Improving Attack Detection in IoV with Class Balancing and Feature Selection

Thierry Widyatama Azhari [1]*, Ifan Rizqa [2]*, Fauzi Adi Rafrastara [3]*
* Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Dian Nuswantoro, Semarang
111202113807@mhs.dinus.ac.id [1], risqa.ifan@dsn.dinus.ac.id [2], fauziadi@research.dinus.ac.id [3]

Article Info

Article history:

Received 2025-01-15
Revised 2025-01-24
Accepted 2025-01-30

Keyword:

Internet of Vehicles (IoV),
DoS Attack,
Spoofing Attack,
Ensemble Learning.

ABSTRACT

The Internet of Vehicles (IoV) represents a specialized application of the Internet of Things (IoT), enabling vehicles to communicate with their surrounding infrastructure to enhance transportation safety and efficiency. However, IoV systems are susceptible to various cyberattacks, including Denial of Service (DoS) and spoofing attacks, which necessitate effective and efficient detection mechanisms. This study investigates the enhancement of detection efficiency for DoS and spoofing attacks in IoV by employing Ensemble Learning methods combined with feature selection techniques. The selected feature selection methods include Information Gain Ratio, Chi-Square (X^2), and Fast Correlation-Based Filter (FCBF). The CICIoV2024 dataset, utilized in this study, was balanced using the Random Under Sampling technique to address data imbalance issues. The ensemble algorithms evaluated in this research comprise Random Forest, Gradient Boosting, and XGBoost. Results indicate that all three algorithms achieved high accuracy and F1 scores, reaching 0.985. Moreover, the application of feature selection significantly reduced computational time without compromising detection performance. These findings are expected to contribute to the advancement of IoV security systems in the future.

I. Introduction

The Internet of Things (IoT) is an evolving concept that offers innovative approaches to connecting electronic devices and sensors via the internet, enabling efficient communication between devices without the need for human intervention. Through IoT, devices such as temperature, pressure, and light sensors can autonomously collect, process, and share data, making this technology highly valuable across various domains of human life, including smart homes, transportation, and healthcare [1][2]. This technology not only enhances efficiency and convenience but also facilitates faster and more accurate data-driven decision-making processes in diverse sectors, ranging from business and government to industry [2][3]. In the industrial realm, IoT enables the automation of production processes and real-time asset management, aiding companies in reducing operational costs and improving productivity [3][4]. Furthermore, IoT significantly contributes to the development of smart cities, where interconnected infrastructures optimize energy usage, manage traffic, and enhance security [3]. With the annual increase in IoT devices, this technology is projected to become a critical element in driving the Fourth Industrial Revolution (Industry 4.0) and bolstering global economic competitiveness [1][2]. However, despite the numerous benefits offered by IoT, challenges such as

data security and privacy remain key areas of focus in its future development [3].

The Internet of Vehicles (IoV) represents a more specific application of the Internet of Things (IoT), particularly within the transportation sector. This technology is envisioned to realize intelligent transportation systems (ITS), driving advancements in traffic management and mobility solutions. With the rapid development of communication technologies, high-throughput satellites, and cyber-physical systems, IoV enables smart vehicles to connect directly to the internet and interact with surrounding infrastructure components, such as roadways, pedestrians, and other vehicles [5][6]. IoV integrates technologies such as cellular communication, cloud computing, and edge computing to enhance connectivity and safety between vehicles and related infrastructure, thereby improving traffic efficiency and safety [7][8]. Additionally, IoV facilitates the implementation of the Vehicle-to-Everything (V2X) concept, supporting communication between vehicles and infrastructure, including other vehicles, traffic lights, and road sensors [6]. While IoV offers significant benefits, such as improved traffic management and reduced accidents, key challenges surrounding data security and privacy remain critical concerns in its development [8]. The adoption of blockchain technology is also being explored to ensure that IoV data exchange platforms are secure, transparent, and resistant to cyberattacks [6].

The Internet of Vehicles (IoV) plays a significant role in the development of smart cities, where the system is envisioned as an open and integrated network connecting various critical components. Consequently, IoV encompasses multiple types of communication, including Vehicle-to-Vehicle (V2V), Vehicle-to-Infrastructure (V2I), Vehicle-to-Pedestrian (V2P), and Vehicle-to-Network (V2N). This combination of communication types enables the exchange of information that significantly enhances traffic safety and efficiency [2][9]. V2V facilitates the sharing of data between vehicles regarding their position and speed to prevent collisions, while V2I supports communication between vehicles and road infrastructure to optimize traffic flow [2][9]. Moreover, V2P and V2N enable the management of interactions between vehicles and pedestrians and accelerate vehicle communication with broader networks, minimizing delays in responding to traffic conditions [9][10].

Denial of Service (DoS) attacks pose one of the most critical threats to IoV systems. The open and interconnected nature of IoV networks makes them particularly vulnerable to these attacks, as attackers can flood the network with false information. This results in severe disruptions to communication between vehicles and infrastructure, ultimately jeopardizing traffic safety and efficiency [11][12]. Although ongoing efforts aim to strengthen security, the limitations in detecting malicious nodes amidst a surge of fraudulent traffic remain a significant challenge [13]. These security gaps necessitate the development of more effective measures to ensure the system can maintain functionality during an attack, especially given the increasing complexity of IoV networks [12].

When a DoS attack occurs, core functionalities within an IoV system can become paralyzed as vehicle components fail to communicate with one another. Consequently, the data required by the central system becomes disorganized or entirely unavailable. For instance, radar systems responsible for maintaining automated lane safety may generate errors in decision-making processes [9][14]. In the case of Distributed Denial of Service (DDoS) attacks, the threat escalates as the attacks originate from multiple points, ultimately disrupting critical components across the entire network [13][14][11]. A SYN flood attack in the context of IoV, for example, exacerbates this issue by overwhelming the network with unresolved SYN requests, draining target resources and ignoring legitimate connection requests. This can lead to communication failures in autonomous vehicles, significantly impacting traffic safety [9][14]. Furthermore, botnets are frequently employed in DDoS attacks targeting IoV systems, where compromised devices launch simultaneous attacks on servers, denying legitimate user requests and triggering widespread system failures [11][14][15].

This research approach focuses on utilizing sensor data from vehicles, such as speedometer readings, steering angles, and accelerometer outputs, to accurately predict vehicle location shifts. These predictions are then compared with location shifts measured by the Global Navigation Satellite System (GNSS). This method enables faster and more accurate spoofing attack detection, particularly through machine learning approaches like Long Short-Term Memory (LSTM) neural networks, which are designed to identify discrepancies between GNSS data and predicted vehicle location shifts [16]. The LSTM-based detection system also integrates multiple data sources from vehicle sensors to enhance spoofing detection accuracy, demonstrating improved efficiency in attack prediction compared to conventional methods [16]. Furthermore, the integration of data from Inertial Measurement Units (IMU), Controller Area Network (CAN), and GNSS facilitates the development of more robust prediction systems against advanced spoofing attacks, including the detection of movement pattern anomalies caused by spoofing [16].

Research conducted by [17] focused on attack detection within the Internet of Vehicles (IoV) using the CICIoV2024 dataset to evaluate the performance of three algorithms: Decision Tree, Naive Bayes, and Logistic Regression. The dataset comprised real-time IoV communication data, specifically targeting Denial of Service (DoS) and spoofing attacks. The findings revealed that the Naive Bayes algorithm achieved the best results, with an accuracy of 98.10% and an F1-Score of 98.00%. However, a limitation of this study lies in the use of an imbalanced dataset, which may lead to biased analysis and a model that tends to favor the majority class.

A similar study by [18] also utilized the CICIoV2024 dataset but adopted a different approach. In this research, five types of attacks were executed on a 2019 Ford vehicle, focusing on Denial of Service (DoS) and spoofing attacks conducted through the CAN-BUS protocol. The study employed various Machine Learning (ML) algorithms, including Random Forest, AdaBoost, Logistic Regression, and Deep Neural Networks, to detect IoV attacks. The results indicated that Deep Neural Networks outperformed other methods, achieving an accuracy of 95% for binary data and 96% for decimal data. However, the study faced challenges due to imbalanced data and the absence of cross-validation, which increased the risk of overfitting in the resulting model.

Additionally, [17][18] identified a limitation regarding the dataset's coverage, which is restricted to specific vehicle types. The studies emphasized the need for broader testing across various vehicle models and communication protocols to develop a more generalized and robust detection model.

Research by [19] evaluated the performance of the Random Under-Sampling (RUS) technique in cancer classification using multi-omic data from TCGA. The results indicated that while RUS helped balance the data, some algorithms experienced a decline in performance after its application. The PART model achieved an accuracy of 97.4%

The study conducted by [20] compared the effectiveness of several data balancing techniques, including RUS, in the context of educational data classification. Using the High School Longitudinal Study of 2009 dataset, the research evaluated the performance of the Random Forest algorithm with various resampling techniques. The findings revealed that RUS effectively reduced model training time and slightly improved performance on datasets with extreme imbalance. However, the primary drawback remained the loss of information from the majority class, which could impact overall model reliability.

In this research, RUS method was utilized to address the class imbalance problem in the CICIoV2024 dataset. By reducing the number of samples in the majority class to align with those of the minority class, RUS achieves a balanced class distribution [21]. This technique minimizes the risk of bias toward the majority class and ensures that critical yet infrequent attack patterns are not ignored. While RUS inevitably results in the removal of some majority-class data, it is favored in this study for its straightforward implementation and effectiveness in managing significant class imbalances without generating synthetic data. The primary objective of this study is to assess the performance of machine learning models on both the original and RUS-balanced datasets [21].

Despite the numerous benefits that IoV offers, the involvement of internet technology introduces security vulnerabilities that pose significant risks. Attacks targeting IoV systems are emerging and increasing, ranging from Denial of Service (DoS) to Spoofing attacks. Therefore, effective methods are needed to detect these attacks to ensure that IoV systems remain secure and free from failures that could have fatal consequences for users, vehicles, and the surrounding environment. [17]

The effectiveness of class balancing techniques plays a pivotal role in addressing class imbalance, especially in real-world scenarios like IoV attack detection. Although oversampling methods like SMOTE are commonly employed, recent findings by [22] reveal significant drawbacks associated with these approaches. The study indicates that synthetic samples generated through oversampling often fail to capture the true characteristics of the minority class, resulting in poor generalization and inaccurate predictions in practical applications. These challenges highlight the necessity of choosing balancing methods that preserve data authenticity. By utilizing RUS, this study mitigates the risks linked to oversampling and contributes to the development of more robust techniques for managing class imbalance in critical areas such as IoV security.

II. Method

The stages undertaken in this study included data collection, preprocessing, modeling, and evaluation, as illustrated in Figure 1.

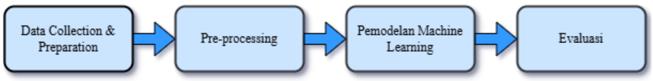

Figure 1. Research stages

A. Hardware and Software

Hardware and software play a crucial role in the process of training machine learning models, especially when dealing with large-scale datasets. An optimal combination of appropriate hardware and software enhances training efficiency, reduces the required time, and improves overall model performance, ensuring more effective and optimal learning outcomes [23].

In this study, the hardware utilized was a personal laptop equipped with an Intel Core i5 11400H processor and an NVIDIA RTX 3050 graphics card. The software used included Orange Data Mining and Microsoft Excel. Orange Data Mining was employed to implement machine learning models on both imbalanced and balanced datasets, while Microsoft Excel was used to systematically and structurally document the results of the algorithm applications.

B. Data Collection and Preparation

This research utilizes a public dataset developed in 2024 by researchers from the University of New Brunswick, Canada, called the "CIC IoV Dataset 2024," which can be downloaded from their website [17]. The novelty of the dataset is crucial in information security research, considering the constantly evolving attack patterns in the cyber world. This dataset comprises 11 key features and inclusdes six target classes: Benign, DoS (Denial-of-Service), Gas-Spoofing, Steering Wheel-Spoofing, Speed-Spoofing, and RPM-Spoofing. CICIoV2024 focuses on the detection of spoofing and DoS attacks through simulations conducted on a 2019 Ford vehicle, aiming to provide a realistic benchmark for developing cybersecurity solutions in the IoV environment. This dataset is expected to facilitate advancements in cybersecurity systems for smart vehicles in the future [18].

C. Data Pre-processing

Data preprocessing is a critical step in data analysis, aimed at ensuring the quality and consistency of the dataset before applying it to machine learning models. In this study, preprocessing was essential to address challenges in the CICIoV2024 dataset, particularly its highly imbalanced class distribution. The dataset contains a significantly higher number of benign instances compared to attack classes such as DoS and spoofing. Without proper preprocessing, the resulting model may exhibit bias toward the majority class, ultimately reducing the model's accuracy and effectiveness [24]. Normalization plays a vital role in standardizing diverse data formats, minimizing potential errors during processing [24]. Through appropriate preprocessing, the model becomes more accurate in detecting cyberattacks within IoV environments [24].

The CICIoV2024 dataset comprises six files encompassing various types of attacks, with a total of over 1 million data instances. The benign class dominates the dataset with 1,223,737 instances, while other classes, such as gas spoofing, contain only 9,991 instances. This significant imbalance in the number of instances per class indicates a vast disparity between classes, which may cause the model to bias toward the majority class. Figure 2 illustrates the class imbalance in the dataset.

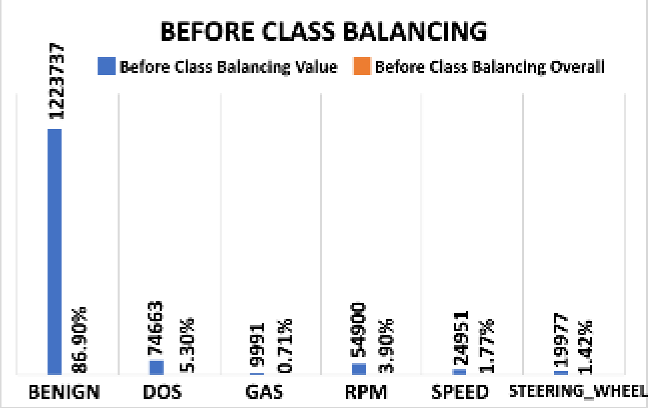

Figure 2. Graph before class balancing

To address this issue, class balancing steps were implemented using the Random Under-Sampling (RUS) method. This technique reduces the number of instances in the majority class to balance the dataset, enabling machine learning algorithms to provide more accurate predictions for minority classes [20]. The RUS method was implemented using orange software, with the fixed data sample feature set to 9,991 instances for each file. Subsequently, the six data files were combined using the concatenate feature.

After the merging process, the columns id, category, and specify_class were removed, leaving only the data columns from DATA_0 to DATA_7. The target was then set to specify_class, and the data was normalized using the min-max method within the range [0, 1].

In this study, feature selection was conducted using the rank method to reduce computational time without compromising model accuracy. The feature selection process involved a series of experiments, testing subsets of features ranging from the top-ranked single feature to the top eight. Several ranking techniques were applied, including Information Gain Ratio, Chi-square (X^2), and Fast Correlation-Based Filter (FCBF), each chosen for their ability to identify the most relevant features. The Information Gain Ratio was used to evaluate the contribution of each feature to the target variable by quantifying the amount of information it provides. The Chi-square test assessed the statistical independence of each feature with respect to the target, identifying those with significant associations. FCBF was employed to address feature redundancy by selecting features that are highly correlated with the target, while ensuring minimal overlap. This approach, combining multiple ranking techniques, aimed to optimize model performance by reducing the dimensionality of the dataset. By retaining only the most statistically relevant features, we were able to improve computational efficiency and reduce the risk of overfitting. Ultimately, the goal was to enhance model interpretability, generalization, and robustness, ensuring that the final model would perform well on unseen data while being computationally efficient.

In this study, experiments were conducted using ranked features. An illustration of these experiments can be seen in Figure 3.

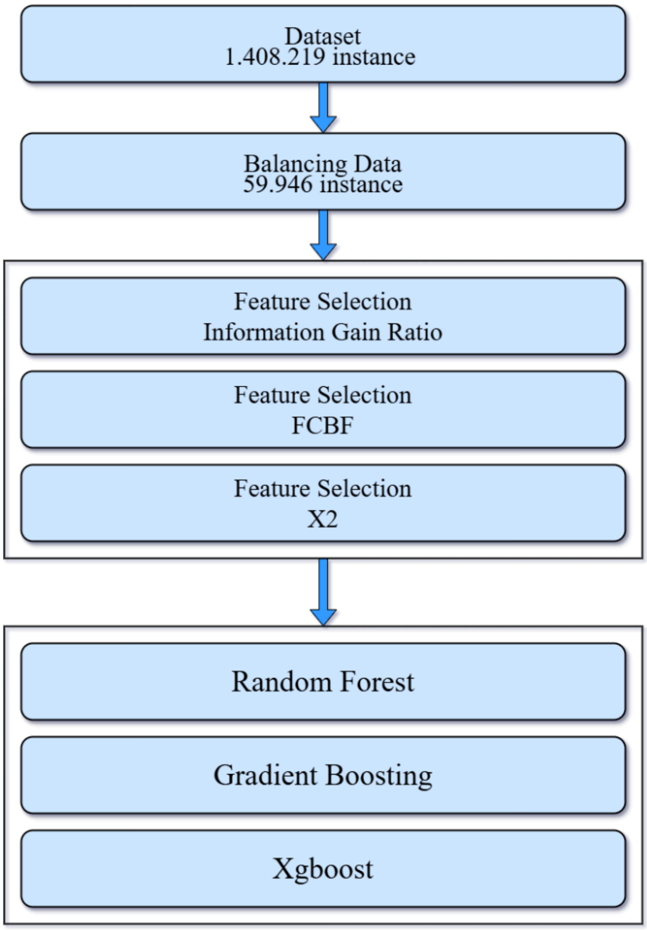

Figure 3. Feature Selection

D. *Machine Learning Modelling*

In this study, three classification algorithms were compared to evaluate their performance and identify the best algorithm. The three algorithms analyzed were Random Forest, Gradient Boosting, and Gradient Boosting (XGBoost).

Random Forest is a widely used machine learning algorithm for classification and regression tasks, leveraging the concepts of bootstrap aggregating (bagging) and decision trees. The following diagram provides an overview of the general structure of a Random Forest, illustrating its core processes and interactions [25].

In this algorithm, each decision tree is constructed from a random subset of the data, and predictions are made based on majority voting (for classification) or averaging (for regression), which enhances predictive accuracy. The primary advantage of Random Forest lies in its ability to reduce variance and handle noisy data, making it robust against overfitting.

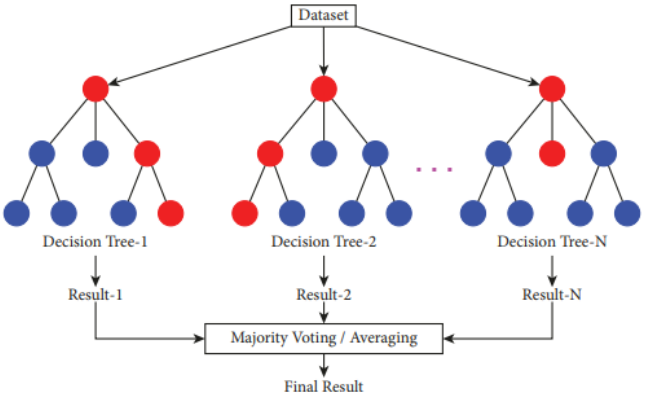

Figure 4. Illustration of Random Forest

Research in [16] highlights that the success of Random Forest is largely attributed to the randomization elements that act as an implicit form of regularization, particularly effective in scenarios with low signal-to-noise ratios. Parameters such as mtry, which determines the number of features considered at each split, play a critical role in reducing model variance [24][25]. This algorithm offers several advantages, including high accuracy, the ability to handle large and complex datasets, and resilience to outliers and noise, making it reliable for a wide range of classification and regression scenarios. Additionally, Random Forest effectively mitigates overfitting by combining multiple independent decision trees [27]. However, it is not without limitations, which include relatively high computational time, difficulty in interpretation due to model complexity, and large model size that can demand substantial computational resources [23][24].

Gradient Boosting is a powerful machine learning algorithm widely used for both classification and regression tasks. The following diagram provides an overview of the Gradient Boosting process, illustrating its iterative structure and the flow of residual error corrections [28].

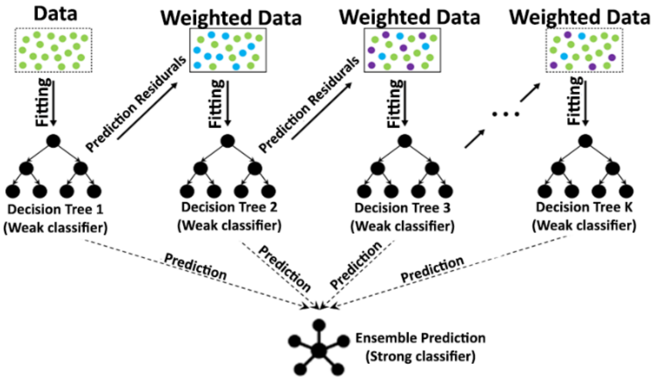
Data
Weighted Data
Weighted Data
Weighted Data
Fitting
Fitting
Fitting
Fitting
Prediction Residurals
Prediction Residurals
Decision Tree 1 (Weak classifier)
Decision Tree 2 (Weak classifier)
Decision Tree 3 (Weak classifier)
Decision Tree K (Weak classifier)
Prediction
Prediction
Prediction
Prediction
Ensemble Prediction (Strong classifier)

Figure 5. Illustration of Gradient Boosting

This algorithm operates iteratively, where each new model aims to correct the prediction errors of the previous model. Gradient Boosting begins with an initial decision tree, followed by subsequent iterations that add new trees focused on addressing the residual errors from prior iterations. This process continues until the model optimally minimizes the loss function [29]. The primary strength of Gradient Boosting lies in its ability to produce highly accurate models, especially when applied to complex datasets [30]. The algorithm is also versatile, as it can be employed with various types of data. However, its drawbacks include a higher risk of overfitting if hyperparameter tuning is not properly managed, as well as significant computational and resource demands due to the large number of trees built during the iterative process [25], [26][27].

XGBoost (Extreme Gradient Boosting) is an algorithm based on the Gradient Boosting Decision Tree (GBDT) method, designed to enhance computational efficiency and model performance for classification and regression tasks. The following diagram illustrates the core components and workflow of the XGBoost algorithm, highlighting its computational optimizations and tree-based structure [31].

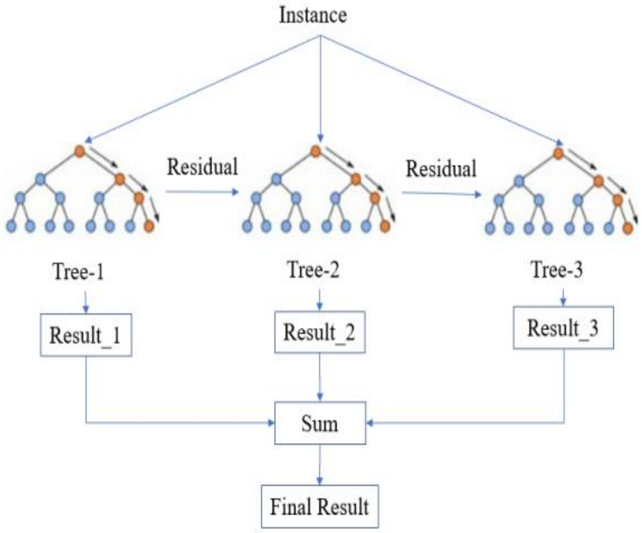

Figure 6. Illustration of XGBoost

One of XGBoost's standout features is its ability to perform efficient parallel processing and pruning, enabling the model to mitigate overfitting by reducing excessive tree complexity. Additionally, XGBoost can handle large-scale data more efficiently due to computational optimizations, including the effective utilization of CPU cores during the training process [32]. The primary advantage of XGBoost lies in its ability to control overfitting through mechanisms such as tree pruning and regularization, maintaining a balance between bias and variance, thereby producing more stable models. Its efficiency is further demonstrated by leveraging parallel processing, making it significantly faster in handling large and complex datasets. Moreover, its performance can be greatly improved through hyperparameter tuning techniques like grid search and random search. However, XGBoost has certain limitations. The hyperparameter tuning process required to achieve optimal performance can be intricate and resource-intensive. The algorithm is also sensitive to noisy data, potentially leading to suboptimal results if the data is not preprocessed effectively. Additionally, XGBoost demands substantial memory, particularly when dealing with very large datasets, due to its computationally intensive processes [28][29].

E. Machine Learning Model Evaluation

Evaluation is a crucial phase in data mining aimed at assessing the performance of the developed model. The purpose of model evaluation is to ensure that the model achieves high levels of accuracy, reliability, and relevance in addressing the targeted problem. Prior to evaluation, the model must undergo validation. One of the validation methods utilized in this study is Cross-Validation with k = 10 (10-fold Cross-Validation).

In this method, the dataset is divided into 10 equally sized parts. Each part is used once as a test set, while the remaining 9 parts are used for model training. This process is repeated 10 times, with each subset serving as the test set in turn. This method is preferred over Split Validation as it effectively reduces the risk of overfitting [17].

The next step is to evaluate the model using appropriate metrics. In classification tasks, commonly used metrics include accuracy, recall, precision, and F1-Score. Each metric measures different aspects of model performance. In this study, accuracy and F1-Score are selected as the primary metrics to assess the model's performance.

Accuracy is a metric that measures how often the model makes correct predictions overall. In the accuracy formula (Equation 1), TP (True Positive) represents the number of correct predictions for the positive class, while TN (True Negative) reflects correct predictions for the negative class. FP (False Positive) denotes the number of incorrect predictions where the positive class is mistakenly predicted, and FN (False Negative) occurs when the negative class is misclassified. In the context of attack detection in the Internet of Vehicles (IoV), accuracy provides a general overview of how well the model distinguishes various types of traffic, such as benign traffic, DoS, Gas-Spoofing, Steering Wheel-Spoofing, Speed-Spoofing, and RPM-Spoofing [17] (Equation 1).

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

Precision measures the proportion of positive predictions that are truly correct, helping to evaluate the model's effectiveness in avoiding false positives [34] (Equation 2).

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

Recall measures how well the model identifies all true positive instances. It reflects the proportion of positive cases correctly detected out of the total actual positive cases in the dataset [34] (Equation 3).

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

The F1-Score provides a comprehensive view of the balance between precision and recall. It helps to harmonize these two metrics, particularly in scenarios where errors in both positive and negative predictions have significant consequences, such as in attack detection within the IoV [34] (Equation 4).

$$F1 = 2\ x\ \frac{Presisi\ x\ Recall}{Presisi+Recall} \tag{4}$$

Evaluation results are used to compare various models, select the most optimal one, and identify aspects that require improvement. Effective evaluation ensures the development of high-quality data mining models and positively impacts further advancements.

III. RESULT AND DISCUSSION

This study utilizes the CICIoV2024 dataset, which consists of six separate files in decimal format. Each file contains 12 columns, with one column, specific_class, designated as the target variable, while the remaining columns serve as features. The specific_class column comprises six classes: benign, DoS, GAS, Speed, RPM, and Steering Wheel. Overall, the dataset includes 1,408,219 instances. The six files were combined using the concatenate feature available in Orange. Irrelevant columns, such as ID, Label, and Category, were removed to reduce complexity and dataset size. Prior to balancing, the dataset exhibited significant class imbalance. The benign class accounted for 86.90%, dominating other classes such as DoS (5.30%), GAS (0.71%), Speed (1.77%), RPM (3.90%), and Steering Wheel (1.42%).

To address the data imbalance, the Random Under Sampling (RUS) method was applied. This method reduces the number of instances in the majority class, creating a more balanced class distribution. After applying RUS, the total number of instances in the dataset decreased from 1,408,219 to 59,946 instances. As a result, each class—BENIGN, DoS, GAS, RPM, SPEED, and STEERING_WHEEL—now contains an equal number of instances, specifically 9,991 instances, representing 16.67% of the total dataset. This more balanced class distribution is expected to assist machine learning models in making more accurate and fair predictions across all classes, as illustrated in Figure 7. This step was crucial in ensuring that no single class dominated the training process, which could potentially lead to biased model performance.

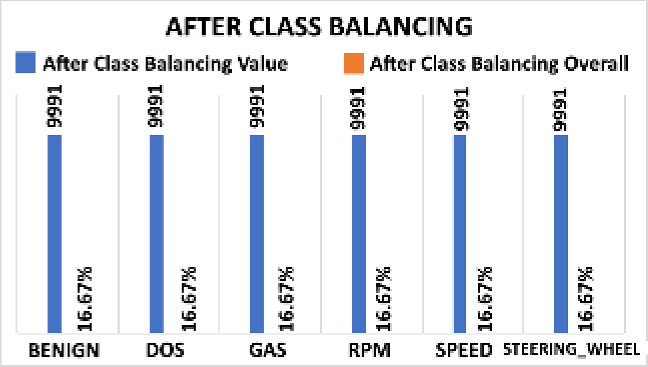

Figure 7. Graph after class balancing

After balancing the dataset distribution, the next step involves feature scaling by applying data normalization using the Min-Max method (0,1). This method transforms all values in the dataset from their original range to a scale between 0 and 1.

This study evaluates the performance of three classification algorithms: This study evaluates the performance of three classification algorithms: Random Forest, Gradient Boosting, and XGBoost. For Random Forest, the number of trees is set to 50, while Gradient Boosting is implemented with 100 trees. Similarly, the XGBoost algorithm is configured with 100 trees. These configurations represent the optimal version of each algorithm and method. The numbers are the best for each algorithm when adjusted for these methods; changing these numbers either upward or downward will drastically affect the test time, training time, and accuracy—some will experience performance gains, while others will see a decline across all methods.

To minimize the risk of overfitting, evaluation was conducted using the Cross-Validation method with a 10-fold scheme. The testing of these three algorithms was performed without applying feature selection, aiming to measure the models' performance comprehensively without reducing the data dimensionality. The results of this evaluation are presented in Table 1.

TABLE I
WITHOUT FEATURE SELECTION

Algorithm	Accuracy	F1 – Score
Random Forest	0.985	0.985
Gradient Boosting	0.985	0.985
XGboost	0.983	0.983

Further testing was conducted using the same algorithms, but with the application of feature selection through several ranking methods: Information Gain Ratio, Chi-square (χ^2), and Fast Correlation-Based Filter (FCBF). The results showed that accuracy and F1-Score remained consistent, even when only the top 5 features were used. In some cases, the model's performance improved compared to testing without feature selection.

Notably, in the case of the XGBoost algorithm, feature selection contributed to increased accuracy and efficiency compared to previous tests. The complete results of this testing are presented in Table 2.

TABLE II
WITH FEATURE SELECTION

Algorithm	Feature	Accuracy			F1 - Score		
		(IGR)	(FCBF)	(X^2)	(IGR)	(FCBF)	(X^2)
Random Forest	5	0.985	0.985	0.985	0.985	0.985	0.985
Gradient Boosting	5	0.985	0.985	0.985	0.985	0.985	0.985
XGBoost	5	0.985	0.985	0.985	0.985	0.985	0.985

In addition to evaluating accuracy and F1-Score, this study places significant emphasis on train time and test time as key metrics for assessing the computational efficiency of each algorithm. Train time plays a critical role in the development of machine learning models, as it reflects the duration required for a model to process and learn from the training data, which directly impacts the feasibility of deploying the model in resource-constrained environments [35].

Similarly, test time measures the speed at which the trained model can generate predictions on new, unseen data, highlighting its efficiency in real-world applications [36]. In time-sensitive domains such as the Internet of Vehicles (IoV), where rapid decision-making is essential to address potential security threats and ensure safety, minimizing test time becomes a crucial consideration. The ability to quickly process data and respond to threats can significantly enhance the effectiveness of IoV systems.

Table 3 provides a comprehensive overview of the train and test times recorded for the algorithms, both before and after applying feature selection techniques, offering valuable insights into the trade-offs between computational cost and model performance.

TABLE III
COMPLETE RESULT OF THE EXPERIMENT

Algorithm	**Train Time**				**Test Time**			
		(IGR)	(FCBF)	(X^2)		(IGR)	(FCBF)	(X^2)
Random Forest	3.006	2.041	2.192	2.188	0.387	0.228	0.238	0.222
Gradient Boosting	186.544	127.999	129.256	130.441	0.837	0.685	0.686	0.711
XGBoost	10.711	6.403	6.403	6.831	0.182	0.125	0.133	0.121

Table 3 provides a detailed summary of the experiment, including the training and testing times for Random Forest, Gradient Boosting, and XGBoost across three feature selection methods: Information Gain Ratio (IGR), Fast Correlation-Based Filter (FCBF), and Chi-Square (X^2). For Random Forest, the training times were 3.006 seconds (IGR), 2.041 seconds (FCBF), and 2.192 seconds (X^2). Its testing times were 0.387 seconds (IGR), 0.228 seconds (FCBF), and 0.238 seconds (X^2). In the case of Gradient Boosting, the training times were notably higher, at 186.544 seconds (IGR), 127.999 seconds (FCBF), and 129.256 seconds (X^2). Its testing times, however, were 0.837 seconds (IGR), 0.685 seconds (FCBF), and 0.686 seconds (X^2). Lastly, XGBoost showed a more efficient performance, with training times of 10.711 seconds (IGR), 6.403 seconds (FCBF), and 6.831 seconds (X^2). Its testing times were the shortest, recorded at 0.182 seconds (IGR), 0.125 seconds (FCBF), and 0.133 seconds (X^2).

These results highlight the differences in computational efficiency and speed among the algorithms, providing insights into their practical applications. With feature selection and data balancing, the algorithms in this study outperformed those in previous research, achieving higher accuracy and F1-Score. The comparison is shown in Table 4.

TABLE IV
COMPARISON WITH PREVIOUS RESEARCH

Algorithm	**Method**	**Feature Selection**	**Accuracy**	**F1-Score**
Random Forest	RUS & Feature Selection	IGR	0.985	0.985
		FCBF	0.985	0.985
		X2	0.985	0.985
Gradient Boost		IGR	0.985	0.985
		FCBF	0.985	0.985
		X2	0.985	0.985
XGBoost		IGR	0.985	0.985
		FCBF	0.985	0.985
		X2	0.985	0.985
Naïve Bayes [17]	Manual Selection	-	0.981	0.98
Decision Tree [17]		-	0.975	0.971
Logistic Regression [17]		-	0.876	0.842
AdaBoost [18]	-	-	0.92	0.51
Random Forest [18]	-	-	0.96	0.76
Logistic Regression [18]	-	-	0.89	0.49
Deep Neural Network [18]	-	-	0.96	0.78

To provide a clearer depiction of the accuracy performance comparison for each algorithm, the findings of this study are visualized in the form of a graph. This graph highlights the superiority of the proposed methods in this study compared to the best methods suggested in previous research. The feature selection techniques are directly compared with those from prior studies, as illustrated in Figure 5.

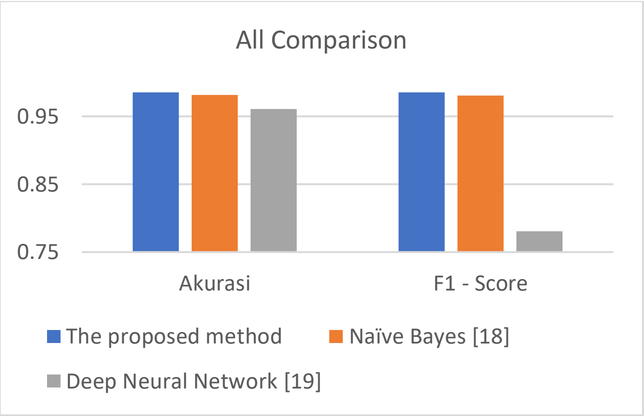

Figure 8. Comparison chart with previous researcher

The visualization results indicate that the proposed method, which combines data balancing and feature selection, achieved higher and consistent accuracy at a value of 0.985 compared to other algorithms tested in previous studies. The graph facilitates readers' understanding of the advantages of the approach used in this study in enhancing the accuracy of DoS and spoofing attack detection in IoV, while also highlighting the computational efficiency achieved through feature selection.

IV. CONCLUSION

This study successfully demonstrates that Ensemble Learning methods, particularly Random Forest, Gradient Boosting, and XGBoost, are effective in detecting DoS and spoofing attacks in IoV networks. Data balancing and feature selection improved computational efficiency without compromising accuracy and F1-Score, with XGBoost showing the best efficiency. With an accuracy and F1-Score of 0.985, this solution can enhance IoV security against cyber threats.

Future research should focus on developing a self-collected, balanced dataset from real-time IoV data to better represent real-world conditions. This approach would address overfitting risks and improve the model's generalizability to diverse vehicle types and protocols.

BIBLIOGRAPHY

[1] A. Korte, V. Tiberius, and A. Brem, “Internet of Things (IoT) Technology Research in Business and Management Literature: Results from a Co-Citation Analysis,” *JTAER*, vol. 16, no. 6, pp. 2073–2090, Aug. 2021, doi: 10.3390/jtaer16060116.

[2] S. Ahmetoglu, Z. Che Cob, and N. Ali, “A Systematic Review of Internet of Things Adoption in Organizations: Taxonomy, Benefits, Challenges and Critical Factors,” *Applied Sciences*, vol. 12, no. 9, p. 4117, Apr. 2022, doi: 10.3390/app12094117.

[3] A. M. Rahmani, S. Bayramov, and B. Kiani Kalejahi, “Internet of Things Applications: Opportunities and Threats,” *Wireless Pers Commun*, vol. 122, no. 1, pp. 451–476, Jan. 2022, doi: 10.1007/s11277-021-08907-0.

[4] S. Kumar, P. Tiwari, and M. Zymbler, “Internet of Things is a revolutionary approach for future technology enhancement: a review,” *J Big Data*, vol. 6, no. 1, p. 111, Dec. 2019, doi: 10.1186/s40537-019-0268-2.

[5] A. Talpur and M. Gurusamy, “Machine Learning for Security in Vehicular Networks: A Comprehensive Survey,” *IEEE Commun. Surv. Tutorials*, vol. 24, no. 1, pp. 346–379, 2022, doi: 10.1109/COMST.2021.3129079.

[6] M. B. Mollah *et al.*, “Blockchain for the Internet of Vehicles towards Intelligent Transportation Systems: A Survey,” *IEEE Internet Things J.*, vol. 8, no. 6, pp. 4157–4185, Mar. 2021, doi: 10.1109/JIOT.2020.3028368.

[7] K. Al Marri, F. A. Mir, S. A. David, and M. Al-Emran, Eds., *BUiD Doctoral Research Conference 2023: Multidisciplinary Studies*, vol. 473. in Lecture Notes in Civil Engineering, vol. 473. Cham: Springer Nature Switzerland, 2024. doi: 10.1007/978-3-031-56121-4.

[8] E. Alalwany and I. Mahgoub, “Security and Trust Management in the Internet of Vehicles (IoV): Challenges and Machine Learning Solutions,” *Sensors*, vol. 24, no. 2, p. 368, Jan. 2024, doi: 10.3390/s24020368.

[9] L. Khoukhi, H. Xiong, S. Kumari, and N. Puech, “The Internet of vehicles and smart cities,” *Ann. Telecommun.*, vol. 76, no. 9–10, pp. 545–546, Oct. 2021, doi: 10.1007/s12243-021-00891-7.

[10] C. Storck and F. Duarte-Figueiredo, “A 5G V2X Ecosystem Providing Internet of Vehicles,” *Sensors*, vol. 19, no. 3, p. 550, Jan. 2019, doi: 10.3390/s19030550.

[11] T. Christensen, S. B. Mandavilli, and C.-Y. Wu, “The Dark Side of The Internet of Vehicles: A Survey of the State of IoV and its Security Vulnerabilities”.

[12] P. Sharma, M. Patel, and A. Prasad, “A systematic literature review on Internet of Vehicles Security,” Dec. 16, 2022, *arXiv*: arXiv:2212.08754. Accessed: Oct. 23, 2024. [Online]. Available: http://arxiv.org/abs/2212.08754

[13] O. A. Albishi and M. Abdullah, “DDoS Attacks Detection in IoV using ML-based Models with an Enhanced Feature Selection Technique,” *IJACSA*, vol. 15, no. 2, 2024, doi: 10.14569/IJACSA.2024.0150282.

[14] M. A. H. Zamrai, K. Mohamad Yusof, and A. Azizan, “Dissecting Denial of Service (DoS) Syn Flood Attack Dynamics and Impacts in Vehicular Communication Systems,” *ITM Web Conf.*, vol. 63, p. 01008, 2024, doi: 10.1051/itmconf/20246301008.

[15] K. B. Adedeji, A. M. Abu-Mahfouz, and A. M. Kurien, “DDoS Attack and Detection Methods in Internet-Enabled Networks: Concept, Research Perspectives, and Challenges,” p. 57, Jul. 2023, doi: 10.3390/jsan12040051.

[16] S. Dasgupta, M. Rahman, M. Islam, and M. Chowdhury, “Prediction-Based Gnss Spoofing Attack Detection For Autonomous Vehicles”.

[17] F. A. Rafrastara, W. Ghozi, and A. Wardoyo, “Deteksi Serangan berbasis Machine Learning pada Internet of Vehicle,” vol. 2024, 2024.

[18] E. C. P. Neto *et al.*, “CICIoV2024: Advancing realistic IDS approaches against DoS and spoofing attack in IoV CAN bus,” *Internet of Things*, vol. 26, p. 101209, Jul. 2024, doi: 10.1016/j.iot.2024.101209.

[19] Y. Yang and G. Mirzaei, “Performance analysis of data resampling on class imbalance and classification techniques on multi-omics data for cancer classification,” *PLoS ONE*, vol. 19, no. 2, p. e0293607, Feb. 2024, doi: 10.1371/journal.pone.0293607.

[20] T. Wongvorachan, S. He, and O. Bulut, “A Comparison of Undersampling, Oversampling, and SMOTE Methods for Dealing with Imbalanced Classification in Educational Data Mining,” *Information*, vol. 14, no. 1, p. 54, Jan. 2023, doi: 10.3390/info14010054.

[21] M. Kim and K.-B. Hwang, “An empirical evaluation of sampling methods for the classification of imbalanced data,” *PLoS ONE*, vol. 17, no. 7, p. e0271260, Jul. 2022, doi: 10.1371/journal.pone.0271260.

[22] A. S. Tarawneh, A. B. Hassanat, G. A. Altarawneh, and A. Almuhaimeed, “Stop Oversampling for Class Imbalance Learning: A Review,” *IEEE Access*, vol. 10, pp. 47643–47660, 2022, doi: 10.1109/ACCESS.2022.3169512.

[23] L. Shen, Y. Sun, Z. Yu, L. Ding, X. Tian, and D. Tao, “On Efficient Training of Large-Scale Deep Learning Models: A Literature Review,” Apr. 07, 2023, *arXiv*: arXiv:2304.03589. Accessed: Oct. 30, 2024. [Online]. Available: http://arxiv.org/abs/2304.03589

[24] M. K. Diallo and O. Karahan, “Intrusion detection system using Optimized Machine Learning Algorithms for cyberattacks in the Internet of Vehicles (IoV),” in *Cognitive Models and Artificial Intelligence Conference Proceedings*, SETSCI, Jul. 2019, pp. 13–19. doi: 10.36287/setsci.17.1.0013.

[25] M. Y. Khan, A. Qayoom, M. S. Nizami, M. S. Siddiqui, S. Wasi, and S. M. K.-R. Raazi, “Automated Prediction of Good Dictionary EXamples (GDEX): A Comprehensive Experiment with Distant Supervision, Machine Learning, and Word Embedding-Based Deep Learning Techniques,” *Complexity*, vol. 2021, no. 1, p. 2553199, Jan. 2021, doi: 10.1155/2021/2553199.

[26] L. Mentch and S. Zhou, “Randomization as Regularization: A Degrees of Freedom Explanation for Random Forest Success”.

[27] R.-C. Chen, C. Dewi, S.-W. Huang, and R. E. Caraka, “Selecting critical features for data classification based on machine learning methods,” *J Big Data*, vol. 7, no. 1, p. 52, Dec. 2020, doi: 10.1186/s40537-020-00327-4.

[28] H. Deng, Y. Zhou, L. Wang, and C. Zhang, “Ensemble learning for the early prediction of neonatal jaundice with genetic features,” *BMC Med Inform Decis Mak*, vol. 21, no. 1, p. 338, Dec. 2021, doi: 10.1186/s12911-021-01701-9.

[29] S. E. Suryana, B. Warsito, and S. Suparti, “Penerapan Gradient Boosting Dengan Hyperopt Untuk Memprediksi Keberhasilan Telemarketing Bank,” *J.Gauss*, vol. 10, no. 4, pp. 617–623, Dec. 2021, doi: 10.14710/j.gauss.v10i4.31335.

[30] H. Firmansyah and Z. Abidin, “Penerapan Algoritma Gradient Boosted Decision Trees Pada Adaboost Untuk Klasifikasi Status Desa,” vol. 1, no. 1, 2022.

[31] W. Wang, G. Chakraborty, and B. Chakraborty, “Predicting the Risk of Chronic Kidney Disease (CKD) Using Machine Learning Algorithm,” *Applied Sciences*, vol. 11, no. 1, p. 202, Dec. 2020, doi: 10.3390/app11010202.

[32] G. Abdurrahman, H. Oktavianto, and M. Sintawati, “Optimasi Algoritma XGBoost Classifier Menggunakan Hyperparameter Gridesearch dan Random Search Pada Klasifikasi Penyakit Diabetes,” *INFORMAL*, vol. 7, no. 3, p. 193, Dec. 2022, doi: 10.19184/isj.v7i3.35441.

[33] T. Z. Jasman, M. A. Fadhlullah, A. L. Pratama, and R. Rismayani, “Analisis Algoritma Gradient Boosting, Adaboost dan Catboost dalam Klasifikasi Kualitas Air,” *JuTISI*, vol. 8, no. 2, Aug. 2022, doi: 10.28932/jutisi.v8i2.4906.

[34] T. Hu and X.-H. Zhou, “Unveiling LLM Evaluation Focused on Metrics: Challenges and Solutions,” Apr. 14, 2024, *arXiv*: arXiv:2404.09135. Accessed: Oct. 31, 2024. [Online]. Available: http://arxiv.org/abs/2404.09135

[35] M. Guimarães *et al.*, “Predicting Model Training Time to Optimize Distributed Machine Learning Applications,” *Electronics*, vol. 12, no. 4, p. 871, Feb. 2023, doi: 10.3390/electronics12040871.

[36] J. Hübotter, S. Bongni, I. Hakimi, and A. Krause, “Efficiently Learning at Test-Time: Active Fine-Tuning of LLMs,” Oct. 10, 2024, *arXiv*: arXiv:2410.08020. Accessed: Oct. 31, 2024. [Online]. Available: http://arxiv.org/abs/2410.08020