An Innovative Feature Selection Approach for CAN Bus Data Leveraging Constant Value Analysis

Muzun AlThunayyan[1,2], Amir Javed[1], and Omer Rana[1]

[1] Cardiff University, School of Computer Science & Informatics, CF24 3AA, UK
[2] Majmaah University, Al Majma'ah 15362, Saudi Arabia
AlthunayyanMS@cardiff.ac.uk

Abstract. Intrusion detection systems (IDS) serve as effective security measures within in-vehicle networks. There is a growing demand for lightweight and computationally efficient IDS solutions compatible with systems constrained by limited computing and storage resources. One way to achieve this is to use feature selection methods to reduce computational costs. However, selecting a subset of features introduces a potential vulnerability, allowing adversaries to exploit unselected features for new unknown attacks and bypass IDS. To address this concern, we propose a novel observation-driven feature selection approach for CAN bus data. This approach selects the most important features without losing valuable information and prevents adversaries from exploiting unselected features. We validate our observations using three benchmark datasets. We assess the impact of our proposed approach on the number of trained parameters, false positives, false negatives, and F1 score. We illustrate how our approach effectively addresses the risks associated with adversaries exploiting unselected features. Experimental results demonstrate that our approach reduces the number of trained parameters by approximately 44% in a machine learning model and by 14.24% in a deep learning model. Moreover, the results show that our approach helps the model detect around 8% of unknown attacks. Our approach reduces computational overhead, thereby improving overall computational efficiency. It demonstrates promising results by reducing computational resources and minimising their vulnerability to potential malicious traffic injection, thereby enhancing vehicle security.

Keywords: CAN Bus · Machine Learning · Intrusion Detection System· Feature Selection.

1 Introduction

Intrusion detection systems (IDS) are highly effective security measures for identifying malicious attacks within in-vehicle networks. There is an increasing demand for IDS solutions that are lightweight, computationally efficient, and compatible with systems that possess limited computing and storage resources. Feature selection is a pre-processing step in machine learning methods with the

aim of reducing computational time complexity. It eliminates irrelevant features while improving or maintaining the performance of IDS [1]. Feature selection offers numerous benefits, including avoiding overfitting, facilitating data visualisation, reducing model training time, and minimising storage requirements [2, 6]. However, selecting a subset of features may introduce a vulnerability because adversaries could exploit the unselected features to launch previously unknown attacks. These attacks pose a serious threat that could ultimately result in catastrophic consequences, including loss of life [10]. To address this issue, we propose a novel feature selection approach based on observations of CAN bus data that can be used with any machine learning model. The primary objective of the proposed approach is to reduce the computational costs associated with CAN bus IDS while retaining all critical information about CAN bus messages. This observation-driven feature selection not only reduces computational overhead, but also improves the understanding of researchers of CAN bus data to facilitate the development of improved solutions. Furthermore, it helps mitigate the risks associated with learning from noise, generally improves computational efficiency and simplifies machine learning models, rendering them more interpretable. This approach helps to focus the model on the most significant features, potentially improving its generalisation capability. To our knowledge, no previous work has proposed a way to reduce computational costs without losing important information about the CAN bus or creating opportunities for adversaries to introduce new manipulation attacks, such as evasion attacks and IDS bypasses [11]. The key contributions of this paper are as follows.

- This paper introduces a novel feature selection approach that prevents adversaries from exploiting unselected features to launch previously unknown attacks.
- We propose a novel feature selection approach that generalises over different datasets by validating it using different benchmark datasets.
- We propose a novel feature selection model that generalises over different machine learning algorithms.
- To measure the effect of our proposed approach in terms of the number of trained parameters and detection capability, we selected a traditional machine learning algorithm, random forest (RF) and a deep learning algorithm, artificial neural network (ANN), for evaluation, both with and without the proposed feature selection approach.
- We compare our approach with six other feature-selection methods to assess how they handle constant features. The results reveal that all tested methods treat constant values as important.

The remainder of this paper is organised as follows. Section 2 presents background information about the CAN bus data. Section 3 presents the proposed feature selection approach. Section 4 presents and discusses the experimental results and comparative analysis. Finally, section 5 concludes the paper.

2 Background

In this section, we provide background information on the CAN bus data.

2.1 Controller Area Network

The controller area network (CAN) bus protocol is the primary communication method between multiple electronic control units (ECUs). Robert Bosch developed the CAN protocol in 1985 to reduce the weight, complexity, and cost of wires. Due to its high speed and efficiency, this protocol is widely used as the default communication system in connected and autonomous vehicles. The CAN bus protocol is a message-based broadcast protocol in which ECUs transmit data in a predefined data frame as messages. The message is sent to all the ECUs because the CAN system uses a broadcast protocol. Despite its importance, the CAN bus protocol lacks security features, making it vulnerable to confidentiality, integrity, and availability attacks [7]. Figure 1 shows the standard CAN frame format, which is made up of various fields, including the following:

- **Start of Frame (SOF):** The purpose of this field is to synchronise the transmission of the CAN message with all nodes and to signal the initiation of its transmission.
- **Arbitration Field (ID):** This field, also known as the CAN ID, is used to specify the destination address of the designated ECU. It also determines the priority of the message, where a lower value generally indicates a higher priority. The ID field is 11 bits in size.
- **Data Length Code (DLC):** This field provides information about the length of the data field in bytes.
- **Data Field:** This field, also known as the payload, includes the actual vehicle parameter values, which are interpreted by the received ECU and its size can vary from 0 to 8 bytes.
- **Cyclic Redundancy Check (CRC):** This field detects errors and maintains data integrity during message transmission with a fixed size of 16 bits.
- **Acknowledge Field (ACK):** This field obtains confirmation from the receiving node that the CAN message was received correctly.
- **End of Frame (EOF):** This field signifies the completion of CAN message transmission.

In this paper, our objective is to answer the following questions.

1. Are the constant values in CAN bus data fields consistent across different CAN IDs?
2. What are the advantages associated with the removal of these constant values?

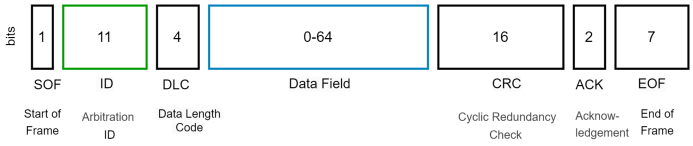


Fig. 1: CAN Data Frame

3. How can this consistency be used to reduce the risk of adversarial evasion?
4. How do other feature-selection methods handle these constant values?

3 Proposed Approach

3.1 Fundamental Idea

The fundamental idea behind the proposed feature selection approach emerged from a close examination of the data from the CAN bus. Figure 2 illustrates the stages we followed and the findings that formulated our proposed approach. We noticed that each CAN ID exhibits distinct patterns of data field values (D0, D1, D2, D3, D4, D5, D6, D7). For example, as shown in Table 1, CAN ID 399 consistently exhibits zeros in its data fields D2, D3, D4, D6, and D7, which remain constant. Meanwhile, in CAN ID 320, the data fields that always include zero are D1, D2, and D3. Table 1 highlights these constant zero values in blue, representing data that remain unchanged throughout the dataset. In contrast, the data highlighted in yellow signify changing values, while the data marked in green indicate constant non-zero values. In this paper, we focus on constant zero values, which represent data that remain unchanged throughout the dataset. This observation has two significant implications. First, significant features (possessing varying values) for one CAN ID may not hold relevance for another, making the selection of a consistent subset of important features for all CAN IDs impractical due to the inherent nature of CAN bus data. Second, this observation may be valuable for IDSs that build specific models for each CAN ID. Reduce computational resources required while preserving important information about the CAN bus. As shown in Table 2, various CAN IDs have different constant zero features, ranging from none, as seen in CAN ID 608, to all features being constant zeros, as observed in CAN ID 1072. To ensure the validity of our observation across various scenarios, we validate it using three separate datasets. In this paper, we focus on CAN IDs with constant-zero features ranging from one to seven constant-zero features. We exclude CAN IDs with no constant feature values because this approach cannot be applied to them and does not reduce the dimensionality of the model. Similarly, CAN IDs for which all data field values are zero will not have input features. Removing consistently zero feature values from the model input reduces the complexity of the model and prevents learning from noise without losing essential information about the CAN bus message. The concept here is that in IDSs designed for each CAN ID, if we

have 70 CAN IDs, we would traditionally require 70 models, each with eight data field features as input. However, by applying our approach, we can reduce the number of input features for each CAN ID by removing constant zero features during model training. However, by eliminating constant features from the input, there is a potential vulnerability, as adversaries could inject values into these removed features. Therefore, we propose a safeguard: first, verify whether the constant values are indeed zeros, as expected. If they are, they can be safely removed from the feature inputs. However, if they are not zeros, this will trigger an alert before the data enter the model. This can be applied using a rule-based model. Moreover, combining a rule-based model with a machine learning model has been shown to result in an improved ability to detect various attacks [8]. The workflow of the proposed approach is shown in Figure 3.

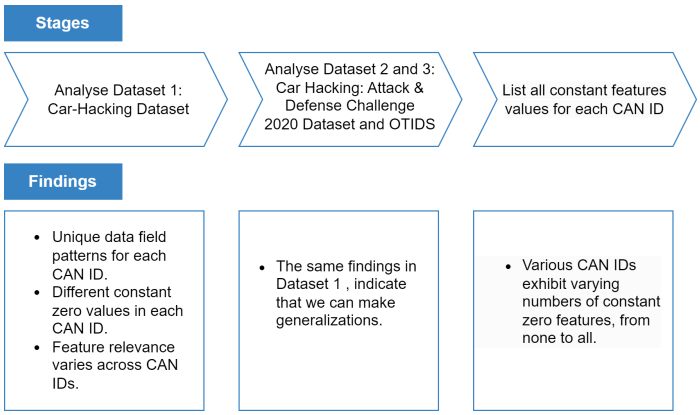


Fig. 2: Followed Stages and Findings

Table 1: Constant and Changing Data in CAN Message

CAN ID	D0	D1	D2	D3	D4	D5	D6	D7
399	254	59	00	00	00	60	00	00
399	254	60	00	00	00	61	00	00
.	.	.	00	00	00	.	00	00
.	.	.	00	00	00	.	00	00
.	.	.	00	00	00	.	00	00
399	254	94	00	00	00	74	00	00
399	254	95	00	00	00	75	00	00

Table 2: Samples of Number of Constant Zeros in Each CAN ID

Car Hacking Dataset [9]		**Car Hacking: Attack & Defense Challenge 2020 [4]**		**OTIDS Dataset** [5]	
CAN ID	**Data Field**	**CAN ID**	**Data Field**	**CAN ID**	**Data Field**
2	D[0], D[1], D[2], D[3], D[4]	67	D[1], D[5], D[6]	24	D[1],D[5]
160	D[4], D[7]	304	D[4]	66	D[1],D[4],D[5],D[6],D[7]
161	D[2], D[3], D[5], D[6], D[7]	320	D[2],D[5]	67	D[1],D[2], D[3], D[4], D[5],D[6],D[7]
261	D[1],D[3]	339	D[4]	68	D[0],D[1],D[2],D[5],D[6],D[7]
305	D[3]	854	D[0], D[1], D[2],D[5], D[6],D[7]	129	D[3],D[4],D[5],D[6]
320	D[1], D[2], D[3]	871	D[2],D[5]	160	D[4], D[6]
339	D[0], D[4], D[6], D[7]	872	D[0],D[3]	161	D[5],D[6],D[7]
399	D[2], D[3], D[4], D[6], D[7]	897	D[3],D[4]	272	D[4], D[5],D[6],D[7]
497	D[1], D[2], D[3], D[4], D[5], D[6], D[7]	903	D[4],D[7]	339	D[0], D[4]
672	D[1], D[7]	909	D[0], D[1], D[3]	356	D[0],D[2], D[3], D[4], D[5]
704	D[1], D[2], D[3], D[4], D[5], D[6], D[7]	913	D[0], D[1], D[2],D[3], D[4],D[5]	357	D[3], D[4], D[5]
790	D[6]	916	D[3]	399	D[0], D[3], D[4], D[6],D[7]
809	D[6]	1040	D[3], D[5], D[6], D[7]	848	D[5], D[6]
848	D[5], D[6]	1042	D[1], D[2], D[5], D[6], D[7]	880	D[2], D[5], D[6]
880	D[0], D[2],D[4], D[5], D[6], D[7]	1056	D[7]	898	D[3], D[4], D[5], D[6]
1072	D[0], D[1], D[2], D[3], D[4], D[5], D[6], D[7]	1057	D[2],D[6]	1087	D[7]
1088	D[1], D[2], D[3], D[7]	1069	D[0], D[1], D[2], D[3]	1088	D[2],D[3],D[7]
1201	D[4], D[5], D[6]	1151	D[4],D[6]	1264	D[0],D[1],D[3], D[4]
1264	D[0], D[4]	1162	D[0], D[1]	1265	D[2], [3], D[4], [5], D[6],D[7]
1349	D[2], D[4], D[5], D[6], D[7]	1164	D[3], D[4], D[5], D[6], D[7]	1266	D[0],D[1],D[4], D[5], D[6]
1440	D[0], D[1], D[2], D[3], D[4], D[5], D[6], D[7]	1168	D[0], D[1], D[4]	1306	D[0],D[2], D[3], D[4], D[5],D[6],D[7]
1442	D[1], D[4], D[5], D[6], D[7]	1170	D[3], D[4], D[5], D[6]	1349	D[2]
1680	D[0], D[1], D[3], D[6], D[7]	1173	D[0], D[1]	1415	D[0],D[1],D[2], D[3], D[4], D[5],D[6]
1697	D[0], D[1], D[2], D[3], D[4], D[5]	1183	D[2], D[3], D[4], D[6], D[7]	1435	D[0],D[1],D[2], D[3], D[4], D[5],D[6]
2009	D[4], D[5], D[6], D[7]	1280	D[2], D[3], D[5]	1680	D[1], D[5]
2024	D[4], D[5], D[6], D[7]	1322	D[1], D[2], D[3], D[5]	1201	D[0],D[2], D[3], D[4], D[5],D[6]

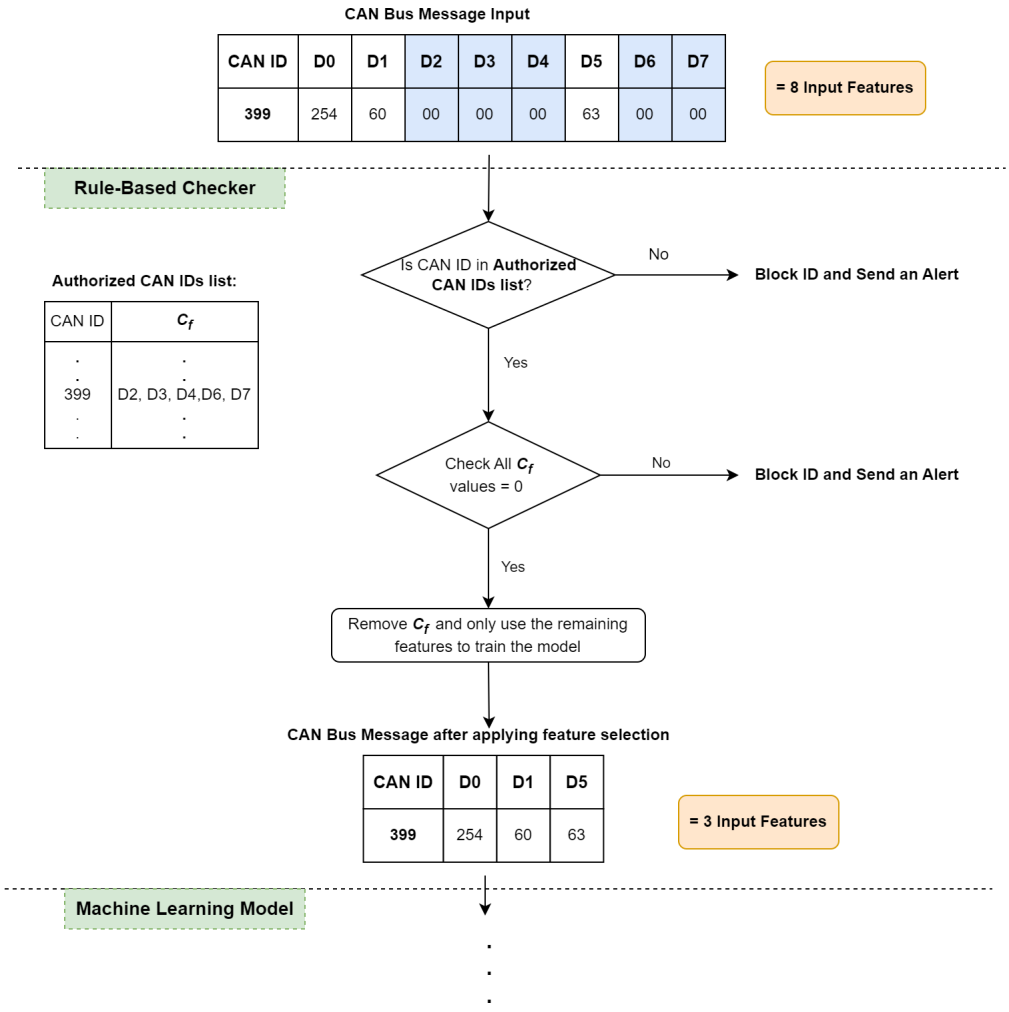


Fig. 3: Workflow of the Proposed Feature Selection Approach

3.2 Rule-Based Checker

This step requires an engineer to set it up during the design phase. The engineer will specify the authorised CAN IDs for the vehicle. For example, the list of authorised CAN IDs could include (320, 339, 399, 797, 608, 672, 704, 790, 809, 848, 880, and 1072). This practice prevents attacks involving the injection of unauthorised CAN IDs, such as denial of service (DoS) and fuzzy attacks. Any attempt to inject CAN IDs not in this list will trigger an alarm. Furthermore, for each CAN ID, the engineer will identify features with constant zero values denoted by C_f to be checked in a preliminary step before data enter the machine

learning model. This is essential because the constant zero features differ for each distinct CAN ID. For example, for CAN ID 339, where the constant features are D0, D4, D6, and D7, the input values of these features will be verified first. If they are equal to zero, the remaining features (D1, D2, D3, and D5) will be used as input features to train the model. This approach reduces the number of input features in the machine learning model, in this case, to only four, which is nearly half of the original eight data fields. This selection approach can then be applied as a pre-processing stage for any machine learning model conducted for a specific CAN ID. The pseudocode for our proposed feature selection approach is presented in Algorithm 1.

3.3 Machine Learning Algorithms

To measure the reduction in the number of trained parameters and the detection capability of our proposed feature selection approach, we select RF to represent traditional machine learning algorithms and ANN to represent deep learning algorithms, both with and without applying the proposed feature selection approach. Any machine learning algorithm can be used with our proposed approach.

Algorithm 1 Proposed Feature Selection Approach

```
Initialize authorised CAN IDs with associated constant features
authorised_CAN_IDs ← {
    CAN_ID1 : C_f1,
    CAN_ID2 : C_f2,
    ...
}
while True do
    CAN_ID, features ← receive_CAN_data()
    if CAN_ID ∉ authorised_CAN_IDs then
        Block unauthorised CAN ID and raise an alarm
        block_CAN_ID()
        raise_alarm("Unauthorised CAN ID detected!")
    else
        constant_features ← authorised_CAN_IDs[CAN_ID]
        if all feature == 0 for feature in constant_features then
            Remove constant features and send non-constant features to the model
            non_constant_features ← features − constant_features
            send_to_model(CAN_ID, non_constant_features)
        else
            Raise an alarm for non-zero constant features
            raise_alarm()
        end if
    end if
end while
```

3.4 Dataset

To ensure the robustness of our observations across diverse scenarios, we validated them using three widely used benchmark datasets in automotive security research: the Car Hacking dataset [9], the Car Hacking: Attack & Defense Challenge 2020 dataset [4], and the CAN dataset for Intrusion Detection (OTIDS) [5]. For the experiments, we utilised the Car Hacking Dataset, which includes normal traffic and four types of attacks: DoS, fuzzy, RPM engine spoofing, and drive gear spoofing [9]. Additionally, the Car Hacking: Attack & Defense Challenge 2020 dataset was used, including normal traffic and four types of attacks: DoS, fuzzy, spoofing, and replay [4].

These datasets cover scenarios with both normal and attack data, providing comprehensive information for each CAN message, including timestamp, CAN ID, DLC, data field, and flag. The timestamp indicates the precise recording time of each message from start-up. The CAN ID plays a crucial role in determining the priority of multiple messages, with lower values given precedence over higher values. In addition, the DLC specifies the length of the data field in bytes, up to 8 bytes, while the flag distinguishes whether the message is normal or an attack. In this paper, we focus on using CAN ID to categorise the data and the data field to extract features that exhibit variability (non-constant features).

4 Results and Discussion

4.1 Performance Metrics

To evaluate our proposed approach, we consider performance metrics, including false positives (FPs), false negatives (FNs), and the F1 score, along with the number of trained parameters. FPs occur when the model incorrectly identifies normal traffic as an attack. However, FNs occur when the model fails to recognise an actual attack, incorrectly classifying it as normal. The F1 score, a crucial metric for imbalanced datasets, offers a comprehensive assessment that considers precision and recall while accounting for both FPs and FNs.

The F1 score is calculated as follows:

$$F1score(F1) = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{1}$$

Where *Precision* is the ratio of true positive predictions (TP) to the sum of the TP and FP predictions, and *Recall* is the ratio of TP predictions to the sum of TP and FN instances in the dataset.

4.2 Experimental Results

FPs, FNs, F1 Score: To assess the FPs, FNs, and F1 of the models before and after applying our feature selection approach, we evaluated various models

from each dataset with different constant features. Each model corresponded to specific CAN IDs; for example, we chose CAN IDs (2, 160, 399, 1201, 1442) from the Car Hacking dataset [9] and CAN IDs (67, 320, 871, 872, 903, 909, 1164, 1208) from the Attack & Defense Challenge 2020 dataset [4].

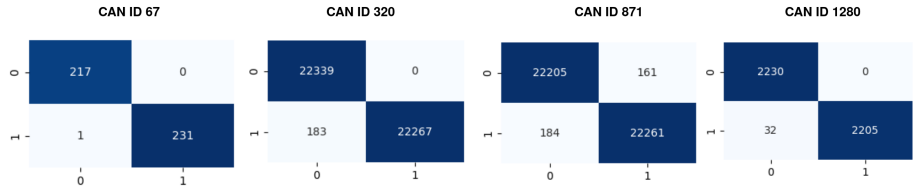
In the first dataset, before applying the feature selection approach, we trained and tested the data with all the features. All models accurately classified the data into normal or attacks without encountering any FPs or FNs, achieving a 1.0 F1 score. Subsequently, after applying our selection approach, we trained and tested each model with its selected features, and all models accurately classified the data into normal or attacks without encountering FPs or FNs. The F1 score for each CAN ID model remained consistent at 1.0 before and after applying our feature selection approach, indicating that there is no discernible impact on detection performance even after feature selection on these models.

In the second dataset, before applying the selection approach, we trained and tested the data with all features. Figure 4(a) displays the confusion matrix of the models (67, 320, 871, 1208), indicating some FPs and FNs. Subsequently, after applying our selection approach, we trained and tested each model with its selected features. As illustrated in Figure 4 (b), even with feature selection, there is a significant decrease in both FPs and FNs in these models. As a result, the F1 scores for the models (67, 320, 871, 1208) before applying the feature selection approach were 0.998, 0.996, 0.992, and 0.993, respectively. After applying the feature selection approach, they increased to 1.0, 0.996, 0.994, and 0.999, indicating that applying our feature selection approach improves the results, as it does not remove any important information.

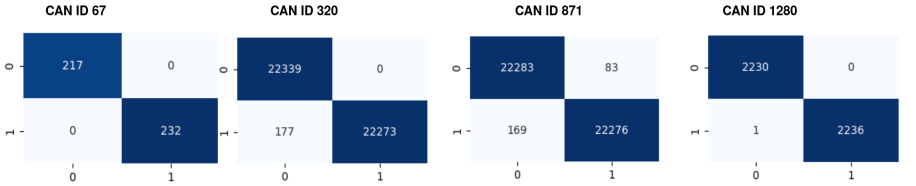
On the other hand, in the case of the other models (872, 903, 909, 1164), although the rule-based checkers filtered out most of the attacks, the model failed to classify all data correctly. There was a decrease in FNs and an increase in FPs after applying the feature selection approach. The F1 scores for the models (872, 903, 909, 1164) before applying the feature selection approach were 0.995, 0.997, 0.998, and 0.985, respectively. After applying the feature selection approach, they became 0.996, 0.995, 0.998, and 0.998. This may be attributed to the model's failure to learn the data pattern, indicating a need for further improvements.

Number of Trained Parameters: In the machine learning model, RF, each model has 90 parameters. With a total of 24 models, there are combined 2,160 parameters across all 24 CAN IDs models. The total number of trained parameters after applying the feature selection approach is 1,200. This reduces the number of parameters by approximately 44%. Figure 5 shows the number of parameters for each CAN ID before and after selection of features in the RF model.

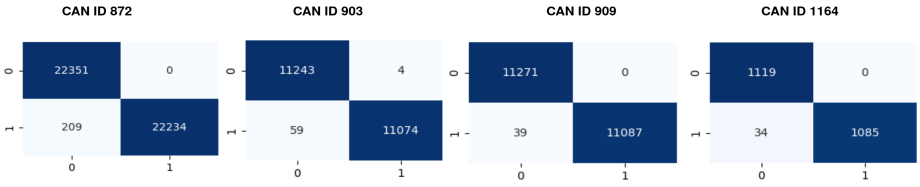
In the deep learning model, ANN, each model has 865 parameters. With 24 models, there are combined 20,760 parameters across all 24 CAN IDs models. The total number of trained parameters after applying the feature selection



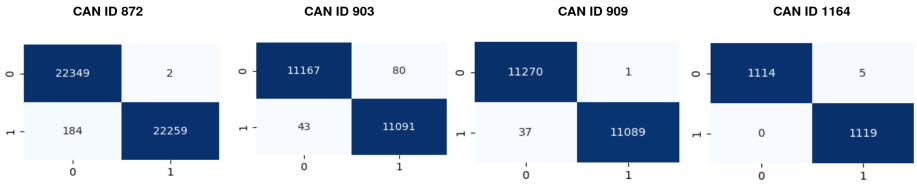
(a) Before



(b) After



(c) Before



(d) After

Fig. 4: Confusion Matrix Before and After Applying Feature Selection

approach is 17,802. This represents a reduction of 14.24%. Figure 6 shows the number of parameters for each CAN ID before and after feature selection in the ANN model. Figure 7 shows a comparison between the number of parameters before and after feature selection in machine learning and deep learning models. Furthermore, reducing the number of features reduces the dimensionality of the data, making it easier to explain, explore, and visualise.

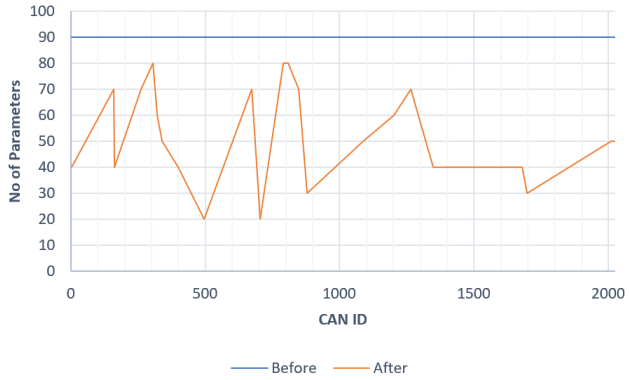


Fig. 5: Number of Parameters for Each CAN ID Before and After Feature Selection in RF Model

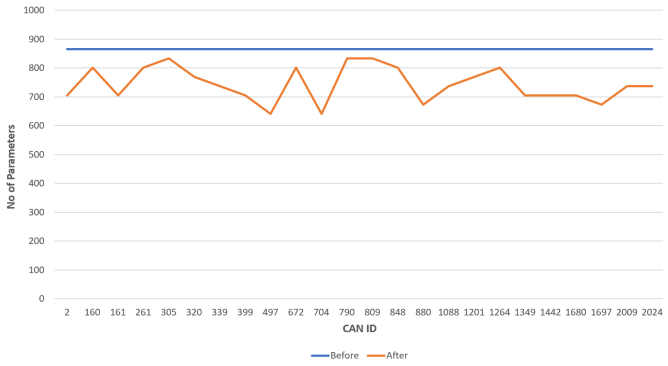


Fig. 6: Number of Parameters for Each CAN ID Before and After Feature Selection in ANN Model

Detecting Unknown Attacks To demonstrate the effectiveness of our proposed method in mitigating the risks posed by adversaries that exploit unselected

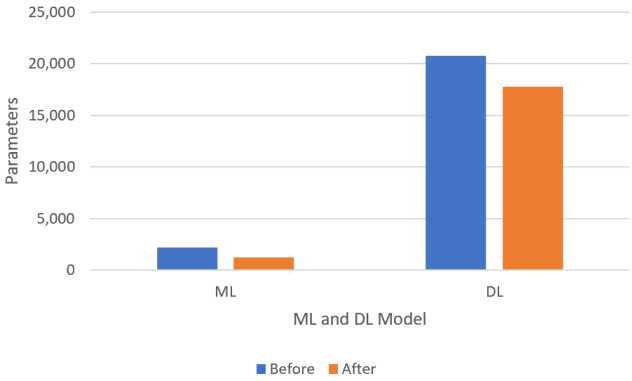


Fig. 7: Comparison between Number of Parameters in ML and DL Before and After Feature Selection

features to launch unknown attacks, we used two baseline ANN models for CAN ID 160. The choice of CAN ID 160 was deliberate because it is an authorised ID, and the dataset encompasses attacks with the same ID that carry malicious payloads, representing a potential challenge for the model.

The first model lacks our feature selection approach, while the second incorporates our methodology. To test our proposed method in detecting unknown attacks, we followed the methodology outlined in [3]. We define an unknown attack as an attack that the model has not encountered during training. To assess this, each type of attack was treated as an unknown attack by excluding its labeled data from the training set. For example, if the unknown attack is fuzzy, we trained the model with labeled data, including normal, DoS, gear, and RPM attacks, and then tested the model on fuzzy attacks.

As shown in Table 3, in the case of fuzzy attacks, where the total number is 491,847, the first model misclassifies 40,567 instances (8.25%) as normal. Conversely, in the second model that incorporates our proposed feature selection approach, the feature selection mechanism successfully identifies all fuzzy attacks. In the case of DoS attacks, all attack data bypassed the first model and was classified as normal, while the second model successfully identified all DoS attacks. In gear and RPM attacks, both models were able to detect them correctly. These findings highlight that our proposed method enhances the detection capability by minimising the risks associated with adversaries exploiting unselected features to launch unknown attacks.

4.3 Comparative Analysis

In this section, our aim is to determine whether other feature selection methods, including univariate selection, feature importance, information gain, recursive feature elimination (RFE), chi-squared, and Pearson correlation, can identify

Table 3: Unknown Attacks Results

Unknown Attack	No of Instances	Model 1		Model 2	
		TP	FN	TP	FN
Fuzzy	491,847	451,280	40,567	491,847	0
DoS	587,521	0	587,521	587,521	0
Gear	597,252	597,252	0	597,252	0
RPM	654,897	654,897	0	654,897	0

constant features in each CAN ID data as important features. We chose these feature selection methods because they are the most widely used. We apply these different feature selection methods to each CAN ID and compare the results with the non-constant features selected by our approach. Table 4 presents the important features selected by each method, including our approach. We applied these methods to a subset of five CAN IDs (2, 120, 399, 1201, and 1442) as samples to conserve space. For example, for CAN ID:2, our approach indicates that the best (non-constant) values to extract from the data fields are D[5], D[6], and D[7]. To determine whether constant features have been selected by other feature selection methods, we choose an equal number of features. For example, if our selection approach selects four features, we also select the four top features of the other methods. In univariate selection, Pearson correlation, and chi-squared methods, the important selected characteristics are D [2], D [3], and D [4], all of which are constant. In the feature importance method, the crucial features include D[7], D[3], and D[6], with D[3] being constant. Furthermore, the information gain method highlights significant features, such as D[3], D[0], and D[1], while the RFE method identifies important features as D[2], D[6], and D[7]. Based on the results, all comparable feature selection methods choose some or all of their important features, encompassing constant zero features designed to handle irrelevant information.

4.4 Discussion

While feature selection is a crucial pre-processing step before applying machine learning algorithms, it can inadvertently create opportunities for adversaries to inject malicious data into the least important and removed features. Our feature selection approach serves multiple purposes. Reduce feature size, eliminating constant zero values that could lead to overfitting and noise in the model. Additionally, it improves security by preventing adversaries from injecting new manipulation attacks. Our approach promptly examines these features when the data enter the network to determine if they are zeros, and triggers an alarm if necessary. Furthermore, if adversaries attempt to inject new manipulation attacks into the constant zero features, our rule-based checker quickly identifies and blocks these attacks from entering the model. This not only improves security, but also reduces latency in detecting unknown attacks, which is a persistent challenge for IDSs. The main finding of this paper is that for each CAN ID, there

are constant values that never change in the CAN bus data. Compared to all feature selection methods we tested, these zero-constant features are chosen as important and remove genuinely important features instead. Therefore, security solution designers should carefully choose the right feature selection method. We leverage this observation to ensure that we selected the most important features, limiting the adversary's ability to inject in the removed features and reducing the complexity of the model without losing crucial information about the data. Also, choosing a feature selection for all CAN IDs can lead to loss of important information since some features that are important for one CAN ID would be zero-constant for other CAN IDs. An application of our proposed method is intended for use in IDSs that employ a separate model for each CAN ID.

Table 4: Important Features selected by each Feature Selection Method

Feature Selection Method	**Selected Features**				
	ID: 2	ID: 160	ID: 399	ID: 1201	ID: 1442
Univariate Selection	D[3],D[4],D[2]	D[3],D[7], D[1], D[4], D[0], D[2]	D[0], D[3],D[7]	D[3], D[4], D[5], D[6], D[2]	D[2], D[3], D[7]
Feature Importance	D[7],D[3], D[6]	D[1], D[3], D[4], D[6], D[0], D[7]	D[0], D[3],D[2],	D[3], D[4],D[7], D[5], D[2]	D[3], D[7], D[4]
Information Gain	D[3],D[0],D[1]	D[3], D[1], D[0], D[7], D[4], D[2]	D[0], D[1],D[2],	D[3], D[7], D[0], D[1],D[4]	D[3], D[7], D[0]
Pearson correlation	D[1], D[2],[3]	D[2], D[3], D[4], D[5], D[6], D[7]	D[2],D[3],D[4]	D[1], D[2], D[4], D[5], D[6]	D[2],D[3],D[4]
RFE	D[2],D[6],D[7]	D[1], D[3], D[4], D[5], D[6], D[7]	D[0], D[4],D[7]	D[3], D[4], D[5], D[6],D[7]	D[3], D[6], D[7]
chi-squared	D[2],D[3],D[4]	D[0], D[1], D[2], D[3], D[4], D[7]	D[0], D[3], D[7]	D[2], D[3], D[4], D[5],D[6]	D[2], D[3], D[7]
Our approach	D[5],D[6],D[7]	D[0], D[1], D[2], D[3], D[5], D[6]	D[0],D[1],D[5]	D[0], D[1], D[2], D[3], D[7]	D[0],D[2],D[3]

5 Conclusion

In conclusion, our paper introduces an innovative feature selection approach for CAN bus data without losing important data or causing overfitting. This method effectively safeguards against adversaries that exploit unselected features for unknown attacks and bypass IDS. Additionally, our work contributes to a deeper understanding of CAN bus data, facilitating better solutions for enhanced vehicle security. Through a comparative analysis, our approach, unlike others, successfully identifies and removes irrelevant constant zero features, leading to a significant reduction in trained parameters for machine and deep learning models. A limitation is observed, as some CAN IDs lack zero-constant features and may not benefit from the proposed approach. Nevertheless, our method shows promising results by decreasing computational resource requirements and bolstering resilience against potential malicious traffic injection.

Our future work involves deploying the approach in dynamic real-world environments to enhance its robustness and applicability. Furthermore, we plan to assess how our feature selection affects algorithm computation time, investigating whether the computational cost outweighs the benefits and offering practical insights.

References

1. Chandrashekar, G., Sahin, F.: A survey on feature selection methods. Computers & Electrical Engineering **40**(1), 16–28 (2014)
2. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. Journal of machine learning research **3**(Mar), 1157–1182 (2003)
3. Hoang, T.N., Kim, D.: Detecting in-vehicle intrusion via semi-supervised learning-based convolutional adversarial autoencoders. Vehicular Communications **38**, 100520 (2022)
4. Kang, H., Kwak, B.I., Lee, Y.H., Lee, H., Lee, H., Kim, H.K.: Car hacking and defense competition on in-vehicle network. In: Workshop on Automotive and Autonomous Vehicle Security (AutoSec). vol. 2021, p. 25 (2021)
5. Lee, H., Jeong, S.H., Kim, H.K.: Otids: A novel intrusion detection system for in-vehicle network by using remote frame. In: 2017 15th Annual Conference on Privacy, Security and Trust (PST). pp. 57–5709. IEEE (2017)
6. Mwangi, B., Tian, T.S., Soares, J.C.: A review of feature reduction techniques in neuroimaging. Neuroinformatics **12**, 229–244 (2014)
7. Paul, A., Islam, M.R.: An artificial neural network based anomaly detection method in can bus messages in vehicles. In: 2021 International Conference on Automation, Control and Mechatronics for Industry 4.0 (ACMI). pp. 1–5. IEEE (2021)
8. Rajapaksha, S., Kalutarage, H., Al-Kadri, M.O., Petrovski, A., Madzudzo, G., Cheah, M.: Ai-based intrusion detection systems for in-vehicle networks: A survey. ACM Computing Surveys **55**(11), 1–40 (2023)
9. Seo, E., Song, H.M., Kim, H.K.: Gids: Gan based intrusion detection system for in-vehicle network. In: 2018 16th Annual Conference on Privacy, Security and Trust (PST). pp. 1–6. IEEE (2018)
10. Young, C., Zambreno, J., Olufowobi, H., Bloom, G.: Survey of automotive controller area network intrusion detection systems. IEEE Design & Test **36**(6), 48–55 (2019)
11. Zhang, F., Chan, P.P., Biggio, B., Yeung, D.S., Roli, F.: Adversarial feature selection against evasion attacks. IEEE transactions on cybernetics **46**(3), 766–777 (2015)