Jurnal Informatika: Jurnal pengembangan IT

Vol. 10, No. 2, 2025
ISSN : 2477-5126, DOI:10.30591/jpit.v9ix.xxx

Performance Improvement of Machine Learning Algorithm using PCA on IoV Attack

Octaviano Ryan Eka Putra Hartanto [1], Wildanil Ghozi [2], Fauzi Adi Rafrastara [3], Cinantya Paramita [4]
[1,2,3,4]Informatics Engineering Study Program, Faculty of Computer Science, Dian Nuswantoro University, Semarang
Jl. Imam Bonjol No. 207, Semarang, 50131, Indonesia

Info Artikel

Riwayat Artikel:

Received 2024-12-12
Revised 2025-04-30
Accepted 2025-05-03

Corresponding Author:

Wildanil Ghozi

Email:

wildanil.ghozi@dsn.dinus.ac.id

Abstract *– In the transportation industry, the Internet of Vehicles (IoV) is an advancement of the Internet of Things (IoT), allowing automobiles to connect to networks to provide a range of features. This connectivity transforms traditional vehicles into intelligent systems, fostering innovations like autonomous driving and traffic optimization. However, this increased connectivity exposes IoV to cybersecurity threats, particularly because the networks utilized are often public and lack robust security measures. Cyberattacks targeting IoV can involve data packet modification, traffic flooding, or spoofing, potentially disabling critical vehicle components, compromising passenger safety, and increasing the risk of accidents. Consequently, accurate and efficient attack detection systems are essential to counter these threats and ensure IoV security. This study leverages the CICIoV2024 dataset and applies Principal Component Analysis (PCA) to enhance computational efficiency in detecting IoV attacks. The algorithms employed in this research include Random Forest, AdaBoost, Logistic Regression, and Deep Neural Networks. Experimental results demonstrate that implementing PCA significantly improves computational efficiency across all algorithms while maintaining consistent accuracy and F1-Score, highlighting its effectiveness in securing IoV systems.*

Keywords*: Internet of Things; Internet of Vehicles; CICIoV2024; Principal Component Analysis; attack detection.*

Abstrak *– Dalam industri transportasi, Internet of Vehicles (IoV) merupakan pengembangan dari Internet of Things (IoT) yang memungkinkan kendaraan terhubung ke jaringan untuk menyediakan berbagai fitur. Konektivitas ini mengubah kendaraan konvensional menjadi sistem cerdas yang mendukung inovasi seperti kendaraan otonom dan optimalisasi lalu lintas. Namun, peningkatan konektivitas ini juga membuka celah bagi ancaman keamanan siber, terutama karena jaringan yang digunakan sering kali bersifat publik dan tidak memiliki sistem keamanan yang kuat. Serangan siber pada IoV dapat berupa modifikasi paket data, pembanjiran lalu lintas, atau spoofing yang berpotensi merusak komponen kendaraan penting, mengancam keselamatan penumpang, dan meningkatkan risiko kecelakaan. Oleh karena itu, sistem deteksi serangan yang akurat dan efisien sangat dibutuhkan untuk menjaga keamanan IoV. Penelitian ini memanfaatkan dataset CICIoV2024 dan menerapkan metode Principal Component Analysis (PCA) untuk meningkatkan efisiensi komputasi dalam mendeteksi serangan pada IoV. Algoritma yang digunakan meliputi Random Forest, AdaBoost, Logistic Regression, dan Deep Neural Network. Hasil eksperimen menunjukkan bahwa penerapan PCA secara signifikan meningkatkan efisiensi komputasi pada semua algoritma tanpa mengurangi akurasi dan nilai F1-Score, sehingga membuktikan efektivitas pendekatan ini dalam menjaga keamanan sistem IoV.*

Kata kunci*: Internet of Things, Internet of Vehicles, CICIoV2024, Principal Component Analysis, Deteksi Serangan Siber.*

I. INTRODUCTION

The Internet of Things (IoT) is an innovative technological concept that modifies everyday objects to connect to the internet, enabling remote control and monitoring[1]. Over time, advancements in IoT have extended into the transportation sector, resulting in the Internet of Vehicles (IoV). IoV allows vehicles to operate autonomously without manual driving [2]. Furthermore, IoV facilitates data exchange and information gathering with other vehicles and nearby infrastructure, aiming to enhance driving experiences by reducing traffic congestion, improving traffic management, and ensuring road safety [3].

One significant challenge in IoV development is the vulnerability of internet-connected devices and systems to cyberattacks. These attacks can exploit the networks linked to such devices. For instance, Denial of Service (DoS) and spoofing attacks pose risks by disabling vehicle sensors and disrupting device communication [4]. These disruptions can lead to accidents due to compromised sensor systems [5].

Building on Vehicular Ad-Hoc Networks (VANETs), IoV is a branch of IoT, particularly in the automotive area. Vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I), vehicle-to-roadside unit (V2R), vehicle-to-sensors (V2S), and vehicle-to-everything (V2X) are among the various communication paradigms that it integrates [6]. IoV enables vehicles to share critical information, such as accident locations, road hazards, or dangerous areas, with other IoV-enabled vehicles [7].

Security remains a significant concern in IoV implementation. According to [8], fabricated data, denial of service (DoS), eavesdropping, hardware tampering, impersonation, and message suspension are the six categories into which security threats fall. These issues highlight the need for robust network services to remain operational even under attack. Among various types of attacks, Distributed Denial of Service (DDoS) is the most prevalent and effective method to disrupt IoV. It floods networks with large-scale internet traffic, jeopardizing the IoV ecosystem [9][10].

Previous research by Euclides Carlos Pinto Neto [11] on attack detection for IoV utilized the CICIoV2024 dataset [12], dividing the data into binary and decimal formats with a specific_class target. The study evaluated four algorithms: AdaBoost, Random Forest, Deep Neural Network (DNN), and Logistic Regression. The experiments revealed that DNN achieved the highest scores for both binary and decimal data formats, with an accuracy of 0.95 and an F1-score of 0.63 for binary data, and an accuracy of 0.96 with an F1-score of 0.78 for decimal data. However, the study faced challenges with imbalanced data, potentially leading to inaccurate results. Additionally, the absence of cross-validation methods for separating training and testing datasets increased the risk of overfitting.

Similarly, research by Fauzi Adi Rafrastara [4] employed the same dataset, CICIoV2024 [12], focusing on decimal data and the specific_class target. Unlike prior studies using complex algorithms, this research opted for simpler approaches like Naive Bayes and Decision Tree. The results indicated that Naive Bayes achieved the highest accuracy, with an F1-score of 0.981 and an accuracy of 0.98. However, the use of imbalanced data posed limitations, including reduced precision in accuracy results and an increased likelihood of models predicting the majority class.

Another study by Devaj Ramsamooj [13] also utilized the CICIoV2024 dataset [12], emphasizing security issues in VANET, particularly in vehicle-to-infrastructure (V2I) communication, which has received less attention than vehicle-to-vehicle (V2V) communication. The study used GenVRAM, a simulation dataset containing normal scenarios and attack scenarios like bogus messages and black hole attacks. These scenarios involve RSUs behaving abnormally by disseminating false information or blocking critical messages. The research demonstrated that deep learning models outperformed other methods in detecting attacks on VANET systems, especially in identifying anomalous RSUs. The highest accuracy was achieved using supervised K-Nearest Neighbor (KNN), with accuracy rates ranging from 88% to 95%.

Furthermore, a study by Chengpeng Yao [14] developed an anomaly detection method for wireless sensor network (WSN) traffic using Principal Component Analysis (PCA) and Deep Convolutional Neural Network (DCNN). This study employed datasets like KDDCup99, NSL-KDD, and UNSW-NB15. PCA was applied for dimensionality reduction by eliminating redundant features, while DCNN combined traditional convolution with depthwise separable convolution to reduce parameters without compromising feature extraction capabilities. Global Average Pooling was implemented to prevent overfitting, and an attention mechanism replaced pooling layers to retain critical information. The study achieved the highest performance on the KDDCup99 dataset, with an accuracy of 97.83% and an F1-score of 97.97% using DCNN.

This research aims to compare the use of PCA on a model in detecting an attack on IoV. This research starts with dataset collection, pre-processing, building an efficient model, and evaluating with related research. The use of PCA is expected to make the model more efficient in detecting attacks.

II. METHOD

This research was conducted through several main stages, including dataset collection, dataset merging, data preprocessing, modeling, and model evaluation.

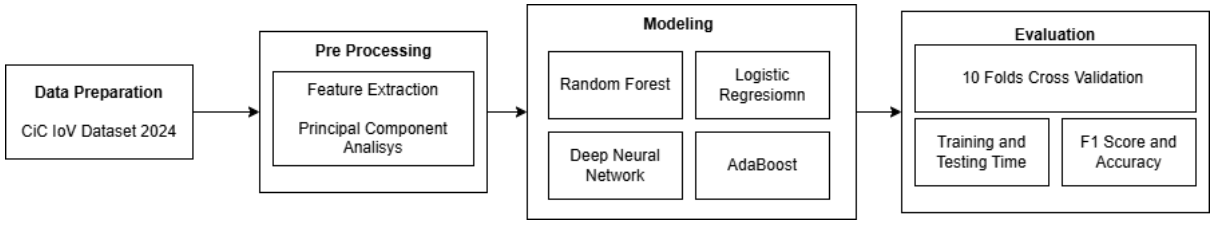

Figure 1.The Process of Developing an IoV Attack Detection Model

A. Hardware

This research will develop a model using a huge amount of datasets. The gear utilized is an Intel Core i7-13620H with 16 GB RAM. Hardware is not constrained, allowing for a more efficient model creation process.

B. Data Preparation

The dataset used in this research is a publicly available dataset provided by the University of Brunswick, Canada, named CIC IoV 2024 [12], which was created in 2024. It contains 1,408,219 records and has 11 features. The dataset is labeled with two primary labels and divided into three main categories. This dataset comprises six separate files, each grouped based on the following specific classes DOS, Spoofing Gas, Spoofing RPM, Spoofing Speed, Spoofing Steering Wheel, dan Benign. The initial step involves merging the datasets from these six files to prepare the data for model development. The merging process aims to create a more structured dataset and facilitate more effective model training. Detailed description of the dataset, is presented in Table (1).

TABLE 1
DESCRIPTION DATASET

Feature Name	Description
ID	specific type of attack class
DATA_0	Byte 0 of the data sent
DATA_1	Byte 1 of the data sent
DATA_2	Byte 2 of the data sent
DATA_3	Byte 3 of the data sent
DATA_4	Byte 4 of the data sent
DATA_5	Byte 5 of the data sent
DATA_6	Byte 6 of the data sent
DATA_7	Byte 7 of the data sent
label	The detection of benign or malicious traffic.
Category	The detection the traffic categorization.
Specific_class	The detection distinct traffic classes.

All the separate dataset files were imported into the Orange software, where the data type for each dataset was checked to ensure no errors. Features DATA_0, DATA_1, DATA_2, DATA_3, DATA_4, DATA_5, DATA_6, and DATA_7 were identified as numerical, while the label was identified as categorical. After importing the data, the next step was to concatenate all datasets. The Concatenate node in Orange was used to merge the six dataset files into a unified dataset. The purpose of this merging process was to combine all relevant variables from each dataset for preprocessing and modelling. This process also removed any duplicate or redundant data to improve accuracy [15]. Next, unnecessary and irrelevant features were removed. Features such as ID, category, and specific_class were eliminated as they were unrelated to the research objective. The study used the label as the target variable for classification modelling. There were two types of labels: BENIGN (normal traffic) and ATTACK (malicious traffic).

C. Pre-Processing

Following data preparation, preprocessing was conducted. This involved feature extraction using Principal Component Analysis (PCA). PCA is a multivariate analysis technique based on linear transformation. It is widely used to reduce data dimensions, extract critical information from large datasets, and analyze the structure of variables [16]. Combining PCA with an appropriate model can produce a robust and adaptable model [17].

D. Modelling

Modelling is a critical stage in building a classification model. In this stage, suitable machine learning algorithms were selected to achieve high accuracy and F1-Score.

1) Random Forest:

Random Forest (RF) is an ensemble method that consists of several independent decision trees, all originating from the same distribution and aggregated through a voting process (the majority voting) to obtain a classification of prediction. RF features minimize correlations, which helps reduce errors when predicting a given problem [18]. The advantages of using a RF model include its ability to handle complex variable dependencies and interactions, as well as its effectiveness in mitigating overfitting. Additionally, RF demonstrates relatively low error rates, improving classification performance, rapidly processing huge volumes of training data, and providing as an excellent approach for predicting missing data [19].

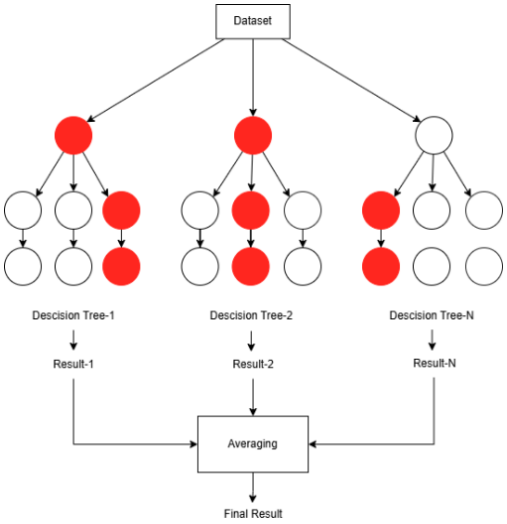

Figure 2.Overview Random Forest Algorithm

2) Logistic Regression:

Logistic Regression (LR) is one of the classification algorithms utilized to discover the relationship between discrete/continuous features and the probability of obtaining a specific discrete outcome, represented by a sigmoid function that models the data [20]. The advantage of using this model is its flexibility in handling both categorical and continuous independent variables [21].

$$h_\theta(X) = \frac{1}{1+e^{-\theta^T X}} \qquad (1)$$

Where:

- e: The base of the natural logarithm
- θ: Parameter vector
- X: Input vector

3) Neural Network:

Neural Network (NN) is a classification algorithm inspired by the functioning of neural networks in the human brain, where each neuron is connected and information flows through each network [22]. The advantages of using this model include its high tolerance for errors, ability to store information across the entire network, capability to function even with incomplete knowledge/data, and its ability to process tasks in parallel [23].

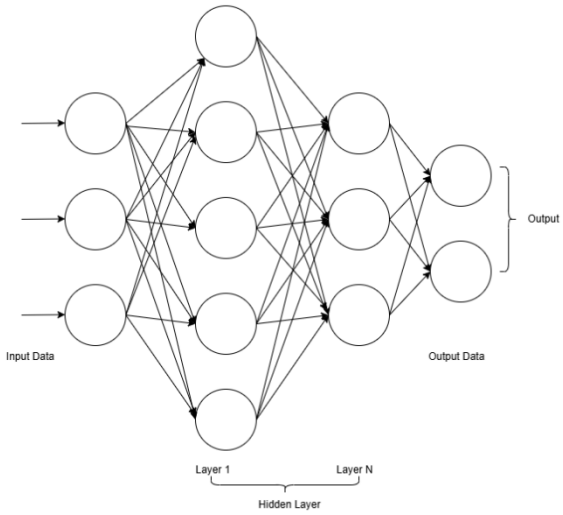

Figure 3.Overview Neural Network Algorithm

4) AdaBoost:

AdaBoost (AB) is an ensemble algorithm that generates classifiers iteratively and combines them to create a strong classifier [24][25]. The advantages of using the AB algorithm include its ability to reduce the gap between bias values and weak classifiers, good generalization skills, and the ability to transform AB output into logarithms using the nearest ratio. AB can also be used for feature selection with a principled strategy (minimizing the upper bound of errors), and it approximates linear decision-making [26]. The AdaBoost equation is shown in the equation below [26] (2).

$$H(x) = \sum_{t=1}^{T} \propto_t h_t(x) \quad (2)$$

Where:

- $h_t(x)$: Weak or base classifier
- $\propto_t$: Learning rate
- $H(x)$: Output result

E. Hyper Parametes

Hyperparameters set the learning process and decide the model's final parameters[27]. Each algorithms have their own parameter. The hyperparameters for each algorithm utilized in this investigation are shown in Table (2).

TABLE 2
ALGORITHM PARAMETER

Model	Parameter
RF	n_estimators = 100
AB	n_estimators = 50 , Learning rate = 1,0 , Algoritma = SAMME.R
LR	C = 1.0
DNN	hidden_layer_sizes = (16,16,16,16), solver = 'adam', alpha = 0.0001 , max_iter = 200

F. Evaluation

Evaluation is the final step in data mining, aimed at assessing the performance of the developed model. The purpose of evaluating a model is to ensure that it is accurate, reliable, and effective in solving the problem at hand. Before evaluating a model, validation is required using either split validation or cross-validation. In this study, cross-validation with a number of folds set to 10 was employed. This method involves randomly dividing the data into multiple folds, enabling the identification of the model with the highest accuracy and determining the optimal model configuration [28]. Using cross-validation, the dataset is divided into 10 equally sized groups. The test process is then conducted iteratively for 10 runs, where each group undergoes nine training sessions and one testing session [29].

After performing cross-validation, the next step is to evaluate the model using a confusion matrix. A confusion matrix is a crucial table used to measure the performance of classification models. It is particularly utilized to evaluate multi-class, single-label classification models [30]. This matrix includes four types of prediction outcomes:

- True Positive (TP) : Data belonging to the positive class is accurately predicted as positive.
- True Negative (TN) : Data belonging to the negative class is accurately predicted as negative.
- False Positive (FP) : Data from the negative class is mistakenly predicted as positive.
- False Negative (FN) : Data from the positive class is mistakenly predicted as negative.

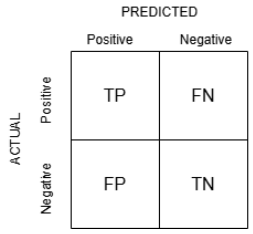

Figure 4.Confusion Matrix

Based on Figure (4), the evaluation metrics derived include accuracy, precision, recall, and F1-score. Accuracy is used to measure the overall correctness of predictions [31]. As represented in the accuracy equation (Equation 2), accuracy provides a general understanding of the model's performance in detecting traffic during IoV attack detection, regardless of whether the traffic is classified as benign or attack.

The F1-score is employed to examine the balance between precision and recall [31]. This metric is particularly useful when there is an imbalance in the dataset, as it provides a harmonic mean of precision and recall, ensuring that both metrics are adequately accounted for in the evaluation process.

$$Accuracy = \frac{TP+TN}{(TP+FP+TN+FN)} \quad (3)$$

$$Precision = \frac{TP}{TP+FP} \quad (4)$$

$$Recall = \frac{TP}{TP+FN} \quad (5)$$

$$F1-Score = \frac{2\times(Precision\times Recall)}{Precision+Recall} \quad (6)$$

In this study, computational costs such as training and testing time were also considered. Testing time refers to the time required to evaluate the model's performance, while training time is the duration needed to train and build the model. Both training and testing times are crucial to ensure the model achieves reliability and can be consistently dependable.

III. RESULTS AND DISCUSSION

The dataset used in this study initially consisted of six separate files. These datasets were merged into a single dataset containing 12 columns with a total of 1,408,219 records. In the model development process, the label was selected as the dependent variable (target). The distribution of the amount of data can be seen in figure (5).

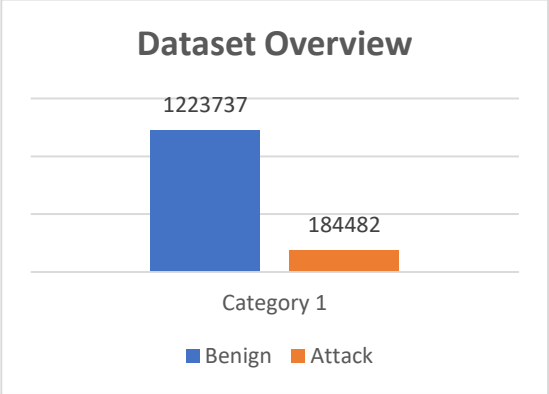

Figure 5.Dataset Overview

The preprocessing step was carried out using the PCA method. PCA works by transforming a set of correlated variables into a new set of uncorrelated variables [32]. In other words, PCA reduces the dimensionality of the data while retaining the variance information from the original dataset [14]. PCA was applied with the expectation of achieving faster testing and training times compared to models built without PCA. This study used a single component (PC1).

The next step involved modelling, which was based on prior research conducted by [11]. The modeling process employed four classification algorithms: Random Forest, AdaBoost, Logistic Regression, and Deep Neural Network. The parameters for each algorithm were adopted precisely as specified in the previous study [11]. The parameters for each algorithm are detailed in Table (2).

The final stage of this study is evaluation, where a cross-validation method with 10 folds was employed to minimize overfitting in the model. The algorithm testing results are presented in Table (3), showcasing both detection effectiveness (accuracy and F1-Score) and computational efficiency (training and testing time). Accuracy and F1-Score metrics are pivotal in the discussion as they relate to the model's predictive accuracy in identifying IoV attacks, classifying them as either attack or benign.

TABLE 3
RESULTS AND COMPARISON OF DETECTION EFFECTIVENESS WITH PREVIOUS STUDIES

Model	Accuracy		F1-Score	
	Non-PCA [11]	PCA	Non-PCA [11]	PCA
RF	1.0	1.0	1.0	1.0
AB	1.0	1.0	1.0	1.0
LR	0.902	0.902	0.898	0.898
DNN	0.98	0.98	0.962	0.962

According to Table (3), the accuracy and F1-Score of models involving PCA are identical to those from previous research [11], which did not utilize PCA. This can be explained by the fact that, while PCA is effective for dimensionality reduction, it may not improve performance when the original features are already highly informative [33]. The models using Random Forest and AdaBoost algorithms achieved an accuracy and F1-Score of 1.0, and Deep Neural Network obtained an accuracy of 0.98 and F1-Score 0.962, while the Logistic Regression model obtained an accuracy of 0.902 and an F1-Score of 0.898. Consequently, this study demonstrates no improvement or degradation in detection effectiveness compared to the previous study [11].

To ensure accurate results for training and testing times, the computational efficiency testing was iterated three times, and the average results were calculated. The differences in computational efficiency are summarized in Table (4).

TABLE 4
RESULTS AND COMPARISON OF COMPUTATIONAL EFFICIENCY WITH PREVIOUS STUDIES

Model	Train Time		Test Time	
	Non-PCA [11]	PCA	Non-PCA[11]	PCA
RF	304,777	276,017	14,257	13,653
AB	30,543	31,937	2,743	2,183
LR	842,613	842,34	6,493	5,693
DNN	10,6	10,41	0,567	0,54

From Table (4), it is evident that applying PCA reduced training time for Random Forest, AdaBoost, and Deep Neural Network models, whereas Logistic Regression experienced an increase in training time. Despite this, PCA implementation increased testing time across all algorithms. Based on this analysis, it can be concluded that PCA enhanced computational efficiency in this study compared to the previous research [11], which did not utilize PCA.

From the results presented in Table (3) and Table (4), it can be concluded that incorporating PCA into the model does not impact the effectiveness of attack detection but significantly enhances computational efficiency. This demonstrates how PCA increases the model's processing efficiency while maintaining detection accuracy. Although there is no distinct advantage in detection metrics such as accuracy and F1-score, dimensionality reduction obtained using PCA can greatly reduce training time and computer resource usage. This is especially useful when working with high-dimensional IoT datasets or distributing models to edge devices with limited computing resources. Previous research has demonstrated that PCA reduces duplication and accelerates convergence in machine learning workflows without necessarily compromising model accuracy when the initial features are already reduced[33].

IV. CONCLUSION

This study utilized the CICIoV2024 dataset, selecting the target from the label column. Experiments were conducted by combining PCA (specifically PC1) with several classification models: Random Forest, AdaBoost, Logistic Regression, and Deep Neural Network. Cross-validation with 10 folds (k=10) was employed for model evaluation in this experiment. After model development, the results indicated that PCA did not affect detection effectiveness but significantly improved computational efficiency. Therefore, the use of PCA has been shown to significantly enhance computational efficiency by 4,43% without compromising detection accuracy. This study's evaluation was done on a particular dataset with a somewhat well-structured feature set, which might not accurately represent more diverse or noisy real-world data.

BIBLIOGRAPHY

[1] F. Susanto, N. Komang Prasiani, and P. Darmawan, "IMPLEMENTASI INTERNET OF THINGS DALAM KEHIDUPAN SEHARI-HARI," Online, 2022. [Online]. Available: https://jurnal.std-bali.ac.id/index.php/imagine

[2] N. Anwar, D. Rosian Adhy, N. Widiyasono, R. Hermawan, M. A. Hadi, and M. Tarigan, "Internet of Things ; Model Moda Layanan Sistem Transportasi Internet of Vehicle," QoS.

[3] E. S. Ali *et al.*, "Machine Learning Technologies for Secure Vehicular Communication in Internet of Vehicles: Recent Advances and Applications," 2021, *Hindawi Limited*. doi: 10.1155/2021/8868355.

[4] F. A. Rafrastara, W. Ghozi, and A. Wardoyo, "Deteksi Serangan berbasis Machine Learning pada Internet of Vehicle," *Seminar Nasional Informatika-FTI UPGRIS*, vol. 2, 2024.

[5] S. Abbas, M. A. Talib, A. Ahmed, F. Khan, S. Ahmad, and D. H. Kim, "Blockchain-based authentication in internet of vehicles: A survey," Dec. 01, 2021, *MDPI*. doi: 10.3390/s21237927.

[6] M. Arif Hakimi Zamrai, K. Mohamad Yusof, M. Afizi Azizan, M. Azam Asri Azman, and S. Mazhar Hussain, "A Survey on Internet of Vehicle (IoV): Applications & Comparison of VANETs, IoV and SDN-IoV," vol. 20, no. 3, pp. 26–31, 2021, [Online]. Available: www.elektrika.utm.my

[7] B. P. Rimal, C. Kong, B. Poudel, Y. Wang, and P. Shahi, "Smart Electric Vehicle Charging in the Era of Internet of Vehicles, Emerging Trends, and Open Issues," *Energies (Basel)*, vol. 15, no. 5, Mar. 2022, doi: 10.3390/en15051908.

[8] J. Li, Z. Xue, C. Li, and M. Liu, "RTED-SD: A real-time edge detection scheme for sybil DDoS in the internet of vehicles," *IEEE Access*, vol. 9, pp. 11296–11305, 2021, doi: 10.1109/ACCESS.2021.3049830.
[9] 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, Feb. 2024, doi: 10.1016/j.adhoc.2023.103330.
[10] B. Kwapong Osibo, C. Zhang, C. Xia, G. Zhao, and Z. Jin, "Security and Privacy in 5G Internet of Vehicles (IoV) Environment," *Journal on Internet of Things*, vol. 3, no. 2, pp. 77–86, 2021, doi: 10.32604/jiot.2021.017943.
[11] E. C. P. Neto *et al.*, "CICIoV2024: Advancing realistic IDS approaches against DoS and spoofing attack in IoV CAN bus," *Internet of Things (Netherlands)*, vol. 26, Jul. 2024, doi: 10.1016/j.iot.2024.101209.
[12] Canadian Institute for Cybersecurity, "CIC IoV Dataset 2024."
[13] D. Ramsamooj, P. Sharma, and H. Liu, "GenVRAM: Dataset Generator for Vehicle to Roadside Attacks and Misbehavior," *IEEE Access*, vol. 12, pp. 86176–86193, 2024, doi: 10.1109/ACCESS.2024.3416840.
[14] C. Yao, Y. Yang, K. Yin, and J. Yang, "Traffic Anomaly Detection in Wireless Sensor Networks Based on Principal Component Analysis and Deep Convolution Neural Network," *IEEE Access*, vol. 10, pp. 103136–103149, 2022, doi: 10.1109/ACCESS.2022.3210189.
[15] J. W. Cho, A. Korchmaros, J. T. Vogelstein, M. P. Milham, and T. Xu, "Impact of concatenating fMRI data on reliability for functional connectomics," *Neuroimage*, vol. 226, Feb. 2021, doi: 10.1016/j.neuroimage.2020.117549.
[16] A. S. Ritonga and I. Muhandhis, "TEKNIK DATA MINING UNTUK MENGKLASIFIKASIKAN DATA ULASAN DESTINASI WISATA MENGGUNAKAN REDUKSI DATA PRINCIPAL COMPONENT ANALYSIS (PCA)."
[17] J. R. Beattie and F. W. L. Esmonde-White, "Exploration of Principal Component Analysis: Deriving Principal Component Analysis Visually Using Spectra," Apr. 01, 2021, *SAGE Publications Inc.* doi: 10.1177/0003702820987847.
[18] M. A. As Sarofi, Irhamah, and A. Mukarromah, "Identifikasi Genre Musik dengan Menggunakan Metode Random Forest," *URNAL SAINS DAN SENI ITS*, vol. 9, no. 1, 2020.
[19] H. Tantyoko, D. Kartika Sari, and A. R. Wijaya, "PREDIKSI POTENSIAL GEMPA BUMI INDONESIA MENGGUNAKAN METODE RANDOM FOREST DAN FEATURE SELECTION," 2023. [Online]. Available: http://jom.fti.budiluhur.ac.id/index.php/IDEALIS/indexHenriTantyoko|http://jom.fti.budiluhur.ac.id/index.php/IDEALIS/index|
[20] G. P. P.B and R. Febryansyah, "KLASIFIKASI PERSETUJUAN PERMOHONAN PINJAMAN PADA KOPERASI SIMPAN PINJAM MENGGUNAKAN ALGORITMA LOGISTIC REGRESSION," 2022.
[21] D. Vitona, A. Fitrianto, and M. Alwi Aliu, "ANALISIS FAKTOR YANG MEMPENGARUHI INDEKS PEMBANGUNAN MANUSIA DI INDONESIA MENGGUNAKAN MODEL REGRESI LOGISTIK BINER," vol. 5, no. 2, 2024, doi: 10.46306/lb.v5i2.
[22] R. Ridwan, H. Lubis, and P. Kustanto, "Implementasi Algoritma Neural Network dalam Memprediksi Tingkat Kelulusan Mahasiswa," *JURNAL MEDIA INFORMATIKA BUDIDARMA*, vol. 4, no. 2, p. 286, Apr. 2020, doi: 10.30865/mib.v4i2.2035.
[23] M. M. Mijwil, "Artificial Neural Networks Advantages and Disadvantages," *Mesopotamian Journal of Big Data*, vol. 2021, pp. 29–31, Aug. 2021, doi: 10.58496/mjbd/2021/006.
[24] A. N. Iman, A. G. Putrada, S. Prabowo, and D. Perdana, "Peningkatan Kinerja AMG8833 sebagai Thermocam dengan Metode Regresi AdaBoost untuk Pelaksanaan Protokol COVID-19," *Jurnal Elektro dan Telekomunikasi Terapan*, vol. 8, no. 1, p. 978, Aug. 2021, doi: 10.25124/jett.v8i1.3894.
[25] F. Syah, H. Fajrin, A. N. Afif, R. Saeputra, D. Mirranty, and D. D. Saputra, "Analisa Sentimen Terhadap Twitter IndihomeCare Menggunakan Perbandingan Algoritma Smote, Support Vector Machine, AdaBoost dan Particle Swarm Optimization," *Jurnal Teknologi Informasi dan Komunikasi)*, vol. 7, no. 1, 2023, doi: 10.35870/jti.
[26] R. T. Febianto, D. Suranti, and R. T. Alinse, "PENERAPAN ALGORITMA ADABOOST DALAM MENGETAHUI POLA PENGGUNA KB DI PUSKESMAS TANJUNG HARAPAN," 2024. [Online]. Available: http://jurnal.goretanpena.com/index.php/JSSR
[27] Y. A. Ali, E. M. Awwad, M. Al-Razgan, and A. Maarouf, "Hyperparameter Search for Machine Learning Algorithms for Optimizing the Computational Complexity," *Processes*, vol. 11, no. 2, Feb. 2023, doi: 10.3390/pr11020349.
[28] R. Rizqi Robbi Arisandi, B. Warsito, and A. Rachman Hakim, "APLIKASI NAÏVE BAYES CLASSIFIER (NBC) PADA KLASIFIKASI STATUS GIZI BALITA STUNTING DENGAN PENGUJIAN K-FOLD CROSS VALIDATION," vol. 11, no. 1, pp. 130–139, 2022, [Online]. Available: https://ejournal3.undip.ac.id/index.php/gaussian/
[29] M. A. N. Anargya, W. Ghozi, and F. A. Rafrastara, "Random Under Sampling for Performance Improvement in Attack Detection on Internet of Vehicles Using Machine Learning," *Jurnal Informatika: Jurnal Pengembangan IT*, vol. 10, no. 1, pp. 11–19, Jan. 2025, doi: 10.30591/jpit.v10i1.8034.
[30] D. Krstinić, M. Braović, L. Šerić, and D. Božić-Štulić, "Multi-label Classifier Performance Evaluation with Confusion Matrix," Academy and Industry Research Collaboration Center (AIRCC), Jun. 2020, pp. 01–14. doi: 10.5121/csit.2020.100801.
[31] K. L. Kohsasih, Z. Situmorang, and I. Artikel, "Analisis Perbandingan Algoritma C4.5 Dan Naïve Bayes Dalam Memprediksi Penyakit Cerebrovascular," *JURNAL INFORMATIKA*, vol. 9, no. 1, pp. 13–17, 2022, [Online]. Available: http://ejournal.bsi.ac.id/ejurnal/index.php/ji
[32] D. Sartika and I. Saluza, "Penerapan Metode Principal Component Analysis (PCA) Pada Klasifikasi Status Kredit Nasabah Bank Sumsel Babel Cabang KM 12 Palembang Menggunakan Metode Decision Tree."
[33] S. Haque, Z. Eberhart, A. Bansal, and C. McMillan, "Semantic Similarity Metrics for Evaluating Source Code Summarization," in *IEEE International Conference on Program Comprehension*, IEEE Computer Society, 2022, pp. 36–47. doi: 10.1145/nnnnnnn.nnnnnnn.