National College of Ireland

Enhancing Vehicle Security: Intrusion Detection Using Machine Learning

Practicum Part 2
MSc Cyber Security

Anusha Varghese
Student ID: 23217693

School of Computing
National College of Ireland

Supervisor: Mr. Vikas Sahni

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Anusha Varghese
Student ID:	23217693
Programme:	Msc Cyber Security **Year:** 2024-2025
Module:	Practicum Part 2
Supervisor:	Mr. Vikas Sahni
Submission Due Date:	12 December 2024
Project Title:	Enhancing Vehicle Security: Intrusion Detection Using Machine Learning
Word Count:	6452 **Page Count: 20**

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Anusha Varghese

Date: 12 December 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Enhancing Vehicle Security: Intrusion Detection Using Machine Learning

Anusha Varghese
23217693

Abstract

Today's Connected vehicles open completely new dimensions in the risk of cyberattacks due to the integration of complex electronic control units and communication within the vehicle. This paper presents several different machine learning-based methods for enhancement of accuracy and detection efficiency for an intrusion detection system for improved vehicular security. First, in-vehicle network data was collected and pre-processed, and critical features were selected for intrusion detection. The ensemble learning methods employed, including Random Forest and Voting Classifier, enhanced detection accuracy and adaptability. The main challenges addressed in this paper were high false-positive rates, computational overhead, and real-time processing in designing a scalable and adaptable IDS architecture. The proposed system was designed to be effective for different vehicle models and types of cyberattacks, enhancing security and reliability in modern vehicles.

Keywords: *Intrusion Detection, Vehicle Security, Machine Learning, Cybersecurity, In-Vehicle Networks, Random Forest, Real-time Detection, Ensemble Learning*

1 Introduction

1.1 Background

The increasing automotive connectivity has grown with the integration of more sophisticated Electronic Control Units (ECU) and internal vehicle communications, thereby increasing the compute functionality of modern vehicles considerably. However, this exposes them to several different forms of cyber-attacks. Additionally, as more vehicles connect to external networks, including the Internet, the landscape of vehicular cybersecurity threats has expanded. In this context, intrusion detection systems (IDS) have emerged as a key defense mechanism against cyber threats in vehicular networks (Wang, et al., 2022). Despite extensive research, most of the existing solutions for IDS in in-vehicle networks still suffer from several fundamental limitations, such as high false positives, inefficiency in real-time detection, and a lack of scalability over different models and types of attacks. This implies that ongoing research in increasingly sophisticated cyber-attacks calls for more robust and adaptive IDS frameworks that can help safeguard the security and privacy of modern vehicles.

1.2 Motivation

It has become a critical requirement as the frequency of cyberattacks on vehicles has increased, and the most well-known incident is the hack of Hyundai cars to control them through exploited vulnerabilities. This poses severe consequences in terms of safety, privacy, and financial wellbeing. So far, IDS solutions have failed to keep pace with the dynamic nature of cyber threats, especially zero-day attacks and emerging attack vectors. Most of the existing IDS models are static, hence limiting versatility across different vehicle architectures and communication protocols (Moulahi, et al., 2021). Literature has significantly lagged behind in developing scalable, real-time, low-overhead IDS solutions, which is still in its embryonic stage. This research investigates how machine learning techniques, especially ensemble learning, can be used to develop better detection accuracy, adaptability, and scalability in IDS of vehicular networks.

The efficiency of the IDS in vehicular networks is influenced by factors such as the nature and complexity of cyberattacks, real-time processing capability, and scalability across different vehicle models. In designing the machine learning models, intrusion detection accuracy has to be implemented along with computational efficiency so as not to impact vehicle performance significantly. Besides, IDS have to be designed to adapt to the constantly changing methods of attack with the modulus of long-term security in a never-changing threat environment. The performance of the intrusion detection system it is greatly influenced by the selected machine learning algorithms, feature selection methods, and quality of data.

1.3 Research Question

How can real-time speed and minimal overhead be combined with an intrusion detection system's ability to efficiently identify and counteract cyberattacks on in-car networks?

1.4 Research Objectives

The research aims to address several core objectives to answer the above research question comprehensively:

1. State of the Art: The paper reviews the state of the art in IDS for vehicular networks, considering machine learning techniques applied for intrusion detection.
2. Implementation of Machine Learning Techniques: Use machine learning algorithms including Ensemble Learning: Random Forest and Voting Classifier to improve the detection rate of the IDS.
3. Performance Evaluation of IDS: Check the performance of the system in terms of accuracy and real-time processing for two datasets.

1.5 Contribution

The major contribution this research provided is toward the provision of a robust, scalable IDS framework for vehicular networks, embedding machine learning techniques to enhance detection accuracy and adaptability. The system is designed for real-time execution with low computational overhead so that protection would not degrade the vehicle's performance. The research substrated insights into features and attack vectors that were

highly useful, important for vehicle security, and contributed much value to the scientific repository on vehicular cybersecurity. The other improvements that the research sought to address included scalability across models of vehicles and the capability for adaptation to new kinds of cyber-attacks.

1.6 Structure of the Report

The paper is structured as follows: after an in-depth review of state-of-the-art literature on intrusion detection systems and machine learning applications in vehicular networks, in second section the methodologies are discussed, henceforth presenting data collection, feature selection, and the machine learning algorithms themselves. The fifth and sixth sections describes the implementation and performance evaluation of the proposed IDS, which covers the performance metrics: accuracy, precision, recall, and computational overhead. The paper concludes by summarizing the findings, contributions, and potential directions for future research in vehicular cybersecurity.

2 Related Work

Connected and autonomous vehicle improvements have brought unsurpassed development in transportation with a great deal of cyber security threats, especially within in-vehicle networks like the Controller Area Network bus (CAN). The CAN bus is a vehicle communication network that is robust and features microcontrollers and devices in the vehicle that can communicate with each other without the use of a host computer.

The main focus of research, therefore, is placed on IDS based on machine learning and deep learning techniques in spotting such threats. This literature review describes the state-of-the-art of ML and DL-based IDS for automotive security, identifies strengths and limitations of different approaches, explains challenges in real-time applications, and points to possible solutions with hybrid models.

2.1 Machine Learning and Deep Learning in CAN Bus Security

According to Almehdhar et al. (2024), several deep learning techniques have demonstrated high accuracies, such as autoencoders, GANs, and transformers in threat detection on CAN bus networks. However, these techniques are mainly restricted for real-time applications due to their high computational overhead. This paper advocates shifting from the classical signature-based IDS to AI-driven anomaly detection systems and further proposes hybrid techniques that integrate deep learning with federated learning for better adaptability. In a similar vein, Narasimhan et al. (2021) argue that unsupervised deep learning models, like autoencoders combined with Gaussian Mixture Models, outperform prior techniques while still being too resource-intensive and hence inapplicable for real-time applications.

Yang et al. (2020) and Aldhyani and Alkahtani (2022) target the cybersecurity vulnerabilities regarding the in-vehicle CAN bus. Yang et al. propose an RNN-LSTM model to detect spoofing attacks runtime using an ECU fingerprint signal. The proposed model shows very high computational efficiency running on FPGA platforms. On the other hand, Aldhyani and Alkahtani propose a hybrid model of CNN-LSTM for detecting several types of attacks on the CAN bus; the performance has been excellent, showing an accuracy of

97.30%. Both of these works stress deep learning to upgrade the security level of the CAN bus. Yang et al., however, focused on fingerprint signals, while Aldhyani and Alkahtani did work with an emphasis on message attack detection.

Lin et al. (2022) presented a deep learning-based IDS for IVNs by using VGG16 and XGBoost classifiers, which attained detection accuracies of 97.82% and 99.99%, respectively, on the HCRL Car-Hacking dataset.[1] The authors emphasize the detection of different DoS and spoofing threats in IVNs. Similarly, Hossain et al. (2022) utilize an LSTM-based IDS to help in mitigating attacks against a CAN bus network. The authors report 99.99% detection accuracy with their private dataset. Both papers pinpoint the effectiveness of deep learning in IVN security, giving considerable emphasis to improving threat detection in automotive networks that are continuously becoming complex.

2.2 Hybrid Approaches for Intrusion Detection

Hybrid IDS models address the current challenges with machine learning or deep learning models alone. Researchers Alsarhan et al. (2021) proposed a hybrid IDS based on rule-based filters, Bayesian learners, and Dempster-Shafer theory for intrusion detection in VANETs. This model runs with reduced false positives as the trust-based decisions, but it still suffers from the challenge of scalability in various vehicle models. Similarly, Zhang and Ma (2022) have developed a hybrid IDS that combines rule-based approaches with machine learning to reduce computational overhead to a minimum while gaining enhanced detection accuracy.

Among the reviewed papers, Basavaraj and Tayeb (2022) proposed a lightweight IDS for invehicle networks, which can target the detection of attacks like reconnaissance, DoS, and Fuzzing against a vehicle's CAN bus. Their solution leveraged real-time CAN data, which showed very good performance with respect to accuracy and other metrics considered for evaluation. Bozdal et al. (2020) also dwell on the issues of security in CAN bus, stating that the lack of encryption and authentication exposes the protocol to a variety of cyberattacks. They give a notice call for urgent need in advanced IDS solutions. Dong et al. (2023) expand on earlier work, presenting an intrusion detection system with a multiple observation hidden Markov model in CAN bus anomaly detection. Their model provides a very significant performance improvement in the detection of several attack scenarios. Finally, Pascale et al. propose, in 2021, an IDS based on a Bayesian network. The proposed system aims at the cyber-attacks on the connected vehicles. The system embeds spatial and temporal message analysis hence has effective detection in a number of attack scenarios. On the other hand, it has low accuracy under specific conditions. Cumulatively, these papers raise awareness of the escalating demand to create advanced IDS models for the protection of vehicular communication systems against various growing cyber threats in both connected and autonomous vehicles.

[1] HCLR Car-Hacking dataset https://ocslab.hksecurity.net/Dataset/CAN-intrusion-dataset

2.3 Real-Time Detection and Computational Challenges

The main obstacle of deploying deep learning-based IDS in automotive networks is computational complexity. As Bangui and Buhnova (2021) mentioned, the employment of deep learning models significantly enhances vehicle network security, while their high demands in computation and energy make it challenging for real-time implementation. Cheng et al. (2022) propose STC-IDS, a vehicle intrusion detection system, specifically designed with spatial-temporal correlation and attention-based networks to enhance the accuracy of anomaly detection compared to predecessors. The multi-frame model assurance for real-time detection, but regarding limitation, it remains unknown attack patterns. In the same vein, Cheng et al. (2022) devise TCAN-IDS using temporal convolutional neural networks combined with global attention to detect intrusion in vehicular networks. Spatial-temporal details are captured, and false positives are reduced, at the cost of extremely high computational complexity that may impede large-scale real-time deployment.

Bi et al. (2022) put forth the message and time transfer matrix-based intrusion detection method to break or mitigate computational and accuracy constraints in the ECUs. The proposed approach of the authors achieves high accuracy with optimized computational resources. It is further effective even in high-frequency attack injections and so may provide an enhancement to traditional approaches.

Mourad et al. (2020) introduced a VEC fog-enabled scheme in 2021, whose main focus is on overcoming the intensive computational needs of traditional intrusion detection systems in intelligent vehicles. The idea behind this economic system is to offload intrusion detection tasks to neighborhood-federated vehicles for latency, energy consumption, and survivability enhancement. Their solution demonstrates effective performance within real-world vehicular fog environments.

Ma et al. (2022) propose a lightweight neural network system that performs real-time intrusion detection in the CAN bus using a GRU-based architecture. The system leverages the power of invehicle embedded devices with open datasets for low-latency, high classification performance. The real-time performance and deployment efficiency of the system are demonstrated within the study and are highlighted as a strong solution for CAN intrusion detection in modern automotive.

2.4 Enhancing Security in Vehicle Networks

Several works show the potential of machine and deep learning in improving vehicle network security. Karthiga et al. (2022) have highlighted an IDS that combines ANFIS and CNN with 98.6% detection accuracy, especially for DoS attacks. A hybrid deep learning model combining LSTMs and GRU reached an accuracy of 99.5% for real-time DDoS detection. Bakhsh et al. (2023) proposed a deep learning IDS for IoT, showing 99.93% accuracy. Alladi et al. (2022) reviewed some blockchain applications that guarantee better decentralization and transparency. Makarfi et al. (2020) investigated RIS for enhancing the physical layer security. Zhang et al. (2021) proposed an ensemble learning algorithm in 6G vehicular IDS, which reduced false positives.

The key papers are summarized in the Table 1 below

Paper Title	Authors	Focus Area	Key Methods	Research Gaps/Limitations	Proposed Improvements (Compared to Proposed Project)
Deep Learning in the Fast Lane: A Survey on Advanced Intrusion Detection Systems for Intelligent Vehicle Networks	Almehdhar et al. (2024)	Deep learning methods for IVN security; Emphasis on CAN protocol	Deep learning, anomaly-based detection, federated learning, transformers	High computational overhead in deep learning models	Lightweight ensemble learning for real-time processing
Unsupervised Deep Learning Approach for In-Vehicle Intrusion Detection	Narasimhan et al. (2021)	Unsupervised deep learning for CAN intrusion detection	Autoencoders, Gaussian Mixture Model (GMM)	Real-time application is limited due to computational complexity	Focus on Random Forest and Voting Classifier for real-time detection
Machine Learning-driven Optimization for Intrusion Detection in Smart Vehicular Networks	Alsarhan et al. (2021)	Hybrid IDS using rule-based filters and Bayesian learners	Rule-based filters, Bayesian learning, Dempster-Shafer theory	Scalability across different vehicle models not explored	Continuous learning and scalable models for diverse vehicle models
Recent Advances in Machine-Learning Driven Intrusion Detection in Transportation: Survey	Bangui & Buhnova (2021)	Machine learning IDS in VANET and UAV-aided networks	Machine learning, anomaly detection, UAV-aided IDS	Challenges in big data analysis and high computational complexity	Focus on reducing false positives and enhancing real-time adaptability

Table 1: Literature Review Table

2.5 Research Gap and Difference

The fundamental difference in this project with respect to existing research is that it focuses on lightweight models that are optimized for real-time detection. While the deep learning models of prior related studies emphasized high demands on computational resources, this particular project takes efficiency into consideration, hence being best suited for automotive systems that require real-time processing. This, if anything, tames the scaling problems of prior studies, such as Narasimhan et al. (2021) which narrowed down to attack

types or vehicle model varieties. In contrast, the present study enables the proposed method with more scalability across heterogeneous types of vehicles and attack vectors to come up with adaptable ensemble learning models. It also handles the generalisability issue, which Almehdhar et al. illustrated in 2024, by applying mutual information and select best for feature selection to enhance robustness across different datasets. This now provides major steps forward in reducing computational overhead, a common issue in deep learning-based systems, by using efficient machine learning models that keep memory and processing requirements low.

This also introduces the possibility of improved attack adaptability based on a continuous learning system able to evolve with new threats. Ensemble learning can make the system more robust, since these benefits in combining accuracy and robustness do not come with a significant computational cost. There are still some challenges related to the management of ensemble complexity and the assurance of smooth model updates in strict real-time, resource-constrained contexts, an interesting point for future research work.

3 Research Methodology

3.1 Research Design

This work employed a quantitative experimental design to ascertain whether machine learning algorithms can effectively identify attack-free datasets from DoS attack datasets. The experimental design allowed iterative model testing and fine-tuning of its parameters to deliver robust results.

3.2 Data Sets

The datasets used in this research were real-time automotive network data. Further, the data consisted of attack-free[2] records and those labeled as DoS attacks[3]. The attack-free dataset contained over 2.2 million entries, while that of the DoS was over 650,000 entries.

3.3 Preprocessing of Data

The initial exploration showed the presence of inconsistencies, such as missing values and unprocessed formats; thus, cleaning and transformation were called for:

- For missing values in the Data column of the DoS dataset, a default hexadecimal string was imputed that indicates null payloads. The columns of data were normalized to be compatible for processing.
- For example, the Data column was made into lists of integers by hexadecimal decoding.

[2] Acttack_free_dataset https://www.dropbox.com/scl/fo/8kll7yvbgogkp0vahowvm/AKBKujyHfjh202zzLpBcdb0/Attack_free_dataset.txt?rlkey=43n570cnodtq6yls139r4yvn7&e=1&st=vcb22fsi&dl=0

[3] DoS_attrack_dataset https://www.dropbox.com/scl/fo/8kll7yvbgogkp0vahowvm/ADhDIC8LRFL8wHUexib3C3w?e=1&preview=DoS_attack_dataset.txt&rlkey=43n570cnodtq6yls139r4yvn7&st=iys945ng&dl=0

Then, insignificant columns like RTR within the attack-free data were removed to save the computation cost and reduce noise.

3.4 Exploratory Data Analysis

EDA was performed to understand the distribution and relationship present in the data. Descriptive statistics were created for relevant features: timestamp and DLC. These provide the visual of the histogram distribution in both attack-free and DoS datasets with regard to such attributes, showing variations that may inform an approach toward modeling. Trends of missing data were analyzed and handled.

3.5 Feature Engineering

Feature engineering was done in the interest of both model explainability and improvement; the steps taken included:

1. Transformation of Data Column: The values of Payload were transformed into numeric features.
2. Feature Selection: Mutual information scores were among the statistical techniques used to rank the strengths of the engineered features.

3.6 Data Balancing

Class imbalance was dealt with in this dataset, with instances of attack-free significantly outweighing instances of DoS attack cases. This was handled using the Synthetic Minority Over-sampling Technique.

3.7 Model Development

Two machine learning classifiers were developed and evaluated:

- Random Forest Classifier: This is one of the strongest ensemble learning algorithms using bagging, thus overfitting would be reduced to a minimum, and accuracy would be quite high. Class weights were adjusted to handle residual imbalances.
- Voting Classifier: The hard voting ensemble is taking predictions from a Random Forest and Logistic Regression classifier.

The model development process included: Splitting: Data is divided into 70% training and 30% test subsets. Feature Scaling: The standardScaler technique was used for the normalization of the numerical features.

3.8 Model Evaluation

A classifier's performance was calculated based on the following performance metrics:

- Accuracy: It is the ratio of correctly identified instances.
- Precision: Fraction of the positive instances correctly predicted.
- Recall: The percentage of actual positive instances detected.
- F1-Score: Harmonic average of precision and recall.
- Confusion Matrices: It displays the outcome of the prediction in terms of true positives, true negatives, false positives, and false negatives.

3.9 Ethical Issues

The datasets used were anonymized in order to narrow down those that comply with the standards of privacy. Secondly, there is active monitoring for bias in model outputs through feature importance analysis on balanced dataset performance. High regard was given to transparency about any preprocessing and modeling done, to guarantee replicability.

3.10 Limitation

Despite preprocessing and feature engineering, some limitations remained. This synthetic data, created with SMOTE, is unlikely to exactly give a true distribution so far as realistic scenarios go, further deteriorating generalization. Feature selection was based on mutual information scores; as such, subtle but relevant attributes may have been excluded.

4 Design Specification

4.1 Overview of System Design

This section describes the architectural and methodological framework used in the design specification for the detection of DoS attacks in automotive networks. Each subcomponent of the system is connected with others for the overall data ingestion, preprocessing, feature extraction, model training, and classification. These diverse components have to work together in cohesion to ensure valid and effective detection capability of malicious activities present in network traffic.

4.2 System Requirements

The following functional and non-functional requirements are addressed by the system:

Functional Requirements:

- Support for heterogeneous automotive data sets, including variable structure payloads.
- Handles imbalanced datasets with the help of advanced resampling techniques.
- Provide feature engineering and selection to enhance interpretability and accuracy.
- Deployment of high classification-performing machine learning algorithms.
- Output of the full metrics, including confusion matrices and performance scores.

Non-functional Requirements:

- High computational efficiency for real-time or near real-time detection.
- Scalability to datasets with millions of entries.
- Robustness to incomplete or noisy input.
- Modularity for maintainability by design.

4.3 Framework Architecture

The key elements for the detection framework are as follows:

1. **Data Ingestion Layer:**
    - It is responsible for loading datasets from storage and normalizing raw inputs for further processing.
    - Includes handling of missing or corrupted entries to get a clean dataset for analysis.

2. **Preprocessing Module:**
    - Cleans and transforms raw data by:
    - Addressing missing values with default placeholders.
    - Standardize and normalize numeric fields, such as Timestamp and DLC.
    - It encodes the Data field into numerical lists for computational processing.

3. **Feature Engineering Module:**
    - Extract statistical features from payload data minimum, maximum, mean, and standard deviation.
    - Filter columns by variance and significance, removing RTR and other such irrelevant or redundant information.

4. **Data Balancing Module:**
    - Uses the Synthetic Minority Over-sampling Technique (SMOTE) to handle class imbalances.
    - Generates artificial examples of minority class to match the distribution of the majority class.

5. **Model Training and Optimization Layer:**
    - Implements machine learning algorithms, including:
        - **Random Forest Classifier**
        - **Voting Classifier**
    - Performs hyperparameter tuning for the modification and fine-tuning of algorithm performance.

6. **Evaluation and Reporting Layer:**
    - It is used to evaluate the models using various metrics including accuracy, precision, recall, and F1-score.
    - Generates visualizations of the confusion matrices and feature importance rankings.

4.4 Algorithmic Design

A feature extraction mechanism-based detection is done through supervised learning algorithms. The process flow is as follows:

1. **Data Preparation:**
    - Loading raw datasets and cleaning.
    - Encode payload data into numerical representations.
    - Balance classes of the dataset using SMOTE to ensure their equitable learning across categories.

2. **Feature Engineering:**
    - Extract statistical features from these preprocessed payloads, which may improve model interpretability and predictive power.

3. **Model Implementation:**
   - **Random Forest Classifier:**
     - Generates multiple decision trees during the training process.
     - Aggregates outputs from individual trees to determine the final classification.
     - Handles imbalanced datasets by using class weights, which allow it to focus more on minority classes.
   - **Voting Classifier:**
     - This model combines the predictions of the classifiers Random Forest and Logistic Regression.
     - Uses hard voting to predict the majority class label.

4. **Performance Evaluation:**
   - Split data into training and testing subsets for validation of models.
   - Normalize features using StandardScaler technique.
   - Evaluate predictions against actual labels for the model, presenting detailed metrics and visualizations.

4.5 Systems Architecture Diagram

The architecture embeds the components mentioned above into a smooth pipeline, starting from data intake to actionable classified output. This pipeline provides flexibility for different datasets and machine learning models in a deployable environment.

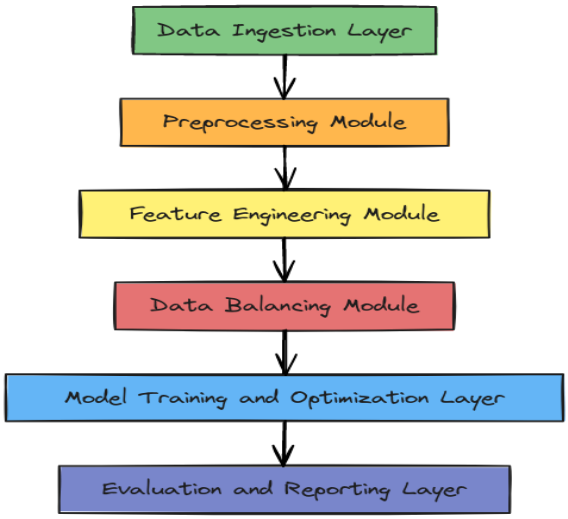

Figure 1: System Architecture Diagram

4.6 Design Considerations

The proposed system design takes into consideration a number of challenges that normally characterize data to be analyzed, such as class imbalance, high data dimensionality, and noise. Machine learning model selection and preprocessing techniques are informed by scalability, efficiency, and accuracy considerations. This will enable the easy integration of new algorithms or features in a modular design.

4.7 Summary

The methodologies, architecture, and technologies presented in this section are the backbone for designing and developing a DoS detection system. The architecture will make the system reliable, scalable, and accurate, thus laying a solid platform for practical implementations and industrial applications.

5 Implementation

This chapter elaborates on the implementation details of the procedure to be followed for detection using a machine-learning technique for DoS attacks in automotive networks. The whole implementation falls into various phases: data ingestion, preprocessing, feature engineering, the training of the model, and developing an evaluation framework. Each stage here is designed considering scalability, robustness, and precision.

5.1 Data Handling

Data Loading and Exploration

Loading of both attack-free and Dos-attack representation datasets was done as the first step. Each dataset consisted of, though not limited to, the fields: timestamp, identifier, payload length, and raw hexadecimal payload. Data ingestion was performed by using Python's Panda's library in structured formats known as DataFrames, which thus enabled seamless manipulation of the data.

Descriptive statistics and exploratory analyses were done to understand the distributions, correlations, and anomalies of the data. Missing values in the DoS dataset have been identified and treated accordingly to ensure the integrity of the data for further processing.

Handling Missing Data

The Data column was missing in the DoS dataset. These were replaced by a placeholder hexadecimal string indicating an empty payload.

5.2 Preprocessing

Data Transformation

The raw payloads, stored in hexadecimal format, first needed to get transformed into numerical representations. Python's ast module was helpful to convert string representations into lists of integers. It was this transformation that prepared the data for numerical operations that were to follow for feature extraction.

Data Cleaning

The highly sparse columns-for example, the RTR column from the attack-free dataset-were removed to reduce the computational burden. This also reduces noise and thus could improve model performance.

Dataset Balancing

SMOTE-a technique used to balance the class imbalance in a dataset-was applied. This method generates synthetic examples for the minority class, namely DoS attacks, balancing the data so models can learn equitably across classes.

5.3 Feature Engineering

Feature Extraction

To extract meaningful attributes, statistical features were extracted from the numerical payload data:

- Min Value: The minimum value of the payload.
- Maximum Value: The highest value of payload.
- Mean Value: The average of payload values.
- Variance / Standard Deviation: A measure of dispersion in payload values.

These features were representatives of the characteristics of the payload, thus allowing appropriate classification.

Feature Selection

The method of SelectKBest based on mutual information was used to review feature relevance. Scores were computed with the view of prioritizing features so that only the most significant attributes contributed toward the training of the model.

5.4 Model Development

Training and Testing Split

The balanced dataset was split into a training set of 70% and a test set of 30% using train_test_split. This separation ensured that model evaluation would be fairly performed on data the model had not seen.

Machine Learning Algorithms

Two machine learning models are implemented and fine-tuned:

- **Random Forest Classifier:** Another popular ensemble technique, training a lot of decision trees and combining outputs during predictions. The class_weight parameter was modified with small residual class imbalances to take into consideration getting equal model attention across the labels.
- **Voting Classifier:** Basically, it includes the Random Forest and Logistic Regression classifiers into a model combination. As per the ensemble approach, both models complement each other with different strengths. It performed the hard vote to return the majority class prediction from within the output.

5.5 Training the Model

Training the Random Forest

The Random Forest Classifier will be trained on a scaled feature set. Its two most important hyperparameters are optimized - the number of trees and depth - to reach an optimal spot where computation efficiency meets prediction accuracy.

Training Voting Classifier

The Vote Classifier combined the predictions of both the Random Forest and Logistic Regression models. This turned out to be tough since Logistic Regression contributes its linear decision boundary to the nonlinear capability of Random Forest.

5.6 Implementation Workflow

Integrated Pipeline

An integrated pipeline that automates this workflow was built:

- Data Ingestion: It can load and structure different datasets automatically.
- Preprocessing: Cleaning, transformation, and balancing performed sequentially in a chain.
- Feature Engineering: Statistical feature extraction and selection.
- Model Training: Classification by random forest and voting classifiers, followed by automated hyperparameter tuning.
- Evaluation Framework: Metric output and confusion matrix on validation.

5.7 Challenges Addressed

- **Dataset imbalance:** One of the very important parts of the work in reducing the biases of the model predictions comes with balancing the datasets. Then, SMOTE proved efficient in synthesizing new samples for the minority class, which in turn will prevent the classifier from being biased toward the majority class.
- **High-Dimensionality:** Feature extraction reduced the dimensionality of payload data whereby this allowed one to train without a loss of performance efficiently.
- **Noise in Data:** Cleaning steps such as low variance column removal, and missing value treatments balanced the noise in the data.

5.8 Summary

Implementation means translating the conceptual model into an actual system that performs the detection of the DoS attack while cleaning and structuring data proper feature engineering is performed to train a generalizable model that will be able to adapt to real-world scenarios. This pipeline is more general and gives the grounds for the evaluation and deployment phases.

6 Evaluation

This section provides a detailed analysis of the experimental results related to intrusion detection on in-vehicle CAN bus data while under attack and attack-free conditions. Performance metrics and implications involving machine learning model performance are presented herein to identify anomalies in the CAN bus data caused by cyberattacks, especially involving DoS attacks.

6.1 Data Distribution Analysis

This section aims to understand the characteristics of the data sets used in this study. It performs statistical and graphical analyses on both attack-free and DoS attack data sets.

6.1.1 Timestamp Distribution

The timestamp distribution of the attack-free dataset showed an even trend, indicating that the data are transmitted on a regular periodic basis. On the contrary, the timestamp distribution of

the DoS attack dataset shows an uneven mode with bursts that signal the abnormal traffic behavior caused by the attack. The differences in timestamp distribution represent one potential characteristic for distinguishing between normal and attack.

6.1.2 DLC (Data Length Code) Distribution

The DLC values of the attack-free dataset were mainly concentrated on 8, which was attributed to the traditional practice of data transmission. In the DoS attack dataset, the DLC distribution also showed a dominance of 8, though with some abnormalities and lower values. These differences further pinpoint the impact of DoS attacks on network communication.

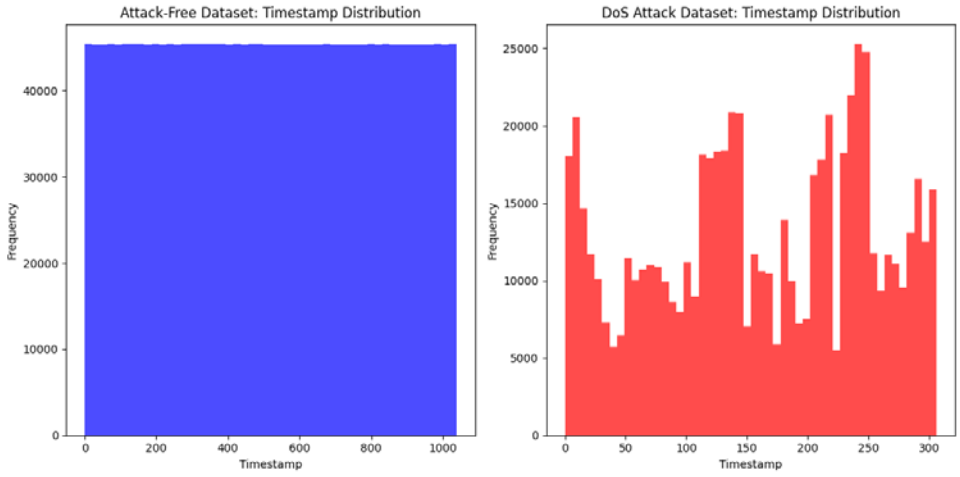

Figure 2: Timestamp Distributions

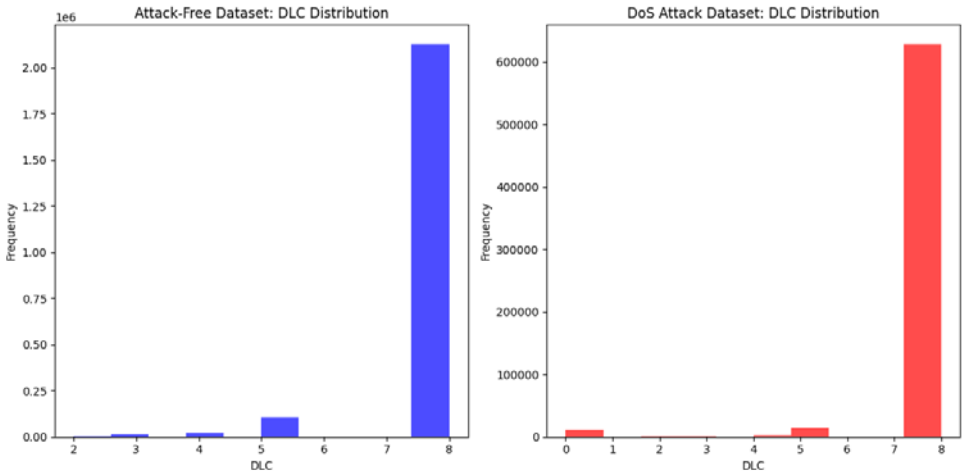

Figure 3: DLC Distributions

6.2 Feature Engineering and Preprocessing

Feature engineering was the most elementary step in transforming raw CAN bus data into a meaningful input technical tool to be used by the machine learning model. Some of the major features extracted include:

- Min Data Value: Smallest byte value in the data frame.
- max data value: Maximum byte value in the data frame.
- Mean Data Value: The mean value at any byte position in the data frame.
- Std Dev of Data (Std Data): Variance in the value of bytes.

Feature selection through mutual information ranked max_data, mean_data, and std_data in descending order of importance for the task of telling apart attack and non-attack situations, which confirmed our hypothesis that anomalies in data patterns are the most indicative of network anomalies.

To handle class imbalance between attack-free and DoS attack samples, SMOTE was employed, preparing a balanced dataset of both classes with equal representation. This balanced dataset ensures unbiased model training and evaluation.

```
Class distribution before SMOTE:
 label
0    2268519
1     656579
Name: count, dtype: int64
Class distribution after SMOTE:
label
0    2268519
1    2268519
Name: count, dtype: int64
```

Figure 2: SMOTE Applied

6.3 Model Evaluation - Random Forest Classifier

The Random Forest Classifier was chosen because it can handle complex, nonlinear relationships and is not seen to overfit on high-dimensional datasets easily. The model achieved the following performance:

```
Random Forest Results:
Accuracy: 0.896103333157007
Precision: 0.9124289163479785
Recall: 0.8764132902896051
F1-Score: 0.8940585433021611
Confusion Matrix:
 [[622984  57270]
 [ 84145 596713]]
```

Figure 3: Random Forest Results

The confusion matrix of this optimal RF classifier gives very good performances of the classifier with low values of false positives and false negatives. Similarly, these balanced precisions and recalls are indicative of better performance of the model in identifying the DoS attack without overestimation toward benign instances.

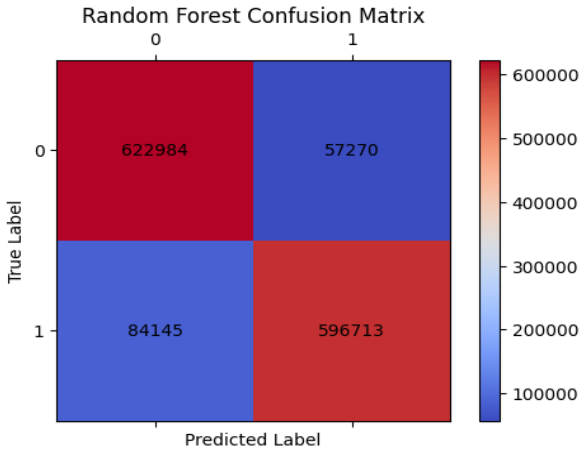

Figure 4: RF Confusion Matrix

6.4 Model Evaluation - Voting Classifier

A Voting Classifier was then used to combine the Random Forest with the Logistic Regression to take full advantage of their respective complementary strengths. However, this model gave a high precision of (88.84%), whereas the recall was (59.86%), which is quite a bit worse than the above Random Forest model, thus yielding:

```
Voting Classifier Results:
Accuracy: 0.761607420991072
Precision: 0.8884197447439265
Recall: 0.5986079329316832
F1-Score: 0.7152726407999909
Confusion Matrix:
 [[629066  51188]
 [273291 407567]]
```

Figure 5: Voting Classifier Results

The confusion matrix also showed a high rate of false negatives, meaning the model failed to detect any instance of an attack. That could suggest further optimization may be necessary for the ensemble methods to function effectively in a real-time CAN-bus environment.

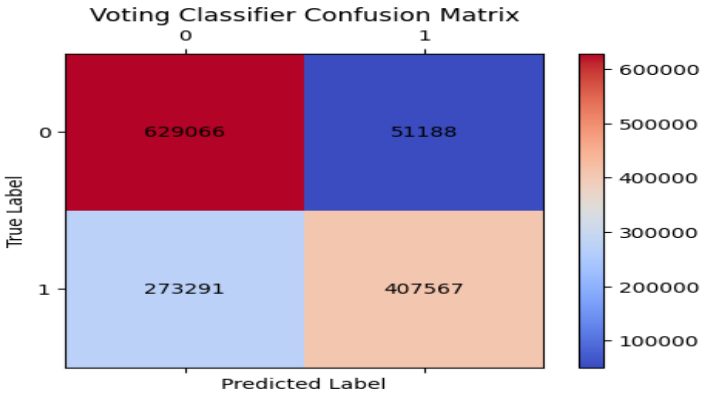

Figure 6: Voting Classifier Confusion Matrix

6.5 Discussion

These results are important for gaining insight into the performance and shortcomings of machine learning-based intrusion detection systems for CAN bus networks. Evaluations

revealed that the Random Forest Classifier outperformed the Voting Classifier in detecting Dos attacks with higher recall and F1-scores. This further builds on existing literature on how effective Random Forest is in modeling nonlinear relationships common within CAN bus traffic. Critical areas for further development were found in this study, mainly for a practical deployment in real scenarios.

Class imbalance was dealt with the help of SMOTE. This resulted in the introduction of artificial data into the dataset that can potentially make the model even less generalized to real life. For this, possible alternatives could be either using ADASYN or instead weighted loss function. What badly performed was the performance of the Voting Classifier and it was mainly caused because of low recall, enlightening a number of weaknesses associated with combining such models, some of which could also have weighted voting as its remedy or using Gradient Boosting.

The study placed a major focus on the study of DoS attacks; hence, generalization of results to other attacks like spoofing or replay attack is limited. In the future, Yang et al. propose extending IDS coverage for a wide variety of attack types through research. Besides, the high computational cost of Random Forest prevents its real-time deployment in resource-constrained environments. The lightweight architectures and FPGA acceleration discussed by Ma et al. 2022 and Yang et al. 2020 could overcome these limitations. While the results align with state-of-the-art advancements, scalability, efficiency, and broader attack coverage remain open challenges that require a hybrid approach and real-time optimizations toward practical automotive IDS deployment.

7 Conclusion and Future Work

The efficiency of various machine learning algorithms was reviewed in this paper for improving intrusion detection systems in vehicular networks. It described intrusion detection, focusing on the issue of DoS attacks in the CAN bus system, and how to combine real-time speed and minimum overhead with the capability of IDS in the detection and neutralization of cyberattacks.

Among those, the best performance was done by Random Forest. Its accuracy reached 89.61%, while Recall attained 87.64% and the F1-score 89.41%. The good result here shows it can handle the non-linear variability typical of feature engineering for CAN bus traffic-Metric means: mean_data, standard: std_data or max: max_data- will improve. Statistic handcrafted features led to poorer adaptability against new threats in performances obtained using Voting Classifier; the task becomes worse.

These findings reveal that machine learning-based IDS significantly enhances vehicular cybersecurity anomaly detection in real time. However, scalability issues, feature generalization, and computational efficiency keep them off the road to practical applications. Further emphasis on research was placed on the automatic extraction of features using deep learning, lightweight neural architecture for resource-constrained situations, and hybrid approaches for robustness against a wide range of attack types and zero-day threats. Such gaps, when addressed, would lead to significant advances in IDSs and wider adoption in automotive cybersecurity.

References

Aldhyani, T.H. and Alkahtani, H., 2022. Attacks to automatous vehicles: A deep learning algorithm for cybersecurity. Sensors, 22(1), p.360.

Alladi, T., Chamola, V., Sahu, N., Venkatesh, V., Goyal, A. and Guizani, M., 2022. A comprehensive survey on the applications of blockchain for securing vehicular networks. IEEE Communications Surveys & Tutorials, 24(2), pp.1212-1239.

Almehdhar, M., Albaseer, A., Khan, M.A., Abdallah, M., Menouar, H., Al-Kuwari, S. and Al-Fuqaha, A., 2024. Deep learning in the fast lane: A survey on advanced intrusion detection systems for intelligent vehicle networks. IEEE Open Journal of Vehicular Technology.

Alsarhan, A., Al-Ghuwairi, A.R., Almalkawi, I.T., Alauthman, M. and Al-Dubai, A., 2021. Machine learning-driven optimization for intrusion detection in smart vehicular networks. Wireless Personal Communications, 117, pp.3129-3152.

Bakhsh, S.A., Khan, M.A., Ahmed, F., Alshehri, M.S., Ali, H. and Ahmad, J., 2023. Enhancing IoT network security through deep learning-powered Intrusion Detection System. Internet of Things, 24, p.100936.

Bangui, H. and Buhnova, B., 2021. Recent advances in machine-learning driven intrusion detection in transportation: Survey. Procedia Computer Science, 184, pp.877-886.

Basavaraj, D. and Tayeb, S., 2022. Towards a lightweight intrusion detection framework for in-vehicle networks. Journal of Sensor and Actuator Networks, 11(1), p.6.

Bi, Z., Xu, G., Xu, G., Tian, M., Jiang, R. and Zhang, S., 2022. Intrusion Detection Method for In-Vehicle CAN Bus Based on Message and Time Transfer Matrix. Security and Communication Networks, 2022(1), p.2554280.

Bozdal, M., Samie, M., Aslam, S. and Jennions, I., 2020. Evaluation of can bus security challenges. Sensors, 20(8), p.2364.

Cheng, P., Han, M., Li, A. and Zhang, F., 2022. STC-IDS: Spatial–temporal correlation feature analyzing based intrusion detection system for intelligent connected vehicles. International Journal of Intelligent Systems, 37(11), pp.9532-9561.

Cheng, P., Xu, K., Li, S. and Han, M., 2022. TCAN-IDS: intrusion detection system for internet of vehicle using temporal convolutional attention network. Symmetry, 14(2), p.310.

Dong, C., Wu, H. and Li, Q., 2023. Multiple observation HMM-based CAN bus intrusion detection system for in-vehicle network. IEEE Access, 11, pp.35639-35648.

Hossain, M.D., Inoue, H., Ochiai, H., Fall, D. and Kadobayashi, Y., 2020. LSTM-based intrusion detection system for in-vehicle can bus communications. Ieee Access, 8, pp.185489-185502.

Karthiga, B., Durairaj, D., Nawaz, N., Venkatasamy, T.K., Ramasamy, G. and Hariharasudan, A., 2022. Intelligent intrusion detection system for VANET using machine learning and deep learning approaches. Wireless Communications and Mobile Computing, 2022(1), p.5069104.

Lin, H.C., Wang, P., Chao, K.M., Lin, W.H. and Chen, J.H., 2022. Using deep learning networks to identify cyber attacks on intrusion detection for in-vehicle networks. Electronics, 11(14), p.2180.

Ma, H., Cao, J., Mi, B., Huang, D., Liu, Y. and Li, S., 2022. A GRU-Based Lightweight System for CAN Intrusion Detection in Real Time. Security and Communication Networks, 2022(1), p.5827056.

Makarfi, A.U., Rabie, K.M., Kaiwartya, O., Li, X. and Kharel, R., 2020, May. Physical layer security in vehicular networks with reconfigurable intelligent surfaces. In 2020 IEEE 91st vehicular technology conference (VTC2020-Spring) (pp. 1-6). IEEE.

Moulahi, T., Zidi, S., Alabdulatif, A. and Atiquzzaman, M., 2021. Comparative performance evaluation of intrusion detection based on machine learning in in-vehicle controller area network bus. IEEE Access, 9, pp.99595-99605.

Mourad, A., Tout, H., Wahab, O.A., Otrok, H. and Dbouk, T., 2020. Ad hoc vehicular fog enabling cooperative low-latency intrusion detection. IEEE Internet of Things Journal, 8(2), pp.829-843.

Narasimhan, H., Ravi, V. and Mohammad, N., 2021. Unsupervised deep learning approach for in-vehicle intrusion detection system. IEEE Consumer Electronics Magazine, 12(1), pp.103-108.

Pascale, F., Adinolfi, E.A., Coppola, S. and Santonicola, E., 2021. Cybersecurity in automotive: An intrusion detection system in connected vehicles. Electronics, 10(15), p.1765.

Wang, K., Zhang, A., Sun, H. and Wang, B., 2022. Analysis of recent deep-learning-based intrusion detection methods for in-vehicle network. IEEE Transactions on Intelligent Transportation Systems, 24(2), pp.1843-1854.

Yang, Y., Duan, Z. and Tehranipoor, M., 2020. Identify a spoofing attack on an in-vehicle CAN bus based on the deep features of an ECU fingerprint signal. Smart Cities, 3(1), pp.17-30.

Zhang, L. and Ma, D., 2022. A hybrid approach toward efficient and accurate intrusion detection for in-vehicle networks. IEEE Access, 10, pp.10852-10866.

Zhang, Z., Cao, Y., Cui, Z., Zhang, W. and Chen, J., 2021. A many-objective optimization based intelligent intrusion detection algorithm for enhancing security of vehicular networks in 6G. IEEE Transactions on Vehicular Technology, 70(6), pp.5234-5243.