Jurnal Informatika: Jurnal pengembangan IT

Vol. 10, No. 1, 2025
ISSN : 2477-5126, DOI:10.30591/jpit.v10i1.8034

Optimizing IoV Attack Detection using Random Under Sampling Techniques

Muhammad Alden Nayef Anargya , Wildanil Ghozi , Fauzi Adi Rafrastara
Informatics Engineering Study Program, Faculty of Computer Science, Universitas Dian Nuswantoro, Semarang
Jl. Imam Bonjol No. 207, Semarang, 50131, Indonesia

Info Artikel

Riwayat Artikel:

Received 2024-12-09
Revised 2024-12-19
Accepted 2024-12-22

Corresponding Author:

Wildanil Ghozi

Email:

wildanil.ghozi@dsn.dinus.ac.id

Abstract *– The Internet of Vehicles (IoV) technology is one of the advancements derived from the Internet of Things (IoT) in the transportation sector, benefiting its users. This technology allows various vehicles to connect with each other and share data in real-time to improve transportation efficiency and safety. However, the development of this technology cannot be separated from various security threats, particularly Denial of Service (DoS) and spoofing attacks. System disruptions caused by these attacks can degrade service quality and jeopardize the safety of vehicle users. Given these threats, it is crucial to continuously develop methods used for detecting attacks on IoV systems. Several researchers have conducted research related to attacks and threats on IoV systems, and one such study resulted in a dataset called CICIoV2024. This dataset has an imbalanced class distribution. This study aims to examine the implementation of Random Under-Sampling to improve the performance of classification algorithms in detecting attacks on IoV systems. The algorithms used in this study include Decision Tree, K-Nearest Neighbors (KNN), and Random Forest. The test results show that the Random Forest algorithm achieved the best results with an accuracy of 98.5% and an F1-Score of 98.5%.*

Keywords: *Attack Detection, Internet of Vehicles, Machine Learning, Random Under-Sampling, Random Forest.*

Abstrak *– Teknologi Internet of Vehicles (IoV) merupakan salah satu hasil pengembangan dari Internet of Things (IoT) di bidang transportasi yang bermanfaat bagi penggunanya. Teknologi ini memungkinkan berbagai kendaraan untuk terhubung satu sama lain dan berbagi data secara real-time untuk meningkatkan efisiensi dan keamanan transportasi. Namun, perkembangan teknologi ini tidak dapat terlepas dari berbagai ancaman keamanan, terutama terkait serangan Denial of Services (DoS) dan spoofing. Gangguan pada sistem yang diakibatkan oleh serangan-serangan tersebut dapat menurunkan kualitas layanan dan membahayakan keselamatan pengguna kendaraan. Mengingat ancaman tersebut, sangat penting untuk terus mengembangkan metode yang digunakan dalam mendeteksi serangan pada sistem IoV. Penelitian terkait serangan dan ancaman pada sistem IoV pernah dilakukan oleh beberapa peneliti dan salah satu penelitian tersebut menghasilkan dataset bernama CICIoV2024. Dataset yang dihasilkan tersebut memiliki kelas yang tidak seimbang. Penelitian ini bertujuan untuk menguji implementasi Random Under-Sampling terhadap peningkatan performa algoritma klasifikasi dalam mendeteksi serangan pada sistem IoV. Algoritma yang digunakan dalam penelitian ini yaitu Decision Tree, K-Nearest Neighbors (KNN), dan Random Forest. Hasil pengujian menunjukkan bahwa algoritma Random Forest memperoleh hasil terbaik dengan nilai akurasi 98.5% dan nilai F1-Score 98.5%.*

Kata Kunci*: Deteksi Serangan, Internet of Vehicles, Pembelajaran Mesin, Random Under-Sampling, Random Forest.*

I. INTRODUCTION

The internet network has become an integral part of modern life [1]. Numerous technologies have been developed to facilitate human tasks by utilizing internet connectivity, one of which is IoT (Internet of Things). IoT enables the connection of objects or smart devices with one another, allowing interaction between these devices through the internet [2]. IoT is not only applied to physical devices equipped with sensors, such as mobile phones, household appliances, and other gadgets, but it can also be implemented in vehicles, leading to the emergence of a new technology known as IoV (Internet of Vehicles) [3].

IoV (Internet of Vehicles) is the implementation of IoT (Internet of Things) in the transportation sector [4]. It evolved from the existing Vehicular Ad-Hoc Network (VANET), where the internet connects vehicles with their infrastructure to ensure seamless communication [5]. Several types of communication can be applied within IoV, including vehicle-to-vehicle (V2V), vehicle-to-everything (V2X), vehicle-to-infrastructure (V2I), vehicle-to-surroundings (V2S), and vehicle-to-ecosystem (V2E) [6]. Integrating these various communication types facilitates information exchange, which enhances traffic safety and efficiency [7].

As a technology that relies on internet infrastructure, IoV is not immune to various threats targeting its systems, particularly concerning security and privacy, which can jeopardize the safety of vehicle users in traffic [8]. Various attacks on the system, regardless of whether sophisticated or not, can still compromise the integrity of IoT networks

[9]. Threats to IoV systems, such as Denial of Service (DoS), illusion attacks, and spoofing, have increased over time [10], [11].

According to [12], a Denial of Service (DoS) attack involves sending a large amount of unnecessary data to the system to disrupt its activities, rendering it inaccessible. Due to the high volume of data traffic, IoV system resources such as network bandwidth and CPU are automatically exhausted [13]. As a result, the IoV system fails to receive accurate and organized data, leading to severe consequences.

Spoofing is another attack that disrupts the security and privacy of IoV systems, a passive attack where an intruder impersonates a legitimate device on the network to carry out malicious activities [14]. A component of IoV that is particularly vulnerable to spoofing attacks is GPS, where intruders can falsify location data, potentially causing traffic congestion or accidents for vehicle users [15]. As reported by [16], GPS signal spoofing incidents have occurred over Iraq, particularly near the Iranian border, leading to navigation failures for several aircraft.

A study by [11] focused on attack detection in IoV, producing the CICIoV2024 dataset with two data types, binary and decimal, and a target labelled as specific_class. The study evaluated the performance of four algorithms, that is Logistic Regression, AdaBoost, Deep Neural Network (DNN), and Random Forest (RF). The Deep Neural Network algorithm achieved the highest scores across both data types, with an accuracy of 0.95, an F1-score of 0.63 for binary data, and an accuracy of 0.96 and an F1-score of 0.78 for decimal data. However, the dataset used was imbalanced, which risks drawing incorrect conclusions due to the dominance of a specific class. Additionally, the study did not apply cross-validation during the split between training and testing data, potentially leading to overfitting.

A similar study was conducted by [17] using the same decimal dataset from CICIoV2024 with the target specific_class. Unlike the previous research that employed complex algorithms, this study opted for simpler algorithms, namely Naïve Bayes (NB) and Decision Tree. The results showed that NB achieved the highest scores, with an accuracy of 0.981 and an F1-score of 0.98. However, the study still used imbalanced data, which could lead to analytical errors, such as reduced accuracy reliability and a tendency for the model to predict the majority class.

In the study by [18], the Edge IIoT dataset was used to detect DoS attacks on IoT systems. The research applied Random Under-Sampling (RUS) and SMOTE for class balancing within the dataset. The tested algorithms demonstrated excellent performance, notably the Neural Network algorithm, which achieved an accuracy of 99.62% and an F1-score of 95.59% with SMOTE-based class balancing. On the other hand, the XGBoost algorithm also delivered competitive results, with an accuracy of 98.61% and an F1-score of 95.55% using RUS for class balancing. Despite the high accuracy and F1-scores, these tests required significant computational resources due to the complexity of the algorithms. Additionally, these algorithms are prone to overfitting, especially when the dataset is imbalanced or too small.

A study related to IoT attack detection was conducted by [19] using the IoT-data-driven-defense dataset. The research tested binary and multiclass classification using several machine learning algorithms, including bagging, RF, KNN, J84, Multilayer Perceptron (MLP), and Logistic Regression (LR). Because the dataset was imbalanced, the study employed the Random Under-Sampling (RUS) method to balance the dataset before testing. The results showed that the Random Forest (RF) algorithm achieved impressive performance for binary and multiclass classifications. RF attained an accuracy of 99.59% and an F1-score of 99.5% for binary classification while achieving an accuracy of 98.87% and an F1-score of 98.9% for multiclass classification.

This research aims to evaluate class balancing techniques on machine learning algorithms for detecting attacks that threaten system integrity, particularly Denial of Service (DoS) and spoofing attacks. The study will involve data collection, data preprocessing, and machine learning modelling to evaluate the algorithms. The evaluation of machine learning algorithms will measure their effectiveness in detecting attacks on IoV systems. The findings of this study are expected to contribute to developing a safer and more efficient Internet of Vehicles (IoV) system.

II. METHOD

Several stages will be carried out in this research, as illustrated in Figure 1.

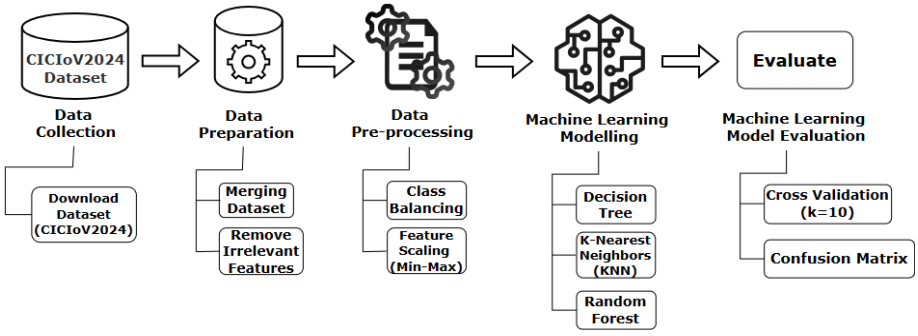


Figure 1. Research Stages

A. Hardware and Software

The research involves a dataset with more than 1 million instances. The role of hardware and software significantly impacts the training process of machine learning models. Higher hardware specifications, combined with appropriate software, can enhance the efficiency and performance of model training, ultimately leading to improved machine learning outcomes [20]. Therefore, the proper selection of hardware and software is vital for enabling successful model training [21].

B. Data Collection and Preparation

The CICIoV2024 dataset is a public dataset developed by the University of New Brunswick (UNB), Canada, in 2024. It was proposed to support research related to developing cybersecurity solutions for the Internet of Vehicles (IoV) using various techniques [22]. The dataset consists of two types, that is binary and decimal. The dataset can be downloaded from UNB's official page, and upon downloading, it provides six separate files. Each file represents specific conditions of the IoV system, named as follows, that is Spoofing-RPM, Spoofing-GAS, Spoofing-STEERING_WHEEL, Spoofing-SPEED, DoS, and Benign. The dataset contains two attack classifications, that is spoofing and Denial-of-Service (DoS), with 11 features and six classes (Speed-Spoofing, Gas-Spoofing, Steering Wheel-Spoofing, RPM-Spoofing, DoS, and Benign).

After downloading the dataset, the next step is data preparation, which includes merging the datasets and removing unused features. The merging process combines the six separate files that have been downloaded into a single dataset.

The next step is to remove features that will not be used, as these features are considered irrelevant to the research being conducted. This step is beneficial because it reduces the complexity and size of the dataset, thereby optimizing the machine learning model's performance [23].

C. Data Pre-processing

The pre-processing stage is crucial in this research as it involves transforming raw data into a form ready for modeling [24]. Therefore, pre-processing is vital in data mining to create more accurate machine learning models and improve performance. The pre-processing steps in this research include class balancing and normalization (feature scaling).

One factor that can influence the performance of a classification prediction model is class imbalance [25]. Imbalanced class data can cause bias in machine learning models, leading to inaccurate results [26]. Therefore, the class balancing step is necessary to address this issue. In this study, Random Under-Sampling (RUS) will be employed as the approach for balancing the classes.

Random Under-Sampling (RUS) is a technique for balancing classes by randomly eliminating some samples from the dominant class until it aligns with the minority class in terms of ratio [25], [27]. This method helps avoid poor classification results caused by noise in the data by eliminating samples from the majority class [28]. RUS is typically used when the majority class significantly dominates the minority class.

Data normalization is essential to eliminate inconsistent value ranges that could cause bias in certain machine learning algorithms and to accelerate the optimization process [29]. It also prepares the data for computation and narrows the range of values [30]. This research employs the Min-Max Normalization method, which scales all data values from their original range to a range between 0 and 1, thereby improving speed and accuracy performance [31]. The formula for Min-Max Normalization can be seen below [32]:

$$\boldsymbol{X_{new}} = \frac{X - \min(X)}{max(X) - min(X)} \quad (1)$$

Where:

- X_{new} = New scaled value
- X = Original value
- Min(X) = Minimum value of the dataset
- Max(X) = Maximum value of the dataset

D. Machine Learning Modelling

This research will assess the performance of three algorithms, including K-Nearest Neighbors (KNN), Decision Tree, and RF, following the pre-processing phase. The algorithm with the best performance among the three will be identified based on the evaluation results.

The Decision Tree algorithm is a data analysis technique that builds a tree-like model to make predictions [33]. It is versatile, capable of handling both classification and regression problems. Additionally, the Decision Tree algorithm is easy to understand because its process resembles real-world decision-making. The structure of the Decision Tree algorithm consists of a root node, internal nodes, and terminal nodes [34]. The root and internal nodes represent variables or features in the dataset, while the terminal nodes represent class labels. An illustration of a Decision Tree can be seen in Figure 2.

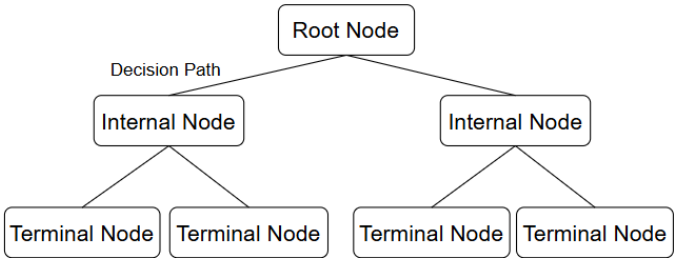


Figure 2. Illustration of Decision Tree

The second algorithm to be tested for its performance is K-Nearest Neighbors (KNN). KNN is a classification method in data mining that categorizes data by considering the closeness of neighboring data points to the target point [35]. The Euclidean distance can be used to calculate the distance between two data points, specifically the point with coordinates (x_1, y_1), and the point with coordinates (x_2, y_2). The formula for Euclidean distance in a 2-dimensional space, which is commonly applied, is as follows:

$$\boldsymbol{d} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (2)$$

To calculate the Euclidean distance between two points in an n-dimensional space, compute the following formula.:

$$\boldsymbol{d(x_{i_1}, x_{i_2})} = \sqrt{\sum_{j=1}^{n} (x_{i_1,j} - x_{i_2,j})^2} \quad (3)$$

The third algorithm is RF. RF is an algorithm that makes predictions by generating decision trees randomly and combining the prediction results from each tree using majority voting (for classification) or averaging (for regression) to produce the final decision [36]. This algorithm can handle complex data with stable results; however, it requires a relatively long processing time due to its high model complexity and lower interpretability [37]. Figure 3 illustrates the Random Forest algorithm [36].

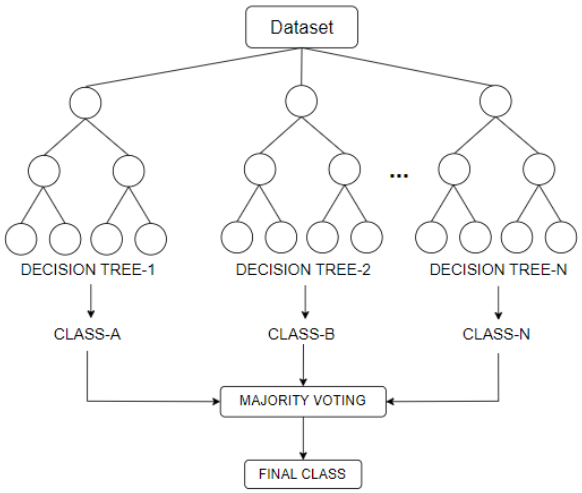


Figure 3. Illustration of Random Forest

E. Machine Learning Model Evaluation

Model performance will be measured and analyzed during the evaluation phase to determine how well the model predicts. Guaranteeing that the machine learning model delivers precise results to solve the issue is essential. To evaluate the model, this study employs the Cross-Validation method to minimize the risk of overfitting. This method uses several folds, precisely 10 (k = 10). Consequently, the dataset will be divided into 10 parts, and testing will be conducted 10 times. In each iteration, one part will be used for testing, while the remaining 9 will be used for training. Each iteration will use a different part as testing data, ensuring all parts are used as testing data exactly once. An illustration of how Cross-Validation works is shown in Figure 4.

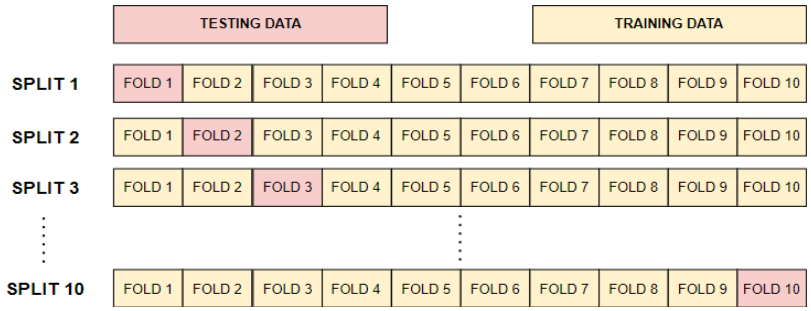


Figure 4. Illustration of Cross Validation

Furthermore, the confusion matrix will be used to evaluate the machine learning model's performance by showing the number of correctly and incorrectly classified test data [38]. The confusion matrix will display four combinations of predicted and actual values, that is True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN), where TP represents the count of positive instances correctly identified as positive, TN refers to the negative instances accurately classified as negative, FP indicates the negative instances mistakenly identified as positive, and FN refers to the positive instances incorrectly classified as negative [39]. Various evaluation metrics derived from the confusion matrix can be used to assess the performance of the trained model. Commonly used metrics include accuracy, precision, recall, and F1-score.

Accuracy measures the proportion of correct predictions relative to the total predictions made by the model. A higher accuracy indicates better performance in correctly predicting the test data. This accuracy is helpful for precisely predicting attacks on IoV systems. The formula for accuracy is as follows:

$$\boldsymbol{Accuracy} = \frac{TP+TN}{(TP+TN+FP+FN)} \qquad (4)$$

Precision is a metric to determine how many positive predictions are correctly classified as positive [40]. It evaluates the accuracy of the model in identifying the positive class. Precision is useful for ensuring that attacks on IoV systems are predicted correctly and not errors. The formula for precision is as follows:

$$\textbf{\textit{Precision}} = \frac{TP}{TP+FP} \quad (5)$$

Recall is a metric used to measure how many positive instances are correctly identified as positive [40]. Recall is helpful in ensuring that the model can detect all attacks directed at the IoV system. The formula for recall is as follows:

$$\textbf{\textit{Recall}} = \frac{TP}{TP+FN} \quad (6)$$

The F1-score metric reflects the balance between precision and recall. Therefore, precision and recall values are required before calculating the F1-score. An imbalance between these two can lead to inaccuracies in the model's predictions. The F1-score is helpful in ensuring the model's balance between correctly detecting attacks (recall) and the predicted attacks' accuracy (precision). The formula for F1-score is as follows:

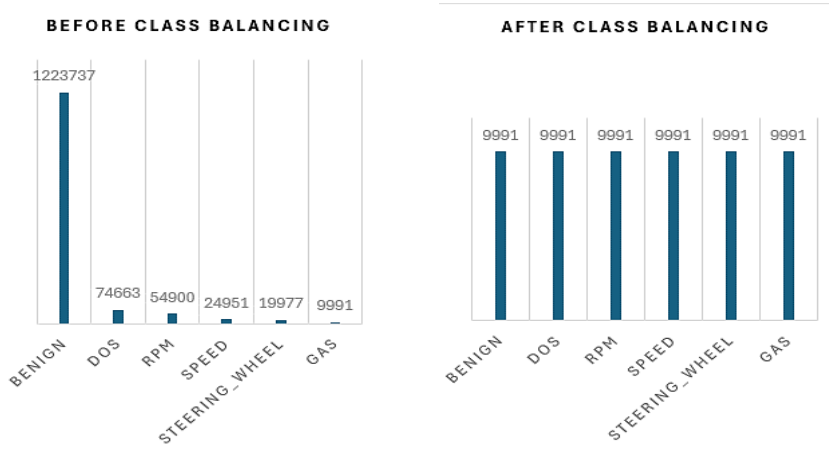
$$\textbf{\textit{F1-score}} = \frac{2\ x\ (precision\ x\ recall\ )}{precision+recall} \quad (7)$$

The evaluation results will be used to compare various models, determine the best model, and identify aspects that need improvement. A well-conducted evaluation will result in a high-quality data mining model and contribute positively to its development.

III. RESULTS AND DISCUSSION

The dataset used in this study is called CICIoV2024, which is available on the official page of the University of New Brunswick (UNB) [22]. The type of dataset downloaded is the decimal dataset, consisting of six separate files. Each file contains 12 columns, with one of the columns named specific_class designated as the target, while the other columns are set as features. The target column includes six classes, that is benign, DoS, GAS, Speed, RPM, and Steering Wheel. The total dataset comprises 1,408,219 instances. Dataset merging will be conducted to combine the six separate files using the concatenate widget in Orange. In the next step, several columns deemed irrelevant for this research, such as ID, Label, and Category, will be removed. These columns are removed to simplify the dataset and decrease its overall size.

There are two steps conducted during the pre-processing stage, that is class balancing and feature scaling (normalization). Because the dataset in this study has imbalanced classes, class balancing will be performed using the Random Under-Sampling (RUS) method. The number of instances will be reduced for several classes, that is BENIGN (1,223,737 instances), DOS (74,663 instances), RPM (54,900 instances), SPEED (24,951 instances), and STEERING_WHEEL (19,977 instances). An illustration of the difference in class distribution before and after balancing is shown in Figure 5.



(a) (b)

Figure 5. Illustration of the distribution difference (a) class distribution before applying the RUS method and (b) class distribution after applying the RUS method

With the implementation of the class balancing step, the dataset, which initially consisted of 1,408,219 instances, was reduced to 59,946 instances. An illustration comparing the number of dataset instances can be seen in Figure 6.

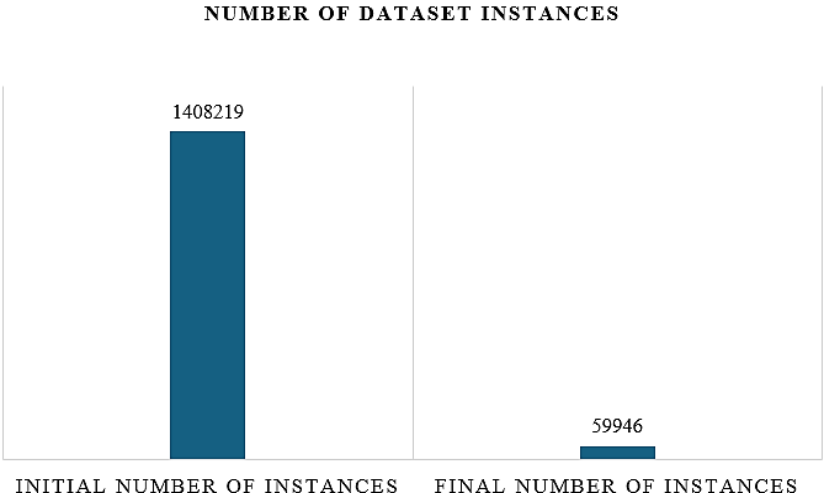


Figure 6. Comparison of the number of dataset instances

Next, in the feature scaling step, data normalization will be applied using the Min-Max method (0,1), where all values in the dataset will be transformed from their original range to a range between 0 and 1. Figure 7 illustrates normalization using the Min-Max method (0,1).

DATA_1	DATA_2
1	2
3	2
3	2
3	2
1	2

(a)

DATA_1	DATA_2
0.00392157	0.00784314
0.0117647	0.00784314
0.0117647	0.00784314
0.0117647	0.00784314
0.00392157	0.00784314

(b)

Figure 7. Illustration of dataset (a) before normalization, and (b) after normalization

After completing the pre-processing stage, the performance of three classification algorithms was tested, that is Decision Tree, KNN, and RF. The KNN algorithm utilized 7 nearest neighbours (k=7) and the Euclidean distance to calculate the distance between two data points, whereas the Random Forest algorithm used 25 decision trees. For the evaluation stage, this study employed Cross-Validation with 10 folds to minimize the risk of overfitting. This method divides the dataset into training and testing data before performing algorithm performance testing. After implementing these three algorithms, testing was conducted to observe their overall performance. The results of the tests are presented in Table 1.

TABLE 1
TESTING RESULTS OF THE PROPOSED METHOD

Algorithm	Accuracy	F1-score	Precision	Recall
Decision Tree	97.7%	97.7%	97.8%	97.7%
KNN	96.6%	96.6%	97.2%	96.6%
Random Forest	98.5%	98.5%	98.6%	98.5%

The testing results of the three classification algorithms, that is Decision Tree, KNN, and Random Forest, will be compared with the results of previous studies to provide a broader context. The comparison of these results can be seen in Table 2.

TABLE 2
COMPARISON WITH STATE OF THE ART

Algorithm	Method	Accuracy	F1-score
Decision Tree	RUS	97.7%	97.7%
KNN	RUS	96.6%	96.6%
Random Forest	**RUS**	**98.5%**	**98.5%**
AdaBoost [11]	-	92%	51%
Random Forest [11]	-	96%	76%
Logistic Regression [11]	-	89%	49%
Deep Neural Network [11]	-	96%	78%
Naive Bayes [17]	-	98.1%	98%
Decision Tree [17]	-	97.5%	97.1%
Logistic Regresion [17]	-	87.6%	84.2%

Based on Table 2, the testing results using the CICIoV2024 dataset indicate that the Random Forest (RF) classification algorithm, which employed the Random Under-Sampling (RUS) method for class balancing, achieved the best performance with an accuracy of 98.5% and an F1-Score of 98.5%. These results surpass those of studies that did not apply the RUS method to address class imbalance, such as the study by [11], where the Random Forest (RF) algorithm achieved only 96% accuracy and an F1-Score of 76%, and the study by [17], where the Naive Bayes (NB) algorithm achieved 98.1% accuracy and an F1-Score of 98%. This demonstrates that implementing the RUS method to handle a class imbalance in the dataset can enhance model performance.

IV. CONCLUSION

This study utilized the CICIoV2024 dataset to test the performance of several classification algorithms, such as Decision Tree, KNN, and RF. Pre-processing steps included class balancing using the Random Under-Sampling (RUS) method and feature scaling with the Min-Max (0,1) method. Cross Validation with 10 folds (k=10) was applied to evaluate the models. Additionally, a confusion matrix was used to assess the performance of the machine learning models. After undergoing these steps, the classification algorithm testing results revealed that the Random Forest algorithm achieved the best scores with an accuracy of 98.5% and an F1-Score of 98.5%. Thus, testing the performance of the Random Forest algorithm with the RUS method to address class imbalance proved effective in enhancing performance, as evidenced by scores that surpassed those of previous studies.

BIBLIOGRAPHY

[1] W. Apriyanti *et al.*, "SOSIALISASI PENGGUNAAN INTERNET YANG SEHAT BAGI ANAK – ANAK DI YAYASAN DOMYADHU," *Abdi Jurnal Publikasi*, vol. 1, no. 1, pp. 13–17, Sep. 2022.
[2] F. Trisnawati, "SEMMUDIK : Selamat Mudik Menggunakan Helm Berbasis Internet of Things (IoT," *JICTEE*, vol. 1, no. 1, Sep. 2020, doi: 10.33365/jictee.v1i1.696.
[3] S. Kumar and J. Singh, "INTERNET OF VEHICLES (IOV) OVER VANETS: SMART AND SECURE COMMUNICATION USING IOT," *Scalable Computing: Practice and Experience*, vol. 21, no. 3, pp. 425–440, 2020.
[4] N. Anwar, D. R. Adhy, N. Widiyasono, R. Hermawan, M. A. Hadi, and M. Tarigan, "Internet of Things ; Model Moda Layanan Sistem Transportasi Internet of Vehicle," *Seminar Nasional Pengaplikasian Telematika (SINAPTIKA)*, vol. 2, no. 1, pp. 51–61, 2022.
[5] S. S. Kanumalli, A. Ch, and P. Sri, "Secure V2V Communication in IOV using IBE and PKI based Hybrid Approach," *IJACSA*, vol. 11, no. 1, pp. 466–472, 2020, doi: 10.14569/IJACSA.2020.0110157.
[6] M. S. Korium, M. Saber, A. Beattie, A. Narayanan, S. Sahoo, and P. H. J. Nardelli, "Intrusion detection system for cyberattacks in the Internet of Vehicles environment," *Ad Hoc Networks*, vol. 153, p. 103330, Feb. 2024, doi: 10.1016/j.adhoc.2023.103330.
[7] E. Zadobrischi and M. Dimian, "Vehicular Communications Utility in Road Safety Applications: A Step toward Self-Aware Intelligent Traffic Systems," *Symmetry*, vol. 13, no. 3, p. 438, Mar. 2021, doi: 10.3390/sym13030438.
[8] S. M. Karim, A. Habbal, S. A. Chaudhry, and A. Irshad, "Architecture, Protocols, and Security in IoV: Taxonomy, Analysis, Challenges, and Solutions," *Security and Communication Networks*, vol. 2022, pp. 1–19, Oct. 2022, doi: 10.1155/2022/1131479.
[9] 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.
[10] N. Tabassum and C. R. K. Reddyy, "Review on QoS and security challenges associated with the internet of vehicles in cloud computing," *Measurement: Sensors*, vol. 27, p. 100562, Jun. 2023, doi: 10.1016/j.measen.2022.100562.
[11] 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.
[12] T. Garg, N. Kagalwalla, P. Churi, A. Pawar, and S. Deshmukh, "A survey on security and privacy issues in IoV," *IJECE*, vol. 10, no. 5, p. 5409, Oct. 2020, doi: 10.11591/ijece.v10i5.pp5409-5419.
[13] K. M. Sai, B. B. Gupta, F. Colace, and K. T. Chui, "A lightweight Anomaly based DDoS flood attack detection for Internet of vehicles," in *CEUR Workshop Proceedings (CEUR-WS.org)*, CEUR Workshop Proceedings, Dec. 2021.
[14] F. Khan *et al.*, "Development of a Model for Spoofing Attacks in Internet of Things," *Mathematics*, vol. 10, no. 19, p. 3686, Oct. 2022, doi: 10.3390/math10193686.
[15] M. Mehta and K. Patel, "Experimental Study of Location Spoofing and Identity Spoofing Attack in Internet of Things Network:," *International Journal of Intelligent Information Technologies*, vol. 18, no. 3, pp. 1–13, Sep. 2022, doi: 10.4018/IJIIT.309587.

[16] K. Lynch, "Ops, Security Experts Alert of GPS Spoofing Near Iran," Aviation International News (AIN). Accessed: Oct. 01, 2024. [Online]. Available: https://www.ainonline.com/aviation-news/air-transport/2023-09-26/ops-security-experts-alert-gps-spoofing-near-iran
[17] F. A. Rafrastara, W. Ghozi, and A. Wardoyo, "Deteksi Serangan berbasis Machine Learning pada Internet of Vehicle," in *Inovasi AI dalam Mewujudkan Kemajuan Berkelanjutan*, Universitas PGRI Semarang: Prosiding Seminar Nasional Informatika, 2024.
[18] B. R. Kikissagbe, M. Adda, P. Célicourt, I. T. Haman, and A. Najjar, "Machine Learning for DoS Attack Detection in IoT Systems," *Procedia Computer Science*, vol. 241, pp. 195–202, 2024, doi: 10.1016/j.procs.2024.08.027.
[19] M. Aljabri *et al.*, "Machine Learning-Based Detection for Unauthorized Access to IoT Devices," *JSAN*, vol. 12, no. 2, p. 27, Mar. 2023, doi: 10.3390/jsan12020027.
[20] S. Shi, Q. Wang, P. Xu, and X. Chu, "Benchmarking State-of-the-Art Deep Learning Software Tools," Feb. 17, 2017, *arXiv*: arXiv:1608.07249. Accessed: Oct. 06, 2024. [Online]. Available: http://arxiv.org/abs/1608.07249
[21] B. Rezaeianjouybari and Y. Shang, "Deep learning for prognostics and health management: State of the art, challenges, and opportunities," *Measurement*, vol. 163, no. 107929, 2020, doi: https://doi.org/10.1016/j.measurement.2020.107929.
[22] "CIC IoV dataset 2024: Advancing Realistic IDS Approaches against DoS and Spoofing Attack in IoV CAN bus." 2024. Accessed: Oct. 06, 2024. [Online]. Available: https://unb.ca/cic/datasets/iov-dataset-2024.html
[23] A. Patle and G. L. Prajapati, "An Effectual Hybrid Feature Selection Method Towards the Classification Performance with Support Vector Machine," *NEUROQUANTOLOGY*, vol. 20, no. 11, pp. 2602–2612, Sep. 2022.
[24] K. Maharana, S. Mondal, and B. Nemade, "A review: Data pre-processing and data augmentation techniques," *Global Transitions Proceedings*, vol. 3, no. 1, pp. 91–99, Jun. 2022, doi: 10.1016/j.gltp.2022.04.020.
[25] J. Prasetya, "Penerapan Klasifikasi Naive Bayes dengan Algoritma Random Oversampling dan Random Undersampling pada Data Tidak Seimbang Cervical Cancer Risk Factors," *LBZ*, vol. 2, no. 2, pp. 11–22, Jul. 2022, doi: 10.59632/leibniz.v2i2.173.
[26] S. Susan and A. Kumar, "The balancing trick: Optimized sampling of imbalanced datasets—A brief survey of the recent State of the Art," *Engineering Reports*, vol. 3, no. 4, p. e12298, Apr. 2021, doi: 10.1002/eng2.12298.
[27] 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," *JPIT*, vol. 8, no. 2, pp. 113–118, May 2023, doi: 10.30591/jpit.v8i2.5207.
[28] N. Rodríguez, D. López, A. Fernández, S. García, and F. Herrera, "SOUL: Scala Oversampling and Undersampling Library for imbalance classification," *SoftwareX*, vol. 15, p. 100767, Jul. 2021, doi: 10.1016/j.softx.2021.100767.
[29] H. A. Ahmed, P. J. Muhammad Ali, A. K. Faeq, and S. M. Abdullah, "An Investigation on Disparity Responds of Machine Learning Algorithms to Data Normalization Method," *ARO*, vol. 10, no. 2, pp. 29–37, Sep. 2022, doi: 10.14500/aro.10970.
[30] V. Putri, "Normalisasi Data Dengan Menggunakan Model Min Max Untuk Klasifikasi Nasabah Potensial Pada Bidang Pembelian Properti Menggunakan Algoritma K-Nearest Neighbor," *Informasi dan Teknologi Ilmiah (INTI)*, vol. 11, no. 3, pp. 111–119, 2024.
[31] A. Ambarwari, Q. J. Adrian, and Y. Herdiyeni, "Analisis Pengaruh Data Scaling Terhadap Performa Algoritme Machine Learning untuk Identifikasi Tanaman," *JURNAL RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 4, no. 1, pp. 117–122, 2020.
[32] H. Henderi, "Comparison of Min-Max normalization and Z-Score Normalization in the K-nearest neighbor (kNN) Algorithm to Test the Accuracy of Types of Breast Cancer," *IJIIS : Int. J. Inform. Inform. Systems.*, vol. 4, no. 1, pp. 13–20, Mar. 2021, doi: 10.47738/ijiis.v4i1.73.
[33] F. M. Hana, "Klasifikasi Penderita Penyakit Diabetes Menggunakan Algoritma Decision Tree C4.5," *siskom-kb*, vol. 4, no. 1, pp. 32–39, Oct. 2020, doi: 10.47970/siskom-kb.v4i1.173.
[34] F. S. Pattiiha and H. Hendry, "Perbandingan Metode K-NN, Naïve Bayes, Decision Tree untuk Analisis Sentimen Tweet Twitter Terkait Opini Terhadap PT PAL Indonesia," *Jur. Ris. Kom.*, vol. 9, no. 2, p. 506, Apr. 2022, doi: 10.30865/jurikom.v9i2.4016.
[35] N. K. K. Ardana *et al.*, "Perbandingan Metode KNN, Naive Bayes, dan Regresi Logistik Binomial dalam Pengklasifikasian Status Ekonomi Negara," *Jambura J. Math*, vol. 5, no. 2, Aug. 2023, doi: 10.34312/jjom.v5i2.21103.
[36] R. Leonardo, J. Pratama, and C. Chrisnatalis, "Perbandingan Metode Random Forest Dan Naïve Bayes Dalam Prediksi Keberhasilan Klien Telemarketing," *JUTIKOMP*, vol. 3, no. 2, pp. 455–459, Oct. 2020, doi: 10.34012/jutikomp.v3i2.1321.
[37] K. Faturrahman and A. Sucipto, "OPTIMASI RANDOM FOREST DENGAN TEKNIK PRUNING UNTUK PREDIKSI DATA NASABAH BMT AL-HIKMAH PERMATA," *Jurnal Informatika Teknologi dan Sains (JINTEKS)*, vol. 6, no. 3, pp. 767–775, Aug. 2024.
[38] D. Normawati and S. A. Prayogi, "Implementasi Naïve Bayes Classifier Dan Confusion Matrix Pada Analisis Sentimen Berbasis Teks Pada Twitter," *Jurnal Sains Komputer & Informatika (J-SAKTI)*, vol. 5, no. 2, pp. 697–711, 2021.
[39] A. R. Purnajaya, R. Jelita, E. Tesvara, M. Nestelrody, and J. Irwansyah, "Penerapan Metode Radial Basis Function (RBF) dalam Mengklasifikasikan Penyakit Demam Berdarah," *Journal of Digital Ecosystem for Natural Sustainability (JoDENS)*, vol. 3, no. 1, pp. 1–4, 2023.
[40] F. Kurniawati and D. Brahma Arianto, "Analisis Implementasi Seleksi Fitur Pada Klasifikasi Diabetes dengan Metode Corellation Matrix dan Algoritma Logistic Regression," *Informatik : Jurnal Ilmu Komputer*, vol. 19, no. 3, pp. 157–164, Dec. 2023, doi: 10.52958/iftk.v19i3.6019.