[16] M. Torabi, N. I. Udzir, M. T. Abdullah, and R. Yaakob, “A review on feature selection and ensemble techniques for intrusion detection system,” *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 5, 2021.

[17] B. A. Durga and N. Mangla, “A novel network intrusion detection system based on semi-supervised approach for iot,” *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 4, 2023.

[18] S. A. Hussein, A. A. Mahmood, and E. O. Oraby, “Network intrusion detection system using ensemble learning approaches,” *Technology*, vol. 18, pp. 962–974, 2021.

[19] M. K. Hooshmand and D. Hosahalli, “Network anomaly detection using deep learning techniques,” *CAAI Transactions on Intelligence Technology*, vol. 7, no. 2, pp. 228–243, 2022.

[20] E. U. H. Qazi, M. H. Faheem, and T. Zia, “Hdlnids: hybrid deep-learning-based network intrusion detection system,” *Applied Sciences*, vol. 13, no. 8, p. 4921, 2023.

[21] M. K. Hooshmand, M. D. Huchaiah, A. R. Alzighaibi, H. Hashim, E.-S. Atlam, and I. Gad, “Robust network anomaly detection using ensemble learning approach and explainable artificial intelligence (xai),” *Alexandria Engineering Journal*, vol. 94, pp. 120–130, 2024.

[22] S. S. Shafin, “An explainable feature selection framework for web phishing detection with machine learning,” *Data Science and Management*, 2024.

[23] S. H. Almotiri, “Ai driven iomt security framework for advanced malware and ransomware detection in sdn,” *Journal of Cloud Computing: Advances, Systems and Applications*, vol. 14, 2025. [Online]. Available: https://journalofcloudcomputing.springeropen.com/articles/10.1186/s13677-025-00745-w

[24] S. Kaushik, A. Bhardwaj, A. Almogren, S. Bharany, A. Altameem, A. U. Rehman, S. Hussen, and H. Hamam, “Robust machine learning based intrusion detection system using simple statistical techniques in feature selection,” *Scientific Reports*, vol. 15, no. 1, p. 3970, 2025.

[25] M. A. O. Ahmed, Y. Abdelsatar, R. Alotaibi, and O. Reyad, “Enhancing internet of things security using performance gradient boosting for network intrusion detection systems,” *Alexandria Engineering Journal*, vol. 116, pp. 472–482, 2025.

[26] L. Shi, Q. Yang, L. Gao, and H. Ge, “An ensemble system for machine learning iot intrusion detection based on enhanced artificial hummingbird algorithm,” *The Journal of Supercomputing*, vol. 81, no. 1, p. 110, 2025.

[27] M. Ouhssini, K. Afdel, E. Agherrabi, M. Akouhar, and A. Abarda, “Deepdefend: A comprehensive framework for ddos attack detection and prevention in cloud computing,” *Journal of King Saud University-Computer and Information Sciences*, vol. 36, no. 2, p. 101938, 2024.

[28] dhoogla. Cicids collection. Kaggle. [Online]. Available: https://www.kaggle.com/datasets/dhoogla/cicidscollection

[29] W. Zhu, R. Qiu, and Y. Fu. Comparative study on the performance of categorical variable encoders in classification and regression tasks. arXiv. [Online]. Available: https://arxiv.org/abs/2401.09682

[30] Scikit learn Developers, “Scikitlearn: sklearn.featureselection.selectkbest,” https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectKBest.html, 2025, accessed: 2025-10-21.

[31] ——, “Scikitlearn: sklearn.featureselection.fclassif,” https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.f_classif.html, 2025, accessed: 2025-10-21.

[32] A. A. Bhutta, M. un Nisa, and A. N. Mian, “Lightweight real-time wifi-based intrusion detection system using lightgbm,” vol. 30, no. 2, pp. 749–761, 2024. [Online]. Available: https://doi.org/10.1007/s11276-023-03516-0

[33] N. Nabil, N. Najib, and J. Abdellah, “Leveraging artificial neural networks and lightgbm for enhanced intrusion detection in automotive systems,” *Arabian Journal for Science and Engineering*, vol. 49, no. 9, pp. 12 579–12 587, 2024. [Online]. Available: https://link.springer.com/article/10.1007/s13369-024-08787-z

[34] L. Shi, Q. Yang, L. Gao, and H. Ge, “An ensemble system for machine learning iot intrusion detection based on enhanced artificial hummingbird algorithm,” 2025, online first (Nov 2024); assigned to 2025 issue. Add volume/number/pages when available. [Online]. Available: https://link.springer.com/article/10.1007/s11227-024-06475-1

[35] C. Lu, Y. Cao, and Z. Wang, “Research on intrusion detection based on an enhanced random forest algorithm,” vol. 14, no. 2, p. 714, 2024. [Online]. Available: https://doi.org/10.3390/app14020714

[36] X. Yu, W. Meng, Y. Liu, and F. Zhou, “Tridentshell: An enhanced covert and scalable backdoor injection attack on web applications,” *Journal of Network and Computer Applications*, vol. 223, pp. 1–12, 2024. [Online]. Available: https://doi.org/10.1016/j.jnca.2023.103823