Journal of Applied Informatics and Computing (JAIC)
Vol.9, No.2, April 2025, pp. 250~257
e-ISSN: 2548-6861

Balancing CICIoV2024 Dataset with RUS for Improved IoV Attack Detection

Muhammad David Firmansyah [1]*, Ifan Rizqa [2]*, Fauzi Adi Rafrastara[3]*, Wildanil Ghozi[4]*
* Teknik Informatika, Universitas Dian Nuswantoro
mdavidfrsy@gmail.com [1], risqa.ifan@dsn.dinus.ac.id [2], fauziadi@research.dinus.ac.id [3], wildanil.ghozi@dsn.dinus.ac.id [4]

Article Info

Article history:

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

Keyword:

Internet of Things,
Internet of Vehicle,
Imbalanced Dataset,
Machine Learning,
Random Under Sampling.

ABSTRACT

This study addresses the cybersecurity challenges within the Internet of Vehicles (IoV) by exploring the efficacy of Random Under-Sampling (RUS) in balancing the class distribution of the CICIoV2024 dataset for improved intrusion detection. IoV technology connects vehicles to digital infrastructure, fostering communication and enhancing safety but is simultaneously vulnerable to cyber threats such as Denial of Service (DoS) and spoofing attacks. This research employed RUS to mitigate data imbalance within the CICIoV2024 dataset, which often impedes effective threat detection in machine learning models. Four machine learning classifiers Random Forest, AdaBoost, Gradient Boosting, and XGBoost were evaluated on both imbalanced and balanced datasets to compare their performance. Results demonstrated that RUS significantly enhances model accuracy, precision, recall, and F1-score, reaching perfect scores across all classifiers post-balancing. Additionally, RUS contributed to substantial reductions in training and testing times, thereby boosting computational efficiency. These findings underscore the potential of RUS in addressing data imbalance in IoV cybersecurity, establishing a foundation for future research aimed at safeguarding IoV systems against evolving cyber threats.

I. INTRODUCTION

Internet of Vehicles (IoV) is a technology that connects vehicles to digital infrastructure through the internet, enabling communication between vehicles (V2V), vehicles and infrastructure (V2I), and vehicles and pedestrians (V2P) [1]. This technology is expected to enhance transportation efficiency, reduce accidents, and support smart mobility in modern cities [2]. However, alongside these advancements, IoV has also become a target for cyberattacks that threaten the safety and security of the system [3]. Attacks such as Denial of Service (DoS) and spoofing can cause significant disruptions to the IoV network, ranging from manipulating vehicle data to blocking critical access, potentially leading to severe accidents [4].

In the context of the Internet of Vehicles (IoV), ensuring network security is crucial due to cyber threats that can compromise the integrity and functionality of the system. One of the most effective ways to protect the IoV network is by implementing an Intrusion Detection System (IDS) [5]. IDS serves as a protective layer by detecting suspicious or unauthorized activities within the network and preventing them before more serious damage occurs [6]. Without a reliable Intrusion Detection System (IDS), IoV systems are susceptible to attacks, including Denial of Service (DoS) and Spoofing [7].

The application of machine learning in intrusion detection within the Internet of Vehicles (IoV) offers significant advantages in identifying and responding to cyber threats [8]. This method is highly effective in analyzing large volumes of data, enabling real-time detection of attacks such as Denial of Service (DoS) and spoofing, which are crucial for maintaining system integrity [9]. Its ability to learn from historical data and update predictive models makes it more responsive than traditional approaches [10]. In the dynamic IoV environment, where attack patterns continuously evolve, machine learning plays a pivotal role in enhancing the overall security of the system [11].

Applying machine learning in IoV faces significant challenges related to data volume and sensor complexity [12]. One of the obstacles is class imbalance in datasets such as CiCIoV2024, where anomalous data is scarce, making it

difficult for algorithms to detect them [13]. The primary challenge in such datasets lies in balancing detection accuracy with computational efficiency for practical implementation [14]. Data imbalance can cause machine learning models to favor the majority class, increasing the risk of failing to detect rare but highly dangerous cyberattacks [15].

In this study, the researchers applied the Random Under-Sampling (RUS) method to address the issue of data imbalance in the CICIoV2024 dataset. RUS balances class distribution by reducing the instances in the majority class to match the minority class [16]. This approach prevents the model from being biased toward the majority class and ensures that rare but critical attack patterns are not overlooked. Although RUS may lead to a loss of some majority-class data, it remains a practical choice for this study due to its simplicity and effectiveness in handling severe class imbalance without introducing synthetic data. The goal of this research is to evaluate the performance of machine learning models on both the original and balanced datasets using RUS.

The research conducted by [9] aimed at creating a realistic CICIoV2024 dataset to identify attacks like Denial of Service (DoS) and spoofing in Internet of Vehicles (IoV) systems. This dataset was collected from the Controller Area Network (CAN) system of a stationary 2019 Ford vehicle to ensure safety during data collection. Various machine learning algorithms, including Logistic Regression (LR), Random Forest (RF), AdaBoost (AB), and Deep Neural Network (DNN), were used to analyze the attack data. The results indicated that DNN and RF models delivered the best performance with the highest F1-scores, reaching 0.63 for binary classification and 0.74 for decimal classification. However, the main limitation of this study was the significant class imbalance in the dataset, particularly with attacks like speed spoofing, which were challenging to detect, thereby reducing model accuracy as the number of analyzed classes increased.

Research on the Internet of Vehicles (IoV) using the CICIoV2024 dataset has been conducted by [17], which evaluated three algorithms: Naïve Bayes, Decision Tree, and Logistic Regression for detecting attacks. The results showed that Naïve Bayes achieved the highest accuracy of 98.10% and an F1-score of 98.00. This study concentrated on detecting attacks, such as Denial of Service (DoS) and spoofing, which can disrupt communication between devices in the IoV environment. The research used specify_class as the target in classification. Innovation in the dataset used is crucial in security research, considering that cyberattack patterns continue to evolve over time [18].

The study by [19] demonstrated the success of applying the RUS method in enhancing the performance of the Random Forest algorithm on imbalanced datasets. Using a dataset from the UCI Machine Learning Repository with a class ratio of malware to goodware at 1:9.5, this method effectively balanced the data. It resulted in significant improvements in model performance. Random Forest with RUS achieved accuracy, recall, and specificity of 98.3%, showcasing its superiority over other algorithms such as kNN, Naïve Bayes, and Logistic Regression. This success illustrates the potential of RUS to address severe class imbalance challenges and serves as the foundation for its application in this research to optimize model performance on the CICIoV2024 dataset.

The reliability of class balancing techniques is a critical factor in addressing class imbalance, particularly in real-world applications like IoV attack detection. While oversampling methods such as SMOTE are widely used, recent research by [20], highlights their significant limitations. The study demonstrates that synthetic samples generated by oversampling methods often fail to represent the true characteristics of the minority class, leading to poor model generalization and inaccurate predictions in practical use cases. This limitation underscores the importance of selecting a balancing method that maintains data authenticity. By opting for RUS, this research avoids the risks associated with oversampling and contributes to advancing techniques for handling class imbalance in critical domains such as IoV security.

The study by [21] aimed to identify Distributed Denial-of-Service (DDoS) attacks on the CICDDoS 2019 dataset using boosting algorithms, namely LightGBM and XGBoost. The XGBoost algorithm delivered the best results with an accuracy of 94.89% and superior processing time efficiency, thanks to its implementation of parallel processing and the use of L1 and L2 regularization. With its proven ability to handle large datasets and reduce noise, XGBoost has become an ideal choice for detecting large-scale attacks in real-world applications. This research builds upon such findings by integrating RUS-balanced datasets with advanced algorithms like XGBoost to achieve high accuracy and efficiency in detecting IoV attacks.

II. METHODOLOGY

As illustrated in Figure 1, the stages conducted in this study include data collection, pre-processing, modeling, and evaluation.

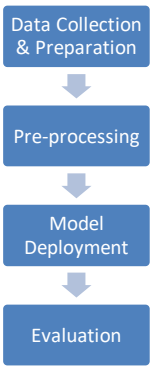

Figure 1.Research Stages

A. *Hardware and Software*

This study utilized a dataset comprising over 1 million instances. The selection of appropriate hardware and software is a key factor in the success of research. Without suitable

software, even high-specification hardware will not deliver optimal results. Similarly, effective software will face limitations if not supported by adequate hardware [22].

This study used a personal laptop equipped with an Intel Core i7 11800H processor and an NVIDIA RTX 3050ti graphics card. The software used included Orange Data Mining (downloaded from https://orangedatamining.com/) and Microsoft Excel. Orange Data Mining applied machine learning models to both imbalanced and balanced datasets. At the same time, Microsoft Excel was used to systematically and structurally record the results of the algorithm implementation.

B. Data Collection and Preparation

Cyberattack patterns continue to evolve. Therefore, the novelty of datasets is crucial in the context of network security [18]. In this study, the authors utilized the CICIoV2024 dataset, published by researchers from the University of New Brunswick, Canada, 2024 as the primary data source [13]. This dataset includes two main attack classifications: spoofing and Denial-of-Service (DoS), and consists of 11 features that support security analysis. Additionally, the dataset provides six distinct classes: Benign, Gas-Spoofing, RPM-Spoofing, Speed-Spoofing, Steering Wheel-Spoofing, and DoS. Details of the dataset used are presented in Table 1.

TABLE I
DETAIL DATASET

Dataset Name	CIC IoV Dataset 2024
Release Year	2024
Number of Features	11
Number of Instances	1408219 (1,223,737 Benign Class and 184,482 Attack Class)
Number of Class	6 (Benign, Gas-Spoofing, RPM-Spoofing, Speed-Spoofing, Steering Wheel-Spoofing, and DoS)

In the data preparation stage, the dataset was divided into two parts for two experiments. The first experiment used imbalanced data, where the benign and attack class data quantities remained unchanged. The second experiment divided the dataset into two classes: benign and attack. The benign class data was extracted from a single file and balanced using the RUS method to prevent this class from dominating the model.

Meanwhile, the attack class data consisted of five files: DoS, Steering Wheel-Spoofing, Gas-Spoofing, Speed-Spoofing, and RPM-Spoofing. Both classes, benign and attack, were combined and balanced using the RUS method, resulting in a balanced dataset for use in the second experiment. Figure 2 illustrates the dataset preparation.

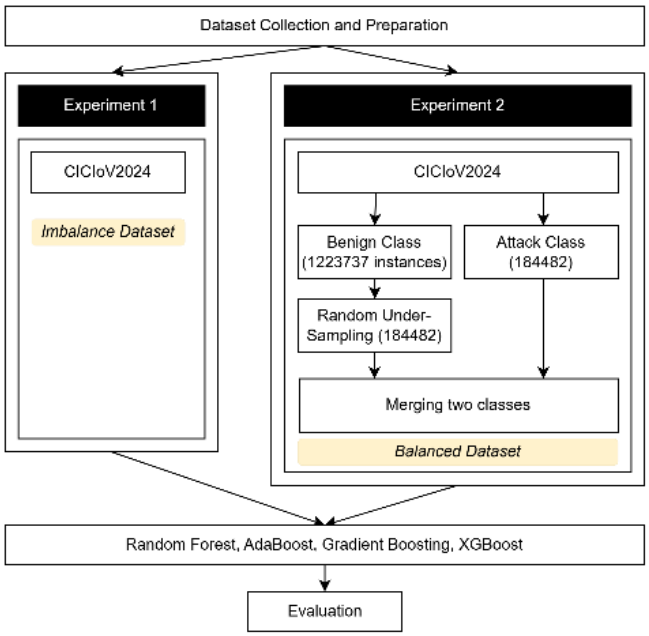

Figure 2.Dataset Preparation

C. Data Pre-processing

Data preprocessing is a crucial step in data analysis aimed at improving the quality and consistency of the dataset before applying it to machine learning models [23]. In this study, preprocessing was necessary to address the challenges posed by the CICIoV2024 dataset, which exhibits a highly imbalanced class distribution. With proper preprocessing, the resulting model can avoid producing biased and unrepresentative outcomes, ultimately reducing the accuracy and effectiveness of the analysis [24].

The CICIoV2024 dataset consists of six files covering various types of attacks, with over 1 million rows of data. The benign class dominates with 1,223,737 rows, while other classes, such as gas spoofing, have only 9,991 rows. The significant imbalance in the number of instances across classes highlights their vast disparity, which may lead the model to become biased toward the majority class. A data balancing process was conducted using the RUS method to address this issue.

RUS is an effective technique for handling class imbalances in datasets. It randomly reduces the instances in the majority class to equal the number of instances in the minority class [16]. The RUS method was implemented using "Orange" software, with the fixed data sample widget set on the file with the benign class to 184,482 instances. Subsequently, the six data files were combined using the concatenate widget. After merging, the id, category, and specify_class columns were removed, leaving only the data columns from DATA_0 to DATA_7. Data normalization was performed using the min-max normalization method as the final step in the preprocessing process. This method transforms feature values into the range [0, 1], ensuring all features have a uniform scale.

In this study, two experiments were conducted to evaluate the performance of machine learning models. The first experiment used an imbalanced dataset, where the benign class significantly dominated. The second experiment used a balanced dataset created with the RUS method. The objective of these two experiments was to compare the model's performance under both conditions and determine whether balancing the data could improve model accuracy

D. Machine Learning Modelling

This study will compare four classification algorithms to evaluate their performance and determine the best algorithm. The four algorithms to be analyzed are Random Forest, Adaptive Boosting, Gradient Boosting, and Extreme Gradient Boosting.

The Random Forest algorithm is an ensemble method that builds multiple decision trees using random subsets of the training data, then combines their predictions through majority voting (for classification) or averaging (for regression) to enhance accuracy [25]. Random Forest was selected for its ability to address overfitting, which often occurs in single decision trees, and to handle data with numerous or complex variables [26]. One of the main advantages of Random Forest is its effectiveness in dealing with imbalanced datasets [27]. This algorithm leverages random subset selection and voting to make the model fairer to minority classes and reduce bias toward majority classes [27].

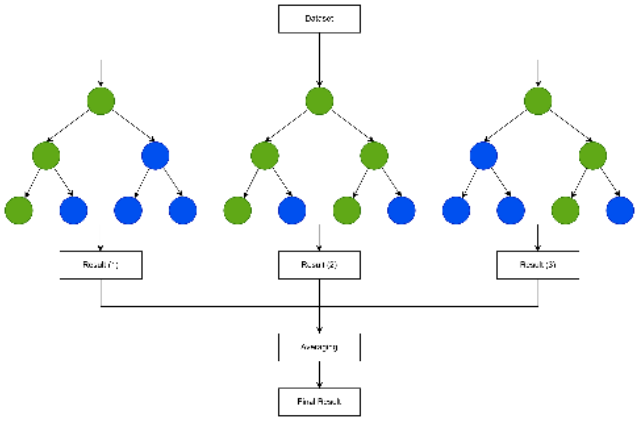

Figure 3.Illustration of Random Forest

The Adaptive Boosting algorithm, or AdaBoost, is a machine learning technique that merges several simple models to create a more robust model [28]. This algorithm works by assigning higher weights to data that are difficult to classify, so each new model is built based on the errors of the previous model [29]. The main advantage of AdaBoost is its ability to improve accuracy without causing overfitting, especially on data with low noise [30]. This algorithm was chosen for its ease of implementation and effectiveness, mainly when used on imbalanced data [31].

The Gradient Boosting algorithm is a machine learning method that incrementally builds predictive models by combining multiple simple models, typically decision trees [32]. This algorithm works by minimizing a loss function using a gradient-based approach, where each new model is created to reduce the prediction errors of the previous model [32]. The main advantage of Gradient Boosting is its ability to handle complex data and deliver accurate predictions [33]. This algorithm is often chosen for its flexibility in handling data with uneven distributions and outliers [34].

The Extreme Gradient Boosting (XGBoost) algorithm is an ensemble-based machine learning algorithm that uses boosting techniques to improve prediction accuracy [35]. Unlike Gradient Boosting, XGBoost optimizes computation through parallelism and better memory management, enabling it to handle large datasets more efficiently Additionally, XGBoost incorporates L1 and L2 regularization, which helps prevent overfitting, making it more robust for complex data [36]. This algorithm is also designed to handle imbalanced data and sparsity more effectively, making it a stronger choice for large and diverse datasets [37].

E. Machine Learning Evaluation

The evaluation assesses how well a model can predict or classify new data based on the training data. The assessment ensures the model performs well on known data and can generalize to unseen data. This study used a cross-validation method with a value of k=10 as the initial step of the evaluation process. The 10-fold cross-validation method divides the dataset into ten subsets or "folds." This method seeks to repeatedly partition the data into testing and training sets, offering a more precise assessment of model performance and minimizing the likelihood of overfitting. In each iteration, 9-fold are used as training data, while 1-fold is used as validation data. This process is repeated ten times, with each fold used as the validation data once. After ten iterations, the average results from each iteration are used to provide an overall view of the model's performance. A visual illustration of the cross-validation process is shown in Figure 4.

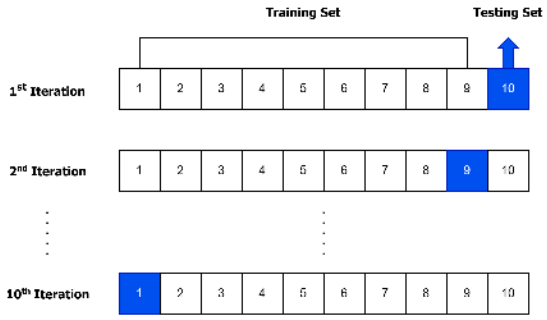

Figure 4.Illustration of 10-Fold Cross-Validation

The next step is to evaluate the model using various appropriate metrics. This study will assess the model using two categories of evaluation metrics: effectiveness and efficiency. Effectiveness metrics, such as accuracy, precision, recall, and F1-score, focus on how well the model performs

prediction or classification tasks. Meanwhile, efficiency metrics measure the "cost" or "time" required to achieve these results, including train and test times.

Accuracy is a metric used to evaluate the overall correctness of a model's predictions. In the accuracy formula (Equation 1), TP (True Positive) represents correct predictions for positive examples, while TN (True Negative) refers to accurate predictions for negative examples. False Positive (FP) happens when the model mistakenly classifies an instance as positive, while False Negative (FN) occurs when the model incorrectly classifies an instance as unfavorable. In the context of attack detection in the Internet of Vehicles (IoV), accuracy provides an overview of how effectively the model identifies benign or attack instances. Accuracy is calculated using the formula in Equation (1):

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

In addition to accuracy, precision is a metric used to measure the accuracy of the model's optimistic predictions. Precision indicates the extent to which the model correctly predicts whether an instance belongs to the benign or attack class. Precision is calculated using the formula in Equation (2):

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

Recall, also known as sensitivity, is a metric that measures how well the model can detect positive instances out of all existing positive instances. Recall evaluates how effectively the model can detect actual cyber threats between benign and attack classes, representing the ratio of threats successfully detected to the total threats present. Recall is calculated using the formula in Equation (3):

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

Finally, the F1-score is a metric that integrates precision and recall into a single value. The F1-score helps evaluate the model more balanced, ensuring that the model not only focuses on detecting all threats (high recall) but also maintains detection quality by reducing false alarms (high precision). The F1-score is determined as the harmonic mean of precision and recall, as shown in the formula in Equation (4):

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

In addition to predictive performance measured by accuracy, precision, recall, and F1-score, computational efficiency is also a crucial consideration, especially in real-world applications that require fast processing times. Therefore, this study also evaluates training time (train time) and testing time (test time) to assess how efficiently the model performs in large datasets.

Train time measures the time required to train the model from start to finish using the training data. Train time is an important indicator when applying the model to large datasets, especially when the methods involve numerous or complex iterations, as in Gradient Boosting and XGBoost algorithms. Training time is calculated using a time function that measures the duration of model training in seconds or minutes. The formula for calculating training time is provided in Equation (5):

$$Train\ Time = t_{end\ train} - t_{start\ train} \quad (5)$$

Test time measures the time the model requires to predict outcomes on test data. Test time is significant when the model is applied in applications requiring fast or real-time predictions, such as Internet of Vehicles (IoV) threat detection. Testing time is calculated in the same manner as training time. The formula for calculating testing time is provided in Equation (6):

$$Test\ Time = t_{end\ test} - t_{start\ test} \quad (6)$$

By measuring train time and test time and comparing the models applied to balanced and imbalanced datasets, conclusions can be drawn about how effective the balancing methods are in improving the model's performance and efficiency. The use of a balanced dataset is expected to enhance the model's performance on minority classes and maintain or improve computational efficiency.

III. Result and Discussion

The CICIoV2024 dataset exhibits a significant imbalance, with the benign class containing 1,223,737 instances compared to 184,482 instances in the attack classes. This imbalance results in a ratio of 6.63:1, heavily biasing model predictions toward the majority class and reducing detection rates for the minority class (Figure 5). Under these conditions, machine learning models often fail to identify rare but critical patterns in the minority class, significantly affecting their ability to detect cyberattacks.

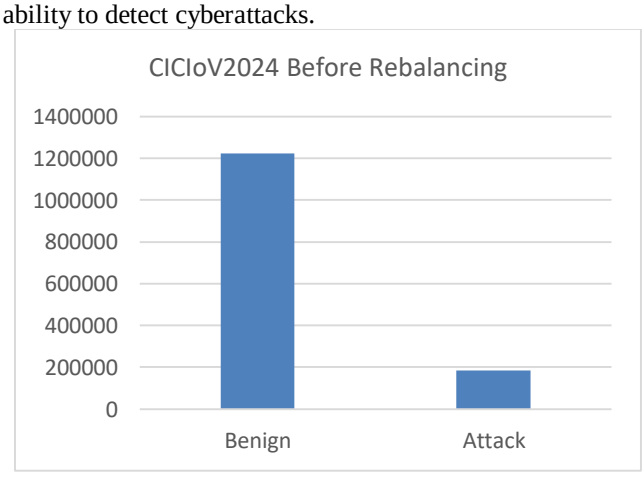

Figure 5.Comparison Between the Benign Class and the Attack Class

To address this issue, the Random Under-Sampling (RUS) method was applied. This method reduces the number of instances in the benign class to match the size of the attack class, resulting in a balanced distribution across all classes. After applying RUS, the dataset was restructured, with each class containing an equal number of 184,482 instances. This balancing ensures that all classes are equally represented during the training process, enhancing the model's ability to detect patterns in the minority class. A visualization of the balanced dataset is presented in Figure 6, clearly showing the equal distribution of instances across all classes.

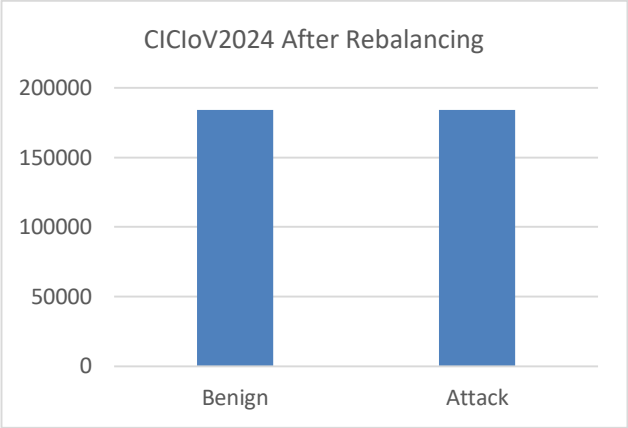

Figure 6. Comparison of the Dataset After Balancing

TABLE II
EFFECTIVENESS PERFORMANCE BEFORE REBALANCING

Model	Acc	Precision	Recall	F1-Score
Random Forest [9]	0.96	0.76	0.76	0.76
XGBoost	1	1	1	1
AdaBoost [9]	0.92	0.48	0.66	0.51
Gradient Boosting	1	1	1	1

Before balancing, the performance of the machine learning models on the CICIoV2024 dataset varied significantly across effectiveness metrics. Random Forest achieved an accuracy of 0.96, indicating its ability to correctly classify the majority of instances. However, its precision, recall, and F1-score were limited to 0.76, suggesting that the model struggled to accurately identify and generalize patterns in the minority class. Similarly, AdaBoost demonstrated a high accuracy of 0.92, but its recall (0.66) and F1-score (0.51) revealed a significant bias toward the majority class. This discrepancy between high accuracy and lower recall and F1-score highlights the limitations of these models in detecting rare attack patterns, as accuracy primarily reflects the dominance of the benign class in the dataset.

In contrast, Gradient Boosting and XGBoost achieved perfect scores (1.00) across all effectiveness metrics, but this result likely reflects the influence of overfitting on the imbalanced dataset. The dominance of the benign class may have allowed these models to correctly classify most instances by focusing on the majority class, resulting in seemingly flawless performance metrics. However, such metrics do not necessarily indicate effective minority class detection, especially in real-world scenarios where data distribution is often unpredictable.

TABLE III
EFFICIENCY PERFORMANCE AFTER REBALANCING

Model	Train Time	Test Time
Random Forest [9]	278.678	15.665
XGBoost	32.271	0.683
AdaBoost [9]	439.548	16.937
Gradient Boosting	833.339	3.678

From an efficiency perspective, significant variation was observed across the models. AdaBoost exhibited the longest training time at 439.548 seconds, followed by Gradient Boosting at 833.339 seconds, which was the slowest model in the experiment. Random Forest required 278.678 seconds for training, while XGBoost was notably more efficient with a training time of only 32.271 seconds. In terms of testing time, XGBoost maintained its efficiency at 0.683 seconds, outperforming Gradient Boosting (3.678 seconds), Random Forest (15.665 seconds), and AdaBoost (16.937 seconds). These results indicate that while some models demonstrate better computational efficiency, others require extensive resources, limiting their applicability in real-time scenarios.

TABLE IV
EFFECTIVENESS PERFORMANCE AFTER REBALANCING

Model	Acc	Precision	Recall	F1-Score
Random Forest [9]	1	1	1	1
XGBoost	1	1	1	1
AdaBoost [9]	1	1	1	1
Gradient Boosting	1	1	1	1

The application of Random Under-Sampling (RUS) brought significant changes to the performance metrics of all models evaluated on the CICIoV2024 dataset. Before balancing, the models demonstrated varied effectiveness and efficiency, with significant limitations in their ability to detect minority class patterns. For instance, Random Forest achieved an accuracy of 0.96 but had lower precision, recall, and F1-scores at 0.76, reflecting bias toward the majority class. Similarly, AdaBoost exhibited a high accuracy of 0.92 but struggled with recall and F1-score, indicating its difficulty in recognizing attack patterns effectively. Gradient Boosting and XGBoost, while achieving perfect scores before balancing, likely overfit the imbalanced dataset by relying heavily on patterns from the dominant benign class.

After balancing the dataset using RUS, all models achieved perfect scores (accuracy, precision, recall, and F1-score of 1.00), indicating their complete ability to classify both benign and attack classes. This improvement suggests that balancing successfully eliminated bias toward the majority class, allowing models to focus equally on both classes. Additionally, the reduction in data imbalance likely simplified the learning process, enabling models like Random Forest and AdaBoost to perform at the same level as XGBoost and Gradient Boosting.

However, while the perfect metrics are noteworthy, they also raise concerns about the generalizability of these results. The balanced dataset may have introduced a level of uniformity that does not reflect real-world IoV scenarios, where data often contain noise and diverse attack patterns. This uniformity could lead to overfitting, especially for models like XGBoost and Gradient Boosting, which are sensitive to data characteristics. Thus, while the balanced dataset improves performance within this controlled experiment, further testing on more complex or noisy datasets is necessary to validate these findings.

TABLE V
EFFICIENCY PERFORMANCE AFTER REBALANCING

Model	Train Time	Test Time
Random Forest [9]	43.047	3.142
XGBoost	5.610	0.092
AdaBoost [9]	139.590	5.031
Gradient Boosting	189.807	0.953

From an efficiency perspective, balancing the dataset reduced the computational demands for all models. For instance, the training time for Random Forest decreased from 278.678 seconds to 43.047 seconds, while XGBoost's already efficient training time improved further to 5.610 seconds. Gradient Boosting, although maintaining high computational costs (189.807 seconds for training), demonstrated improved test time efficiency at 0.953 seconds compared to its pre-balancing performance. These improvements highlight the dual benefits of balancing: enhanced predictive performance and reduced computational complexity.

IV. CONCLUSION

This study utilized the RUS technique to balance the CICIoV2024 dataset and reduce bias toward the majority class. Experimental results showed that this balancing significantly improved model performance, with accuracy, recall, precision, and F1-score achieving perfect values across all algorithms. Balancing using RUS also reduced train and test time, leading to enhanced computational efficiency. Therefore, RUS effectively improves model performance on imbalanced datasets, and this method holds potential as a relevant solution for improving security in IoV applications.

REFERENCES

[1] S. M. Hussain, K. M. Yusof, R. Asuncion, S. A. Hussain, and A. Ahmad, "An Integrated Approach of 4G LTE and DSRC (IEEE 802.11p) for Internet of Vehicles (IoV) by Using a Novel Cluster-Based Efficient Radio Interface Selection Algorithm to Improve Vehicular Network (VN) Performance," in *Sustainable Advanced Computing*, vol. 840, S. Aurelia, S. S. Hiremath, K. Subramanian, and S. Kr. Biswas, Eds., in Lecture Notes in Electrical Engineering, vol. 840. , Singapore: Springer Singapore, 2022, pp. 569–583. doi: 10.1007/978-981-16-9012-9_46.

[2] C. Abdelaziz Kerrache, M. Amadeo, S. H. Ahmed, and C. Liang, "Future Internet of Vehicles," Trans. Emerg. Telecommun. Technol., vol. 31, no. 5, p. e3975, May 2020, doi: 10.1002/ett.3975.

[3] T. Guan, Y. Han, N. Kang, N. Tang, X. Chen, and S. Wang, "An Overview of Vehicular Cybersecurity for Intelligent Connected Vehicles," Sustainability, vol. 14, no. 9, p. 5211, Apr. 2022, doi: 10.3390/su14095211.

[4] H. Taslimasa, S. Dadkhah, E. C. P. Neto, P. Xiong, S. Ray, and A. A. Ghorbani, "Security issues in Internet of Vehicles (IoV): A comprehensive survey," Internet Things, vol. 22, p. 100809, Jul. 2023, doi: 10.1016/j.iot.2023.100809.

[5] J. Asharf, N. Moustafa, H. Khurshid, E. Debie, W. Haider, and A. Wahab, "A Review of Intrusion Detection Systems Using Machine and Deep Learning in Internet of Things: Challenges, Solutions and Future Directions," Electronics, vol. 9, no. 7, p. 1177, Jul. 2020, doi: 10.3390/electronics9071177.

[6] P. Vanin et al., "A Study of Network Intrusion Detection Systems Using Artificial Intelligence/Machine Learning," Appl. Sci., vol. 12, no. 22, p. 11752, Nov. 2022, doi: 10.3390/app122211752.

[7] Z. Jiang, K. Zhao, R. Li, J. Zhao, and J. Du, "PHYAlert: identity spoofing attack detection and prevention for a wireless edge network," J. Cloud Comput., vol. 9, no. 1, p. 5, Dec. 2020, doi: 10.1186/s13677-020-0154-7.

[8] J. Nagarajan et al., "Machine Learning based intrusion detection systems for connected autonomous vehicles: A survey," Peer--Peer Netw. Appl., vol. 16, no. 5, pp. 2153–2185, Sep. 2023, doi: 10.1007/s12083-023-01508-7.

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

[10] P. Dey and D. Bhakta, "A New Random Forest and Support Vector Machine-based Intrusion Detection Model in Networks," Natl. Acad. Sci. Lett., vol. 46, no. 5, pp. 471–477, Oct. 2023, doi: 10.1007/s40009-023-01223-0.

[11] S. Salmi and L. Oughdir, "Performance evaluation of deep learning techniques for DoS attacks detection in wireless sensor network," J. Big Data, vol. 10, no. 1, p. 17, Feb. 2023, doi: 10.1186/s40537-023-00692-w.

[12] E. S. Ali et al., "Machine Learning Technologies for Secure Vehicular Communication in Internet of Vehicles: Recent Advances and Applications," Secur. Commun. Netw., vol. 2021, pp. 1–23, Mar. 2021, doi: 10.1155/2021/8868355.

[13] "IoV Dataset 2024 | Datasets | Research | Canadian Institute for Cybersecurity | UNB." Accessed: Oct. 31, 2024. [Online]. Available: https://www.unb.ca/cic/datasets/iov-dataset-2024.html

[14] L. Dube and T. Verster, "Enhancing classification performance in imbalanced datasets: A comparative analysis of machine learning models," Data Sci. Finance Econ., vol. 3, no. 4, pp. 354–379, 2023, doi: 10.3934/DSFE.2023021.

[15] T. H. M. Le and M. A. Babar, "Mitigating Data Imbalance for Software Vulnerability Assessment: Does Data Augmentation Help?," Jul. 15, 2024, arXiv: arXiv:2407.10722. Accessed: Oct. 31, 2024. [Online]. Available: http://arxiv.org/abs/2407.10722

[16] 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.

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

[18] M. A. Ferrag, O. Friha, D. Hamouda, L. Maglaras, and H. Janicke, "Edge-IIoTset: A New Comprehensive Realistic Cyber Security Dataset of IoT and IIoT Applications for Centralized and Federated Learning," IEEE Access, vol. 10, pp. 40281–40306, 2022, doi: 10.1109/ACCESS.2022.3165809.

[19] F. A. Rafrastara, C. Supriyanto, C. Paramita, Y. P. Astuti, and F. Ahmed, "Performance Improvement of Random Forest Algorithm for Malware Detection on Imbalanced Dataset using Random Under-Sampling Method," J. Inform. J. Pengemb. IT, vol. 8, no. 2, pp. 113–118, May 2023, doi: 10.30591/jpit.v8i2.5207.

[20] 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.

[21] V. Kumar, A. Kumar, S. Garg, and S. R. Payyavula, "Boosting Algorithms to Identify Distributed Denial-of-Service Attacks," J.

Phys. Conf. Ser., vol. 2312, no. 1, p. 012082, Aug. 2022, doi: 10.1088/1742-6596/2312/1/012082.
[22] F. A. Rafrastara, C. Supriyanto, A. Amiral, S. R. Amalia, M. D. Al Fahreza, and F. Ahmed, "Performance Comparison of k-Nearest Neighbor Algorithm with Various k Values and Distance Metrics for Malware Detection," *J. MEDIA Inform. BUDIDARMA*, vol. 8, no. 1, p. 450, Jan. 2024, doi: 10.30865/mib.v8i1.6971.
[23] K. Maharana, S. Mondal, and B. Nemade, "A review: Data pre-processing and data augmentation techniques," Glob. Transit. Proc., vol. 3, no. 1, pp. 91–99, Jun. 2022, doi: 10.1016/j.gltp.2022.04.020.
[24] A. Moscovich and S. Rosset, "On the cross-validation bias due to unsupervised pre-processing," J. R. Stat. Soc. Ser. B Stat. Methodol., vol. 84, no. 4, pp. 1474–1502, Sep. 2022, doi: 10.1111/rssb.12537.
[25] S. K. Wildah, A. Latif, A. Mustopa, S. Suharyanto, M. S. Maulana, and A. Sasongko, "Klasifikasi Penyakit Daun Kopi Menggunakan Kombinasi Haralick, Color Histogram dan Random Forest," J. Sist. Dan Teknol. Inf. JustIN, vol. 11, no. 1, p. 35, Jan. 2023, doi: 10.26418/justin.v11i1.60985.
[26] L. Mentch and S. Zhou, "Randomization as Regularization: A Degrees of Freedom Explanation for Random Forest Success," Aug. 2020.
[27] F. O. Aghware et al., "Enhancing the Random Forest Model via Synthetic Minority Oversampling Technique for Credit-Card Fraud Detection," J. Comput. Theor. Appl., vol. 1, no. 4, pp. 407–420, Mar. 2024, doi: 10.62411/jcta.10323.
[28] W. Fan, Z. Ding, R. Huang, C. Zhou, and X. Zhang, "Improved AdaBoost for virtual reality experience prediction based on Long Short-Term Memory network," Appl. Comput. Eng., vol. 77, no. 1, pp. 158–163, Jul. 2024, doi: 10.54254/2755-2721/77/20240678.
[29] F. Aziz and B. L. E. Panggabean, "Klasifikasi Nasabah Potensial menggunakan Algoritma Ensemble Least Square Support Vector Machine dengan AdaBoost," J. Inform. J. Pengemb. IT, vol. 8, no. 3, pp. 269–274, Sep. 2023, doi: 10.30591/jpit.v8i3.5675.
[30] Z. G. Modarres, M. Shabankhah, and A. Kamandi, "Making AdaBoost Less Prone to Overfitting on Noisy Datasets," in 2020 6th International Conference on Web Research (ICWR), Tehran, Iran: IEEE, Apr. 2020, pp. 251–259. doi: 10.1109/ICWR49608.2020.9122292.
[31] N. Novianti, M. Zarlis, and P. Sihombing, "Penerapan Algoritma Adaboost Untuk Peningkatan Kinerja Klasifikasi Data Mining Pada Imbalance Dataset Diabetes," J. MEDIA Inform. BUDIDARMA, vol. 6, no. 2, p. 1200, Apr. 2022, doi: 10.30865/mib.v6i2.4017.
[32] B. Fuhrer, C. Tessler, and G. Dalal, "Gradient Boosting Reinforcement Learning," Jul. 11, 2024, arXiv: arXiv:2407.08250. Accessed: Oct. 31, 2024. [Online]. Available: http://arxiv.org/abs/2407.08250
[33] P. Messer and T. Schmid, "Gradient Boosting for Hierarchical Data in Small Area Estimation," Jun. 06, 2024, arXiv: arXiv:2406.04256. Accessed: Oct. 31, 2024. [Online]. Available: http://arxiv.org/abs/2406.04256
[34] A. F. Cruz, C. Belém, S. Jesus, J. Bravo, P. Saleiro, and P. Bizarro, "FairGBM: Gradient Boosting with Fairness Constraints," Mar. 03, 2023, arXiv: arXiv:2209.07850. Accessed: Oct. 31, 2024. [Online]. Available: http://arxiv.org/abs/2209.07850
[35] T. Wahyuningsih, A. Iriani, H. D. Purnomo, and I. Sembiring, "Predicting students' success level in an examination using advanced linear regression and extreme gradient boosting," Comput. Sci. Inf. Technol., vol. 5, no. 1, pp. 29–37, Mar. 2024, doi: 10.11591/csit.v5i1.pp29-37.
[36] C. Qin, Y. Zhang, F. Bao, C. Zhang, P. Liu, and P. Liu, "XGBoost Optimized by Adaptive Particle Swarm Optimization for Credit Scoring," Math. Probl. Eng., vol. 2021, pp. 1–18, Mar. 2021, doi: 10.1155/2021/6655510.
[37] W. Chimphlee and S. Chimphlee, "Hyperparameters optimization XGBoost for network intrusion detection using CSE-CIC-IDS 2018 dataset," IAES Int. J. Artif. Intell. IJ-*AI*, vol. 13, no. 1, p. 817, Mar. 2024, doi: 10.11591/ijai.v13.i1.pp817-826.