Received: 1 September 2025 | Revised: 22 December 2025 | Accepted: 20 January 2026 | Published online: 22 April 2026

RESEARCH ARTICLE

Journal of Data Science and Intelligent Systems
2026, Vol. 4(2) 193–201
DOI: 10.47852/bonviewJDSIS62027552

BON VIEW PUBLISHING

Correlation-Based Feature Selection for Efficient Intrusion Detection: Comparative Evaluation of Machine Learning Models on CICIDS-2017

Bilal Rafique[1], Sania Kanwal[1] and Xuyang Shi[1,*]

[1]*School of Information and Control Engineering, Southwest University of Science and Technology, China*

Abstract: The increased growth of network traffic has enabled digital communication, global connectivity, driving advances in e-commerce, and cloud computing. However, this growth also increases the risk of cyberattacks, making effective and efficient intrusion detection systems (IDS) essential. Although many studies have applied machine learning (ML) and deep learning (DL) to improve detection accuracy, the majority pay less attention to computational efficiency, which is critical for real-time deployment. The proposed study evaluates three widely used ML models, namely, Random Forest (RF), Extreme Gradient Boosting (XGBoost), and Multilayer Perceptron (MLP), on the CICIDS-2017 dataset, with and without correlation-based feature selection (CFS). Results show that RF and XGBoost achieved the highest accuracy (0.998) with very low false positive rates, while MLP lagged behind in both detection and runtime performance. Notably, applying CFS reduced RF's training time by 35% without sacrificing accuracy. The proposed study's findings confirm that ensemble models, particularly RF and XGBoost, provide a strong balance between accuracy and efficiency. Moreover, feature selection emerges as a simple yet effective strategy to lower computational cost, making IDS more practical for large-scale, real-time network environments.

Keywords: intrusion detection system, XGBoost, Random Forest, MLP, feature selection

1. Introduction

The rapid expansion of communication networks, driven by the growth of the Internet, cloud technologies, and connected devices, has dramatically increased the diversity and volume of network traffic. This vast interconnected network enhances global communication and accessibility but introduces numerous cybersecurity threats. Modern networks are continually targeted by different types of cyber threats such as denial-of-service (DoS), distributed denial-of-service (DDoS), port scanning, botnets, and stealthy man-in-the-middle (MITM) attacks [1–3].

In the real world, several cyberattack incidents have caused severe disruptions and financial damage. For instance, the NotPetya cyberattack in 2017 triggered a cascading supply-chain crisis that resulted in an estimated USD 7.3 billion in losses, far exceeding the immediate damage to directly compromised firms [4]. Similarly, in 2022, during the Russia-Ukraine conflict, the malware attack on Viasat's KA-SAT satellite network disabled thousands of modems and disrupted internet communication across Ukraine and large parts of Europe, demonstrating how cyberattacks can severely affect both civilian and military connectivity [5, 6].

Consequently, intrusion detection systems (IDS) have become indispensable for modern cybersecurity architectures because they enable real-time differentiation between benign and malicious traffic. Although existing IDS research extensively evaluates performance indicators such as accuracy, F1-score, recall, and precision standard metrics for assessing detection capability [7–9], these measures alone do not fully capture how well an IDS can operate under real-world, high-throughput network conditions. In practice, a model must be not only accurate but also computationally efficient, particularly in environments where millions of flows per second must be analyzed [10, 11].

Despite this reality, very few studies systematically analyze how feature reduction techniques such as correlation-based selection (CFS) affect the computational cost of machine learning (ML) models. Training and testing times are often overlooked, although they directly influence IDS performance in large-scale networks such as cloud environments and enterprise infrastructures. A model with extremely high accuracy but long training or slow inference is impractical because it cannot capture the incoming network attacks in real time [12–15].

For example, the widely used CICIDS-2017 dataset [16] includes 78 flow-based features, many of which are redundant or highly correlated, which, if not properly preprocessed, can lead to prolonged training times and poor generalization [17].

To address these limitations, the present study conducts a systematic comparison of three commonly used ML models, namely, Random Forest (RF), XGBoost, and Multilayer Perceptron (MLP), on the CICIDS-2017 dataset. We apply the CFS feature selection technique to reduce redundant attributes and evaluate its impact on both computational efficiency (training and testing times) and classification performance (accuracy, precision, recall, and F1-score).

The following are the main contributions of the proposed study:

1) A comparative evaluation of RF, XGBoost, and MLP under both full-feature and reduced-feature settings.
2) Empirical evidence that feature selection significantly reduces training and inference times while maintaining competitive detection performance.

*Corresponding author: Xuyang Shi, School of Information and Control Engineering, Southwest University of Science and Technology, China. Email: xuyangshi@swust.edu.cn

2. Methodology

This section explains the methodology implemented in the proposed study, including dataset description, preprocessing of data, balancing datasets, feature selection technique, and implemented ML models. Figure 1 presents the overall flow of the proposed methodology.

2.1. Dataset description

The widely used CICIDS-2017 dataset [16] is a well-established benchmark for network intrusion detection research. It includes 78 flow-based features, many of which are redundant or highly correlated, and if not properly preprocessed, such characteristics may lead to prolonged training times and poor generalization [17]. These features capture diverse network flow characteristics, including packet timing, size distributions, flow duration, and statistical measures, which require careful selection to mitigate the curse of dimensionality.

The dataset was generated in 2017 by the Canadian Institute for Cybersecurity (CIC) and contains both benign and malicious traffic collected over five days of real network activity. It encompasses multiple attack scenarios, including PortScans, brute force attacks, DoS, DDoS, botnets, and web-based attacks, thereby representing a comprehensive range of contemporary cyber threats. In general, the dataset comprises approximately 2.8 million bidirectional network flow records extracted using CICFlowMeter [16].

2.2. Data preprocessing

Before model implementation, the dataset is preprocessed to ensure data quality and consistency. Null and infinite values are systematically identified and removed to prevent computational errors and to maintain data integrity during the training process. Categorical and protocol-type features are transformed into numerical representations using label encoding techniques, ensuring compatibility with machine learning algorithms that require numerical input. Numerical features are standardized using z-score normalization (mean = 0, standard deviation = 1) to maintain consistency across attributes with different scales and ranges. This normalization step prevents features with larger magnitudes from dominating the learning process and contributes to improved model convergence and stability.

Figure 1
Overall flow of the proposed methodology

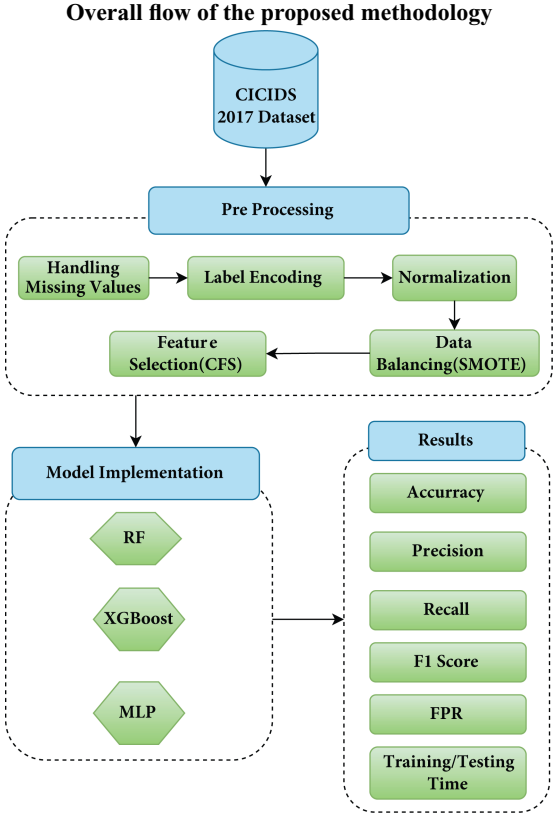
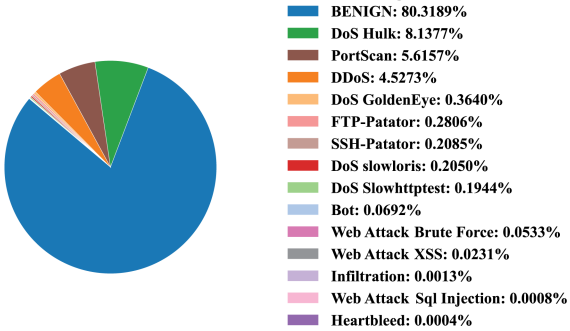


Figure 2
Class distribution before balancing the dataset



2.3. Data balancing

The CICIDS-2017 dataset has a highly imbalanced distribution of attack categories, where the majority classes, such as benign, DoS, and PortScan, dominate the minority classes like Infiltration and Heartbleed. To mitigate this imbalance, the Synthetic Minority Oversampling Technique (SMOTE) is applied to the training set. SMOTE generates synthetic samples for minority classes by interpolating between nearest neighbors [18], thereby balancing the class distribution and improving the ability of classifiers to detect rare attacks. Figure 2 illustrates the class distribution before balancing the datasets, while Figure 3 presents the balanced class distribution after the implementation of the SMOTE technique.

2.4. Correlation-based feature selection

High dimensionality often introduces redundant and highly correlated features, which increase computational overhead and may lead to overfitting by allowing the model to memorize noise rather than learning generalizable patterns. To address this issue, a

Figure 3
Class distribution after balancing the dataset

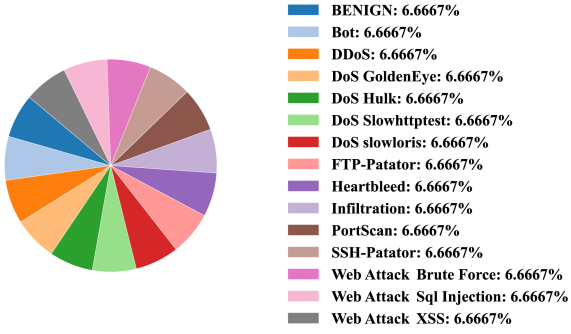
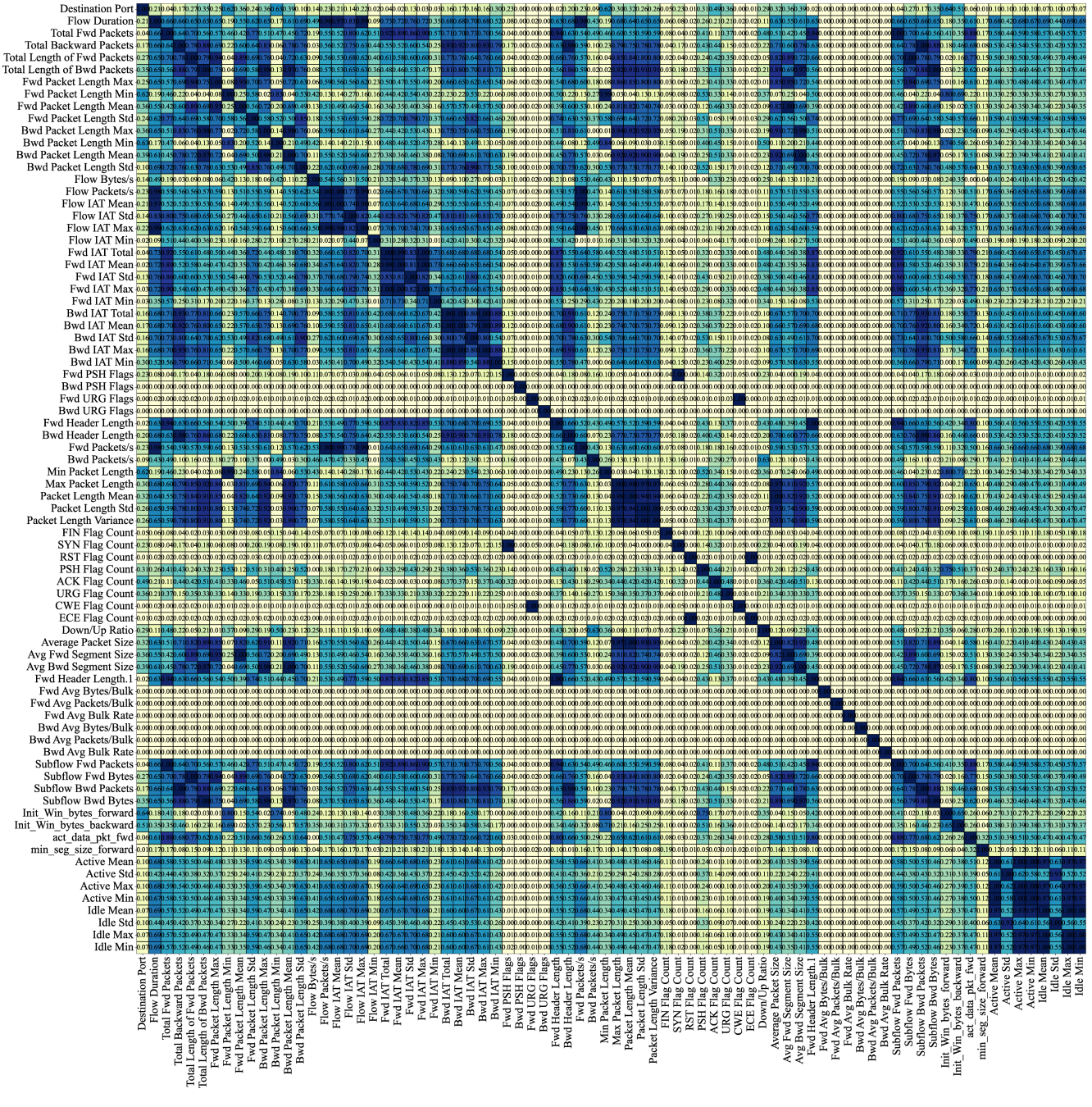


Figure 4
Correlation-based feature selection



CFS technique is employed to identify and eliminate feature redundancy systematically. The pairwise correlation between features is calculated using the Spearman correlation coefficient, which is robust to non-linear relationships and outliers, providing a more comprehensive measure of feature dependency than Pearson correlation [19, 20].

Features exhibiting correlation values greater than 90% are considered redundant and potentially harmful to model performance. In such cases, one feature from each highly correlated pair is retained based on its average correlation with all other features. The feature with a lower overall average correlation (i.e., more independent) is preserved, while the more redundant one is removed. This ensures that the retained features contribute unique information to the model rather than overlapping patterns.

This iterative selection process effectively preserves the most informative and less redundant features, reducing training time and computational complexity while improving generalization by eliminating multicollinearity, which can cause parameter instability and inflated variance in model predictions. For example, Figure 4 shows that 39 features exhibit correlations greater than 90%, demonstrating the substantial redundancy present in the original feature set.

2.5. Machine learning models

Three ML models are implemented to evaluate the effect of feature selection on detection performance and computational cost:

Random Forest (RF): RF is an ensemble learning algorithm that constructs multiple decision trees (DT) using bootstrap samples and aggregates their outputs through majority voting. Its ability to handle high-dimensional data and reduce variance makes it a strong baseline for IDS tasks [21, 22]. Mathematically,

$$\{h_1(x), h_2(x), \ldots, h_T(x)\} \tag{1}$$

are the predictions of T individual trees. The RF output is given by:

$$\hat{y} = \text{mode}\big(h_1(x), h_2(x), \ldots, h_T(x)\big) \tag{2}$$

for classification, where the majority vote determines the final prediction.

Extreme Gradient Boosting (XGBoost): XGBoost is a scalable and regularized version of gradient boosting that iteratively builds DT to minimize a loss function. It incorporates shrinkage, subsampling, and regularization parameters, enabling it to achieve high accuracy while mitigating overfitting [23]. The objective function to be minimized is defined as:

$$\mathscr{L}(\phi) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k), \tag{3}$$

where l is the differentiable loss function (e.g., logistic loss), $\hat{y}_i$ is the prediction for instance i, and f_k is the k-th tree. The regularization term is:

$$\Omega(f_k) = \gamma T + \tfrac{1}{2}\lambda \|w\|^2, \tag{4}$$

which controls model complexity. The additive model at iteration t is:

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + f_t(x_i). \tag{5}$$

Multilayer Perceptron (MLP): The MLP is a feed-forward artificial neural network consisting of an input layer, one or more hidden layers, and an output layer [24, 25]. The output layer uses the Softmax activation function for multiclass classification. Mathematically, for an input vector x, the forward propagation through one hidden layer is:

$$h = \sigma\big(W^{(1)}x + b^{(1)}\big), \tag{6}$$

$$o = \text{Softmax}\big(W^{(2)}h + b^{(2)}\big), \tag{7}$$

where $W^{(1)}$ and $W^{(2)}$ denote the weight matrices, $b^{(1)}$ and $b^{(2)}$ represent the bias terms, and $\sigma(\cdot)$ is the ReLU activation function defined as:

$$\sigma(z) = \max(0, z), \tag{8}$$

where z is the input to the activation layer and o represents the output probability distribution across the classes.

3. Experimental Results and Discussion

This section presents the experimental results obtained using the CICIDS-2017 dataset. The main focus in this section is on detection performance, false alarm behavior, and computational efficiency. The discussion is organized first to present quantitative results and subsequently interpret the observed performance trends.

Table 1
Formulas of performance metrics

Metric	Formula
Accuracy	$\dfrac{TP+TN}{TP+FP+TN+FN}$
Recall	$\dfrac{TP}{TP+FN}$
Precision	$\dfrac{TP}{TP+FP}$
F1-score	$\dfrac{2\cdot\text{Precision}\cdot\text{Recall}}{\text{Precision}+\text{Recall}}$
False positive rate (FPR)	$\dfrac{FP}{FP+TN}$

3.1. Experimental setup and performance evaluation

The experiments for all three AI models are implemented in Python using the NumPy, Pandas, and Scikit-learn libraries. To ensure reproducibility of the reported execution times, all experiments are conducted on a system running Windows 11, equipped with an Intel Core i5 (13th generation) CPU and an NVIDIA GeForce RTX 4050 GPU. Both the RF and XGBoost models are configured with 50 estimators and a maximum tree depth of 35, while all other hyperparameters followed standard settings. The dataset is divided into 80% training and 20% testing sets. Before model training, redundant features are removed using Spearman correlation analysis, and SMOTE is applied to address class imbalance.

The models are evaluated based on standard classification metrics: accuracy, precision, recall, F1-score, and false positive rate (FPR). Table 1 shows the mathematical formula of these performance metrics.

3.2. Overall classification performance

Figure 5 illustrates a comparative analysis of RF, XGBoost, and MLP. RF and XGBoost achieved the highest accuracy of 0.998, demonstrating their strong classification capability. However, when

Figure 5
Comparison of ML models in terms of accuracy, precision, recall, and F1-score

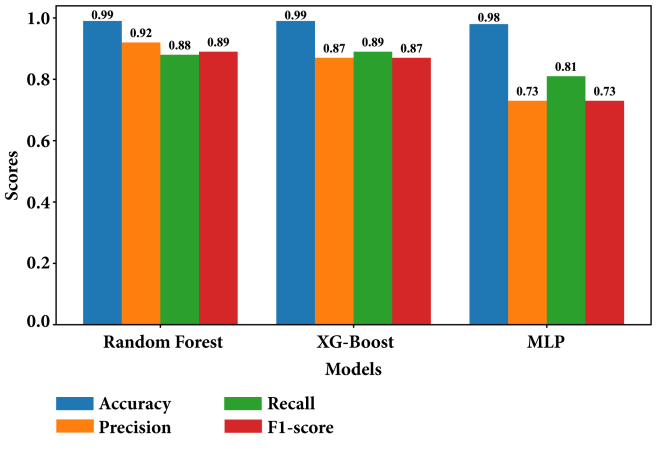
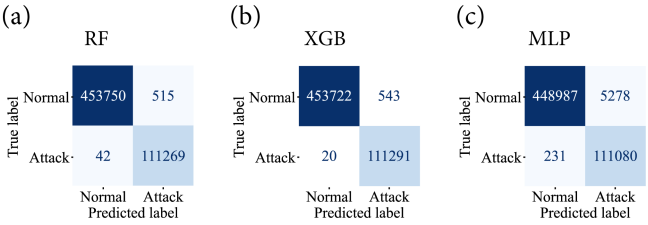


Figure 6
Confusion matrices computed for three AI models



considering other metrics, notable differences emerge. RF achieved the best precision (0.92), indicating its superior ability to minimize false positives and classify benign and attack traffic with higher reliability. In contrast, XGBoost achieved slightly lower precision (0.87) but demonstrated higher recall (0.89), suggesting that it is more effective in identifying a larger proportion of attack instances, even if at the cost of a few additional false positives. The F1-score of RF (0.89) is also higher than that of XGBoost (0.87), showing a better balance between precision and recall. The MLP model significantly underperformed compared to the tree-based approaches, achieving an accuracy of 0.98 with considerably lower precision (0.73) and F1-score (0.73).

3.3. Confusion matrix analysis

The confusion matrices for RF, XGBoost, and MLP in Figure 6 illustrate the comparative classification behavior. RF achieves the smallest number of false positives (515) and false negatives (42), as shown in Figure 6(a), indicating highly reliable detection of both classes. XGBoost shows a similar balanced performance, vwith only marginal increases in misclassified instances. In contrast, Figure 6(c) shows that the MLP model displays noticeably higher false positives (5,278) and false negatives (231), reflecting reduced precision and recall compared to the tree-based models.

3.4. ROC curve analysis

Similarly, the ROC curves for the three models (RF, XGB, and MLP) presented in Figure 7 demonstrate excellent performance, with all models showing a perfect AUC of 1.00. In each case, the true positive rate increases sharply with minimal false positive rates, indicating that the models are able to effectively distinguish between normal and attack instances. This suggests that all three classifiers have a high level of accuracy in detecting network intrusions. The close alignment of the curves with the top-left corner further supports the models' robustness in terms of sensitivity and specificity, particularly for detecting rare attack classes in intrusion detection systems.

3.5. False positive rate analysis

Figure 8 presents the comparison of RF, XGBoost, and MLP in terms of their FPR. This metric is critical in intrusion detection because a high FPR means that a large portion of benign traffic is incorrectly classified as malicious, which can overwhelm security teams with false alarms.

As shown in Figure 8, XGBoost achieved the lowest FPR (0.000099), making it the most reliable model in minimizing false alarms. This suggests that XGBoost can distinguish benign traffic from attack traffic with high precision, an important property for real-world IDS deployments. RF followed closely with an FPR of 0.000108, only slightly higher than XGBoost, indicating that it is also highly effective at controlling false positives. In contrast, the MLP model recorded a much higher FPR (0.000830) compared to the ensemble methods. Although still relatively small in absolute terms, this difference is significant in large-scale network environments where millions of flows are analyzed daily.

3.6. Comparison of computational efficiency and false positive rate

Table 2 presents a comprehensive comparison of RF, XGBoost, and MLP models in terms of their F1-score, training time, and testing time, revealing important trade-offs between detection performance and computational efficiency. Random Forest demonstrates superior performance by achieving the highest F1-score (0.89) while maintaining excellent computational efficiency with a relatively low training time (4.98 seconds) and moderate testing time (0.68 seconds). This combination of high accuracy and efficiency makes RF an ideal balanced solution for intrusion detection tasks where both quick training and reasonably fast inference are critical requirements. XGBoost delivers competitive detection performance with an F1-score of 0.87, although it requires significantly longer training time (21.76 seconds) compared to RF. However, it compensates with the fastest testing time (0.42 seconds) among all models, demonstrating exceptional inference efficiency once the training phase is complete. This characteristic makes XGBoost particularly

Figure 7
ROC computed for three AI models

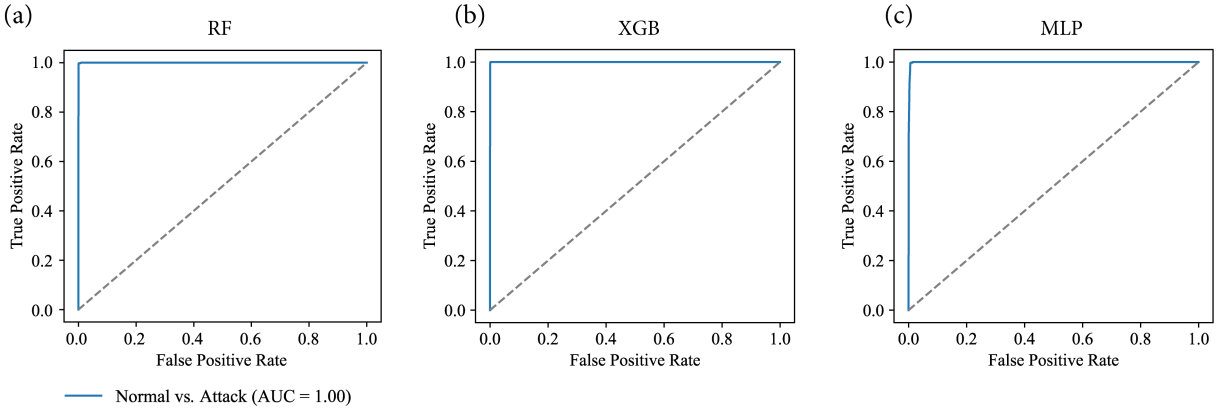
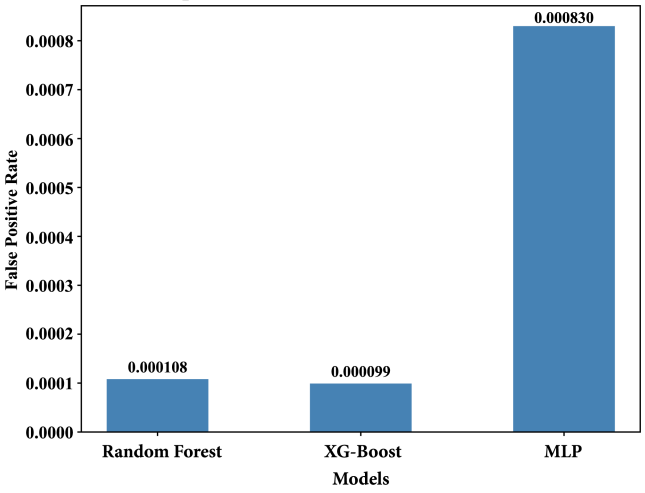


Figure 8
False positive rate (FPR) across models



suitable for scenarios where training can be performed offline during non-peak hours, and fast real-time inference is the primary operational requirement. In contrast, MLP exhibits the weakest overall performance across all metrics. With the lowest F1-score (0.73), extremely high training time (794.17 seconds), and the slowest testing time (1.40 seconds), MLP proves to be both less accurate and computationally intensive, making it less suitable for practical intrusion detection applications.

3.7. Per-class performance evaluation

Similarly, Table 3 comprehensively presents the per-class F1-scores for RF, XGBoost, and MLP, highlighting clear performance differences across the diverse attack categories. RF and XGBoost demonstrate consistently strong performance, achieving near-perfect F1-scores for the majority of the attack types such as DDoS, DoS variants, PortScan, SSH-Patator, and benign traffic. Their performance begins to diverge slightly in more challenging minority classes. For example, it can be seen in Table 3 that RF marginally outperforms XGBoost on Heartbleed (1.00 vs. 0.80), while both models show comparable results on Bot and Infiltration, where scores remain moderate due to class imbalance and subtle feature patterns. In contrast, the MLP model shows considerably weaker per-class performance, particularly for minority and low-frequency classes such as Infiltration (0.31), Web Attack Brute Force (0.28), Web Attack XSS (0.02), and Web Attack SQL Injection, where it fails to detect any instances (0.00). Even in classes where MLP performs reasonably well, its scores remain consistently lower than those of RF and XGBoost. Overall, the per-class comparison demonstrates the superior robustness and generalization of RF and XGBoost across both frequent and rare attack types, while MLP struggles significantly with classes exhibiting limited training samples or subtle behavioral signatures.

Table 2
Comparison of models in terms of F1-score, training time, and testing time

Model	F1-score	Training time (s)	Testing time (s)
Random Forest	0.89	4.983	0.683
XG-Boost	0.87	21.758	0.423
MLP	0.73	794.170	1.406

Table 3
F1-score per class for RF, XGB, and MLP

Class	F1-score		
	RF	XGB	MLP
Benign	1.00	1.00	0.99
Bot	0.79	0.78	0.57
DDoS	1.00	1.00	1.00
DoS GoldenEye	1.00	1.00	0.95
DoS Hulk	1.00	1.00	0.97
DoS Slowhttptest	0.99	0.98	0.92
DoS slowloris	1.00	1.00	0.99
FTP-Patator	1.00	1.00	0.98
Heartbleed	1.00	0.80	1.00
Infiltration	0.73	0.73	0.31
PortScan	1.00	1.00	0.98
SSH-Patator	1.00	1.00	0.96
Web Attack Brute Force	0.77	0.79	0.28
Web Attack Sql Injection	0.67	0.67	0.00
Web Attack XSS	0.37	0.24	0.02

3.8. Statistical significance analysis

To evaluate whether the differences in performance among the three classifiers are statistically meaningful, we conducted a significance analysis using the model-level evaluation metrics (accuracy, precision, recall, F1-score, and FPR). Because the original sample-level predictions are not stored, pairwise error-overlap tests such as McNemar's test could not be applied. Instead, by following standard practice in comparative ML studies, we used the Friedman test, a non-parametric test suitable for repeated measures across multiple models. The test revealed statistically significant performance differences among the three classifiers ($p < 0.05$). Post-hoc Nemenyi analysis showed that RF significantly outperformed both XGBoost and MLP, while XGBoost also achieved significantly better performance than MLP. These results confirm the superiority of the RF model across the evaluated metrics.

3.9. Impact of correlation-based feature selection

Table 4 highlights the effect of applying CFS on the performance and efficiency of the RF model. The detection accuracy remained constant at 0.998 for both settings, confirming that the removal of highly correlated features did not negatively affect the model's predictive capability. In fact, slight improvements in precision (from 0.91 to 0.92) and F1-score (from 0.88 to 0.89) are observed when CFS is applied, suggesting that eliminating redundant features helps the model focus on more informative attributes. From a computational perspective, the benefits of feature selection are more pronounced. The training time decreased from 7.65 seconds (without CFS) to 4.98 seconds (with CFS), indicating a significant reduction in the resources required for model training. Similarly, the testing time improved from 0.704 to 0.683 seconds. Although the difference in testing time is relatively modest, the reduction in training overhead is noteworthy, especially for large-scale IDS deployments where models must be retrained frequently.

3.10. Impact of SMOTE on minority-class detection

The F1-score comparison in Table 5 before and after SMOTE highlights the RF model's behavior on minority attack classes.

Table 4
Impact of feature selection on the training and testing times of the RF model

Model	Acc.	Prec.	Rec.	F1	FPR	Train (s)	Test (s)
RF with-CFS	0.998	0.92	0.88	0.89	0.000108	4.983	0.683
RF without-CFS	0.998	0.91	0.87	0.88	0.000139	7.650	0.704

While major categories such as Benign, DDoS, and PortScan maintained perfect detection (F1 = 1.00), minority classes showed clear improvement after oversampling, for instance, Infiltration (0.60 → 0.73) and Web Attack SQL Injection (0.40 → 0.67). This confirms that SMOTE enhances the detection of rare attacks. However, slight declines in Bot (0.83 → 0.79) and modest gains in Web Attack XSS (0.33 → 0.37) indicate residual misclassification between similar low-frequency attack types.

3.11. Comparison with existing studies

Table 6 presents a comparative analysis between the proposed RF-based implementation and existing IDS models applied on the CICIDS-2017 dataset. The proposed model achieved the highest accuracy of 99.8%, outperforming back propagation DNN [12] (98.3%), support vector machine (SVM) [13] (90%), and Gaussian Naive Bayes (GNB) [13] (83%), as well as a recent DNN-based [14] approach (99.7%). As shown in Table 6, the proposed correlation-based RF framework demonstrates superior accuracy compared to both traditional ML and DL models. This highlights the effectiveness of the proposed approach within the context of recent IDS research.

3.12. Discussion

The experimental results demonstrate that tree-based ensemble models consistently outperform the MLP model in both detection performance and computational efficiency. This superior efficiency is attributed to the inherent robustness of RF and XGBoost against redundant and highly correlated features, as well as the scalability of tree-based ensembles, which enables effective generalization across diverse and high-volume network traffic patterns.

In particular, RF aggregates multiple decorrelated decision trees through a bagging strategy, which reduces variance and yields stable predictions with relatively low computational overhead. This characteristic allows RF to maintain high detection accuracy while achieving faster training and inference times, even when operating on high-dimensional intrusion detection datasets.

XGBoost, on the other hand, employs a boosting-based learning mechanism that sequentially focuses on misclassified samples. This targeted optimization enhances recall by improving the detection of difficult or minority attack instances, albeit at the cost of slightly increased false positives and higher training complexity compared to RF. Nevertheless, once trained, XGBoost demonstrates excellent inference efficiency, making it well suited for real-time intrusion detection scenarios.

In contrast, the inferior performance and efficiency of the MLP model can be attributed to its sensitivity to feature redundancy, class imbalance, and hyperparameter configuration. Deep neural networks generally require extensive tuning and balanced training data to perform optimally on tabular datasets. As a result, MLP exhibits a higher training time, increased false alarms, and reduced robustness when applied to intrusion detection tasks characterized by imbalanced and correlated features.

CFS further enhances the efficiency of ensemble models by eliminating redundant and non-informative features, thereby reducing model complexity and training time without compromising detection accuracy. The observed improvements in both computational efficiency and detection reliability highlight the importance of incorporating feature selection into IDS pipelines, particularly for large-scale and resource-constrained deployment environments.

Although the proposed study does not introduce a new ML algorithm, its novelty lies in the comprehensive and systematic evaluation of correlation-based feature selection within an intrusion detection framework. Unlike many existing studies that primarily emphasize detection accuracy, this work jointly analyzes detection performance, false alarm rate, and computational efficiency, providing a more holistic assessment of IDS effectiveness.

Specifically, the study demonstrates how CFS influences not only classification accuracy but also training time, testing time, and model stability across different classifiers. By integrating feature redundancy analysis with detailed performance metrics and statistical significance testing, the proposed evaluation offers deeper insights into the practical benefits of feature selection for real-world IDS deployment. This perspective extends beyond prior work, which often reports performance gains without explicitly examining the trade-offs between efficiency and detection reliability.

Table 5
F1-score comparison of the RF model before and after SMOTE

Class	F1-score (before)	F1-score (after)	Class	F1-score (before)	F1-score (after)
Benign	1.00	1.00	Heartbleed	1.00	1.00
Bot	0.83	0.79	Infiltration	0.60	0.73
DDoS	1.00	1.00	PortScan	1.00	1.00
DoS GoldenEye	1.00	1.00	SSH-Patator	1.00	1.00
DoS Hulk	1.00	1.00	Brute Force	0.77	0.77
DoS Slowhttptest	0.99	0.99	SQL Injection	0.40	0.67
DoS Slowloris	1.00	1.00	XSS	0.33	0.37
FTP-Patator	1.00	1.00			

Table 6
Comparison of existing studies with the proposed study implemented on CICIDS-2017

Study	Year	Method(s)	Accuracy
[12]	2022	Backpropagation–DNN	98.3%
[13]	2023	SVM / LR / GNB	71% / 90% / 83%
[14]	2024	DNN	99.7%
Proposed	2025	RF	**99.8%**

4. Conclusion

The growing sophistication of network attacks demands IDS that are not only accurate but also computationally efficient for real-time deployment. In this study, CFS is applied to reduce computational cost during feature selection, enabling faster training and testing without sacrificing detection accuracy. The results showed that ensemble-based models, particularly RF and XGBoost, significantly outperformed MLP, achieving accuracies of 99.8% with very low false positive rates. RF provided the most balanced performance across precision, recall, and F1-score, while XGBoost delivered the lowest FPR and fastest testing speed, although with longer training times. The proposed study highlights that combining ensemble learning with correlation-driven feature selection offers a strong balance between accuracy and efficiency.

However, the evaluation is limited to the CICIDS-2017 dataset, which may constrain generalizability to other network environments. To enhance robustness and practical adoption, future work will validate the framework across diverse datasets, implement K-fold cross-validation to improve generalization, conduct multiple runs with different random seeds to assess model stability, and include error bars/confidence intervals to better reflect the variability of model performance. Additionally, the exploration of deep or hybrid models will be considered for broader applicability in real-world intrusion detection systems.

Funding Support

This work is sponsored by the National Natural Science Foundation of China (62572406), Sichuan Science and Technology Program, and the Doctoral Fund Project of Southwest University of Science and Technology (23ZX7136).

Conflicts of Interest

Xuyang Shi is a specialist for *Journal of Data Science and Intelligent Systems* and was not involved in the editorial review or the decision to publish this article. The authors declare that they have no conflicts of interest to this work.

Data Availability Statement

The data that support the findings of this study are openly available in the Canadian Institute for Cybersecurity (CIC), University of New Brunswick (UNB), Canada, at http://cicresearch.ca/CICDataset/CIC-IDS-2017/.

Author Contribution Statement

Bilal Rafique: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data curation, Writing – original draft, Writing – review & editing, Visualization, Project administration. **Sania Kanwal:** Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data curation, Writing – original draft, Writing – review & editing, Visualization, Project administration. **Xuyang Shi:** Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data curation, Writing – original draft, Writing – review & editing, Visualization, Supervision, Project administration, Funding acquisition.

References

[1] Samantaray, M., Barik, R. C., & Biswal, A. K. (2024). A comparative assessment of machine learning algorithms in the IoT-based network intrusion detection systems. *Decision Analytics Journal*, *11*, 100478. https://doi.org/10.1016/j.dajour.2024.100478

[2] Devendiran, R., & Turukmane, A. V. (2024). Dugat-LSTM: Deep learning based network intrusion detection system using chaotic optimization strategy. *Expert Systems with Applications*, *245*, 123027. https://doi.org/10.1016/j.eswa.2023.123027

[3] Elsayed, S., Mohamed, K., & Madkour, M. A. (2024). A comparative study of using deep learning algorithms in network intrusion detection. *IEEE Access*, *12*, 58851–58870. https://doi.org/10.1109/ACCESS.2024.3389096

[4] Lika, R. A., Murugiah, D., Brohi, S. N., & Ramasamy, D. (2018). NotPetya: Cyber attack prevention through awareness via gamification. In *2018 International Conference on Smart Computing and Electronic Enterprise*, 1–6. https://doi.org/10.1109/ICSCEE.2018.8538431

[5] Fyshchuk, I., & Pintsch, A. (2025). Cyber-attacks in Ukraine: Coping with the challenges at the local level in 2022–2024. *Risk, Hazards & Crisis in Public Policy*, *16*(3), e70025. https://doi.org/10.1002/rhc3.70025

[6] Dovbysh, A., Liubchak, V., Shelehov, I., Simonovskiy, J., & Tenytska, A. (2022). Information-extreme machine learning of a cyber attack detection system. *Radioelectronic and Computer Systems*, (3), 121–131. https://doi.org/10.32620/reks.2022.3.09

[7] Ahmad, Z., Khan, A. S., Shiang, C. W., Abdullah, J., & Ahmad, F. (2021). Network intrusion detection system: A systematic study of machine learning and deep learning approaches. *Transactions on Emerging Telecommunications Technologies*, *32*(1), e4150. https://doi.org/10.1002/ett.4150

[8] Samriya, J. K., Kumar, S., Kumar, M., Wu, H., & Singh Gill, S. (2024). Machine learning based network intrusion detection optimization for cloud computing environments. *IEEE Transactions on Consumer Electronics*, *70*(4), 7449–7460. https://doi.org/10.1109/TCE.2024.3458810

[9] Vibhute, A. D., Patil, C. H., Mane, A. V., & Kale, K. V. (2024). Towards detection of network anomalies using machine learning algorithms on the NSL-KDD benchmark datasets. *Procedia Computer Science*, *233*, 960–969. https://doi.org/10.1016/j.procs.2024.03.285

[10] Ali, M. L., Thakur, K., Schmeelk, S., Debello, J., & Dragos, D. (2025). Deep learning vs. machine learning for intrusion detection in computer networks: A comparative study. *Applied Sciences*, *15*(4), 1903. https://doi.org/10.3390/app15041903

[11] Diana, L., Dini, P., & Paolini, D. (2025). Overview on intrusion detection systems for computers networking security. *Computers*, *14*(3), 87. https://doi.org/10.3390/computers14030087

[12] Afolabi, H. A., & Aburas, A. A. (2022). RTL-DL: A hybrid deep learning framework for DDoS attack detection in a big data environment. *International Journal of Computer Networks & Communications*, *14*(6), 51–66. https://doi.org/10.5121/ijcnc.2022.14604

[13] Al Lail, M., Garcia, A., & Olivo, S. (2023). Machine learning for network intrusion detection: A comparative study. *Future Internet*, *15*(7), 243. https://doi.org/10.3390/fi15070243

[14] Osa, E., Orukpe, P. E., & Iruansi, U. (2024). Design and implementation of a deep neural network approach for intrusion detection systems. *e-Prime-Advances in Electrical Engineering, Electronics and Energy*, *7*, 100434. https://doi.org/10.1016/j.prime.2024.100434

[15] Kanwal, S., Amin, W., Khan, A. A., Rafique, B., Huang, Q., Jian, L., & Batool, I. (2025). Enhanced cybersecurity for smart grids: Detecting protocol-specific DDoS attacks on Modbus networks. *Computers and Electrical Engineering*, *127*, 110629. https://doi.org/10.1016/j.compeleceng.2025.110629

[16] Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. In *Proceedings of the 4th International Conference on Information Systems Security and Privacy - Volume 1*, 108–116. https://doi.org/10.5220/0006639801080116

[17] Panigrahi, R., & Borah, S. (2018). A detailed analysis of CICIDS2017 dataset for designing intrusion detection systems. *International Journal of Engineering & Technology*, *7*, 479–482.

[18] Widodo, A. O., Setiawan, B., & Indraswari, R. (2024). Machine learning-based intrusion detection on multi-class imbalanced dataset using SMOTE. *Procedia Computer Science*, *234*, 578–583. https://doi.org/10.1016/j.procs.2024.03.042

[19] Ikhwan, S., Purwanto, P., & Rochim, A. F. (2025). Optimizing intrusion detection in IoT through a combination of feature selection and deep feedforward neural network. *International Journal of Intelligent Engineering & Systems*, *18*(1), 624–637. https://doi.org/10.22266/ijies2025.0229.44

[20] Almelibari, A. A. (2025). Intrusion detection system traffic classification based on machine learning with correlation-based filtering and a genetic algorithm-inspired feature selection method for IoT networks. *Engineering, Technology & Applied Science Research*, *15*(5), 27430–27435. https://doi.org/10.48084/etasr.13511

[21] Anand, V., Senthil Kumar, G., Suresh Kumar, K., & Selvaganesan, C. (2025). Enhancing ransomware detection: A comparative review of XGBoost, random forest, and neural network approaches. In *2025 International Conference on Emerging Systems and Intelligent Computing*, 710–715. https://doi.org/10.1109/ESIC64052.2025.10962609

[22] Yang, B., Jahed Armaghani, D., Fattahi, H., Afrazi, M., Koopialipoor, M., Asteris, P. G., & Khandelwal, M. (2025). Optimized random forest models for rock mass classification in tunnel construction. *Geosciences*, *15*(2), 47. https://doi.org/10.3390/geosciences15020047

[23] Rozam, N. F., & Riasetiawan, M. (2023). XGBoost classifier for DDoS attack detection in software defined network using sFlow protocol. *International Journal on Advanced Science, Engineering and Information Technology*, *13*(2), 718–725. https://doi.org/10.18517/ijaseit.13.2.17810

[24] Ali, A., Assam, M., Khan, F. U., Ghadi, Y. Y., Nurdaulet, Z., Zhibek, A., ..., & Alahmadi, T. J. (2024). An optimized multilayer perceptron-based network intrusion detection using gray wolf optimization. *Computers and Electrical Engineering*, *120*, 109838. https://doi.org/10.1016/j.compeleceng.2024.109838

[25] Hamad, N. A., Jasim, O. N., & Yaser, Z. K. (2025). A hybrid feature learning model to enhance multilayer perceptron for network intrusion detection. *Journal of Education for Pure Science*, *15*(1), 47–55. https://doi.org/10.32792/jeps.v15i1.505

How to Cite: Rafique, B., Kanwal, S., & Shi, X. (2026). Correlation-Based Feature Selection for Efficient Intrusion Detection: Comparative Evaluation of Machine Learning Models on CICIDS-2017. *Journal of Data Science and Intelligent Systems*, *4*(2), 193–201. https://doi.org/10.47852/bonviewJDSIS62027552