electronics

MDPI

Article

Intrusion Detection System for Autonomous Vehicles Using Non-Tree Based Machine Learning Algorithms

Cynthia Anthony [1], Walid Elgenaidi [2] and Muzaffar Rao [1,*]

[1] Department of Electronic and Computer Engineering, University of Limerick, V94 T9PX Limerick, Ireland; 22027831@studentmail.ul.ie

[2] Jaguar Land Rover (JLR), V14 T863 Shannon, Ireland

[*] Correspondence: muzaffar.rao@ul.ie

Abstract: This research work highlights significant achievements in the domain of intrusion detection systems (IDSs) for autonomous vehicles, which are crucial in enhancing their safety, reliability, and cybersecurity. This study introduces an approach that leverages non-tree-based machine learning algorithms, such as K-nearest neighbors and ensemble learning, to develop an IDS tailored for autonomous vehicles. These algorithms were employed because of their ability to process complex and large datasets with less likeliness for overfitting, their scalability, and their ability to adapt to changing conditions in real time. These algorithms effectively handle imbalanced data, enhancing the detection accuracy of both normal and intrusive instances. The IDS's performance was validated through the utilization of three real-world datasets, CAN intrusion, CICIDS2017, and NSL-KDD, where the proposed non-tree-based IDS (NTB-MTH-IDS) was measured with the standard measurement metrics: accuracy, precision, F1-score, and recall, including specificity and sensitivity. Notably, the results indicate that K-nearest neighbors and stacking, as part of NTB-MTH-IDS, has an accuracy of 99.00%, 98.57%, and 97.57%, and F1-scores of 99.00%, 98.79%, and 97.54% in the CICIDS2017, NSL-KDD, and CAN datasets, respectively. The results of this research can lead to establishing a robust intrusion detection framework, thereby ensuring the safety and reliability of autonomous vehicles. Through this achievement, road users, passengers, and pedestrians are safeguarded against the consequences of potential cyber threats.

Keywords: intrusion detection system (IDS); machine learning; autonomous vehicles; imbalanced data; non-tree-based technique; performance metrics; datasets

check for updates

Citation: Anthony, C.; Elgenaidi, W.; Rao, M. Intrusion Detection System for Autonomous Vehicles Using Non-Tree Based Machine Learning Algorithms. *Electronics* **2024**, *13*, 809. https://doi.org/10.3390/electronics13050809

Academic Editors: Piotr Sulikowski, Vasile-Daniel Pavaloaia and Rodrigo Martin-Rojas

Received: 22 December 2023
Revised: 29 January 2024
Accepted: 16 February 2024
Published: 20 February 2024

1. Introduction

The emergence of cutting-edge technologies, such as smarter and more portable computers, has led to an increase in the popularity of autonomous vehicles [1], and sales of these vehicles have increased at an astonishing rate because of rising gasoline prices and growing worries about climate change [2]. Without a doubt, one of the most pressing global concerns of the present time is climate change [3]. There is a need for precisely calibrated actuators, cutting-edge sensors, and top-notch electronic control units (ECUs) as the automotive industry undergoes fast changes to include the most recent mechanical and communication technologies driven by industry demands [4]. These parts are essential for making sure that vehicles run effectively and safely, and, unfortunately, they can be susceptible to attacks.

The motivation for this research is to contribute to the cutting edge of cybersecurity in autonomous vehicles by researching and proposing state-of-the-art machine learning methodologies. This research seeks to set a new standard for detecting and potentially preventing attacks on autonomous vehicles while safeguarding the integrity and safety of the autonomous vehicle ecosystem.

Intrusion detection is the process of keeping track of and examining network or computer system activity to detect intrusion attempts [5]. The process looks for indications of potential intrusions, which might include efforts to bypass security barriers or obtain unauthorized access [5].

As autonomous vehicles become more functional and interconnected, they also become increasingly vulnerable to cyberattacks that target both internal and external networks due to their large attack surfaces. Autonomous vehicles are susceptible to cyber-attacks due to how heavily they rely on computers and communication networks [6]. These systems include flaws that hackers can use to obtain access without authorization, steal information, or even take over the vehicle control [7]. Accidents, fatalities, and severe property damage may be the results of these attacks.

Figure 1 gives an overview of an IDS for autonomous vehicles. It highlights key functionalities, which include real-time filtering and detection capabilities. It can detect both sensor-based and communication-based attacks, which is very important in safeguarding autonomous vehicles.

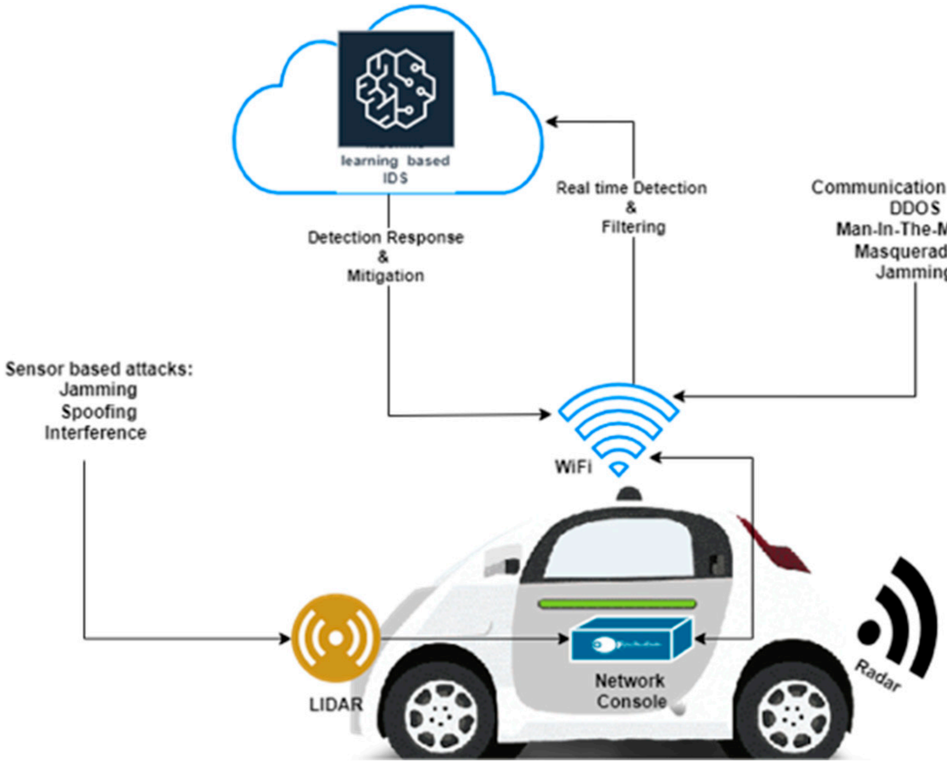
learning based IDS
Real time Detection & Filtering
Communication attacks: DDOS Man-In-The-Middle Masquerading Jamming
Detection Response & Mitigation
Sensor based attacks: Jamming Spoofing Interference
WiFi
LIDAR
Network Console
Radar

Figure 1. Integrated ML-based IDS for autonomous vehicles.

Researchers have focused on developing IDSs that use machine learning (ML) approaches to identify intrusions and safeguard vehicle networks [8]. Therefore, it is crucial to develop an IDS that can recognize and thwart such attempts. The ability of modern, cutting-edge intrusion detection systems for autonomous vehicles to recognize and potentially prevent intrusions is limited as they depend solely on one sort of detection technique, such as anomaly detection or signature-based detection, which is easily evaded by scheming attacks. The broad and intricate makeup of the computer systems and communication networks used by autonomous vehicles is also normally not considered by these intrusion detection systems.

Some existing research [9] proposed a multitier hybrid intrusion detection system (MTH-IDS) to create an IDS for self-driving vehicles. This method integrates several detection techniques, including anomaly detection, signature-based detection, and rule-based detection, to increase detection precision and decrease false detection rates. Additionally, the MTH-IDS technique adopts different tiers of detection algorithms to consider the varied and complex nature of an autonomous vehicle's computer systems and communication networks. To create an improved and effective intrusion detection system based on the

MTH-IDS technique, several issues, such as overfitting, underfitting, and bias, must be resolved. These difficulties include controlling the computing resources needed for the detection methods, creating effective detection algorithms, and balancing the risks of false alarms and accurate detection. Therefore, the goal of this research was to create an improved MTH-IDS approach that can effectively recognize and prevent cyberattacks on autonomous vehicles. This improved approach is referred to as 'NTB-MTH-IDS' in this paper. The proposed IDS takes into consideration the many intricate computer systems and communication networks required by an autonomous vehicle in addition to solving the problems of computing efficiency and detection accuracy. This research employed non-tree-based machine learning techniques as part of an IDS strategy to improve detection accuracy and lower false alarm rates.

The non-tree-based algorithms were selected after an extensive literature review. K-NN, SVM, and naïve Bayes can be computationally efficient, making them suitable for real-world applications in autonomous vehicles, where quick decision making is important. They are able to detect complex patterns and anomalies, which might not be easily detected by tree-based algorithms. Non-tree-based algorithms can handle noise in a robust manner, which is important in real-world scenarios.

Performance was evaluated using standard performance metrics: accuracy, recall, F1-score, and precision, as well as added metrics such as specificity and sensitivity. The goal of this research was to create an optimized IDS and evaluate its effectiveness using information from real-world datasets.

This paper is organized as follows: Section 2 provides a detailed literature review, Section 3 provides details about the data sources and dataset description, Section 4 explains the analytical process and methodology, Section 5 provides the results, Section 6 presents the discussion and Section 7 the conclusion.

2. Literature Review

This section provides a literature review of the latest published work on IDSs for autonomous vehicles. In [9], the authors looked at the weaknesses of both internal and external networks and presented a multi-tiered hybrid IDS that combines a signature-based and an anomaly-based IDS to identify both known and unknown assaults on vehicular networks. The experimental findings demonstrate that the proposed system can identify several known attacks with 99.99% accuracy using the CAN intrusion dataset, which represents data from within the vehicle, and the CICIDS2017 dataset (providing external data on the vehicle). The average processing time of each data packet on a vehicle-level CPU, which is less than 0.6 ms, shows that the proposed system can operate in real time. The given F1-scores (0.963 and 0.800) may be considerably optimized by using other machine learning techniques, and additional performance measures like sensitivity and specificity can be added for more accurate results.

An innovative intrusion detection system (IDS) for CAN was suggested in [10] to discover abnormalities in in-vehicle networks. From the statistical features of assaults, it is possible to infer knowledge about intrinsic incursion patterns and behaviors. The experimental findings showed that denial-of-service (DoS) attacks, such as fuzzy, can be accurately identified by the proposed technique, with minimal false-positive rates. As the attack rate rises, it was also demonstrated that the total error declines for various window widths. The findings showed that the suggested IDS decreases the rate of misclassification and is, hence, more appropriate for in-vehicle networks. According to the experimental results, CAN-BUS systems can efficiently discern between valid and malicious data using the proposed IDS. The IDS scored an F1-score of 99.56% and a recall of 99.64%. By comparing window vectors with reference values of normality, the suggested IDS may identify the presence of an attack on the ECU. However, anomaly diagnosis may be incorrect if the hacked ECU initiates an attack before re-establishing normality values.

In [11], an effective intelligent intrusion detection system (IDS) was developed using machine learning and deep learning techniques, such as the adaptive neuro-fuzzy inference system (ANFIS) and convolutional neural networks (CNNs), respectively. The vehicular ad hoc network (VANET) environment's known attacks are the main focus of the existing approaches. This restriction is lifted by proposing the intelligent IDS and applying soft-computing techniques. Modules called known IDS (KIDS) and unknown IDS (UIDS), which may identify both known and undiscovered threats, are included in the suggested approach. While the UIDS module employs deep learning to find unidentified threats in the VANET, the KIDS module uses ANFIS categorization to identify known destructive assaults. The attack detection rate (ADR) findings showed that 99.7% of port scan attacks, 98.9% of botnet attacks, 93.9% of brute force attacks, and 98.5% of malicious attacks were all detected. The IDS's performance in the VANET was not maximized since deep learning and intelligent key management were not used.

In [12], a novel intrusion detection system based on deep learning that integrates thresholding and error reconstruction methods was proposed. They tested and trained several different neural network topologies while analyzing their performance. The suggested anomaly detection system was put to the test against four distinct types of attacks, denial of service (DoS), fuzzy, RPM spoofing, and gear spoofing, using evaluation criteria including precision, recall, and F1-scores. They also showed reconstruction error distribution charts to provide qualitative knowledge of the proposed system's ability to distinguish between actual and anomalous sequences. In practically every situation, the deep learning-based model performed above 99.90%. With a forecast time of only 128.73 ms, it performed well in terms of prediction times.

The authors of [13] proposed a study that makes use of supervised and semi-supervised datasets to evaluate the effectiveness of each machine learning approach. The semi-supervised dataset was made up entirely of normal data with the anomaly labels intentionally inserted, in contrast with the supervised dataset, which contained both normal and anomalous data. Decision trees, K-nearest neighbors (K-NN), and support vector machines (SVMs) are just a few of the machine learning techniques that the study employed for evaluation. The identification of abnormalities in in-vehicle networks is a pertinent and useful subject in the realm of vehicular cybersecurity that is covered in the study. To understand the benefits and drawbacks of each strategy, the study analyzed the efficiency of supervised and semi-supervised machine learning algorithms in identifying abnormalities. The results of the study demonstrated that the semi-supervised learning method (one-class SVM) outperforms the supervised learning methods (decision tree, random forest, and gradient boosting) in terms of both the detection rate and the false-positive rate. The methodology of the paper was well-designed, and the study employed a variety of machine learning algorithms, making it more robust. However, the lack of a thorough examination of the data in the report makes it challenging to understand the results. The absence of real-world data and the fact that the datasets were artificially manufactured are only two examples of the flaws that were not addressed in this publication. To ascertain whether the findings were in line with state-of-the-art approaches, the study could have compared its findings to those of previous research in the area. Furthermore, neither the paper's computing requirements for the technique nor its potential effects on the performance of the system were covered.

In [14], a novel intrusion detection system (IDS) for CAN to uncover abnormalities in in-vehicle networks was proposed. According to the findings, the in-vehicle networks would benefit more from the suggested IDS since it decreases the rate of misclassification. For XGBoost, the findings from the chi-square and SMOTE techniques were substantial. The mentioned metrics value of FPR was 0.019, and the XGBoost training accuracy was 0.980. The lowering of dimensions was not a topic in the work. Dimensionality reduction may be handled using a deep learning method.

In [15], a unique hybrid method to attack categorization and intrusion detection was proposed for attack categorization and intrusion detection. The recommended technique consists of three phases to manage high false-positive and low false-negative rates. In the first phase, the dataset is pre-processed using the min–max method and data transformation methodology. Second, using the random forest recursive feature removal approach, the best features that improve the performance of the model are identified. The adaptive neuro-fuzzy system (ANFIS) and other forms of support vector machines (SVMs) should then be used to categorize probe, U2R, R2U, and DDOS attacks to identify infiltration. The SVM classification accuracy for the NSL-KDD dataset was 99.3%. Finding data and then effectively recognizing threats are two of the primary issues with the existing method.

In [16], the influence of oversampling procedures on the training sample size of the models was explored, and the smallest practical training sample size was determined. The effectiveness of detection and the complexity of the detection process were also compared, and the implications of information gain and correlation-based feature selection methodologies were given. To enhance the functionality of the NIDS, other ML hyper-parameter optimization techniques were also investigated. The UNSW-NB 2015 and CICIDS2017 datasets (two recent intrusion detection datasets) were utilized to evaluate the effectiveness of the suggested method. The experimental results showed that the proposed model significantly reduces the size of the feature set.

In [17], the authors explored several intrusion detection approaches for detecting and preventing assaults on in-vehicle networks. The approach of the study included a thorough evaluation of the previous research on intrusion detection for in-vehicle networks. The emergence of self-driving vehicles has increased security concerns in in-vehicle networks (IVNs), as well as a lack of cybersecurity considerations in present IVN designs. Intrusion detection systems (IDSs) can efficiently protect IVNs from cyberattacks while also maintaining functional safety and real-time communication assurances. The study highlighted unresolved topics and upcoming research avenues while also presenting the restrictions and peculiarities of IDS designs for IVNs. Improving IDS accuracy and response times, adapting functional safety processes to security engineering, establishing uniform standards for supply chains, utilizing network characteristics across IVN layers, and deploying machine learning algorithms in IVN systems with limited computing resources are some of these topics.

The paper in [18] suggested an intrusion detection system (IDS) for detecting assaults on in-vehicle networks based on a deep convolutional neural network (DCNN). They gathered a dataset of CAN-BUS traffic, separated it into normal and abnormal traffic, and utilized it to train and assess their DCNN model. They compared their suggested method with two existing machine learning algorithms (K-NN and SVM) and assessed each using accuracy, precision, recall, and F1-scores. In the realm of in-vehicle network intrusion detection using DCNNs, the suggested technique is original and creative. The authors gave a full description of the dataset as well as the processes performed to prepare the data for model training. The authors assessed the effectiveness of the suggested strategy using a variety of performance criteria and demonstrated that their study's findings revealed that the suggested DCNN-based IDS beat the other two machine learning algorithms (K-NN and SVM), with an accuracy of 99.97%, precision of 99.98%, recall of 99.94%, and F1-score of 99.96%. This implies that the suggested method is quite good at detecting intrusions in in-vehicle networks. The authors did not address the constraints or problems of their proposed strategy, which may indicate a gap in the literature. Furthermore, the authors did not investigate the influence of various types of attacks on the performance of their suggested strategy, which may be a gap in the literature. Future research could concentrate on filling these gaps in the literature to improve the effectiveness of in-vehicle network intrusion detection systems.

The authors of [19] suggested a tree-structured machine learning model-based intelligent intrusion detection system (IDS). According to the findings of the proposed intrusion detection system installation in the standard datasets, the system can spot various cyberattacks in AV networks. The proposed system can achieve a high detection rate while maintaining a low computing cost because of the ensemble learning and feature selection approaches. The accuracy of the CICIDS2017 dataset is 100% and 99.86%, respectively. The findings in both datasets showed that the proposed system outperforms strategies recommended in the literature in terms of accuracy, detection rate, F1-score, and false alarm rate by 2–3%. The drawbacks include its resource-intensiveness and low scalability, adaptability, and transparency. It is crucial to fully evaluate these drawbacks and investigate alternate strategies, such as deep learning-based IDSs or hybrid IDSs combining other machine learning techniques when developing and deploying a security solution for the IoT for a vehicle.

In [20], the authors proposed an intrusion detection system (IDS) based on machine learning (ML) methods that can identify both spoofing and jamming attacks in a CAV environment. The IDS lessens the possibility of traffic jams and accidents brought on by cyberattacks. The given IDS's detection engine is based on the machine learning (ML) algorithms random forest (RF), k-nearest neighbors (K-NN), and one-class support vector machine (OCSVM) as cross-layer data fusion methods. The implemented IDS's evaluation results showed a high accuracy of over 90% when using training datasets that included both known and new attacks. The IDS yielded 40 false negatives (FNs), 41 false positives (FPs), 527 true positives (TPs), indicating the no-attack class, and 200 true negatives (TNs), indicating the attack class, in this set of experiments using the one-class support vector machine (OCSVM) algorithm after training exclusively with no-attack data.

The study in [21] introduces a novel intrusion detection system (IDS) for in-vehicle networks (IVNs) that uses a remote frame. The suggested OTIDS approach for detecting intrusions in IVNs is described in the study. The remote frame, a type of data frame in the controller area network (CAN) protocol, serves as the foundation of the system. The authors explain how the OTIDS processes remote frames and analyzes the network traffic to identify potential intrusions. The authors also discuss the implementation of the OTIDS and its evaluation using various experiments. One of the strengths of the paper is the novelty of the proposed OTIDS methodology. The use of a remote frame in IVN intrusion detection is a unique approach, and the authors provide a clear explanation of the methodology. Furthermore, the experimental findings revealed that the suggested OTIDS has a high detection rate and a low false alarm rate, implying that it might be an effective IVN IDS. According to the testing results, the suggested OTIDS has a detection rate of more than 99% for various types of attacks such as DoS, ECU spoofing, and message injection. The OTIDS has a low false alarm rate, indicating that it can discern between normal and abnormal network traffic. The authors did not compare the performance of the OTIDS with other current IVN IDSs, which is one of the paper's weaknesses. It is difficult to analyze the relative strengths and shortcomings of the proposed OTIDS without such a comparison. Furthermore, the work lacks a full assessment of the proposed system's shortcomings and potential future research approaches.

A summary of the literature review is given in Table 1 below.

Table 1. Summary of literature review.

Authors	Year	Technology	Strength	Weakness
D. Khan, W. Lim, and Y. S. Kim [10]	2023	Optimized threshold sliding window approach (statistical analysis method)	According to the experimental results, CAN-BUS systems can efficiently discern between valid and malicious data using the proposed IDS. The IDS scored an F1-score of 99.56% and an accuracy, precision, and recall of 99.64%.	By comparing window vectors to reference values of normality, the suggested IDS may identify the presence of an attacking ECU. However, if the compromised ECU launches an attack before establishing normal values, anomaly identification can be inaccurate.

Table 1. *Cont.*

Authors	Year	Technology	Strength	Weakness
L. Yang, A. Moubayed, and A. Shami [9]	2022	Multi-tiered hybrid IDS (signature-based IDS and an anomaly-based IDS)	Using the CAN intrusion dataset and CICIDS2017 dataset, the suggested system can accurately identify a variety of known attacks with detection accuracies of 99.99% and 99.88%, respectively, and with average F1-scores of 0.963 and 0.800 in the CAN intrusion dataset and CICIDS2017 dataset, respectively.	The F1-score is not as high as the accuracy rate, and that can be optimized.
B. Karthiga, D. Durairaj, N. Nawaz, T. K. Venkatasamy, G. Ramasamy, and A. Hariharasudan [11]	2022	Modified LeeNET (MLNET)	Attack detection rate (ADR) results show that 98.5% of malicious attacks, 98.9% of botnet attacks, 99.7% of port scan attacks, and 93.9% of brute force attacks are detected	The performance of the IDS in the VANET was not maximized since intelligent key management and deep learning techniques were not used.
K. Agrawal, T. Alladi, A. Agrawal, V. Chamola, and A. Benslimane [12]	2022	Deep learning	The deep learning-based model achieved above 99.90% in almost every scenario. It has good prediction time results, at just 128.73 ms.	Long prediction time.
Y. Dong, K. Chen, Y. Peng, and Z. Ma [13]	2022	Decision trees, K-nearest neighbors (K-NN), and support vector machines (SVMs)	The semi-supervised learning method achieved a detection rate of 97.08% and a false-positive rate of 2.84%.	The paper does not discuss the limitations of the study, such as the lack of real-world data and the fact that the datasets were artificially created.
A. R. Gad, A. A. Nashat, and T. M. Barkat [14]	2021	Chi-square and SMOTE approaches	The chi-square and SMOTE approaches produced significant results for XGBoost. The FPR was 0.019, the XGBoost training accuracy was 0.980, and all other metrics were 0.978.	The work did not address dimensionality reduction.
M. Mehmood, T. Javed, J. Nebhen, S. Abbas, R. Abid, G. R. Bojja, and M. Rizwan [15]	2021	Support vector machine (SVM) and the adaptive neuro-fuzzy system (ANFIS)	The NSL-KDD dataset's SVM classification accuracy percentage was 99.3%. In testing and validation, the MSEs were 0.08496 and 0.08523, respectively.	Finding data and then effectively detecting attacks is one of the primary challenges faced by the current approach.
M. Injadat, A. Moubayed, A. B. Nassif, and A. Shami [16]	2021	RF classifier with Bayesian optimization using tree Parzen estimator (BO-TPE-RF)	The experimental results showed that the optimized RF classifier with Bayesian optimization using the tree Parzen estimator (BO-TPE-RF) had a high detection accuracy exceeding 99% for both the CICIDS2017 and the UNSW-NB 2015 datasets, and the hyperparameter tweaking improved the model performance.	The computational cost and the limited interpretability caused by using Bayesian optimization, the limited flexibility compared with other algorithms, the possibility of overfitting, and the reliance on the quality of the input data.

Table 1. *Cont.*

Authors	Year	Technology	Strength	Weakness
W. Wu, R. Li, G. Xie, J. An, Y. Bai, J. Zhou and K. Li. [17]	2020	Survey	The paper provides a comprehensive survey of intrusion detection techniques used for in-vehicle networks. This can be useful for researchers and practitioners in the field who want to obtain a quick overview of the current state-of-the-art IDS.	The paper does not discuss the impact of intrusion detection techniques on the performance of in-vehicle networks
H. M. Song, J. Woo, and H. K. Kim [18]	2020	Deep convolutional neural network (DCNN)	The results of the study showed that the proposed DCNN-based IDS achieved an accuracy of 99.97%, a precision of 99.98%, a recall of 99.94%, and an F1-score of 99.96%, which outperformed the other two machine learning algorithms (K-NN and SVM).	The authors did not discuss the limitations or challenges of their proposed approach.
L. Yang, A. Moubayed, I. Hamieh, and A. Shami [19]	2019	Ensemble learning	The accuracy of the CICIDS2017 dataset and the CAN intrusion dataset was 100% and 99.86%, respectively, while the calculation time for each decreased by 73.7% (325.6 s) and 38.6% (2774.8 s).	The drawbacks include resource-intensiveness and low scalability, adaptability, and transparency. It is crucial to carefully evaluate these drawbacks and investigate alternate strategies, such as deep learning-based IDSs or hybrid IDSs combining other machine learning techniques when developing and deploying a security solution for the IoV.
D. Kosmanos, A. Pappas, F. J. Aparicio-Navarro, L. Maglaras, H. Janicke, E. Boiten, and A. Argyriou [20]	2019	Data fusion, random forest (RF), k-nearest neighbors (K-NN), and one-class support vector machine (OCSVM)	The OCSVM algorithm achieved an accuracy of approximately 90%. The IDS yielded 40 false negatives (FNs), 41 false positives (FPs), and 527 true positives (TPs).	The experiment was not conducted on a real-world dataset. The studies were conducted using datasets from a simulated CAV environment with car platooning. OCSVM is unsuitable for large datasets, it has a large training time, more features, and more complexities.
H. Lee, S. H. Jeong, and H. K. Kim [21]	2018	OTIDS	The experimental results showed that the proposed OTIDS had a high detection rate of over 99% for various types of attacks, including DoS, ECU spoofing, and message injection. The false alarm rate of OTIDS was also low, indicating that it can accurately distinguish between normal and abnormal network traffic	One weakness of this paper is that the authors did not compare the performance of OTIDS with other existing IDSs for IVNs. Without such a comparison, it is difficult to assess the relative strengths and weaknesses of the proposed OTIDS.

3. Data Sources and Dataset Description

Understanding the data and data sources involved is crucial when designing an IDS for autonomous vehicles [22]. Data might be organized (structured or unstructured, where structured data are data that are ordered and pre-defined, and unstructured data include text, photographs, and videos). An IDS's data sources might be both external and internal. External inputs include traffic data, weather conditions, and threat intelligence feeds, while internal sources include sensors, cameras, and vehicle control systems. Using analytical tools, researchers and engineers may improve the safety and security of autonomous vehicles, reducing possible dangers and assuring a dependable and trustworthy transportation system in the future. The data used for this research were secondary data sourced from online repositories [23–25]. This research experimented on three datasets for critical evaluation purposes. These datasets included the following:

- The CIC-IDS2017 dataset;
- The NSL-KDD dataset;
- The CAN-BUS dataset.

The Canadian Institute for Cybersecurity Intrusion Detection System (CIC-IDS2017) dataset consists of network traffic data that were acquired in a controlled setting from various network traffic generators and attack tools. It offers a variety of network traffic situations, including both innocent traffic and different kinds of simulated attacks. With the dataset, network traffic is accurately portrayed, facilitating the creation and assessment of intrusion detection systems (IDSs). This dataset contains 1,039,934 rows and 79 features. It contains attacks such as denial of service (DoS), distributed denial of service (DDoS), and SQL injection [26].

The NSL-KDD dataset is a benchmark dataset that is often utilized. The KDD Cup 1999 dataset, which was extensively used to rate intrusion detection systems, has been upgraded. To overcome some of the restrictions and flaws of the KDD Cup dataset, the NSL-KDD dataset was developed [27]. Its main goal is to offer a more rigorous and realistic dataset for assessing intrusion detection systems. It includes network traffic information that simulates both typical computer network activity and other forms of attacks. Both the network traffic attributes and the related class labels indicating whether a certain occurrence is normal or falls under a particular attack type are included in the dataset offering a balanced distribution of attack types. Denial of service (DoS), user to root (U2R), remote to local (R2L), and probing are the four basic types of attacks that it covers. There are 125,973 rows and 43 characteristics in this dataset [27].

The CAN-BUS dataset CAN (controller area network) [28] is a well-liked communication standard in the automotive industry. Simplifying communication between various electronic control units (ECUs) inside a vehicle makes possible functions like engine management, transmission control, brakes, and more [29]. It takes over or influences how the system behaves, including taking advantage of flaws or vulnerabilities in the system. Potential CAN-BUS breaches may have detrimental effects, including jeopardizing the vehicle and its occupants' safety and security. There are 25,192 rows and 42 characteristics in this dataset.

4. Analytics Process and Methodology

Machine learning (ML) and predictive modeling approaches are useful tools for designing successful intrusion detection systems [30]. To categorize and detect prospective intrusions in real time, supervised learning algorithms can be trained on labeled data, where intrusions are tagged as malicious or benign. As seen in the highlighted section in Figure 2, in this research, the architecture of [31] was improved by placing the intrusion detection system in such a way that all key communications, people interactions, and environmental interactions are filtered for intrusion.

Clustering is one unsupervised learning strategy that helps identify abnormal patterns and find undiscovered or zero-day risks [32]. To pick the most effective model for intrusion detection in autonomous vehicles, it is necessary to compare several machine learning models and examine their performance metrics, such as accuracy, precision, recall, and F1-score. These steps' details are given below.

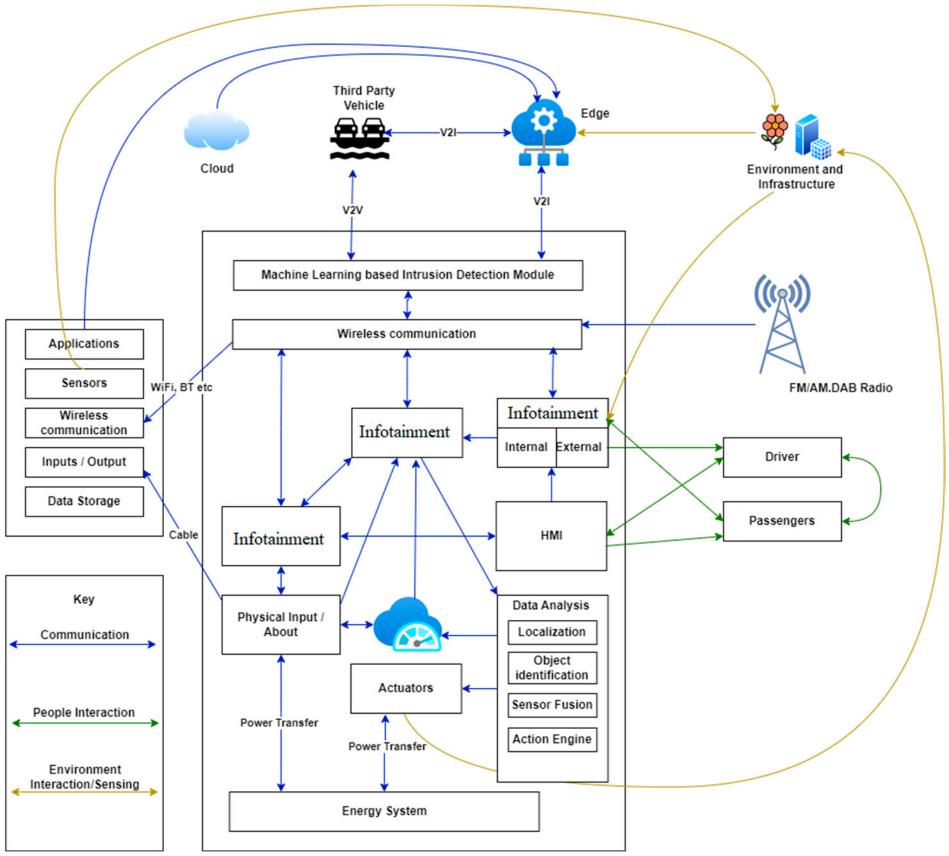

Figure 2. Peripherals and reference architecture of an autonomous vehicle (improved).

4.1. Data Pre-Processing

Preparing unprocessed data for use with a machine learning model is known as data preparation [33]. The first and most crucial step in creating a machine learning model is this one. In this research, the three datasets were cleaned during the data preparation step. Cleaning was performed by eliminating noise from the data. In this stage, this research handled missing values by replacing them with an average value or by deletion. All outliers identified, as well as irrelevant features, were eliminated to make the data usable. Data normalization was performed in this phase by scaling features to a common range if necessary. Normalization is a data preparation method that is used to scale the values of characteristics in a dataset. Standardization involves scaling data to fit a typical normal distribution. A distribution with a mean of 0 and a standard deviation of 1 is known as a standard normal distribution. The dataset was divided into training and testing sets in the last step. Equation (1) designates the standardization:

$$z = \frac{x_i - \mu}{\sigma}(i) \tag{1}$$

where z is the normalized value of x, x_i is the individual data point, μ is the mean of the dataset, and σ is the standard deviation of the dataset.

In all three selected databases, zero missing values were observed. Regarding categorical values, in NSL-KDD, these values were encoded into a machine-readable format, and the target values were also encoded. In CAN, the variables used were numeric data. and the numeric values were transformed using the encoder, while in CICDA2017, one-hot encoding was used to transform the categorical variables.

4.2. Exploratory Data Analysis

Exploratory data analysis is a data analytics strategy that makes use of visual tools to understand the data in greater detail and learn more about them. In this research, data visualization, which, essentially, is a method that allows us to visualize the distribution of target classes (normal or abnormal) in order to understand the imbalance in the dataset visualized, was used. It provided a better comprehension of the material and revealed intriguing tendencies in it. This was a critical method in this research as it was used to conduct initial research on the data to find patterns, identify anomalies, test hypotheses, and verify presumptions using summary statistics and graphical representations.

4.3. Feature Selection

A method for reducing the input variable to the proposed model is feature selection, which involves using only pertinent data and eliminating irrelevant data. It is a method for automatically choosing the right traits for a machine learning model based on the type of problem to be resolved. The three types of feature selection include wrapper approaches (forward, backward, and stepwise selection), filter methods (ANOVA, Pearson correlation, and variance thresholding), and embedded methods (lasso, ridge, and decision tree). This research used recursive feature elimination (RFE), which is a type of wrapper feature selection method. It removes the least relevant feature continuously until the number of desired features is reached.

In NSL-KDD, the principal component analysis technique was used, and the features were reduced to 20. In CAN, RFE was used, and the total number of features used was 10. In CICIDS 2017, 26 features were used.

4.4. Model Selection

The process of choosing one final machine learning model from a group of potential candidates for a training dataset is known as model selection. This research selected a model by splitting the datasets into training and validation sets that simulated previously unobserved data while considering a wide range of non-tree-based models. including logistic regression, SVM, K-NN, and others. Each model was trained using the training dataset and tested on the validation dataset using metrics such as accuracy, precision, recall, F1-score, sensitivity, and specificity. This comparison allowed for a robust model to be selected. Below, the tested machine learning model for each selected database is given (where decision tree and AdaBoost were optimizers):

- NSL-KDD: logistic regression, support vector machine, Gaussian naive Bayes, and AdaBoost.
- CAN: logistic regression, support vector machine, Gaussian naive Bayes, AdaBoost, and decision tree.
- CICIDS2017: logistic regression, support vector machine, and decision tree.

4.5. Training

Giving data to an ML system to help locate and learn appropriate values for all qualities involved is known as “model training” in machine language. There are many kinds of machine learning models, but supervised and unsupervised learning are the most common ones. Supervised learning is achievable when the training data contain both the input and output values. Although there are many alternative training methods, the supervised learning method is the subject of this research. Validation is carried out on the testing dataset, while training is carried out on the training data.

The time complexity was not captured; however, CAN and NSL-KDD were trained for 5 h, while CIDS was trained for 3 days, continuously. The data were divided using 80% for training and 20% for testing.

4.6. Optimization

The optimization of training and testing is a modification made to enhance the final product. Hyper-parameter tuning and sampling (random sampling, SMOTE, etc.) can be used to optimize a system. Iteratively improving a machine learning model's accuracy to lower its level of inaccuracy is known as machine learning optimization. Machine learning algorithms learn to generalize and predict new live data based on information learned from training data. The result is evaluated using a confusion matrix, which consists of true-positive (TP), false-positive (FP), true-negative (TN), and false-negative (FN) results. These results can be used to derive the performance metrics used. The metrics used in this research included the following.

TPs (true positives) correctly predicted positives, TNs (true negatives) correctly predicted negatives, FPs (false positives) incorrectly predicted positives, and FNs (false negatives) incorrectly predicted negatives.

Accuracy: This determined the proportion of occurrences that were successfully predicted for all the examples in the dataset.

$$\text{accuracy} = \frac{(\text{TP} + \text{TN})}{(\text{TP} + \text{TN} + \text{FP} + \text{FN})} \tag{2}$$

Precision: This determined the proportion of accurate positive predictions on all occasions when a positive outcome was expected.

$$\text{Precision} = \frac{(\text{TP})}{(\text{TP} + \text{FP})} \tag{3}$$

Recall: This determined the proportion of genuine positive forecasts in all the actual positive cases, also known as sensitivity.

$$\text{Recall} = \frac{(\text{TP})}{(\text{TP} + \text{FN})} \tag{4}$$

F1-score: The F1-score combined precision and recall in assessing a classification model's accuracy. A balance between recall and accuracy was reached by determining their harmonic mean.

$$\text{F1} - \text{score} = \frac{2 * (\text{recall} * \text{precision})}{(\text{recall} + \text{Precision})} \tag{5}$$

Specificity: This determined the proportion of accurate negative predictions in all of the actual negative cases.

$$\text{Specificity} = \frac{(\text{TN})}{(\text{TN} + \text{FP})} \tag{6}$$

As seen in Figure 3, for the proposed IDS, the dataset was imported from an online repository, after which it was checked for imbalance. If there was a data imbalance, it was handled by splitting the training and testing datasets. The training dataset was trained, and an evaluation was carried out on the testing dataset. If the dataset did not contain any imbalances, the imbalance-handling step was passed.

This research chose K-NN, naïve Bayes, SVM, logistic regression, and AdaBoost as its machine learning algorithms. Additionally, ensemble techniques including voting, stacking, and bagging were used, which are given below.

A machine learning technique called ensemble learning combines predictions from many models to increase the final forecast's accuracy and resilience [34]. A form of machine learning technology known as ensemble methods combines many base models to obtain the single best prediction model. The fundamental principle underlying ensemble learning is to use the collective intelligence of several models to eliminate errors or biases in individual models, leading to a more accurate prediction. Typically, the maximum voting method is used to resolve classification challenges. In this strategy, each data point is forecasted using several models. A "vote" is each model's prediction. The forecasts we obtain from the majority of the models are used to create the final forecast. Bagging integrates aggregation and bootstrapping to produce an ensemble model. From a sample of data, many bootstrapped subsamples are taken. A decision tree is constructed for each of the bootstrapped subsamples. Each subsample decision tree is constructed, and then the decision trees are combined to provide the best effective predictor. Stacking is an ensemble strategy that combines predictions from many model types based on training data to identify a diverse group of members.

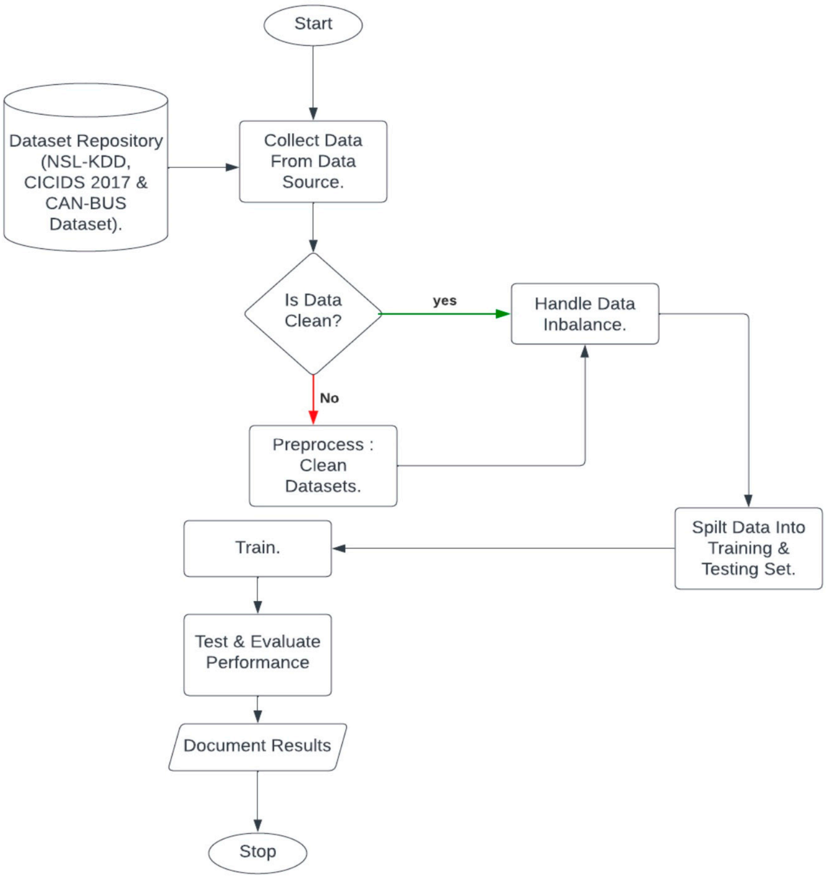
Start
Dataset Repository (NSL-KDD, CICIDS 2017 & CAN-BUS Dataset).
Collect Data From Data Source.
Is Data Clean?
yes
Handle Data Inbalance.
No
Preprocess : Clean Datasets.
Spilt Data Into Training & Testing Set.
Train.
Test & Evaluate Performance
Document Results
Stop

Figure 3. Proposed IDS flowchart.

5. Results

This research work implemented the non-tree-based IDS (NTB-MTH-IDS) approach and evaluated its performance on three datasets: CAN-BUS, NSL-KDD, and CICIDS2017, as mentioned earlier. The results of this research were compared with those of the multitier hybrid intrusion detection system (MTH-IDS) approach [9]. The comparison was based on metrics like accuracy and F1-score. It used real-world datasets, which was a weakness in some of the literature, such as [13,20].

Table 2 below displays the metrics for the different machine learning methods on the CAN-BUS dataset. The accuracy, F1-score, precision, recall, sensitivity, and specificity of each method are displayed.

K-nearest neighbors (K-NN) was the best-performing individual approach on the CAN-BUS dataset because this non-tree-based IDS (NTB-MTH-IDS) achieved a high F1-score and accuracy. For this research, adaptive boosting and stacking also produced excellent results with an accuracy of 99.9% and an F1-score of 99.9%, respectively.

Table 2. CAN-BUS dataset results.

Algorithms	Accuracy	F1-Score	Precision	Recall	Sensitivity	Specificity
K-Nearest Neighbor	97.50%	97.54%	97.54%	97.54%	97.84%	97.50%
Naive Bayes	88.86%	88.80%	88.86%	88.86%	92.31%	88.86%
Support Vector Machine	96.79%	96.78%	96.81%	96.79%	97.97%	96.79%
Logistic Regression	91.96%	91.94%	92.03%	91.96%	93.58%	91.96%
Adaptive boost	98.13%	98.13%	98.13%	98.13%	98.44%	98.13%
Stacking	99.50%	99.50%	99.50%	99.50%	99.50%	99.18%
Voting	97.14%	97.14%	97.17%	97.14%	97.14%	98.04%
Bagging	97.20%	97.19%	97.20%	97.20%	97.20%	98.41%

K-NN also performed well in the MTH-IDS, with an F1-score of 93.4% and an accuracy of 97.4%, respectively. The F1-score in this research was attributed to the selection and combination of machine learning algorithms that better manage the imbalanced data problem, resulting in increased precision and recall for intrusion detection. The adaptive boosting algorithm, which is a tree-based algorithm, served as a good optimizer for boosting weak learners in the non-tree-based IDS ensemble learning implementation. Figure 4 shows the column chart of all the models trained and tested on the CAN intrusion detection dataset.

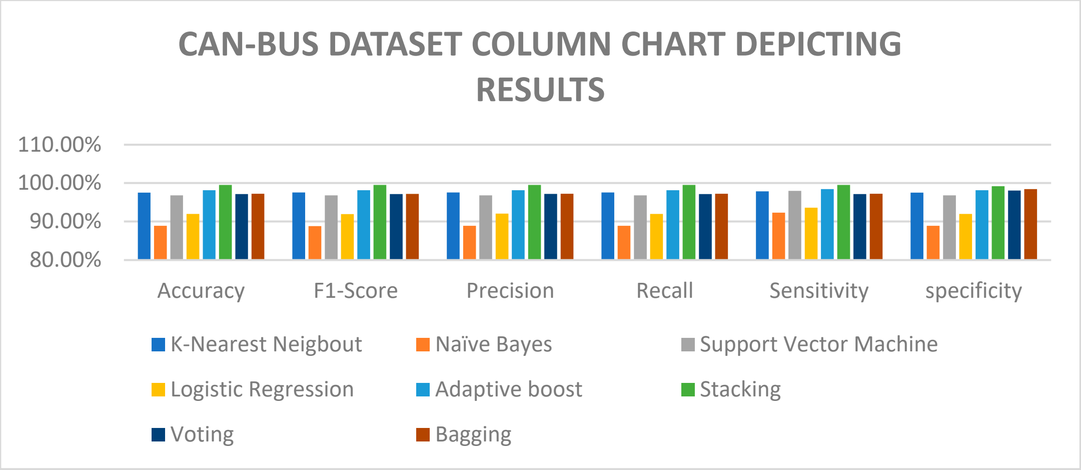

Figure 4. CAN-BUS dataset result chart.

The results of the training and validations of each model on the NSL-KDD dataset are shown in Table 3 below, and Figure 5 shows the column chart of all the models trained and tested on the NSL-KDD dataset. The algorithms' accuracy, F1-scores, precision, recall, sensitivity, and specificity were assessed. This dataset contained labeled network traffic data.

This research performed well on the NSL-KDD dataset, with the maximum accuracy and F1-scores being attained by K-NN, adaptive boosting, and stacking. As opposed to [9], this work achieved an accuracy of 98.87% and an F1-score of 98.79%, which are still excellent results but low in comparison. The higher F1-scores and accuracy may have resulted from this research's (NTB-MTH-IDS) use of non-tree-based methods like K-NN to better manage the dataset's complexity, while adaptive boosting, a tree-based approach, was employed to improve weak learners at the ensemble level.

Table 4 shows how the different machine learning algorithms performed on the CICIDS2017 dataset. This dataset, which comprises network traffic data with tagged instances of legitimate and malicious activity, is used for intrusion detection.

Table 3. NSL-KDD dataset results.

Algorithms	Accuracy	F1-Score	Precision	Recall	Sensitivity	Specificity
K-Nearest Neighbor	98.87%	98.79%	99.02%	98.56%	98.56%	98.75%
Naive Bayes	87.14%	85.34%	91.03%	80.12%	80.12%	84.25%
Support Vector Machine	96.92%	96.67%	97.71%	95.65%	95.65%	95.25%
Logistic Regression	88.05%	87.77%	84.13%	91.74%	91.74%	92.13%
Adaptive boost	98.82%	98.73%	98.94%	98.52%	98.52%	98.71%
Stacking	98.09%	98.09%	98.10%	98.09%	98.09%	97.61%
Voting	93.68%	93.68%	94.27%	93.68%	93.68%	89.64%
Bagging	98.60%	93.55%	94.23%	98.60%	98.60%	89.43%

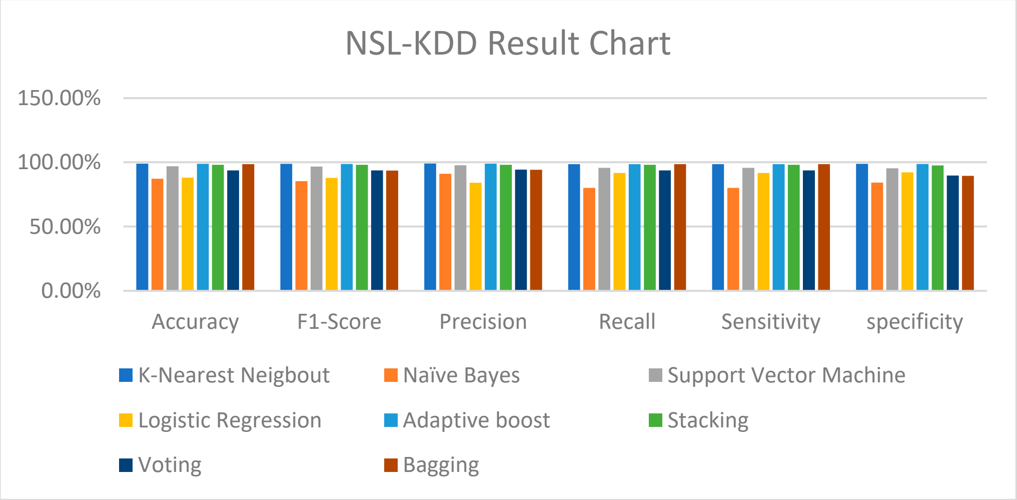

Figure 5. NSL-KDD result chart.

Table 4. CICIDS2017 dataset results.

Algorithms	Accuray	F1-Score	Precision	Recall	Sensitivity	Specificity
K-Nearest Neighbor	99.00%	99.00%	99.00%	99.00%	99.00%	98.70%
Naive Bayes	67.82%	68.00%	68.00%	68.00%	68.00%	52.64%
Logistic Regression	64.50%	64.00%	65.00%	65.00%	65.00%	64.14%
Adaptive boost	98.58%	99.00%	99.00%	99.00%	99.00%	97.88%
Support Vector Machine	72.14%	63.68%	91.80%	48.75%	48.75%	95.63%
Stacking	99.75%	99.75%	99.75%	99.75%	99.75%	99.76%
Bagging	64.38%	63.99%	65.08%	64.38%	64.38%	61.77%
Voting	74.94%	73.87%	79.92%	74.94%	74.94%	67.66%

On the CICIDS2017 dataset, this research achieved high accuracy and F1-score, with K-NN performing exceptionally well as well as tree-based algorithms such as adaptive boosting. In contrast, this work achieved an accuracy of 99.00% and an F1-score of 99.00%. Both approaches displayed impressive results on this dataset, making it challenging to definitively claim one is better than the other. A visualization of the column chart can be seen in Figure 6.

Since the F1-score considers both precision and recall, it is a more accurate tool for assessing performance with imbalanced data. When it comes to intrusion detection for autonomous vehicles, the data are frequently unbalanced, which means that the instances of legitimate occurrences greatly surpass those of illegitimate intrusion cases. In these circumstances, a high degree of accuracy might be attained by categorizing all occurrences as normal (the majority class) and disregarding the minority class of harmful instances. This would, however, result in low recall (missed detections) for the malicious occurrences.

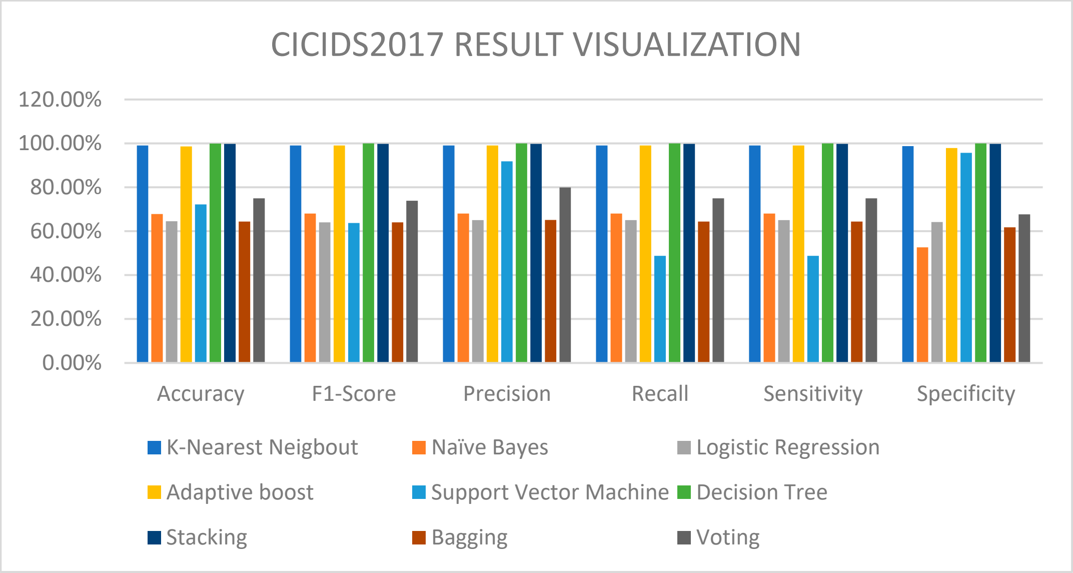

Figure 6. CICIDS2017 result chart.

A balanced evaluation of both false positives (precision) and false negatives (recall) is provided by the F1-score, which is the harmonic mean of precision and recall. While precision indicates the accuracy of the positive predictions, recall is the ability to correctly recognize positive events (intrusions) in the whole collection of positive instances. The harmonic mean of precision and recall is known as the F1-score, which evaluates the model's performance across all classes equally. It is, therefore, a more trustworthy metric for assessing intrusion detection systems in imbalanced data settings because it penalizes models that overfit the majority class and requires them to pay equal attention to the minority class.

6. Discussion

K-nearest neighbors (K-NN) has the best performance for intrusion detection systems, as measured in terms of accuracy, F1-score, precision, recall, and sensitivity on the three datasets used (the CICIDS2017, CAN-BUS, and NSL-KDD datasets). K-NN demonstrated remarkable accuracy, with an F1-score of 97.54%, a precision of 97.54%, a recall of 97.54%, and a sensitivity of 97.84% on the CAN-BUS dataset. Similar results were achieved using K-NN on the NSL-KDD dataset with 98.87% accuracy, a 98.79% F1-score, 99.02% precision, 98.56% recall, and 98.56% sensitivity. K-NN also obtained an F1-score of 99.00%, accuracy of 99.00%, recall of 99.00%, and sensitivity of 99.00% on the CICIDS2017 dataset.

In comparison with ensemble approaches (adaptive boosting and stacking) for intrusion detection in these datasets, the classic algorithms K-nearest neighbors, naïve Bayes, and logistic regression performed well. On all datasets, K-NN performed better than naïve Bayes and logistic regression in terms of accuracy and F1-score. However, ensemble techniques like stacking trees also performed well, particularly when combined with non-tree-based algorithms like stacking and K-NN. Because they may combine the qualities of many base classifiers, ensemble approaches frequently achieve improved accuracy, recall, and sensitivity.

The high accuracy and F1-score results of specific algorithms on these datasets generally translate into high precision, recall, and sensitivity. However, there might be minor trade-offs between these metrics. For example, while achieving a high accuracy and F1-score, naïve Bayes may have relatively lower recall and sensitivity, indicating a tendency to miss certain intrusive instances. Conversely, ensemble methods like stacking often maintain a balanced performance across precision, recall, and sensitivity due to their ensemble nature.

The choice of algorithm significantly impacts the ability to correctly identify normal and malicious network traffic instances in the context of intrusion detection for autonomous vehicles. Algorithms like K-nearest neighbors and ensemble methods such as adaptive boosting and stacking tend to exhibit superior performance in distinguishing between normal and intrusive instances. Their robustness in handling imbalanced data and effectively categorizing both types of instances make them suitable choices for intrusion detection in autonomous vehicles.

Contrasting the effectiveness of machine learning algorithms in spotting intrusions on the CICIDS2017, CAN-BUS, and NSL-KDD datasets reveals the critical role of non-tree-based algorithms, especially K-nearest neighbors and ensemble methods, in achieving high accuracy, precision, recall, and sensitivity. These algorithms demonstrate their potential for effective application in real-world autonomous car environments, providing strong security measures against potential cyber threats. This research underscores the importance of selecting appropriate algorithms for intrusion detection in autonomous vehicles to ensure their safety and reliability in modern transportation networks.

This research successfully achieved its aim and objectives by developing and evaluating an improved intrusion detection system (IDS) for autonomous vehicles. The results demonstrate the effectiveness of non-tree-based machine learning algorithms, especially K-nearest neighbors and stacking, in accurately identifying and preventing cyberattacks. This research contributes valuable insights to the selection and combination of algorithms for building reliable intrusion detection systems for autonomous vehicles, addressing the critical issue of cybersecurity in self-driving vehicles.

The data provided show K-NN had the best performance in the NSL-KDD dataset, while stacking closely followed alongside adaptive boosting. Note that here, the comparison was not between algorithms but between two approaches. The proposed approach used non-tree-based algorithms for ensemble learning, and tree-based algorithms such as DT and AdaBoost were used to optimize their performance. Overfitting is a problem associated with tree-based algorithms. This is the reason why decision tree demonstrated good performance; however, it was only used to optimize the non-tree-based algorithms. In this work, K-NN, stacking, and AdaBoost performed optimally across the three selected datasets.

7. Conclusions

This research compared the performance of various non-tree-based machine learning algorithms on the three datasets. The non-tree-based IDS (NTB-MTH-IDS) approach demonstrated superior performance with high accuracy, F1-scores, precision, recall, and sensitivity on all datasets. The ensemble methods like stacking, along with K-nearest neighbors, consistently outperformed traditional algorithms like naïve Bayes and logistic regression, highlighting the effectiveness of non-tree-based algorithms in handling the complexities of intrusion detection in autonomous vehicles. This research found that ensemble methods, especially stacking, outperformed traditional algorithms in terms of accuracy, F1-score, precision, recall, and sensitivity. K-nearest neighbors also performed well, but naïve Bayes and logistic regression showed comparatively lower performance. This highlights the importance of using ensemble methods and non-tree-based algorithms for more effective intrusion detection in the context of autonomous vehicles. This research showed that in most cases, a high accuracy and F1-score were accompanied by high precision, recall, and sensitivity, indicating a balanced performance of the intrusion detection system. However, there were slight variations in some cases, suggesting that certain algorithms might excel in specific metrics. Nevertheless, the non-tree-based IDS (NTB-MTH-IDS) approach demonstrated consistent performance across all metrics, indicating its reliability in effectively identifying and preventing cyberattacks.

Future works could focus on enhancing this research on non-tree-based IDSs (NTBMTHIDS) by exploring more advanced machine learning algorithms, such as deep learning techniques, for improved detection accuracy and scalability. Another area in which research can be conducted is addressing the challenges of real-time implementation and adaptability to evolving cyber threats in autonomous vehicle environments. Additionally, timing-based attacks, hardware benchmarks, and failure analysis can be conducted in future work.

Author Contributions: Conceptualization, M.R. and W.E.; methodology, C.A.; software, C.A.; validation, M.R. and C.A.; formal analysis, C.A.; investigation, C.A., M.R. and W.E.; resources, M.R. and W.E.; data curation, C.A.; writing—original draft preparation, C.A.; writing—review and editing, M.R.; visualization, C.A. and M.R.; supervision, M.R. and W.E.; project administration, M.R and W.E. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported, in part, by the Higher Education Authority (HEA) under the Human Capital Initiative-Pillar 3 project, Cyberskills.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data that support the findings of this study are available on request from the Authors.

Acknowledgments: Thank you to the Department of Electronic and Computer Engineering at the University of Limerick, Ireland, and Jaguar Land Rover (JLR), Shannon, Ireland, for supporting this work as part of the Masters in Information and Network Security program.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Gram-Hanssen, K.; Georg, S. *Energy Performance Gaps: Promises, People, Practices*; Taylor & Francis: Abingdon, UK, 2018; pp. 1–9.
2. Zhili, D.; Boqiang, L.; Chunxu, G. Development path of electric vehicles in China under environmental and energy security constraints. *Resour. Conserv. Recycl.* **2018**, *143*, 17–26. [CrossRef]
3. Kellstedt, P.M.; Zahran, S.; Vedlitz, A. Personal Efficacy, the Information Environment, and Attitudes Toward Global Warming and Climate Change in the United States. *Risk Anal.* **2008**, *28*, 113–126. [CrossRef] [PubMed]
4. Lavagno, L.; Oppedisano, C.; Cicciarell, S. *Design of a Data Aggregation Circuit for Autonomous Driving LiDAR Sensors*; Politecnico di Torino: Torino, Italy, 2021.
5. Liao, H.-J.; Lin, C.-H.R.; Lin, Y.-C.; Tung, K.-Y. Intrusion detection system: A comprehensive review. *J. Netw. Comput. Appl.* **2013**, *36*, 16–24. [CrossRef]
6. Al-Jarrah, O.Y.; Maple, C.; Dianati, M.; Oxtoby, D.; Mouzakitis, A. Intrusion Detection Systems for Intra-Vehicle Networks: A Review. *IEEE Access* **2019**, *7*, 21266–21289. [CrossRef]
7. Alsulami, A.A.; Abu Al-Haija, Q.; Alqahtani, A.; Alsini, R. Symmetrical Simulation Scheme for Anomaly Detection in Autonomous Vehicles Based on LSTM Model. *Symmetry* **2022**, *14*, 1450. [CrossRef]
8. Bhosale, D.A.; Mane, V.M. Comparative study and analysis of network intrusion detection tools. In Proceedings of the 2015 International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT), Davangere, India, 29–31 October 2015; pp. 312–315.
9. Yang, L.; Moubayed, A.; Shami, A. MTH-IDS: A Multitiered Hybrid Intrusion Detection System for Internet of Vehicles. *IEEE Internet Things J.* **2021**, *9*, 616–632. [CrossRef]
10. Khan, J.; Lim, D.-W.; Kim, Y.-S. Intrusion Detection System CAN-Bus In-Vehicle Networks Based on the Statistical Characteristics of Attacks. *Sensors* **2023**, *23*, 3554. [CrossRef] [PubMed]
11. Karthiga, B.; Durairaj, D.; Nawaz, N.; Venkatasamy, T.K.; Ramasamy, G.; Hariharasudan, A. Intelligent Intrusion Detection System for VANET Using Machine Learning and Deep Learning Approaches. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 1–13. [CrossRef]
12. Agrawal, K.; Alladi, T.; Agrawal, A.; Chamola, V.; Benslimane, A. NovelADS: A Novel Anomaly Detection System for Intra-Vehicular Networks. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 22596–22606. [CrossRef]
13. Dong, Y.; Chen, K.; Peng, Y.; Ma, Z. Comparative Study on Supervised versus Semi-supervised Machine Learning for Anomaly Detection of In-vehicle CAN Network. In Proceedings of the 2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC), Macau, China, 8–12 October 2022; pp. 2914–2919.
14. Gad, A.R.; Nashat, A.A.; Barkat, T.M. Intrusion detection system using machine learning for vehicular ad hoc networks based on ToN-IoT dataset. *IEEE Access* **2021**, *9*, 142206–142217. [CrossRef]
15. Mehmood, M.; Javed, T.; Nebhen, J.; Abbas, S.; Abid, R.; Bojja, G.R.; Rizwan, M. A Hybrid Approach for Network Intrusion Detection. *Comput. Mater. Contin.* **2022**, *70*, 91–107. [CrossRef]
16. Injadat, M.N.; Moubayed, A.; Nassif, A.B.; Shami, A. Multi-Stage Optimized Machine Learning Framework for Network Intrusion Detection. *IEEE Trans. Netw. Serv. Manag.* **2020**, *18*, 1803–1816. [CrossRef]

17. Wu, W.; Li, R.; Xie, G.; An, J.; Bai, Y.; Zhou, J.; Li, K. A survey of intrusion detection for in-vehicle networks. *IEEE Trans. Intell. Transp. Syst.* **2019**, *21*, 919–933. [CrossRef]
18. Song, H.M.; Woo, J.; Kim, H.K. In-vehicle network intrusion detection using deep convolutional neural network. *Veh. Commun.* **2019**, *21*, 100198. [CrossRef]
19. Yang, L.; Moubayed, A.; Hamieh, I.; Shami, A. Tree-Based Intelligent Intrusion Detection System in Internet of Vehicles. In Proceedings of the GLOBECOM 2019—2019 IEEE Global Communications Conference, Waikoloa, HI, USA, 9–13 December 2019; pp. 1–6.
20. Kosmanos, D.; Pappas, A.; Aparicio-Navarro, F.J.; Maglaras, L.; Janicke, H.; Boiten, E.; Argyriou, A. Intrusion Detection System for Platooning Connected Autonomous Vehicles. In Proceedings of the 2019 4th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM), Piraeus, Greece, 20–22 September 2019.
21. Lee, H.; Jeong, S.H.; Kim, H.K. OTIDS: A Novel Intrusion Detection System for In-vehicle Network by Using Remote Frame. In Proceedings of the 2017 15th Annual Conference on Privacy, Security and Trust (PST), Calgary, AB, Canada, 28–30 August 2017; pp. 57–5709.
22. Aloqaily, M.; Otoum, S.; Al Ridhawi, I.; Jararweh, Y. An intrusion detection system for connected vehicles in smart cities. *Ad Hoc Netw.* **2019**, *90*, 101842. [CrossRef]
23. UNB. Brunswick. NSL-KDD Dataset. Available online: https://www.unb.ca/cic/datasets/nsl.html (accessed on 15 May 2023).
24. UNB. Brunswick. Intrusion Detection Evaluation Dataset (CIC-IDS2017). Available online: https://www.unb.ca/cic/datasets/ids-2017.html (accessed on 20 February 2023).
25. HCRL. CAN Dataset for Intrusion Detection (OTIDS). Available online: https://ocslab.hksecurity.net/Dataset/CAN-intrusion-dataset (accessed on 20 February 2023).
26. Ring, M.; Wunderlich, S.; Scheuring, D.; Landes, D.; Hotho, A. A survey of network-based intrusion detection data sets. *Comput. Secur.* **2019**, *86*, 147–167. [CrossRef]
27. Revathi, S.; Malathi, A. A detailed analysis on NSL-KDD dataset using various machine learning techniques for intrusion detection. *Int. J. Eng. Res. Technol.* **2013**, *2*, 1848–1853.
28. Verma, M.E.; Iannacone, M.D.; Bridges, R.A.; Hollifield, S.C.; Moriano, P.; Kay, B.; Combs, F.L. Addressing the lack of comparability & testing in CAN intrusion detection research: A comprehensive guide to CAN IDS data & introduction of the ROAD dataset. *arXiv* **2020**, arXiv:2012.14600.
29. Song, H.M.; Kim, H.R.; Kim, H.K. Intrusion detection system based on the analysis of time intervals of CAN messages for in-vehicle network. In Proceedings of the 2016 International Conference on Information Networking (ICOIN), Kota Kinabalu, Malaysia, 13–15 January 2016; pp. 63–68.
30. Park, S.; Choi, J.-Y. Malware Detection in Self-Driving Vehicles Using Machine Learning Algorithms. *J. Adv. Transp.* **2020**, *2020*, 1–9. [CrossRef]
31. Maple, C.; Bradbury, M.; Le, A.T.; Ghirardello, K. A Connected and Autonomous Vehicle Reference Architecture for Attack Surface Analysis. *Appl. Sci.* **2019**, *9*, 5101. [CrossRef]
32. Piazza, N.P. *A Study on the Effectiveness of Machine Learning Techniques to Detect and Prevent Zero-Day Cyberattacks*; Utica College: Utica, NY, USA, 2020.
33. Brownlee, J. *Data Preparation for Machine Learning: Data Cleaning, Feature Selection, and Data Transforms in Python*; Machine Learning Mastery: Vermont, Australia, 2020.
34. Singh, A. A Comprehensive Guide to Ensemble Learning (with Python Codes). Analyticsvidhya. Available online: https://www.scribd.com/document/408466563/A-Comprehensive-Guide-to-Ensemble-Learning-with-Python-codes-pdf (accessed on 4 July 2023).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.