FOUNDATIONS OF COMPUTING AND DECISION SCIENCES
Vol. 51 (2026) No. 3
ISSN 0867-6356
sciendo DOI: 10.2478/fcds-2026-0014 e-ISSN 2300-3405

Intelligent Multi-Stage Feature Selection for Real-Time Intrusion Detection in IoV Networks: A Machine Learning Framework

Emediong Obot[1], Bliss Utibe-Abasi Stephen[2], Okorafor Nwaji Godfrey[3], Sadiq Thomas[4], Akpasam Joseph Ekanem[5], Philip Asuquo[6]

Abstract: The Internet of Vehicles (IoV) faces critical cybersecurity challenges that require real-time intrusion detection on resource-constrained edge devices. This study proposes a Three-stage Variance-correlation Binary Grey Wolf Optimization (TVC-BGWO) feature selection framework integrated with ensemble learning. The framework combines Variance Threshold filtering, Correlation-based filtering, and Binary Grey Wolf Optimization to achieve 93.59% feature reduction (78→5 features) while maintaining competitive detection performance. Random Forest with TVC-BGWO achieved 99.91% binary accuracy and 99.87% multiclass accuracy 6 attack types plus benign traffic on Canadian Institute for Cybersecurity Intrusion Detection System (CIC-IDS2017). Statistical validation (McNemar's $\chi^2 = 82.01$, $p < 0.001$) confirms a significant improvement. The framework provides 59.0% training time reduction and sub-millisecond inference latency meeting V2V communication requirements. Adversarial evaluation reveals exceptional resilience to data poisoning 99.98% robustness and moderate vulnerability to test-time perturbations. This work contributes to intelligent transportation security by demonstrating competitive intrusion detection performance while achieving substantial computational efficiency gains suitable for edge deployment.

Keywords: Internet of Vehicles, intrusion detection, feature selection, Binary Grey Wolf Optimization, ensemble learning, cybersecurity, edge computing.

[1,2,6] Department of Computer Engineering, University of Uyo, Nigeria, {emediongbassey080@gmail.com, blissstephen@uniuyo.edu.ng, philipasuquo@uniuyo.edu.ng}
[2,6] TETFund Centre of Excellence in Computational Intelligence Research, University of Uyo, Uyo, Nigeria
[3] Federal University of Technology Ikot-Abasi, Akwa Ibom State, Nigeria, godfreyokoro@futia.edu.ng
[4] Department of Computer Engineering, Nile University, Nigeria, sadiqthomas@nileuniversity.edu.ng
[5]Department of Electrical and Electronic Engineering, Akwa Ibom State University, Nigeria, ak-pasamekanem@aksu.edu.ng

1. Introduction

The Internet of Vehicles (IoV) represents a transformative advance in intelligent transportation systems. Modern vehicles are equipped with numerous sensors, embedded software, and advanced communication technologies that enable seamless interactions in multiple domains [31]. These include Vehicle-to-Vehicle (V2V) communication, Vehicle-to-Infrastructure (V2I) communication, Vehicle-to-Pedestrian (V2P) communication, and Vehicle-to-Network (V2N) communication [4]. This interconnected ecosystem enables real-time data exchange and decision making, its applications include adaptive traffic management, collision avoidance, dynamic route optimization, and autonomous driving [3, 8]. By processing and sharing information instantaneously, IoV systems improve road safety, operational efficiency, and provide enhanced user experiences through services such as infotainment and predictive maintenance [18]. IoV has evolved from traditional Vehicular Ad-hoc Networks (VANETs), which focused primarily on short-range communications [28], to integrated frameworks that leverage cloud computing for centralized analytics, edge computing for low-latency processing, and 5G/6G connectivity for ultra-reliable high-bandwidth communication [34, 11]. This evolution enables smarter AI-driven decision making for dynamic traffic scenarios [26]. However, such pervasive connectivity also exposes IoV networks to various cybersecurity threats, including spoofing, denial-of-service (DoS) attacks, distributed denial-of-service (DDoS) attacks, malware injection, and data tampering, which can compromise the integrity, availability, and confidentiality of vehicular communications [2]. Components such as in-vehicle Controller Area Network (CAN) bus systems, designed for reliable internal communication, are inherently vulnerable due to lack of encryption and authentication mechanisms [42]. Autonomous vehicles that rely on sensor fusion and AI-based control face risks from firmware exploitation, sensor manipulation, and unauthorized remote control, which can result in system failures and catastrophic accidents [35].

Machine learning (ML) has emerged as an effective solution for IoV intrusion detection, capable of identifying complex, evolving, and previously unseen attack patterns [5]. Ensemble algorithms like Random Forest (RF), Extreme Gradient Boosting (XGBoost), Adaptive Boosting (AdaBoost), and Gradient Boosting Machine (GBM) have excelled in handling high-dimensional and imbalanced datasets characteristic of network traffic [21]. These methods enables accurate detection of intrusions with high precision and sensitivity. The increasing inter-connectivity and digital dependence of modern vehicular systems have significantly expanded the IoV attack surface [21]. ML-based intrusion detection systems (IDSs) demonstrate strong predictive performance. However, challenges such as feature redundancy, high dimensionality, and suboptimal feature subset selection can limit their real-time applicability in edge-limited vehicular edge devices [36].

Recent IoV cybersecurity research commonly evaluates intrusion detection models on widely recognized benchmark datasets such as CIC-IDS2017 and NSL-KDD, which contain various attack patterns directly applicable to vehicular network security [19, 39]. CIC-IDS2017, contains contemporary network traffic captures including web-based attacks (SQL injection, Cross-Site Scripting), DDoS attacks, infiltration attempts, and benign traffic collected over

five days in a realistic network topology [37]. These attack patterns are directly relevant to IoV security: DDoS attacks simulate resource exhaustion on vehicular communication channels and infrastructure servers [38]; web-based attacks represent threats to cloud connected vehicle services, remote diagnostic platforms, and in-vehicle infotainment systems; infiltration patterns model advanced persistent threats (APTs) targeting vehicle fleet management systems and V2I infrastructure [17, 23].

Despite advances in ML-based intrusion detection for IoV, existing studies focus solely on classification accuracy without adequately addressing computational efficiency and multiclass classification accuracy, robustness against adversarial attacks, including data poisoning during model training and evasion attacks during deployment, remains underexplored in the IoV intrusion detection literature [32, 10, 44, 40]. Consequently, there is a pressing need for an efficient, multi-stage feature selection framework. This study proposes a Three-stage Variance-correlation Binary Grey Wolf Optimization (TVC-BGWO) feature selection framework for IoV intrusion detection, integrating filter-based and wrapper-based strategies with aggregated classifiers. The proposed three-stage consists of: Variance Threshold filtering to eliminate low information features exhibiting near constant behavior across samples [22], Correlation-based filtering to remove highly correlated features and mitigate multicollinearity [20], and Binary Grey Wolf Optimization (BGWO) for optimal feature subset through metaheuristic search [24]. This hierarchical approach progressively reduces dimensionality of the features while preserving discriminative power, ensuring efficient computation, high detection accuracy, and reliable performance across diverse attack categories. The TVC-BGWO framework is evaluated on two benchmark datasets CIC-IDS2017 (primary dataset) and NSL-KDD (secondary dataset) using four ensemble learning algorithms: RF, XGBoost, AdaBoost, and GBM. Evaluation encompasses binary classification and multiclass classification, with comprehensive assessment of classification accuracy, computational efficiency, and adversarial robustness. Statistical validation through McNemar's test and Friedman test with post-hoc pairwise comparisons ensures rigorous performance assessment and establishes the significance of observed improvements.

This study makes the following novel contributions to IoV intrusion detection research:

- Development of a Novel Multi-Stage Feature Selection Framework: The study proposes a novel TVC-BGWO feature selection framework that systematically integrates variance-based filtering, correlation analysis, and metaheuristic optimization. Unlike existing single-stage approaches, the proposed framework progressively eliminates irrelevant and redundant features while preserving discriminative information, advancing feature selection methodology for intrusion detection systems.
- Adversarial Robustness Assessment: Adversarial robustness evaluation of IoV intrusion detection systems under three realistic attack scenarios; data poisoning, Gaussian noise injection, and targeted feature perturbation. These findings establish empirical evidence that ensemble voting mechanisms effectively neutralize corrupted training samples but require additional hardening techniques for deployment in adversarial environments, addressing a critical gap in the IoV security literature.
- Unified Binary and Multiclass Intrusion Detection Evaluation: This study advances the

dual-task intrusion detection capability by demonstrating that extreme dimensionality reduction does not compromise fine-grained attack discrimination. The framework maintains superior performance across both binary classification and multiclass classification using the selected features. Ablation analysis quantifies individual stage contributions.

- Latency-Aware Feature Optimization for IoV Environments: The study demonstrates practical feasibility for real-time deployment on resource-constrained vehicular edge devices through rigorous computational efficiency analysis. The framework achieves substantial training time reduction with inference latency suitable for real-time V2V communication and satisfies stringent V2V communication requirements across all evaluated models.

The remainder of this paper is organized as follows: Section 2 presents related work in IoV intrusion detection and feature selection methodologies; Section 3 describes the proposed TVC-BGWO framework; Section 4 presents comprehensive experimental results; Section 5 concludes the article with key findings and future research directions.

2. Related Work

This section summarizes recent research contributions to intrusion detection within IoV and vehicular cybersecurity systems.

Tanish et al. [32] proposed a framework for detecting intrusion in IoV using ML with integration of Local Interpretable Model-agnostic Explanations (LIME) for model transparency on real time vehicular sensor data. The following pre-processing pipeline was carried out median imputation, one-hot encoding, min–max scaling, Synthetic Minority Over-sampling Technique (SMOTE), and Principal Component Analysis (PCA); they evaluated Random Forest(RF), Extreme Gradient Boosting (XGBoost), Support Vector Classifier (SVC), Decision Tree (DT), and Logistic Regression(LR). Random Forest achieved the highest performance in both binary and multi-class tasks. LIME identified key features that influence the predictions.

Aloqaily et al. [10] developed an Intrusion Detection System (IDS) for Connected and Autonomous Vehicles (CAVs) using the High-Coverage Real-Life Controller Area Network (HCRL CAN-bus) dataset with four types of attacks. Seven ML models were assessed, with RF and Light Gradient Boosting Machine (LightGBM) achieving top performance with RF offering the fastest inference time.

Zhang et al.[44] introduced a lightweight IDS that integrates a Global Redundant Information Pruning and Component Analysis (GRIPCA) technique for dimensionality reduction together with a Dynamic Particle Swarm Optimization (DPSO) and optimized Online Wavelet Extreme Learning Machine (OWELM) for intrusion classification. The proposed system was evaluated using the NSL-KDD dataset and the Canadian Institute for Cybersecurity Intrusion Detection System 2017 (CIC-IDS2017) dataset, achieving high detection accuracy and exceptionally low detection latency.

Ullah et al. [40] proposed Hybrid Deep learning architecture for Intrusion Detection System (HDL-IDS), the framework incorporates an Extremely Randomized Trees (ETC)-based feature selection and Synthetic Minority Over-sampling Technique with K-Nearest Neighbors (SMOTE-KNN) for class imbalance handling. The model achieved an accuracy ranging from 99.85% to 99.99% on DDoS and CAN-bus datasets, while reducing detection latency by nearly 50%.

Jain et al. [25] conducted a comprehensive comparative evaluation of supervised, unsupervised, and deep learning approaches in multiple intrusion detection datasets. The results showed that Long Short-Term Memory (LSTM) networks achieved the highest accuracy of 94.1%, demonstrating an improved ability to capture temporal attack patterns.

Lundberg et al. [30] applied SHapley Additive exPlanations (SHAP) together with a visualization framework (VisExp) to interpret deep neural network predictions on controller area network bus traffic (CAN-bus), while the approach improved expert trust and model interpretability, the evaluation was limited to binary classification scenarios.

Yang et al. [43] introduced a multitiered hybrid intrusion detection system (MTH-IDS), which integrates multi-stage feature engineering using Information Gain, Fast Correlation-Based Filter, and Kernel Principal Component Analysis, combined with Bayesian optimization, stacking ensemble classifiers, and Cluster Labeling K-Means (CL-K-Means) for zero-day attack detection. The framework achieved near-perfect detection accuracy with real-time inference latency between 0.5 and 0.6 ms, even on low-power devices.

Almansour et al. [7] introduced an Adaptive Personalized Federated Learning (APFed) with a Lightweight Depthwise Convolutional Bottleneck Network (LDwCBN) to enable privacy preserving and resource efficient intrusion detection on vehicular devices. The proposed method was evaluated using multiple benchmark datasets, including CIC-IDS2017, CSE-CIC-IDS2018, Car-Hacking, and CAN-Train-Test, in simulated non-independent and Identically Distributed (non-IID) federated environments. Adaptive Personalized Federated Learning (APFed) employed dynamic weight fusion and personalized local updates, allowing each vehicle to adapt the global model to its local data distribution. The Lightweight Dynamic Weighted Collaborative Bayesian Network (LDwCBN) deep learning classifier serves as the core detection model, optimized for low computational overhead while maintaining high detection accuracy.

Khan et al. [27] proposed a deep learning framework based on Crayfish Mother Swarm Optimization. The model integrates advanced encryption standard with a 256-bit key (AES-256) and homomorphic encryption with advanced feature extraction techniques, including Vision Transformers, Wavelet Transform, and Graph Attention Networks. Using the Directed Acyclic Graph–based Security Network classifier, the framework achieved up to 99.1 % accuracy in CAN-bus, CIC-IDS2017, and CICIoV2024 datasets, while significantly reducing false positive and false negative rates.

Alohali et al. [9] introduced a two-stage malware detection framework for IoV environments that integrates deep learning, metaheuristic optimization, and explainable artificial intelligence (XAI). The method employs min–max normalization for preprocessing, Reptile Search Algorithm (RSA) for feature selection, a hybrid Bidirectional Long Short-Term Memory with Multi-Head Self-Attention (BiLSTM-MHSA) model for malware classification,

and the Pelican Optimization Algorithm (POA) for hyperparameter tuning, while SHAP is used to enhance model interpretability. The model was evaluated using a publicly available malware dataset from Kaggle, consisting of 50,000 samples.

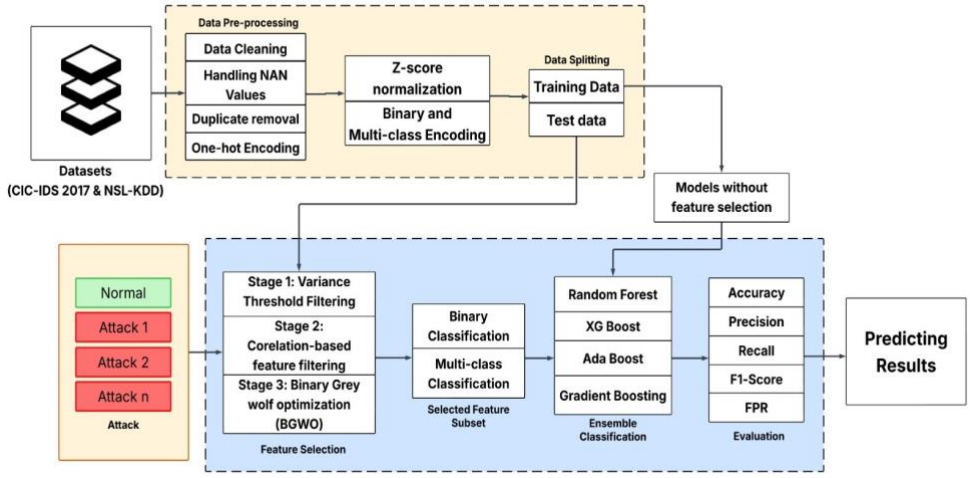


Figure 1. Proposed TVC-BGWO System Architecture

Aziz et al. [12] introduced an explainable neural network for anomaly detection in Internet of Vehicular networks, employing K-Means clustering for feature ranking and an Additive Index Model-based classifier for sequential traffic analysis. The framework achieved accuracies of 99.7% in UNSW-NB15 and 99.3% in CICIDS2019, outperforming the baselines of convolutional, recurrent, and deep neural networks.

The IoV faces critical cybersecurity challenges that existing intrusion detection approaches have inadequately addressed; these limitations include; high feature dimensionality and redundancy degrade performance and increase computational overhead; suboptimal feature selection hinders real-time applicability on resource-constrained edge devices. The study addresses this gap through the TVC-BGWO framework, integrating statistical filtering, correlation analysis, and bio-inspired metaheuristic optimization with ensemble learning, validated across binary and multiclass classification tasks on CIC-IDS2017 and NSL-KDD datasets.

3. Methodology

3.1. Model Design

Figure 1 presents the proposed system architecture which integrates TVC-BGWO feature selection with ensemble machine learning classifiers. The architecture comprises: data

preprocessing, three-stage feature selection combining variance filtering, correlation-based filtering, and BGWO for dimensionality reduction; with RF, XGBoost, AdaBoost, and GBM classifier for binary and multiclass attack detection.

3.2. Dataset Collection

This study adopts a dual-dataset evaluation strategy, using CIC-IDS2017 as the primary dataset for contemporary attack patterns and NSL-KDD as a complementary benchmark validation. Both datasets were obtained from publicly available Kaggle repositories.

1. CIC-IDS2017 Four CSV files derived from July 2017 network traffic captures were integrated, including Thursday-Morning-WebAttacks (Cross-Site Scripting (XSS), Structured Query Language (SQL injection) and brute-force attacks), Thursday-Afternoon-Infiltration (advanced persistent threat patterns), Friday-Afternoon-DDoS (volumetric attacks), and Friday-Morning (benign traffic). Network flows were generated using CICFlowMe-ter, which extracted 79 bidirectional flow-based features that include flow duration, packet statistics, flag information, and protocol-level characteristics. The merged dataset yielded a total of $N = 875,746$ samples.
2. NSL-KDD dataset includes KDDTrain.csv (125,974 records) and KDDTest.csv (22,545 records), each described by 41 features covering four attack classes: Denial of Service (DoS), Remote-to-Local (R2L), User-to-Root (U2R), and probing attacks [13].

3.3. Data Preprocessing

3.3.1. CIC-IDS2017

The data were preprocessed through the following steps; column name standardization, removal of records containing missing values, replacement of infinite numerical values with column-wise medians, and duplicate elimination. The preprocessed dataset shows a class distribution of 83.47% benign traffic and 16.53% attack traffic, retaining 99.52% of the original 875,746 samples after removing invalid and duplicate records. Two labeling methods were applied to generate; binary labels (BENIGN → 0, attack → 1) and multiclass labels using scikit-learn's LabelEncoder. The dataset was divided into training sets (700,597 samples) and testing sets (162,134 samples) using an 80:20 stratified split with a fixed random seed (random_state=42) to ensure reproducibility. All numerical features were standardized using Z-score normalization,

$$z = \frac{\{x - \mu\}}{\{\sigma\}} \tag{1}$$

Where x denotes the original feature value, and μ and σ represent the mean and standard deviation calculated exclusively from the training set to prevent data leakage.

3.3.2. NSL-KDD preprocessing

Initial preprocessing removed the sequential identifier column and the duplicated target column ('difficulty' and 'num_outbound_cmds'), resulting in 40 effective features for modeling. A ColumnTransformer processed categorical and numerical features separately

$$\mathrm{T} = [\text{One Hot Encoding (OHE)}(X_{\text{cat}}), \text{Scale}(X_{\text{num}})], \tag{2}$$

Where categorical attributes (protocol_type, service and flag) were encoded in one-hot way and the numerical features were normalized to the Z-score. This transformation expanded the feature space from 40 to approximately 120 dimensions. Binary and multiclass labels were encoded using scikit-learn's LabelEncoder fitted on the training set, with the same encoding applied to the test set to maintain consistency. All preprocessing artifacts, including scalers, encoders, and transformed datasets, were serialized using NumPy binary formats and joblib to ensure reproducibility and deployment consistency. This preprocessing establishes a reliable foundation for subsequent feature selection and model training.

3.4. Feature Selection (FS)

To reduce feature redundancy while preserving discriminative power, the study used a three-stage hybrid feature selection framework. The **Variance Threshold** was applied in the first stage with a variance threshold filter of ($\tau = 0.01$) to eliminate low-variance features that exhibit near-constant behavior across samples, thereby reducing noise and computational overhead, where τ is the variance threshold parameter; In the second stage, **Pairwise Pearson correlation analysis** was applied to variance-filtered features. Highly correlated features ($|\rho| > 0.95$) were iteratively removed to mitigate multicollinearity and improve model generalization. Where ρ is the Pearson correlation coefficient. Stage 3 employed **Binary Grey Wolf Optimization** with $N_{pop} = 10$ wolves over $T_{max} = 30$ iterations, optimizing the multi-objective fitness function:

$$fitness(S) = 0.99 \times ACC(S) - 0.01 \times \frac{|S|}{|F|} \tag{3}$$

where S represents the selected feature subset, $ACC(S)$ denotes classification accuracy achieved by Random Forest (20 estimators, depth=10) trained on S, $|S|$ is the cardinality of selected features, and $|F|$ is the total number of available features. The fitness weights (0.99 for accuracy, 0.01 for parsimony) prioritize classification performance while encouraging smaller feature subsets. Continuous wolf positions were converted to binary feature selections via sigmoid transfer function:

$$\sigma(x) = \frac{1}{1 + e^{-10(x-0.5)}} \tag{4}$$

where $x \in [0, 1]$ denotes the continuous position value and $\sigma(x) \in [0, 1]$ yields the feature selection probability. A threshold of 0.5 is used to determine binary selection (a feature is selected if $\sigma(x) \geq 0.5$), while the coefficient -10 ensures a steep transition around the threshold, enabling decisive binary selection behavior.

As shown in Figure 2, the top 3 features—Flow Bytes/s, Flow Packets/s, and Total Fwd Packets—account for 80% of cumulative importance in both binary and multiclass tasks, demonstrating highly efficient feature selection. Figure 2: Feature importance analysis of BGWO-selected features on CIC-IDS2017 dataset.

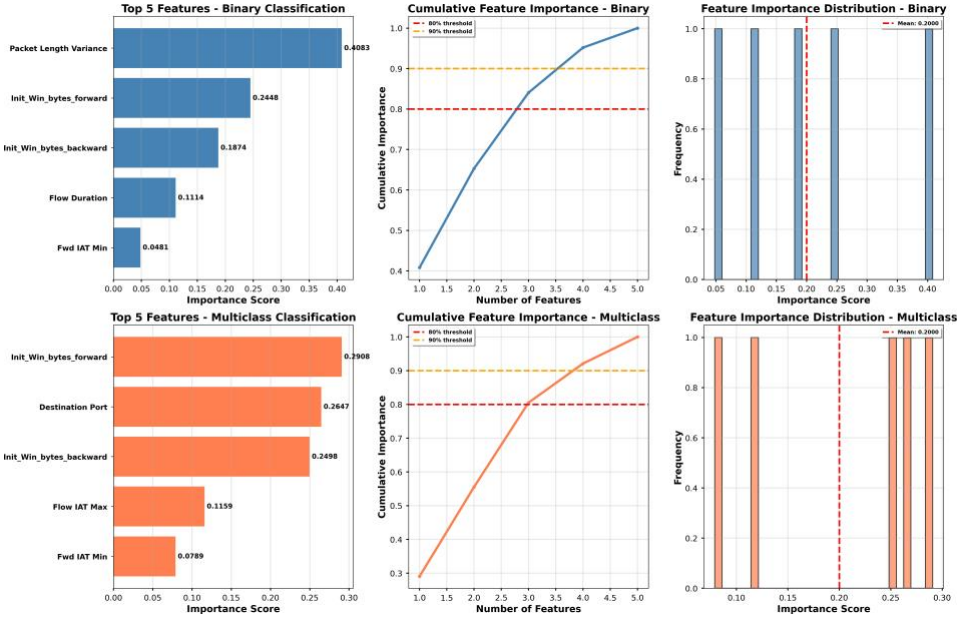


Figure 2. Feature importance analysis of BGWO-selected on CIC-IDS2017 Dataset

3.5. Classification Models

Four tree-based ensemble algorithms were implemented to evaluate intrusion detection performance: RF, XGBoost, AdaBoost, and GBM. These methods offer complementary strengths for handling high-dimensional and class imbalance dataset.

3.5.1. Random Forest (RF)

The algorithm leverages bootstrap aggregation to create multiple decision trees from random subsets of training data, with final predictions determined through majority voting [33]. The RF model was configured with n_estimators = 100, max_depth = 10, min_samples_split = 2, and min_samples_leaf = 1.

3.5.2. Extreme Gradient Boosting (XGBoost)

XGBoost is an optimized gradient boosting framework that minimizes a regularized objective function using second order gradient information, thereby achieving high predictive accuracy and computational efficiency [16]. XGBoost was implemented with n_estimators=100, max_depth=3, and learning_rate=0.1. To prevent overfitting, we employed subsample=0.8 and colsample_bytree=0.8. The model sequentially builds trees that correct errors from previous iterations [15].

3.5.3. Adaptive Boosting (AdaBoost)

AdaBoost is an ensemble learning algorithm that iteratively improves classification performance by adaptively reweighting misclassified instances, thereby focusing subsequent learners on difficult samples [41]. The AdaBoost implementation utilized n_estimators=100 with learning_rate=1.0. For binary classification, we employed the SAMME.R algorithm, while the SAMME algorithm was used for multiclass instances.

3.5.4. Gradient Boosting Machine (GBM)

GBM is an ensemble learning technique that iteratively builds predictive models by minimizing a differentiable loss function using gradient descent in function space [29]. GBM was configured with n_estimators=50, learning_rate=0.1, and max_depth=3.

3.6. Performance Evaluation Metrics

Given the pronounced class imbalance in the CIC-IDS2017 dataset, accuracy alone may be a

misleading indicator of model performance. Consequently, this study adopts a complementary set of metrics, each selected to address a specific limitation of accuracy in the context of imbalanced, security-critical classification.

3.6.1. Classification Accuracy Metrics

1. Accuracy was included to measure the overall proportion of correctly classified instances. However, due to the imbalanced class distribution, accuracy was interpreted alongside other evaluation metrics rather than as the primary indicator of performance. Overall correctness rate computed as the ratio of correct predictions to total predictions. Accuracy is given as:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (5)$$

   Where TP (True Positives), TN (True Negatives), FP (False Positives) and FN (False Negatives) represent confusion matrix.

2. Precision was selected to evaluate the reliability of attack predictions. In practical cybersecurity environments, a high precision indicates that most traffic identified as malicious is genuinely malicious, thereby reducing false alarms, minimizing alert fatigue among security analysts, and lowering the operational cost associated with investigating benign traffic incorrectly classified as attacks. Precision is defined as:

$$Precision = \frac{TP}{TP+FP} \quad (6)$$

   Where TP denotes True positives and FP denotes False positives.

3. Recall was considered one of the most important metrics because it measures the proportion of actual attacks that are successfully detected. In intrusion detection, failing to identify malicious traffic (false negatives) can expose computer networks to severe security breaches. Therefore, maximizing recall is essential for ensuring comprehensive threat detection. Recall is expressed as:

$$Recall = \frac{TP}{TP + FN} \quad (7)$$

Where TP denotes True positives and FN denotes False negatives

4. F1-Score is the harmonic mean of precision and recall providing balanced performance assessment, particularly valuable for imbalanced datasets. F1-Score is given as:

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} = \frac{2TP}{2TP + FP + FN} \quad (8)$$

A model with high recall but low precision generates excessive false alarms, whereas a model with high precision but low recall fails to detect many attacks. The harmonic mean provided by the F1-score offers a balanced evaluation, making it suitable for imbalanced datasets. Where TP denotes True positives, FP donates False negative and FN denotes False negatives.

5. False Positive Rate (FPR): Proportion of benign traffic incorrectly classified as malicious, critical for assessing operational feasibility. FPR was included to assess the operational feasibility of the proposed intrusion detection framework. Although detecting attacks is essential, an IDS that frequently misclassifies legitimate network traffic as malicious becomes impractical for deployment because excessive false alarms reduce analyst confidence and increase response workload. False Positive Rate is expressed as:

$$FPR = \frac{FP}{FP + TN} \quad (9)$$

For multiclass classification, Precision, Recall, and F1-score were computed using the weighted-average approach:

$$Metric_{weighted} = \sum_{i=1}^{c} w_i \times Metric_i, \quad w_i = \frac{n_i}{N} \quad (10)$$

Where n_i is the number of samples belonging to class i, N is the total number of samples, and C is the total number of classes. Weighted averaging was selected to reflect the model's aggregate operational performance across the true class distribution encountered in deployment. We note, however, that because $w_i = \frac{n_i}{N}$ weighted metrics are dominated by high-frequency classes and can mask degraded performance on rare attack types.

3.6.2. Temporal Efficiency Metrics

1. Training Time: Wall-clock time required for complete model training on the training set, measured in seconds. This metric indicates computational complexity and resource requirements for model development and retraining cycles in operational environments.

2. Testing Time: The total prediction time for the entire test set, measured in seconds; reflects batch processing capability relevant for offline analysis and periodic network traffic auditing scenarios.

3. Inference Time: Per-sample prediction latency computed as $t_{inference} = t_{testing}/N_{test}$ and expressed in milliseconds. This metric is essential for real-time intrusion detection systems where sub-second response times are mandatory for inline blocking mechanisms and rapid threat response.

3.7. Confusion Matrix Analysis:

Complete confusion matrices were computed for all experiments, providing granular insight into error distributions across classes. For binary classification, the 2×2 matrix explicitly reveals TP, TN, FP, and FN counts. For multiclass scenarios, the $C \times C$ matrix (where C is the number of attack classes) enables per-class performance analysis, identifying which attack types are most challenging to detect or distinguish.

4. Results and Discussion

4.1. Experimental Setup

The experimental analysis was conducted on a Windows-based personal computer with an Intel Core i5 processor (4.0 GHz), 8 GB RAM and 500 GB storage capacity. All predictive models were implemented using Python 3.11.13 within the Kaggle Notebooks cloud-based computational environment, which provides an integrated interface for coding, visualization, and analysis. The machine learning utilized scikit-learn 1.3.0 for model development and evaluation, XGBoost 1.7.0 for gradient boosting implementation, pandas 1.5.3 and NumPy 1.24.3 for data manipulation and numerical computation, and matplotlib 3.7.1 and seaborn 0.12.2 for visualization and exploratory data analysis. To ensure reproducibility, random seed values (random_state = 42) were fixed in all stochastic processes, and all software dependencies were version controlled and documented.

4.2. Performance on CIC-IDS2017 Dataset

Table 1 presents binary classification results without feature selection in the CIC-IDS2017 dataset. All ensemble models achieve strong predictive performance $\geq$ 99.8% accuracy. XGBoost demonstrates the best baseline accuracy at 99.89 %, followed by RF at 99.82 %. However, computational costs reveal significant differences: AdaBoost requires 410.26 seconds for training with inference time of 0.029 ms per sample, while GBM requires 192.90 seconds for training with 0.002 ms inference time.

From Table 2 Random Forest achieves the highest accuracy of 99.91%, representing a 0.087 percentage point improvement over its baseline performance. More importantly, TVC-BGWO achieves substantial computational efficiency gains for Random Forest: training time decreases by 59.0% (70.18s → 28.74s), testing time reduces by 45.9% (0.99s → 0.53s), and inference latency improves by 45.9% (6.09ms → 3.30ms). XGBoost and GBM achieve training time reductions of 76.9% and 86.5%, respectively, while maintaining sub-millisecond inference

times (0.458 ms and 0.861 µs). XGBoost experiences a slight accuracy reduction from 99.89 % to 99.68 % representing an acceptable trade-off considering the 76.9% reduction in training time. AdaBoost demonstrates an 86.9 % reduction in training time (410.26s → 53.73s) and 66.9 % improvement in inference time (0.029ms → 0.010ms).

Table 1. Binary Classification Performance Without Feature Selection on CIC-IDS2017 Dataset

Model	Accuracy	Precision	Recall	F1-Score	FPR	Training Time (s)	Testing Time (s)	Inference Time (ms)
Random Forest	0.998187	0.998187	0.998187	0.998184	0.000273	70.181855	0.986613	0.006085
XGBoost	0.998865	0.998865	0.998865	0.998865	0.000427	8.596896	0.166878	0.001029
AdaBoost	0.998625	0.998624	0.998625	0.998624	0.000523	410.259037	4.706113	0.029026
Gradient Boosting	0.997231	0.997232	0.997231	0.997223	0.000405	403.1499	0.338494	0.00208

Table 2. Binary Classification Performance With TVC-BGWO Feature Selection on CIC-IDS2017 Dataset

Model	Accuracy	Precision	Recall	F1-Score	FPR	Training Time (s)	Testing Time (s)	Inference Time (ms)
Random Forest	0.999056	0.999056	0.999056	0.999056	0.000251	28.736262	0.534240	0.003295
XGBoost	0.996830	0.996831	0.996830	0.996821	0.000323	1.981825	0.074226	0.000458
AdaBoost	0.993357	0.993370	0.993357	0.993363	0.004377	53.734892	1.557168	0.009604
Gradient Boosting	0.995652	0.995662	0.995652	0.995632	0.000383	54.6083	0.139607	0.000861

Figure 3 presents confusion matrices for all four classifiers with TVC-BGWO feature selection.

4.2.1. Multiclass Classification Results

Table 3. Multiclass Classification Performance Without Feature Selection on CIC-IDS2017 Dataset

Model	Accuracy	Precision	Recall	F1-Score	Train Time (s)	Test Time (s)	Inference Time (ms)
Random Forest	0.997514	0.997539	0.997514	0.997096	71.683790	1.052288	0.006490
XGBoost	0.998557	0.998494	0.998557	0.998315	40.934612	0.545707	0.003366
AdaBoost	0.984211	0.979600	0.984211	0.981788	437.147155	5.186139	0.031987
Gradient Boosting	0.986659	0.997211	0.986659	0.991473	3023.95	2.336906	0.0014413

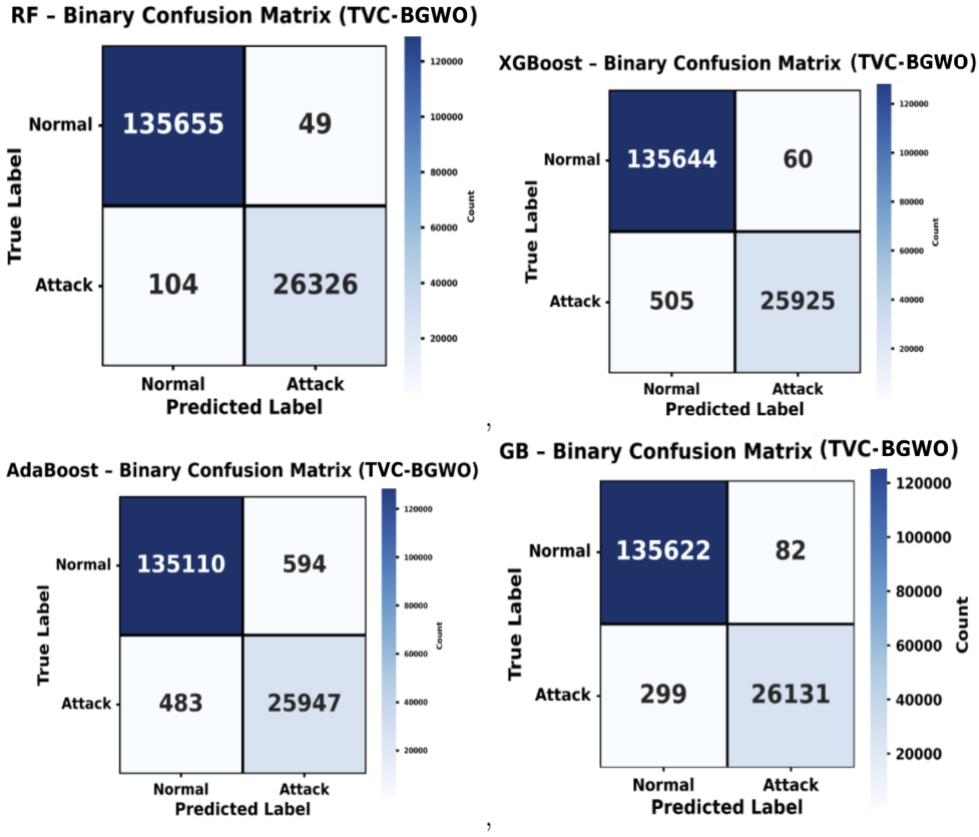


Figure 3. Confusion Matrices for Binary Classification on CIC-IDS2017 Dataset with TVC-BGWO Feature Selection

Table 4. Multiclass Classification Performance With TVC-BGWO Feature Selection on CIC-IDS2017 Dataset

Model	Acc	Pre	Recall	F1-Score	Train Time (s)	Test Time (s)	Inference Time (ms)
Random Forest	0.998748	0.998963	0.998748	0.998440	25.629386	0.710030	0.004379
XGBoost	0.998057	0.997758	0.998057	0.997746	15.649913	0.402112	0.002480
AdaBoost	0.964474	0.964103	0.964474	0.962984	44.898307	1.670902	0.010306
Gradient Boosting	0.996898	0.99785	0.996898	0.997247	247.1732	0.904867	0.005581

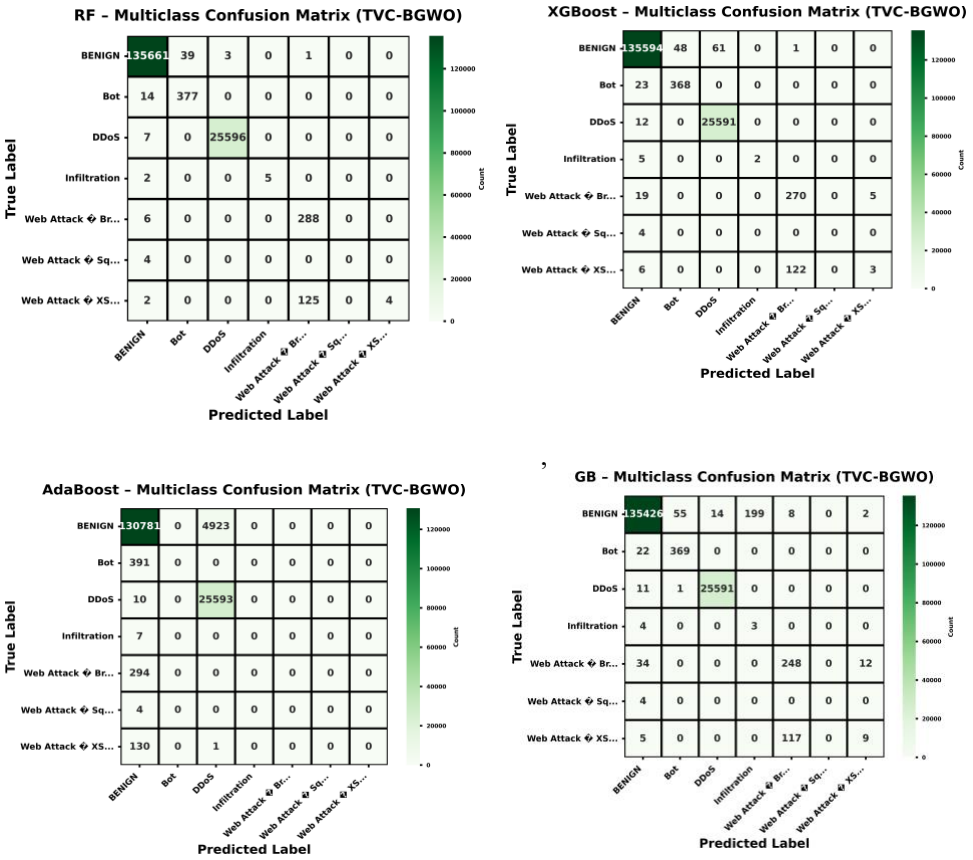


Figure 4. Confusion Matrices for Multiclass Classification on CIC-IDS2017 Dataset With TVC-BGWO Feature Selection

Tables 3 and 4 present multiclass classification performance in the CIC-IDS2017 dataset to distinguish between 6 attack classes plus benign traffic. Without feature selection Table (3), RF and XGBoost achieve strong multiclass discrimination with an accuracy ≥ 99.75%, demonstrating their capability to handle fine grained attack type classification. After TVC-BGWO feature selection (4), all models exhibit marked reductions in training and inference latency while maintaining high detection performance. RF achieves the best overall multiclass performance with 0.9987% accuracy and 0.99844 % F1-score, representing a 0.12% accuracy improvement over baseline alongside a 64.3% reduction in training time (from 71.68s to 25.63s). The improved performance despite using only 5 of 78 features validates TVC-BGWO's ability to identify the most discriminative features for attack type classification. XGBoost maintains competitive accuracy at 0.998057 % with the lowest inference latency of 2.48 ms, representing a 26.3 % improvement over baseline inference time. GBM achieves an

impressive reduction in training time from seconds 3023.95 to 247.1732 seconds while maintaining 99.55 % accuracy, representing only a 0.06 % decrease from baseline.

However, AdaBoost experiences performance degradation in multiclass classification, with accuracy decreasing from 98.42 % to 96.45 % and F1-score dropping from 0.9818 % to 0.9630 %. Although an 89.7 % reduction in training time, the significant accuracy loss suggests that AdaBoost's sequential error-correction mechanism may be more sensitive to feature reduction in multiclass scenarios where class boundaries are more complex.

Figure 4 presents confusion matrices for all classifiers with TVC-BGWO feature selection, revealing per-class detection performance.

4.3. Performance on NSL-KDD Dataset

4.3.1. Binary Classification Results

Table 5. Binary Classification Performance With and Without TVC-BGWO Feature Selection on NSL-KDD Dataset

Model	Feature Selection	Accuracy	Precision	Recall	F1-Score	FPR	Inference Time (ms)
Random Forest	Without FS	0.7673	0.8328	0.7673	0.7638	0.3883	0.00651
XGBoost	Without FS	0.7763	0.8375	0.7763	0.7735	0.3729	0.00088
AdaBoost	Without FS	0.7697	0.8130	0.7697	0.7685	0.3507	0.03028
Gradient Boosting	Without FS	0.7946	0.8465	0.7946	0.7930	0.3401	0.00258
Random Forest	With FS	**0.7932**	**0.8450**	**0.7932**	**0.7916**	**0.3413**	**0.00449**
XGBoost	With FS	**0.8029**	**0.8514**	**0.8029**	**0.8016**	**0.3264**	**0.00063**
AdaBoost	With FS	**0.8016**	**0.8480**	**0.8016**	**0.8005**	**0.3241**	**0.01215**
Gradient Boosting	With FS	**0.8202**	**0.8609**	**0.8202**	**0.8197**	**0.2960**	**0.00150**

Table 5 shows that TVC-BGWO feature selection improves binary classification across all models, with GBM achieving the highest accuracy of 82.02 % and substantially reduced FPR from 34.0 % to 29.60 %. All classifiers show consistent improvements in accuracy (2.59-3.19 %), precision, and inference time efficiency when feature selection is applied. However, the results multiclass classification 6 reveal inconsistent behavior, with AdaBoost experiencing a notable performance accuracy drop from 74.87 % to 65.35 % after feature selection, while RF shows modest improvement and XGBoost remains relatively stable.

4.3.2. Multiclass Classification Results

4.4. Statistical Validation

Model performance was rigorously evaluated using multiple statistical tests to ensure robust validation. McNemar's test was employed to compare RF performance with and without TVC-BGWO feature selection, testing the null hypothesis that error distributions are equivalent. The Friedman test with Bonferroni-corrected post-hoc pairwise comparisons was used to compare

multiple algorithms across 5-fold stratified cross-validation. Confidence intervals (95 %) were computed using the Wilson score method, which provides accurate interval estimation even

Table 6. Multiclass Classification Performance Without and With TVC-BGWO Feature Selection on NSL-KDD Dataset

Model	Feature Selection	Accuracy	Precision	Recall	F1-Score	Inference Time (ms)
Random Forest	Without FS	0.7374	0.8047	0.7374	0.6899	0.0061
XGBoost	Without FS	0.7695	0.8245	0.7695	0.7342	0.0028
AdaBoost	Without FS	0.7487	0.8027	0.7487	0.7011	0.0284
Gradient Boosting	Without FS	0.7827	0.8311	0.7827	0.7492	0.0124
Random Forest	With FS	0.7561	0.8201	0.7561	0.7133	0.0060
XGBoost	With FS	0.7651	0.8231	0.7651	0.7242	0.0028
AdaBoost	With FS	0.6535	0.7547	0.6535	0.6081	0.0092
Gradient Boosting	With FS	0.7547	0.8170	0.7547	0.7195	0.0061

for extreme proportions near 0 or 1. Statistical significance was set at $\alpha = 0.05$ for all tests. Power analysis using G*Power 3.1 confirmed greater than 99 % statistical power to detect differences as small as 0.1 % in accuracy, ensuring adequate sensitivity to detect meaningful performance improvements.

4.4.1. McNemar's Test Results

Table 7. Random Forest Performance Comparison With and Without TVC-BGWO Feature Selection on CIC-IDS2017 Binary Classification

Model	Accuracy (%)	95% CI (%)	Correct (n)	Wrong (n)
RF without FS	99.82	99.80–99.84	161,840	294
RF with TVC-BGWO	**99.91**	**99.89–99.92**	**161,981**	**153**
Improvement	+0.087	–	+141	-141

Table 7 presents McNemar's test results comparing RF performance with and without TVC-BGWO feature selection on CIC-IDS2017 binary classification. Random Forest with TVC-BGWO achieved 99.91 % accuracy (95 % CI: 99.89–99.92 %), significantly outperforming Random Forest without feature selection at 99.82 % (95 % CI: 99.80–99.84 %). McNemar's test confirmed statistical significance ($\chi^2 = 82.01, p < 0.001$) with a large effect size (Cohen's $g = -9.12$).

4.4.2. Friedman Test Results

The Friedman test Table 8 revealed statistically significant differences among the evaluated models ($\chi^2 = 15.00$, $df = 3$, $p = 0.0018$). Random Forest achieved the highest test

accuracy of 99.91%, followed by XGBoost 99.68%, Gradient Boosting 99.56%, and AdaBoost 99.34%. However, Bonferroni-corrected post-hoc pairwise comparisons (adjusted $\alpha = 0.0083$) indicated no statistically significant differences between individual model pairs. This suggests that, although Random Forest demonstrated numerically superior performance, the practical performance differences among the top-performing models are marginal. Additionally, all models exhibited excellent stability, with coefficients of variation below 0.06 %.

Table 8. Performance Comparison of Ensemble Algorithms With TVC-BGWO Feature Selection on CIC-IDS2017 Binary Classification

Model	Test Accuracy (%)	95% CI (%)	CV Mean (%)	CV Std (%)
Random Forest	**99.91**	**99.89–99.92**	**99.90**	**0.019**
XGBoost	99.68	99.66–99.71	99.66	0.018
Gradient Boosting	99.56	99.12–99.18	99.56	0.057
AdaBoost	99.34	99.30–99.37	99.40	0.027

Table 9. Single-Sample Inference Latency Statistics for IoV Intrusion Detection

Configuration	Mean (ms)	P95 (ms)	P99 (ms)	P99.9 (ms)	Std Dev (ms)	CV (%)
Binary – RF (78 features)	4.52	5.18	6.58	8.34	0.47	10.42
Binary – RF (5 features)	4.63	5.56	6.88	9.29	0.58	12.49
Binary – XGB (78 features)	0.31	0.37	0.56	0.70	0.05	17.49
Binary – XGB (5 features)	0.30	0.36	0.42	0.56	0.05	15.09
Multiclass – RF (78 features)	4.43	5.05	6.76	8.14	0.47	10.61
Multiclass – RF (5 features)	4.45	5.12	5.79	7.27	0.38	8.47
Multiclass – XGB (78 features)	0.33	0.38	0.48	0.72	0.04	11.59
Multiclass – XGB (5 features)	0.33	0.40	0.58	0.89	0.06	16.79

4.5. Latency Distribution Analysis

Table 9 presents a comprehensive single-sample inference latency analysis for both binary and multiclass IoV intrusion detection tasks. The results show that all evaluated models satisfy the 100 ms maximum latency requirement specified by 3GPP for LTE-based V2V/V2P communication [1], and remain comfortably within the tighter 5–100 ms range defined for 5G NR V2X advanced driving use cases [6, 14], including the 3–10 ms bounds required for latency-critical services such as cooperative lane change and emergency trajectory alignment [6, 1]. This confirms the framework's suitability for real-time deployment across both LTE C-V2X and 5G NR V2X IoV environments. Random Forest models exhibit mean inference times of approximately 4.4–4.6 ms, with P99 latencies remaining below 7 ms, demonstrating stable and predictable performance under worst-case conditions. XGBoost models achieve significantly lower inference latency, with mean values below 0.35 ms and P99 latencies under 0.6 ms, making them particularly attractive for ultra-low-latency vehicular applications. Feature selection reduces the dimensionality from 78 to 5 features without introducing latency overhead, indicating that inference time is dominated by model complexity rather than input

dimensionality. Overall, the latency results confirm that the proposed intrusion detection framework can operate within real-time constraints while maintaining robustness across both binary and multiclass classification scenarios.

4.6. Adversarial Robustness Evaluation

Beyond standard evaluation metrics, we conducted comprehensive adversarial testing to assess model resilience against three attack scenarios: training data poisoning, test-time Gaussian noise injection, and targeted feature perturbation. These evaluations provide critical insights into model reliability under adversarial conditions that are increasingly relevant as attackers become aware of deployed machine learning defenses.

4.6.1. Data Poisoning Resilience

Table 10. Model Resilience to Data Poisoning Attacks With TVC-BGWO Feature Selection on CIC-IDS2017

Poisoning Ratio (%)	RF Accuracy (%)	XGB Accuracy (%)
0 (Baseline)	**99.80**	**99.72**
1	99.80	99.72
5	99.80	99.71
10	99.79	99.64
15	99.79	99.72
20	99.78	99.66
Max Drop	**0.02%**	**0.08%**
Robustness Score	**99.98**	**99.92**

To evaluate resilience to training data manipulation, we simulated data poisoning attacks Table 10 by randomly flipping class labels in 1–20 % of training samples and retraining models with the corrupted data. Remarkably, both Random Forest and XGBoost demonstrated exceptional robustness across poisoning ratios up to 20%, with maximum accuracy degradation of only 0.02% (Random Forest, at 20% poisoning) and 0.08% (XGBoost, at 10% poisoning)—yielding robustness scores of 99.98 and 99.92.

4.6.2. Gaussian Noise Tolerance

Network traffic features inevitably contain measurement noise from sensor inaccuracies, network jitter, and environmental variance. Table 11 evaluates model performance when Gaussian noise with standard deviation σ is added to all test features. At realistic noise levels $\sigma = 0.01$–0.05, models maintain a performance above 88% accuracy, indicating an acceptable noise tolerance for typical operating conditions. Random Forest demonstrates superior noise

resilience compared to XGBoost at all noise levels, with accuracy degradation of 8.10 % vs. 10.24 % at $\sigma = 0.01$ and 10.45% vs. 11.04 % at $\sigma = 0.05$. However, under extreme noise conditions $\sigma = 1.0$, accuracy drops substantially to 78.95 % Random Forest and 75.75 % XGBoost, revealing vulnerability to severe perturbations.

Table 11. Performance Under Gaussian Noise Attack With TVC-BGWO Feature Selection on CIC-IDS2017

Noise (σ)	RF Acc (%)	Drop (%)	XGB Acc (%)	Drop (%)
Baseline	**99.80**	**0.00**	**99.72**	**0.00**
0.01	91.70	8.10	89.48	10.24
0.05	89.35	10.45	88.68	11.04
0.10	89.02	10.78	88.43	11.29
0.20	88.60	11.20	87.47	12.25
0.50	85.48	14.32	81.75	17.97
1.00	78.95	20.85	75.75	23.97
Robustness Score	**79.15**	-	**76.03**	-

4.6.3. Targeted Feature Perturbation Vulnerability

Table 12. Targeted Feature Perturbation Attack Results With TVC-BGWO Feature Selection on CIC-IDS2017

Perturbation Ratio	RF Acc (%)	Detection Rate (%)	XGB Acc (%)	Detection Rate (%)
Baseline	**99.80**	**100.00**	**99.72**	**100.00**
0.05	89.79	38.14	88.63	33.31
0.10	89.19	34.28	88.47	32.40
0.20	88.85	32.24	87.64	31.72
0.50	85.77	30.50	82.14	32.70
1.00	79.36	31.46	76.25	34.45
Robustness Score	**79.56**	-	**76.52**	-

Table 12 reveals a critical vulnerability: under heavy perturbation ratio = 1.0, attack detection rates collapse to 31.46 % Random Forest and 34.45 % XGBoost, despite maintaining overall accuracy above 76 %. This striking discrepancy occurs because perturbations primarily affect attack samples while benign samples remain correctly classified, resulting in high overall accuracy but severe degradation in the security-critical metric of attack detection.

4.6.4. Comprehensive Robustness Summary

Table 13 synthesizes adversarial testing results across all attack scenarios. Random Forest demonstrates superior overall robustness compared to XGBoost with an average robustness score of 86.23 vs. 84.16. Both models exhibit exceptional resilience to data poisoning (robustness scores > 99.9%) but show moderate vulnerability to test-time perturbations (Gaussian noise and targeted feature manipulation), with robustness scores of 76-79 %. These findings suggest that ensemble models with TVC-BGWO feature selection are highly

Table 13. Adversarial Robustness Summary With TVC-BGWO Feature Selection on CIC-IDS2017

Attack Type	Model	Max Drop (%)	Robustness Score
Data Poisoning	RF	**0.02**	**99.98**
	XGB	0.08	99.92
Gaussian Noise	RF	20.85	79.15
	XGB	23.97	76.03
Feature Perturbation	RF	20.44	79.56
	XGB	23.48	76.52
Average	**RF**	**13.77**	**86.23**
	XGB	**15.84**	**84.16**

resistant to training-time attacks but require additional hardening techniques for deployment in environments where test-time adversarial examples are a concern.

Table 14. Ablation Study Results for the Three-Stage Feature Selection Framework on the CIC-IDS2017 Dataset

Binary Classification	**Features**	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score**	**Training Time (s)**	**Testing Time (s)**
No Feature Selection	78	99.82	99.82	99.82	0.9982	64.50	0.790
Stage 1 Only (Variance)	70	99.83	99.83	99.83	0.9983	70.97	0.782
Stage 2 Only (Correlation)	51	99.82	99.82	99.82	0.9982	53.40	0.658
Stage 1 + Stage 2	43	99.82	99.82	99.82	0.9982	53.88	0.653
All Stages (TVC-BGWO)	**5**	**99.91**	**99.91**	**99.91**	**0.9991**	**26.48**	**0.625**
Improvement over Baseline	**-93.59%**	**+0.087%**	**+0.087%**	**+0.087%**	**+0.086**	**-58.96%**	**-20.91%**
Multiclass Classification	**Features**	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score**	**Training Time (s)**	**Testing Time (s)**
No Feature Selection	78	99.75	99.75	99.75	0.9971	67.30	0.906
Stage 1 Only (Variance)	70	99.75	99.76	99.75	0.9971	72.53	0.845
Stage 2 Only (Correlation)	51	99.77	99.78	99.77	0.9973	56.19	0.821
Stage 1 + Stage 2	43	99.77	99.78	99.77	0.9973	55.46	0.871
All Stages (TVC-BGWO)	**5**	**99.87**	**99.90**	**99.87**	**0.9984**	**22.42**	**0.755**
Improvement over Baseline	**-93.59%**	**+0.123%**	**+0.141%**	**+0.123%**	**+0.135**	**-66.69%**	**-16.68%**

Table 14 quantifies individual stage contributions through systematic ablation. Stage 1 (Variance Threshold) provides minimal reduction (78→70, 10.26 %) with negligible accuracy impact (±0.007 %), establishing baseline noise removal. Stage 2 (Correlation filtering) achieves substantial reduction (78→51, 34.62 % while maintaining accuracy, validating multicollinearity elimination effectiveness.

Stage 3 (BGWO) yields the most significant improvements: from Stage 1+2 baseline (43

features), BGWO reduces dimensionality by 88.37 % (43→5) while improving accuracy by +0.084 % for binary and +0.107 % for multiclass. This counterintuitive gain validates BGWO's capability to identify synergistic feature interactions maximizing ensemble performance through metaheuristic optimization. Computational efficiency gains are substantial: 58.96 % (binary) and 66.69 % (multiclass) training time reduction versus full-feature baselines. Notably, Stage 2 Only (78→51 features) achieves a substantial share of TVC-BGWO's computational efficiency gains while incurring zero accuracy degradation, offering a rapid-deployment alternative when BGWO's optimization overhead is prohibitive. The hierarchical design ensures computational tractability by reducing search space from $2^{78} \approx 3.0 \times 10^{23}$ to $2^{43} \approx 8.8 \times 10^{12}$ possible subsets before metaheuristic optimization.

4.7. Comparison with Recent Methods

Table 15. Comparison of Proposed TVC-BGWO Approach with Recent State-of-the-Art Methods on CIC-IDS2017 Dataset

Study	Feature Selection	Binary Accuracy (%)	Multiclass Accuracy (%)	Dataset	Absolute Improvement
Zhang et. al. (2024) [44]	GRIPCA	94.67	–	CIC-IDS2017	+5.22 pp
El-Gay et al. (2024) [21]	PCA	99.20	–	CIC-IDS2017	+ 0.71 pp
Almansour et al. (2025) [7]	FD-Scheme	98.37	–	CIC-IDS2017	+1.54 pp
Khan et al. (2025) [27]	CMSO	99.10	–	CIC-IDS2017	+0.81 pp
Proposed	**TVC-BGWO**	**99.91**	**99.87**	CIC-IDS2017	

Compared with Zhang et al. [24], which achieved a binary accuracy of 94.67% using GRIPCA, our proposed approach improves performance by +5.24 pp.

Table 15 presents a comparative evaluation of the proposed TVC-BGWO framework against recent methods in intrusion detection in IoV on the CIC-IDS2017 dataset. Compared with Zhang et al. [44], which achieved a binary accuracy of 94.67 % using GRIPCA, our proposed approach improves performance by +5.22 %. Relative to El-Gayer*et al.* [21], which employed PCA and reported an accuracy of 99.20 %, an improvement of +0.71 % is observed. The proposed method also outperforms the FD-based scheme of Almansour *et al.* [7] by +1.54 % and the CMSO-based approach of Khan *et al.* [27] by +0.81 %. Furthermore, unlike the compared studies that focus exclusively on binary classification, the proposed framework additionally achieves a multiclass accuracy of 99.87 %, demonstrating enhanced robustness across diverse attack categories. These results highlight the effectiveness of the multi-stage TVC-BGWO feature selection strategy.

5. Conclusion

This study presented a machine learning-based intrusion detection framework that integrates a novel TVC-BGWO feature selection method with ensemble learning models. Evaluated on CIC-IDS2017 and NSL-KDD across binary and multiclass classification, RF with TVC-BGWO achieved the highest performance on CIC-IDS2017, with McNemar's and Friedman tests confirming significant, consistent improvement over baseline models. A novel contri-

bution is the evaluation of adversarial robustness in three attack scenarios. The framework showed exceptional resistance to data poisoning, but moderate vulnerability to test-time Gaussian noise and feature perturbation.

A key limitation concerns the severe class imbalance that affects multiclass detection reliability for extremely rare attack categories, most notably SQL Injection (21 instances) and Infiltration (18 instances). We attempted to mitigate this via SMOTE oversampling of the minority classes; however, this did not improve detection for either class, likely because the k-nearest-neighbor interpolation underlying SMOTE produces synthetic samples that are near-duplicates of the same few real observations rather than new discriminative information when the source class is this sparse. We therefore report results on the original, unresampled dataset. Future work should therefore address class imbalance, validate the framework on CAN-bus, Car-Hacking, and CICIoV2024 datasets, integrate explainable artificial intelligence for interpretable detection, and extend adversarial evaluation to vehicular-specific threats such as Sybil attacks, wormhole attacks, and coordinated multi-vehicle scenarios.

References

[1] 3rd Generation Partnership Project (2025). LTE; Service requirements for V2X services (3GPP TS 22.185 version 19.0.0 Release 19). Technical Report ETSI TS 122 185 V19.0.0, European Telecommunications Standards Institute (ETSI). 3GPP TS 22.185 V19.0.0, Release 19.

[2] Abreu, Rafael, Emanuel Simão, Carlos Serôdio, Frederico Branco, and António Valente. "Enhancing IoT security in vehicles: A comprehensive review of AI-driven solutions for cyber-threat detection." *Ai* 5, no. 4 (2024): 2279-2299.

[3] Adnan, Iqra, Tariq Umer, Ahmad Arsalan, Maryam M. Al Dabel, Ali Kashif Bashir, and Arooj Ansif. "Data driven vehicular heterogeneity based intelligent collision avoidance system for Internet of Vehicles (IoV)." *Digital Communications and Networks* (2025).

[4] Alalwany, Easa, and Imad Mahgoub. "Security and trust management in the internet of vehicles (IoV): Challenges and machine learning solutions." *Sensors* 24, no. 2 (2024): 368.

[5] Ali, Elmustafa Sayed, Mohammad Kamrul Hasan, Rosilah Hassan, Rashid A. Saeed, Mona Bakri Hassan, Shayla Islam, Nazmus Shaker Nafi, and Savitri Bevinakoppa. "Machine learning technologies for secure vehicular communication in internet of vehicles: recent advances and applications." *Security and Communication Networks* 2021, no. 1 (2021): 8868355.

[6] Ali, Z., Lagén, S., Giupponi, L., and Rouil, R. (2021b). 3gpp nr v2x mode 2: Overview, models and system-level evaluation. *IEEE Access*, 9:89554–89579.

[7] Almansour, Shahad, Kusum Yadav, Lulwah M. Alkwai, Norah Saleh Alghamdi, Wattana Viriyasitavat, and Gaurav Dhiman. "Adaptive personalized federated learning with lightweight depthwise convolutional bottleneck network for novel intrusion detection system in internet of vehicles." *Scientific Reports* 15, no. 1 (2025): 35604.

[8] Almukhalfi, Hanan, Ayman Noor, and Talal H. Noor. "Traffic management approaches using machine learning and deep learning techniques: A survey." *Engineering Applications of Artificial Intelligence* 133 (2024): 108147.

[9] Alohali, Manal Abdullah, Sultan Alahmari, Mohammed Aljebreen, Mashael M. Asiri, Achraf Ben Miled, Sami Saad Albouq, Othman Alrusaini, and Ali Alqazzaz. "Two stage malware detection model in internet of vehicles (IoV) using deep learning-based explainable artificial intelligence with optimization algorithms." *Scientific Reports* 15, no. 1 (2025): 20615.

[10] Aloqaily, Ahmad, Emad E. Abdallah, Hiba AbuZaid, Alaa E. Abdallah, and Malak Al-hassan. "Supervised machine learning for real-time intrusion attack detection in connected and autonomous vehicles: A security paradigm shift." In *Informatics*, vol. 12, no. 1, p. 4. MDPI, 2025.

[11] Altowaijri, Saleh M., and Mohamed Ayari. "The synergistic impact of 5G on cloud-to-edge computing and the evolution of digital applications." *Mathematics* 13, no. 16 (2025): 2634.

[12] Aziz, Saddam, Muhammad Talib Faiz, Adegoke Muideen Adeniyi, Ka-Hong Loo, Kazi Nazmul Hasan, Linli Xu, and Muhammad Irshad. "Anomaly detection in the internet of vehicular networks using explainable neural networks (xnn)." *Mathematics* 10, no. 8 (2022): 1267.

[13] Bala, Ritu, and Ritu Nagpal. "A review on kdd cup99 and nsl nsl-kdd dataset." *International Journal of Advanced Research in Computer Science* 10, no. 2 (2019): 64-67.

[14] Basnayake, Vishaka, Prabhash Rathnayake, and Ella Peltonen. "Vehicle-to-Everything Services in 3GPP 5G Networks: An Empirical Analysis." In *Proceedings of the 8th International Workshop on Edge Systems, Analytics and Networking*, pp. 25-30. 2025.

[15] Chen, Tianqi, and Carlos Guestrin. "Xgboost: A scalable tree boosting system." In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785-794. 2016.

[16] Chen, T. "Package "xgboost"(R version 90 (1-66), 40)." *R Core Team* (2019).

[17] Chen, Yanjiao, Xiaotian Zhu, Xueluan Gong, Xinjing Yi, and Shuyang Li. "Data poisoning attacks in internet-of-vehicle networks: Taxonomy, state-of-the-art, and future directions." *IEEE Transactions on Industrial Informatics* 19, no. 1 (2022): 20-28.

[18] Chunduri, Venkata, Mahmood Alsaadi, Sachin Gupta, Tariq Ahamed Ahanger, Adapa Gopi, Faisal Yousef Alghayadh, Navruzbek Shavkatov, and Ganesh Kumar Mahato. "Blockchain-based secure trust management scheme for Internet of Vehicles over cyber-physical system." *IEEE Transactions on Intelligent Transportation Systems* 26, no. 9 (2024): 14067-14076.

[19] Dasari, Deepthi Reddy, and G. Hima Bindu. "An intelligent intrusion detection system in IoV using machine learning and deep learning models." *International Journal of Communication Systems* 38, no. 10 (2025): e70131.

[20] Deng, Xiongshi, Min Li, Lei Wang, and Qikang Wan. "Rfcbf: enhance the performance and stability of fast correlation-based filter." *International Journal of Computational*

Intelligence and Applications 21, no. 02 (2022): 2250009.

[21] El-Gayar, Mostafa Mahmoud, Faheed AF Alrslani, and Shaker El-Sappagh. "Smart collaborative intrusion detection system for securing vehicular networks using ensemble machine learning model." *Information* 15, no. 10 (2024): 583.

[22] Fida, Muhamma Al Fatih Abil, Tohari Ahmad, and Maurice Ntahobari. "Variance threshold as early screening to Boruta feature selection for intrusion detection system." In *2021 13th International Conference on Information & Communication Technology and System (ICTS)*, pp. 46-50. IEEE, 2021.

[23] Guang, Haoran, Yongling He, Bowen Zheng, Weifeng Gong, Yi Shi, Hao Wu, Yaoguang Cao, Shichun Yang, and Hamid Reza Karimi. "Perspective: A Novel Resilient Cybersecurity Management System for Connected and Automated Vehicles." *Automotive Innovation* 8, no. 2 (2025): 335-367.

[24] Hu, Pei, Jeng-Shyang Pan, and Shu-Chuan Chu. "Improved binary grey wolf optimizer and its application for feature selection." *Knowledge-Based Systems* 195 (2020): 105746.

[25] Jain, Vishal, and Archan Mitra. "Real-time threat detection in cybersecurity: leveraging machine learning algorithms for enhanced anomaly detection." In *Machine Intelligence Applications in Cyber-Risk Management*, pp. 315-344. IGI Global Scientific Publishing, 2025.

[26] Ji, Baofeng, Xueru Zhang, Shahid Mumtaz, Congzheng Han, Chunguo Li, Hong Wen, and Dan Wang. "Survey on the internet of vehicles: Network architectures and applications." *IEEE Communications Standards Magazine* 4, no. 1 (2020): 34-41.

[27] Khan, Hasim, Ghanshyam G. Tejani, Rayed AlGhamdi, Sultan Alasmari, Naveen Kumar Sharma, and Sunil Kumar Sharma. "A secure and efficient deep learning-based intrusion detection framework for the internet of vehicles." *Scientific Reports* 15, no. 1 (2025): 12236.

[28] Lone, Faisal Rasheed, Harsh Kumar Verma, and Krishna Pal Sharma. "Evolution of VANETS to IoV: applications and challenges." *Tehnički glasnik* 15, no. 1 (2021): 143-149.

[29] Lu, Haihao, and Rahul Mazumder. "Randomized gradient boosting machine." *SIAM Journal on Optimization* 30, no. 4 (2020): 2780-2808.

[30] Lundberg, Hampus, Nishat I. Mowla, Sarder Fakhrul Abedin, Kyi Thar, Aamir Mahmood, Mikael Gidlund, and Shahid Raza. "Experimental analysis of trustworthy in-vehicle intrusion detection system using explainable artificial intelligence (XAI)." *Ieee Access* 10 (2022): 102831-102841.

[31] Mishra, Priyanka, and Ghanshyam Singh. "Internet of vehicles for sustainable smart cities: opportunities, issues, and challenges." *Smart Cities* 8, no. 3 (2025): 93.

[32] Patel, Tanish, Rutvij Jhaveri, Dhavalkumar Thakker, Sandeep Verma, and Palash Ingle. "Enhancing cybersecurity in internet of vehicles: a machine learning approach with explainable AI for real-time threat detection." In *Proceedings of the 40th ACM/SIGAPP symposium on applied computing*, pp. 2024-2031. 2025.

[33] Probst, Philipp, Marvin N. Wright, and Anne-Laure Boulesteix. "Hyperparameters and tuning strategies for random forest." *Wiley Interdisciplinary Reviews: data mining and knowledge discovery* 9, no. 3 (2019): e1301.

[34] Raja, Depa Ramachandraiah Kumar, Zuraida Abal Abas, Chandra Sekhar Akula, Yellapalli Dileep Kumar, Goshtu Hemanth Kumar, and Venappagari Eswari. "Artificial intelligence powered internet of vehicles: securing connected vehicles in 6G." *Indonesian Journal of Electrical Engineering and Computer Science* 35, no. 1 (2024): 213-221.

[35] Sarsam, Samer Muthana. "Cybersecurity challenges in autonomous vehicles: Threats, vulnerabilities, and mitigation strategies." *Shifra* 2023 (2023): 34-42.

[36] Shang, Wenli, Xiaojun Yang, Zhaojun Gu, Shuang Wang, and Lei Ding. "An Intrusion Detection System for the Internet of Vehicles Based on Multi-Level Hybrid Feature Analysis." In *2024 10th International Conference on Computer and Communications (ICCC)*, pp. 1976-1981. IEEE, 2024.

[37] Sharafaldin, Iman, Arash Habibi Lashkari, and Ali A. Ghorbani. "Toward generating a new intrusion detection dataset and intrusion traffic characterization." *ICISSp* 1, no. 2018 (2018): 108-116.

[38] Sherazi, Hafiz Husnain Raza, Razi Iqbal, Farooq Ahmad, Zuhaib Ashfaq Khan, and Muhammad Hasanain Chaudary. "DDoS attack detection: A key enabler for sustainable communication in internet of vehicles." *Sustainable Computing: Informatics and Systems* 23 (2019): 13-20.

[39] Singh, Abhijeet Pratap, Brijesh Kumar Chaurasia, and Ashish Tripathi. "Stacking enabled ensemble learning based intrusion detection scheme (SELIDS) for IoV." *SN Computer Science* 5, no. 8 (2024): 1000.

[40] Ullah, Safi, Muazzam A. Khan, Jawad Ahmad, Sajjad Shaukat Jamal, Zil e Huma, Muhammad Tahir Hassan, Nikolaos Pitropakis, Arshad, and William J. Buchanan. "HDL-IDS: a hybrid deep learning architecture for intrusion detection in the Internet of Vehicles." *Sensors* 22, no. 4 (2022): 1340..

[41] Walker, Kevin W., and Zhehan Jiang. "Application of adaptive boosting (AdaBoost) in demand-driven acquisition (DDA) prediction: A machine-learning approach." *The Journal of Academic Librarianship* 45, no. 3 (2019): 203-212.

[42] Wang, Kai, Aiheng Zhang, Haoran Sun, and Bailing Wang. "Analysis of recent deep-learning-based intrusion detection methods for in-vehicle network." *IEEE Transactions on Intelligent Transportation Systems* 24, no. 2 (2022): 1843-1854.

[43] Yang, Li, Abdallah Moubayed, and Abdallah Shami. "MTH-IDS: A multitiered hybrid intrusion detection system for Internet of Vehicles." *IEEE internet of things journal* 9, no. 1 (2021): 616-632.

[44] Zhang, Kaijun, Jiayu Yang, Yangfei Shao, Lehua Hu, Wei Ou, Wenbao Han, and Qionglu Zhang. "Intrusion detection model for internet of vehicles using GRIPCA and OWELM." *IEEe Access* 12 (2024): 28911-28925.

Received 07.02.2026, Accepted 16.08.2026