

An Association Rules-Based Approach for Anomaly Detection on CAN-bus

Gianni D'Angelo, Massimo Ficco, and Antonio Robustelli(✉)

Department of Computer Science, University of Salerno, Salerno, Italy
{giadangelo,mficco,arobustelli}@unisa.it

Abstract. With the rapid development of the Internet of Things (IoT) and the Internet of Vehicles (IoV) technologies, smart vehicles have replaced conventional ones by providing more advanced driving-related features. IoV systems typically consist of Intra-Vehicle Networks (IVNs) in which many Electronic Control units (ECUs) directly and indirectly communicate among them through the Controller Area Network (CAN) bus. However, the growth of such vehicles has also increased the number of network and physical attacks focused on exploiting security weaknesses affecting the CAN protocol. Such problems can also endanger the life of the driver and passengers of the vehicle, as well as that of pedestrians. Therefore, to face this security issue, we propose a novel anomaly detector capable of considering the vehicle-related state over time. To accomplish this, we combine different most famous algorithms to consider all possible relationships between CAN messages and arrange them as corresponding associative rules. The presented approach, also compared with the state-of-the-art solutions, can effectively detect different kinds of attacks (DoS, Fuzzy, GEAR and RPM) by only considering CAN messages collected during attack-free operating scenarios.

Keywords: Anomaly detection · CAN-Bus · Association rules · Internet of Vehicles (IoV) · Machine Learning (ML)

1 Introduction

With the rapid development of the Internet of Things (IoT) and the Internet of Vehicles (IoV) technologies, the vehicle industry has evolved drastically over the last decades resulting in extensive automation and replacement of conventional cars. Typically, IoV systems consist of a mesh of sensors and computational units controlled by thousands of Electronic Control Units (ECUs) [1,18]. These ECUs are specifically designed for the optimal management of a wide array of driving-related features ranging from engine control to Anti-lock Braking (ABS) and Advanced Driver-Assistance Systems (ADAS), respectively.

ECUs are distributed all around the vehicle and communicate with each other via Intra-Vehicle Networks (IVNs) using the Controller Area Network (CAN) protocol, which is the de-facto standard for the IoV industry. Indeed, CAN offers

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2023
O. Gervasi et al. (Eds.): ICCSA 2023 Workshops, LNCS 14105, pp. 174–190, 2023.
https://doi.org/10.1007/978-3-031-37108-0_12

many advantages, such as cost-effective wiring, immunity to electrical interference, self-diagnosing, and error correction [2]. However, despite these functional benefits, the rising vehicle communications render the CAN protocol vulnerable to several hostile activities. For instance, airbags and ABS-related attacks may affect the reputation of manufacturers with substantial financial implications (e.g. the vehicles' recalls) [6], while tampering with ECUs (e.g., the odometers of used cars) is another well-known example that may lead to dire consequences for consumers and manufacturers. In addition, IoV-related intrusions may also concern some privacy issues [3,7]. Indeed, since the CAN protocol was designed to operate within a closed network environment, it is affected by the following two weaknesses: the lack of device authentication (that allows impersonation and masquerading attacks) and the usage of unencrypted traffic (that does not guarantee the confidentiality constraint).

Therefore, several Deep Learning (DL) and Machine Learning (ML)-based approaches have been proposed to counteract such hostile activities that, by exploiting the CAN protocol-related weaknesses, can endanger the life of many people, such as drivers, passengers, and pedestrians, respectively. However, most of the supervised DL-based solutions cannot consider new kinds of attacks [11,25], while those semi-supervised not accurately spot some of the well-known attacks (e.g. they present a considerable number of False Positives or False Negatives) [15,28]. On the other hand, ML-based ones do not analyze correlations among multiple CAN messages [7,26,27]. Consequently, they do not consider a complex and real-world operating scenario, namely the vehicle-related state over time. Instead, they perform only a simple message-by-message detection activity.

For this reason, we propose a novel anomaly detector capable of considering the vehicle's state over time. We accomplish this by using a semi-supervised approach based on several famous algorithms. More precisely, we cluster CAN messages into multi-dimensional regions. Then, we match the derived areas in order to consider all possible relationships between different CAN messages. Hence, by employing a sliding temporal window mechanism, we represent the vehicle's state by arranging such correlations as corresponding associative rules. Finally, we demonstrate the effectiveness of our approach, also compared to some notable ML-based solutions, in detecting several kinds of attacks, namely: DoS, Fuzzy, Gear, and RPM, respectively.

Hence, the main contributions of this paper can be summarized as follows:

1. A novel anomaly detection approach is proposed in order to consider only licit CAN messages and represent the vehicle state over time.
2. The effectiveness of the presented detector is evaluated in detecting several attacks and then compared with some notable ML-based solutions.

The remainder of the paper is organized as follows. Section 2 will present the related works on CAN-bus. Section 3 will report the preliminary concepts related to the CAN protocol and the employed algorithms. Section 4 will describe the proposed approach. Finally, Sect. 5 will report the experimental results, while Sect. 6 will show the conclusions and future work.

2 Related Works

The recent growth and popularity of intelligent vehicles have led to an increase in the development of many DL-based models as effective solutions for IoV intrusion detection and security enhancement [9,20,22]. In this section, we report works related to the CAN bus system that, due to its well-known security weaknesses [8,12,17], can be exploited to accomplish several malicious activities and engage the life of drivers, passengers, and pedestrians.

Hossain et al. [11] proposed a Long Short Term Memory (LSTM)-based model to detect and mitigate the CAN bus network attacks. They accomplished this by injecting different attacks (DoS, Fuzzy, and Spoofing) in a dataset related to a real-world vehicle. Then, they used the derived dataset to train and test the proposed model following a Supervised approach. The obtained results, also compared with those derived using other notable datasets, have always shown an average accuracy of 99%. Similarly, Xiao et al. [25] presented SIMATT-SECCU, which leverages an enhanced RNN cell to perform a time-saving anomaly detection. More precisely, they validated their model by comparing the related results and required time effort with several DL-based models (e.g. RNN, GRU, GIDS, and LSTM).

Also, LSTM layers have been employed by Zhu et al. [28] by proposing a Semi-supervised approach. To accomplish this, they considered the literal binary CAN messages instead of revealing the semantics of CAN messages. Then, they employed two LSTM-based models (multi-dimension and multi-task) trained on only attack-free CAN messages collected in various driving conditions, such as parking, acceleration, deceleration, and steering. The achieved results have proven the effectiveness of the proposed models by showing an average accuracy between 96% and 98%. Instead, Lin et al. [15] introduced an anomaly detection framework that leverages Autoencoder and the Ecogeografic-based optimization algorithm (EBO) to learn sequential patterns existing in the CAN traffic. More precisely, they employed a detection model, trained only on attack-free CAN messages, to identify messages related to three famous attacks (DoS, Fuzzy, and Impersonation). The experimental results indicated that the proposed detector outperformed the most notable Machine Learning (ML)-based models by achieving excellent detection rates.

However, most of the supervised DL-based solutions cannot consider new kinds of attacks, while those semi-supervised achieved good detection results but are affected by a considerable quantity of False Positives (FPs) and False Negatives (FNs).

On the other hand, many ML-based solutions have been proposed to improve the detection rates by following a semi-supervised or unsupervised approach [14,19,23]. For the sake of clarity, we report only the contributions that, for their similarity, can help understand the proposed solution.

D'Angelo et al. [7] provided a new data-driven anomaly detection system based on two algorithms. More precisely, they employed the first algorithm (Cluster-based Learning Algorithm - CLA) to learn the behavior of licit messages passing on the CAN bus, while they used the second one (Data-driven Anomaly Detection Algorithm - DADA) to perform a real-time classification. The experimental results, obtained using data related to the CAN bus, have shown the effectiveness of the proposed detector by also outperforming some famous state-of-the-art approaches.

Instead, Yang et al. [27] presented a novel IDS framework to detect various IoV-related attacks. This framework, named Leader Class and Confidence Decision Ensemble (LCCDE), leverages three advanced ML-based algorithms (XGBoost, LightGBM, and CatBoost) to predict the class related to a specific attack. The experimental results, carried out through two public IoV datasets, highlighted an average accuracy of 99% and a low number of FPs and FNs. Similarly, they proposed a multi-tiered hybrid IDS that employs both a signature-based IDS and an anomaly-based IDS to detect known and unknown attacks [26]. Experimental results have proven that the proposed system can detect various known attacks with an average accuracy of 99%, while it achieved good F-Score values for zero-day attack detection.

However, contrary to our approach, these solutions do not analyze licit correlations among multiple CAN bus messages. Consequently, they cannot perform any anomaly detection activity by considering the vehicle-related state over time.

3 Background

In this section, we provide an overview of the preliminary concepts necessary to understand and appreciate the workflow employed by the proposed approach, namely the CAN protocol and some most famous algorithms, respectively.

3.1 CAN Protocol

The Controller Area Network (CAN) is a message-based protocol designed to enable numerous electric components, e.g., microcontrollers, ECUs, sensors, devices, and actuators, to communicate with each other through a single/dual-wire bus (the CAN bus) [17]. It is one of the most famous standards in automotive and industrial applications due to its low cost and flexible design. Indeed, it is mainly employed to provide many advanced driving-related functions, such as auto start/stop, parking assistance, automatic lane detection, and collision avoidance [12]. In detail, CAN is a time-synchronized broadcast protocol, which means all nodes attached to the CAN bus can receive all packets transmitted [7,14]. For this reason, the sent packets do not have any information about the sender and receiver. The CAN frame can be of 4 types: data frame, error frame, overload frame, and remote frame, respectively.

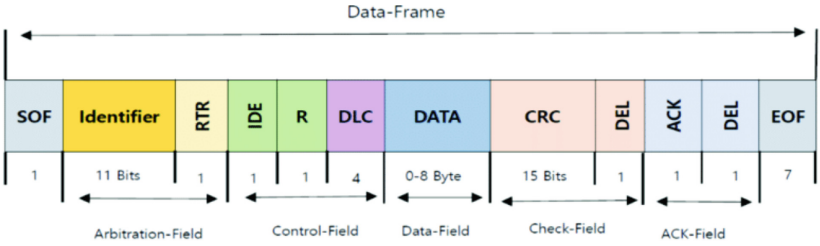

Fig. 1. Structure of the CAN data frame [12].

As shown in Fig. 1, the data frame is composed of seven fields: 1) a Start-of-Frame (SoF) field that denotes the start of a frame; 2) an arbitration field, which contains the identifier of the message and service bits; 3) a control field, which includes the length of the data included in the message and two reserved bits; 4) a data field that contains the transferred data; 5) a CRC field used for error checking; 6) an ACK field employed to acknowledge the receipt of the message; and 7) End-of-Frame (EoF) field that indicates the end of the frame [14].

3.2 K-Nearest Neighbor Algorithm

The K-Nearest Neighbors (KNN) is a supervised learning algorithm used to face both regression and classification tasks. KNN tries to predict the correct class for the test data by calculating the distance between the test data and all the training points. To accomplish this, KNN selects the K number of points closest to the test data. More precisely, the KNN algorithm calculates the probability of the test data belonging to the classes composed of K training data and selects the category holding the highest probability score [10]. Figure 2 shows an iteration of the KNN algorithm using K=5 nearest data points, while its steps can be summarized as follows:

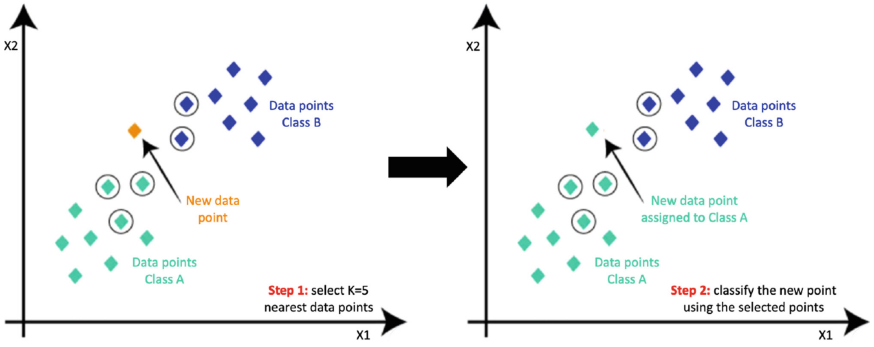

Fig. 2. An iteration of the KNN algorithm with K=5 nearest data points.

1. Select the number K of the neighbors to consider;
2. Derive the distance between each training data point and the new data point.
3. Consider the K nearest neighbors and count the number of the data points in each class.
4. Assign the new data point to that class for which the number of neighbors is maximum.

The success of the KNN algorithm strongly depends on the employed distance and the value K. Typically, the three most famous used distances are Euclidean, Manhattan, and Hamming. Regarding the value of K, it can be chosen randomly or selecting the value that, for instance, gives the minimum error rate [5].

In this study, we employ the KNN algorithm to match the different multi-dimensional regions. Such operation is necessary in order to consider only the licit sequences of CAN messages.

3.3 Apriori Algorithm

The Apriori algorithm, proposed by Agrawal and Srikant in 1994, is one of the most famous and straightforward approaches employed for association rule mining [24]. Apriori is designed to operate on a dataset containing transactions (itemsets) and provide a set of frequent itemsets representing the extracted association rules. To accomplish this, Apriori uses an iterative bottom-up approach and starts by considering every single item in the considered dataset. Then, it generates the itemsets candidates by joining the itemsets generated in the previous step. Finally, the list of candidates is pruned by removing any infrequent itemsets. Apriori repeats this process until no new itemsets can be derived from the previous ones.

The success of the Apriori algorithm strongly depends on the pruning step, in which any itemset that not satisfies a given threshold is pruned. For instance, the support and confidence values represent the most famous threshold values used by the Apriori algorithm. More precisely, given an itemset X, the support is defined as the number of transactions containing X divided by the total number of transactions T.

$$Supp(X) = \frac{count(X)}{T} \tag{1}$$

Instead, given two itemsets X and Y, the confidence is derived as the number of transactions containing X and Y divided by the number of transactions containing X.

$$Conf(X \rightarrow Y) = \frac{count(X\&Y)}{count(X)} \tag{2}$$

Figure 3 shows an execution of the Apriori algorithm using a support value greater than 1.

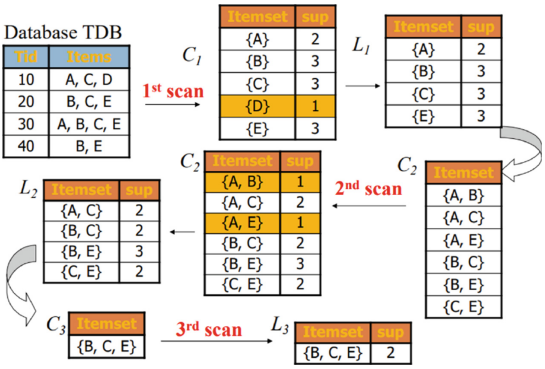

Fig. 3. An execution of the Apriori algorithm [4].

In this study, we employ the Apriori algorithm to mine the association rules as corresponding frequent itemsets. Therefore, we extract only the CAN message sequences useful to represent the vehicle-related state over time.

3.4 Cluster-based Learning Algorithm

The Cluster-based Learning Algorithm (CLA) [7] is an unsupervised approach employed to find a set of behavioral signatures about the CAN bus usage in attack-free scenarios. To this end, CLA follows a multi-step procedure in which the different CAN messages are placed within different subsets and then clustered using the K-Means algorithm. In this way, CLA leads to the definition of multiple centroids (also referred to as gravitational attractors) that individuate, in turn, an admissibility zone. More precisely, such regions represent the CAN commands related to a specific ID within an 8-dimensional space.

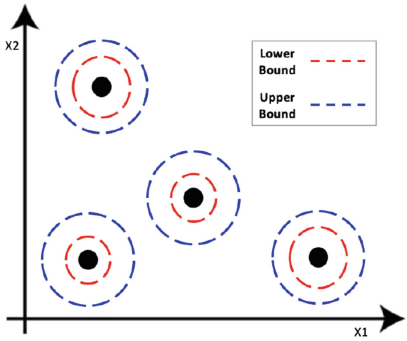

Fig. 4. 2D representation of some centroids and their bounds.

In addition, for each defined cluster, CLA estimates an eligibility area characterized by a lower and an upper bound. Such bounds correspond to the minimum and maximum distance among the points of the considered cluster and its centroid, respectively. Therefore, given an arbitrary CAN ID, this step is fundamental to include the related CAN messages mistakenly excluded by the K-Means algorithm. In a bi-dimensional space, as shown in Fig. 4, this can be represented by two circular crowns around each centroid.

4 The Proposed Approach

This section describes the proposed approach to detect anomalies on CAN-bus by considering the vehicle's state over time. More precisely, since modern cars consist of hundreds of ECUs and sensors, we can define the state as relationships between messages sent in an arbitrary time interval. The following assumption provides the fundaments to perform anomaly detection activities aware of the cooperation between electronic and mechanical components, whose proper functioning is essential to preserve the vehicle's integrity, manufacturers' reputation, and also the people's life [6]. To accomplish this, unlike the most famous ML-based solutions [7,26,27], the proposed approach does not perform a classic message-by-message detection but, instead, follows three main steps, namely: finding the multi-dimensional regions related to each CAN ID, clustering the discovered regions to consider all possible relationships between distinct CAN IDs and extracting the more representative relationships as corresponding association rules, respectively.

Algorithm 1. Training phase

Require: D (attack-free dataset); K (number of clusters related to CLA); N (number of clusters related to KNN); w (size of temporal window); Δ (distance between two temporal windows); *supp* (value of support).

Ensure: I (frequent itemsets containing the selected association rules); Σ_c (clustered multi-dimensional regions)

1: $\Sigma \leftarrow \text{CLA}(D,K)$
2: $\Sigma_c \leftarrow \text{KNN}(\Sigma,N)$
3: $t \leftarrow 0$
4: $I \leftarrow \emptyset$
5: **while** $t + w \leq |D|$ **do**
6: $\quad mess \leftarrow D[t; t + w]$
7: $\quad rules \leftarrow \text{extractRules}(mess,\Sigma_c)$
8: $\quad I \leftarrow I \cup \text{Apriori}(rules,supp)$
9: $\quad t \leftarrow t + \Delta$
10: **end while**
11: **return** $\{I, \Sigma_c\}$

As shown in Alg. 1, the training phase begins by considering only CAN messages related to an attack-free scenario. Therefore, given the dataset D,

the algorithm finds the multi-dimensional regions by running the Cluster-based Learning Algorithm (CLA) with K clusters. this step provides as output Σ, namely a Map associating each CAN ID to the corresponding region. Notice that, as discussed in Sect. 3.4, the CLA algorithm defines each centroid by providing the related 8-dimensional vector of coordinates and bounds, respectively.

Next, the algorithm clusters the derived regions Σ by running the K-Nearest Neighbor (KNN) with N clusters. This step provides Σ_c, the dataset containing the multi-dimensional coordinates associated with the derived regions. Notice that the KNN algorithm is essential in order to consider all possible relationships between distinct CAN IDs. Indeed, the clustered regions can also be seen as sets related to highly correlated messages sent in the same time interval.

After that, the algorithm runs Apriori and provides I, the dataset of the most frequent itemsets containing the selected associative rules. The algorithm accomplishes this by considering the attack-free messages in D through a sliding temporal window of size w and offset Δ. This step provides the messages ($mess$) representing the vehicle's state in the time interval $[t; t+w]$. Then, the algorithm employs such messages to extract the associative rules ($rules$) from the clustered regions Σ_c. Finally, the algorithm updates the set of frequent itemsets I by performing Apriori with a specific threshold value ($supp$).

Algorithm 2. Testing phase

Require: D_{bus} (messages sent on CAN-bus); I (training frequent itemsets); Σ_c (clustered multi-dimensional regions); w (size of temporal window); Δ (distance between two temporal windows); U (list of licit IDs); $supp$ (value of support).
Ensure: R (detection results dataset).

$t \leftarrow 0$
$I_{test} \leftarrow \emptyset$
$R \leftarrow \emptyset$
while $t + w \leq |D_{bus}|$ **do**
  $mess \leftarrow D_{bus}[t; t + w]$
  $\forall \{ID \in mess \mid ID \notin U\} : R \leftarrow R \cup \{(0, 0, 1)\}$
  if checkDistribution($mess$,U) **then**
    $R \leftarrow R \cup \{(0, 0, 1)\}$
  end if
  $rules \leftarrow$ extractRules($mess$,Σ_c)
  $I_{test} \leftarrow$ Apriori($rules$,$supp$)
  $numRules \leftarrow$ rulesMatcher(I_{test},I)
  if $numRules < |I_{test}|$ **then**
    $R \leftarrow R \cup \{(|I_{test}|, numRules, 1)\}$
  else
    $R \leftarrow R \cup \{(|I_{test}|, numRules, 0)\}$
  end if
  $t \leftarrow t + \Delta$
end while
return R

Subsequently, as shown in Alg. 2, the testing phase begins by considering the messages sent in real-time on the CAN-bus (i.e., when the vehicle is in working). Therefore, given the dataset D_{bus}, the algorithm samples the messages in $[t; t+w]$ using a sliding temporal window of size w and offset Δ.

After that, in order to improve the detection performances, the algorithm considers the following two attack scenarios: 1) For each message, it verifies if the related ID is contained within the list U of known IDs, and 2) it verifies if the messages' distribution (i.e., their frequency) significantly changes respect to a known ID. More precisely, as done by the most famous ML-based detectors [7,26,27], the first control (line 6) assumes that an unknown ID is always related to an attack. Instead, the second one (lines 7–9) verifies if several malicious messages are associated with a known ID. Notice that, in both cases, the algorithm updates the detection results R by adding the tuple $(0, 0, 1)$. In detail, the first two values respectively refer to the number of extracted and checked rules, while the third value reports the presence (1) or absence (0) of an attack.

Consequently, only once such tests are passed, the algorithm employs the selected messages ($mess$) to extract the associative rules ($rules$) from the clustered regions Σ_c. Then, the algorithm generates the frequent itemsets I_{test} by performing Apriori with a specific threshold value ($supp$).

Finally, the algorithm checks if the extracted rules (I_{test}) are also present within the training itemsets I. More precisely, the algorithm counts the number of rules ($numRules$) within the training itemsets I. Therefore, if $numRules$ is less than the number of extracted rules $|Itest|$, the algorithm updates R by reporting an attack $(|Itest|, numRules, 1)$.

In addition, for the sake of clarity, the training and testing phases are high-level summarized and reported in Fig. 5, respectively.

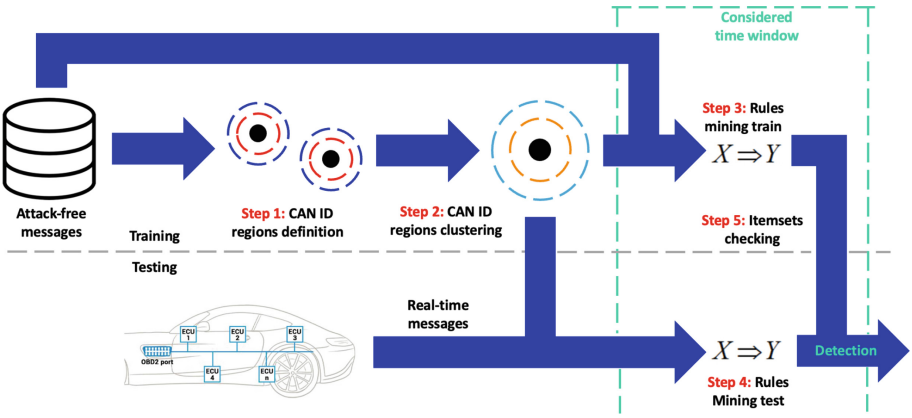

Fig. 5. The high-level workflow.

5 Experimental Results

The goal of the presented experiments is devoted to demonstrating the capabilities of our approach in detecting well-known attacks, namely: DoS, Fuzzy, GEAR and RPM. Also, a comparison with the most famous state-of-the-art solutions is provided in order to remark on the effectiveness of the proposed detector.

5.1 Dataset And Experimental Setting

The experiments have been carried out using the Car-Hacking Dataset provided by the Hacking and Countermeasure Research Lab (HCRL)[1] [20]. More precisely, it consists of five .csv files, that is: 1) attack-free; 2) DoS attack; 3) fuzzy attack; 4) spoofing of the drive GEAR; and 5) spoofing of the RPM gauge, respectively. Each file summarizes around 30–40 minutes of CAN traffic observed from real-world vehicles through the OBD-II port, while the injected messages are performed for 3–5 s for a total of about 300 intrusions. Also, for a DoS attack, the most dominant message (0×000) is injected every 0.3 ms, while random CAN IDs and data values are injected every 0.5 ms in a Fuzzy attack. Finally, for a Spoofing attack (RPM and GEAR), the messages with CAN ID associated with RPM/GEAR are injected every 1 ms. Table 1 provides an overview of the employed dataset.

Table 1. Overview on the involved dataset.

Dataset	Normal mess	Attack mess	Total
GIDS (normal)	988.872	–	988.872
DoS Attack	3.078.250	587.521	3.665.771
Fuzzy Attack	3.347.013	491.847	3.838.860
GEAR Attack	3.845.890	597.252	4.443.142
RPM Attack	3.966.805	654.897	4.612.702
Total	15.226.830	2.331.517	17.558.347

As mentioned earlier, our method makes use of a semi-supervised approach. Accordingly, following the presented Alg. 1, we used only normal messages for training and all traffic (licit and illicit) for testing.

Accordingly, we have run the CLA algorithm with $K = 300$ to find the multi-dimensional regions associated with attack-free CAN messages. Then, we clustered the obtained 8-dimensional regions (centroids), and their bounds, in order to consider all possible relationships between different CAN IDs. To accomplish this, we used the KNN algorithm with $N = 50$. As a result of these steps, the GIDS dataset has been mapped in a new corresponding dataset containing more than 5000 centroids, 1.8 million rows, and 250 columns, respectively.

[1] Dataset available at https://ocslab.hksecurity.net/Datasets/car-hacking-dataset.

Next, we employed a sliding temporal window containing 900 CAN messages and offset = 450 (i.e., the values respectively of w and Δ). We chose 900 because it represents the number of messages sent on the CAN bus in 0.5 s. However, due to the huge quantity of data, we applied an additional quantization step on centroids and their bounds. More precisely, we accomplished this using the K-Means algorithm [13,16]. Hence, we mapped about 5000 centroids into 150 corresponding clusters. Then, we quantized the lower and upper bounds by uniquely assigning them into 100 intervals. Consequently, we reduced the number of columns to 75, in which the first 25 columns represent centroids associated only with some specific IDs, while the remaining 50 represent the corresponding bounds.

Subsequently, we used the quantized dataset to extract the most frequent itemsets. To accomplish this, we have run the Apriori algorithm with minimum support = 0.9. Finally, we employed the K-Fold Cross Validation algorithm, with K = 6, to validate the effectiveness of the mined itemsets and perform a fine-tuning of the considered parameters [21]. More precisely, we have split the training set into six disjointed subsets. Then, the first five partitions have been employed, in turn, as a learning dataset and the remaining one as a testing dataset. The obtained results, shown in Table 2, have proven the effectiveness of such itemsets by achieving all the maximum ratings for each considered metric.

Table 2. Achieved results related to K-Fold cross validation (K=6).

	Acc.	Pre.	Rec.	F1-Mea.
Fold1	1.0000	1.0000	1.0000	1.0000
Fold2	1.0000	1.0000	1.0000	1.0000
Fold3	1.0000	1.0000	1.0000	1.0000
Fold4	1.0000	1.0000	1.0000	1.0000
Fold5	1.0000	1.0000	1.0000	1.0000
Fold6	1.0000	1.0000	1.0000	1.0000

5.2 Evaluation Metrics

To appreciate the effectiveness of the proposed detector, we employed the following evaluation metrics derived from the confusion matrix: Accuracy (Acc.), Precision (Pre.), Recall (Rec.), and F1-Score (F1-Mea.).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

$$F1 - Score = \frac{2 * Sens * Prec}{Sens + Prec} \tag{6}$$

For each considered category (normal or attack), TPs (True Positives) refer to the CAN messages correctly classified, while TNs (True Negatives) refer to the CAN messages correctly identified in another category. Conversely, FPs (False Positives) represent the CAN messages mistakenly identified as the considered category, while FNs (False Negatives) represent the CAN messages mistakenly identified in another category.

5.3 Achieved Results And Comparison

Next, for the testing, we considered all licit and illicit CAN messages according to the datasets reported in Table 1 (Dos, Fuzzy, Gear, and RPM). More precisely, by following the same procedure described in Alg. 2), we mined the most frequent itemsets. Also, to improve the detection capabilities of our approach, we considered the following two attack scenarios: 1) Verify the presence of an unknown CAN ID in the testing messages, and 2) Verify if the distribution of the messages related to a licit CAN ID significantly changes. Tables 3 and 4 report the achieved results related to normal and attack messages, respectively.

Table 3. Achieved results related to normal CAN messages.

Dataset	Acc.	Pre.	Rec.	F1-Mea.
DoS	1.0000	1.0000	1.0000	1.0000
Fuzzy	0.9971	0.9951	1.0000	0.9975
GEAR	0.9905	1.0000	0.9818	0.9908
RPM	0.9911	1.0000	0.9822	0.9910
Avg	0.9947	0.9988	0.9910	0.9948

Table 4. Achieved results related to attack CAN messages.

Dataset	Acc.	Pre.	Rec.	F1-Mea.
DoS	1.0000	1.0000	1.0000	1.0000
Fuzzy	0.9971	1.0000	0.9930	0.9965
GEAR	0.9905	0.9805	1.0000	0.9902
RPM	0.9911	0.9823	1.0000	0.9911
Avg	0.9947	0.9907	0.9983	0.9945

Finally, we compared our approach with some notable ML-based solutions, namely: the CLA and DADA algorithms proposed by D'Angelo et al. (DA-ALG) and the detectors (LCCDE and MTH-IDS) employed by Yang et al. Table 5 summarizes the comparison between our approach and the considered solutions.

Table 5. Comparison with the most notable ML-based approaches.

Approach	Acc.	Pre.	Rec.	F1-Mea.
Proposed	**0.9947**	**0.9907**	**0.9983**	**0.9945**
DA-ALG	0.9998	0.9986	0.9999	–
LCCDE	0.9999	0.9999	0.9999	0.9999
MTH-IDS	0.9999	0.9999	0.9999	0.9999

As shown in Table 5, our approach has achieved excellent ratings for each derived metric. More precisely, they are very close to those derived by the considered ML-based solutions. Therefore, this might highlight that such detectors are better than the proposed ones because they can better reduce the number of FPs and FNs, respectively. However, contrary to our approach, these solutions do not analyze licit correlations among multiple CAN bus messages. Consequently, they cannot perform any anomaly detection activity by considering the vehicle-related state over time.

6 Conclusions And Future Work

In this paper, we proposed a new anomaly detector to counteract the security issues related to the rapid growth of smart vehicles in the last decades. More precisely, we considered the attacks related to the Controller Area Network (CAN) bus that represents the core communication mechanism used by Electronic Control Units (ECUs) in modern Internet of Vehicle (IoV) systems. However, as is well known, the CAN protocol lacks any built-in security mechanisms that, if exploited for malicious purposes, can endanger the life of many people, such as drivers, passengers, and pedestrians. For this reason, we proposed a novel anomaly detector capable of considering the vehicle's state over time. We accomplished this by using a semi-supervised approach based on several famous algorithms. More precisely, we clustered CAN messages into multi-dimensional regions. Then, we matched the derived areas in order to consider all possible relationships between different CAN messages. Finally, by employing a sliding temporal window mechanism, we represented the vehicle's state by arranging such correlations as corresponding associative rules.

The experimental results, obtained using a famous dataset related to a real vehicle, have shown the capabilities of our approach in detecting different kinds of attacks, namely: DoS, Fuzzy, GEAR, and RPM, respectively. Indeed, the derived performance metrics (Accuracy, Precision, Recall, and F1-Score) have achieved excellent ratings for any considered attack. Finally, we also compared the obtained results with those derived from the state-of-the-art solutions by highlighting, once again, the effectiveness of the proposed approach. Therefore, on the basis of the achieved outcomes, we would like to employ our detector to spot the presence of new anomalies and attacks. More precisely, we intend to use our approach in many other application domains where the correlation

among multiple information plays a fundamental role in representing the state of complex systems over time.

Acknowledgments. This work was partially supported by project SERICS (PE00000014) under the NRRP MUR program funded by the EU - NGEU.

CRediT Authorship Contribution Statement
Gianni D'Angelo. Supervision, Methodology, and Reviewing.
Massimo Ficco. Supervision, Methodology, and Reviewing.
Antonio Robustelli. Investigation, Conceptualization, and Writing.

Conflict of interest. The authors declare that they have no conflict of interest.

Availability of Data and Material. The used data is available on https://ocslab.hksecurity.net/Datasets/car-hacking-dataset.

Informed Consent. Informed consent was obtained from all individual participants included in the study.

Ethical Approval. This article does not contain any studies with human participants or animals performed by any of the authors.

References

1. Al-Jarrah, O.Y., Maple, C., Dianati, M., Oxtoby, D., Mouzakitis, A.: Intrusion detection systems for intra-vehicle networks: a review. IEEE Access **7**, 21266–21289 (2019). https://doi.org/10.1109/ACCESS.2019.2894183
2. Bozdal, M., Samie, M., Aslam, S., Jennions, I.: Evaluation of can bus security challenges. Sensors **20**(8) (2020). https://doi.org/10.3390/s20082364, https://www.mdpi.com/1424-8220/20/8/2364
3. Campanile, L., Iacono, M., Levis, A.H., Marulli, F., Mastroianni, M.: Privacy regulations, smart roads, blockchain, and liability insurance: putting technologies to work. IEEE Secur. Priv. **19**(1), 34–43 (2021). https://doi.org/10.1109/MSEC.2020.3012059
4. Chonny: Apriori—association rule mining in-depth explanation and python implementation. https://urlis.net/f2yeu5zt. Accessed 02 Mar 2023
5. Christopher, A.: K-nearest neighbor. https://medium.com/swlh/k-nearest-neighbor-ca2593d7a3c4. Accessed 02 Mar 2023
6. Dürrwang, J., Braun, J., Rumez, M., Kriesten, R.: Security evaluation of an airbag-ECU by reusing threat modeling artefacts. In: 2017 International Conference on Computational Science and Computational Intelligence (CSCI), pp. 37–43 (2017). https://doi.org/10.1109/CSCI.2017.7
7. D'Angelo, G., Castiglione, A., Palmieri, F.: A cluster-based multidimensional approach for detecting attacks on connected vehicles. IEEE Internet of Things J. **8**(16), 12518–12527 (2021). https://doi.org/10.1109/JIOT.2020.3032935
8. EL Madani, S., Motahhir, S., EL Ghzizal, A.: Internet of vehicles: concept, process, security aspects and solutions. Multimedia Tools Appl. **81**(12), 16563–16587 (2022). https://doi.org/10.1007/s11042-022-12386-1

9. Elmasry, W., Akbulut, A., Zaim, A.H.: Evolving deep learning architectures for network intrusion detection using a double PSO metaheuristic. Comput. Netw. **168**, 107042 (2020). https://doi.org/10.1016/j.comnet.2019.107042, https://www.sciencedirect.com/science/article/pii/S138912861930800X
10. Fauzi, M.A., Hanuranto, A.T., Setianingsih, C.: Intrusion detection system using genetic algorithm and k-nn algorithm on dos attack. In: 2020 2nd International Conference on Cybernetics and Intelligent System (ICORIS), pp. 1–6 (2020). https://doi.org/10.1109/ICORIS50180.2020.9320822
11. Hossain, M.D., Inoue, H., Ochiai, H., Fall, D., Kadobayashi, Y.: LSTM-based intrusion detection system for in-vehicle can bus communications. IEEE Access **8**, 185489–185502 (2020). https://doi.org/10.1109/ACCESS.2020.3029307
12. Khatri, N., Shrestha, R., Nam, S.Y.: Security issues with in-vehicle networks, and enhanced countermeasures based on blockchain. Electronics **10**, 893 (2021)
13. Kherbache, M., Espes, D., Amroun, K.: An enhanced approach of the k-means clustering for anomaly-based intrusion detection systems. In: 2021 International Conference on Computing, Computational Modelling and Applications (ICCMA), pp. 78–83 (2021). https://doi.org/10.1109/ICCMA53594.2021.00021
14. Li, X., et al.: Can bus messages abnormal detection using improved SVDD in internet of vehicles. IEEE Internet of Things J. **9**(5), 3359–3371 (2022). https://doi.org/10.1109/JIOT.2021.3098221
15. Lin, Y., Chen, C., Xiao, F., Avatefipour, O., Alsubhi, K., Yunianta, A.: An evolutionary deep learning anomaly detection framework for in-vehicle networks - can bus. IEEE Trans. Ind. Appl. p. 1 (2020). https://doi.org/10.1109/TIA.2020.3009906
16. Liu, J., Liang, B., Ji, W.: An anomaly detection approach based on hybrid differential evolution and k-means clustering in crowd intelligence. Int. J. Crowd Sci. **5**(2), 129–142 (2021). https://doi.org/10.1108/IJCS-07-2020-0013
17. Lokman, S.-F., Othman, A.T., Abu-Bakar, M.-H.: Intrusion detection system for automotive controller area network (CAN) bus system: a review. EURASIP J. Wirel. Commun. Netw. **2019**(1), 1–17 (2019). https://doi.org/10.1186/s13638-019-1484-3
18. Markovitz, M., Wool, A.: Field classification, modeling and anomaly detection in unknown can bus networks. Veh. Commun. **9**, 43–52 (2017). https://doi.org/10.1016/j.vehcom.2017.02.005, https://www.sciencedirect.com/science/article/pii/S2214209616300869
19. Moore, M.R., Bridges, R.A., Combs, F.L., Starr, M.S., Prowell, S.J.: Modeling inter-signal arrival times for accurate detection of can bus signal injection attacks: a data-driven approach to in-vehicle intrusion detection. In: Proceedings of the 12th Annual Conference on Cyber and Information Security Research. CISRC 2017, Association for Computing Machinery, New York, NY, USA, pp. 1–4 (2017). https://doi.org/10.1145/3064814.3064816
20. 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 (2018). https://doi.org/10.1109/PST.2018.8514157
21. Shakya, S., Sigdel, S.: An approach to develop a hybrid algorithm based on support vector machine and Naive Bayes for anomaly detection. In: 2017 International Conference on Computing, Communication and Automation (ICCCA), pp. 323–327 (2017). https://doi.org/10.1109/CCAA.2017.8229836
22. Song, H.M., Woo, J., Kim, H.K.: In-vehicle network intrusion detection using deep convolutional neural network. Veh. Commun. **21**, 100198 (2020). https://

doi.org/10.1016/j.vehcom.2019.100198, https://www.sciencedirect.com/science/article/pii/S2214209619302451
23. Taylor, A., Japkowicz, N., Leblanc, S.: Frequency-based anomaly detection for the automotive can bus. In: 2015 World Congress on Industrial Control Systems Security (WCICSS), pp. 45–49 (2015). https://doi.org/10.1109/WCICSS.2015.7420322
24. Utami, M.P., Nurhayati, O.D., Warsito, B.: Hoax information detection system using Apriori algorithm and random forest algorithm in twitter. In: 2020 6th International Conference on Interactive Digital Media (ICIDM), pp. 1–5 (2020). https://doi.org/10.1109/ICIDM51048.2020.9339648
25. Xiao, J., Wu, H., Li, X.: Internet of Things meets vehicles: Sheltering in-vehicle network through lightweight machine learning. Symmetry **11**(11) (2019). https://doi.org/10.3390/sym11111388, https://www.mdpi.com/2073-8994/11/11/1388
26. Yang, L., Moubayed, A., Shami, A.: MTH-IDS: a multitiered hybrid intrusion detection system for internet of vehicles. IEEE Internet of Things J. **9**(1), 616–632 (2022). https://doi.org/10.1109/JIOT.2021.3084796
27. Yang, L., Shami, A., Stevens, G., de Rusett, S.: LCCDE: a decision-based ensemble framework for intrusion detection in the internet of vehicles. In: GLOBECOM 2022–2022 IEEE Global Communications Conference, pp. 3545–3550 (2022). https://doi.org/10.1109/GLOBECOM48099.2022.10001280
28. Zhu, K., Chen, Z., Peng, Y., Zhang, L.: Mobile edge assisted literal multi-dimensional anomaly detection of in-vehicle network using LSTM. IEEE Trans. Veh. Technol. **68**(5), 4275–4284 (2019). https://doi.org/10.1109/TVT.2019.2907269