CISTER
Research Centre in
Real-Time & Embedded
Computing Systems

Conference Paper

ZeroCAN: Anomaly-Based Zero-Day Attack Detection in Vehicular CAN Bus Networks

Jonathan Rendel
William Balte
Harrison Kurunathan*
Hazem Ali
Alexandre Roque
Wagner Morais
Mahdi Fazeli

*CISTER Research Centre

CISTER-TR-250102

2025

ZeroCAN: Anomaly-Based Zero-Day Attack Detection in Vehicular CAN Bus Networks

Jonathan Rendel, William Balte, Harrison Kurunathan*, Hazem Ali, Alexandre Roque, Wagner Morais, Mahdi Fazeli

*CISTER Research Centre

Polytechnic Institute of Porto (ISEP P.Porto)

Rua Dr. António Bernardino de Almeida, 431

4200-072 Porto

Portugal

Tel.: +351.22.8340509, Fax: +351.22.8321159

E-mail: jhk@isep.ipp.pt, haali@isep.ipp.pt

https://www.cister-labs.pt

Abstract

Zero-day attacks present a significant security threatto vehicular networks, exploiting vulnerabilities at both softwareand hardware levels within such systems that remain undis-covered. Mitigating these threats is essential to ensuring thesafety and security of vehicular systems. Support Vector Machine(SVM) is a good candidate for anomaly detection of zero-dayattacks within vehicular networks because it can handle high-dimensional data and effectively distinguish between normaland abnormal patterns in complex and dynamic environments.A trained SVM on the normal operation data of in-vehicularnetwork can identify flag deviations, thus making it effective inthe detection of any previously unknown attack patterns, whichis a common behaviour of zero-day attacks. In this paper, weintroduce an anomaly detection method called 1dZeroCAN 1d whichmodels the behaviour of every single electronic control unit on thenetwork with a separate SVM and a set of high-level features thatcapture the timing and data payload aspects of CANbus traffic.This approach achieves an anomaly detection rate of over 99%and a false positive rate below 0.01% during normal operationin most cases

2025 33rd Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP) | 979-8-3315-2493-7/25/$31.00 ©2025 IEEE | DOI: 10.1109/PDP66500.2025.00025

ZeroCAN: Anomaly-Based Zero-Day Attack Detection in Vehicular CAN Bus Networks

Jonathan Rendel[1], William Balte[1], Harrison Kurunathan[2], Hazem Ismail Ali[1], Alexandre dos Santos Roque[3], Wagner Ourique de Morais[1], Mahdi Fazeli[1]
[1] Department for Computing and Electronics for Real-time and Embedded Systems (CERES), School of Information Technology (ITE), Halmstad University, Halmstad, Sweden.
[2] CISTER/ISEP, Polytechnic Institute of Porto, Portugal.
[3] Graduate Program of Electrical Engineering, Federal University of Rio Grande do Sul - UFRGS, Porto Alegre, Brazil.
[1] {jonren19, willni19}@student.hh.se, {hazem.ali, wagner.demorais and mahdi.fazeli}@hh.se, **[2]** jhk@isep.ipp.pt, **[3]** as.roque@ufrgs.br

***Abstract*—Zero-day attacks present a significant security threat to vehicular networks, exploiting vulnerabilities at both software and hardware levels within such systems that remain undiscovered. Mitigating these threats is essential to ensuring the safety and security of vehicular systems. Support Vector Machine (SVM) is a good candidate for anomaly detection of zero-day attacks within vehicular networks because it can handle high-dimensional data and effectively distinguish between normal and abnormal patterns in complex and dynamic environments. A trained SVM on the normal operation data of in-vehicular network can identify flag deviations, thus making it effective in the detection of any previously unknown attack patterns, which is a common behaviour of zero-day attacks. In this paper, we introduce an anomaly detection method called "ZeroCAN" which models the behaviour of every single electronic control unit on the network with a separate SVM and a set of high-level features that capture the timing and data payload aspects of CANbus traffic. This approach achieves an anomaly detection rate of over 99% and a false positive rate below 0.01% during normal operation in most cases.**

***Index Terms*—Anomaly detection, CAN bus IDS, Feature engineering, Zero-day**

I. Introduction

Modern vehicles contain an increasingly large number of Electronic Control Units (ECU) that control and monitor almost every aspect of the vehicle [1]. These ECUs are interconnected using Controller Area Networks (CAN bus), which play a pivotal role in vehicle functionality. Each component sends and receives messages across the CAN bus, which operates on a broadcast principle, meaning that all ECUs on the network receive all messages. However, only the intended recipient acts on each message. As vehicles become more connected, they become vulnerable to cyberattacks that can exploit vulnerabilities in CAN bus or ECUs to gain control over critical vehicle functions, such as braking or steering, posing serious safety risks [2].

Though the CAN bus is widely used in automotive, it lacks robust security features, making it vulnerable to zero-day attacks. These attacks exploit previously unknown vulnerabilities, providing little to no time for retaliation against these attacks. CAN bus lacks any authentication or encryption techniques, thus making it susceptible to attacks if an adversary gains access to it. Some exploits in the CAN bus can be carried out through the implementation of the protocol, the ECU firmware, or the network architecture. As the CAN bus in vehicle operation acts as a centralized hub for communication and operation, a successful zero-day attack could disrupt critical functions, resulting in a compromise in vehicle control and safety systems and even catastrophic loss of human life. Exploiting the zero-day vulnerabilities, an attacker can inject malicious messages onto the CAN bus, resulting in unauthorized control of the vehicle. Other attacks, like spoofing, can mislead other ECUs to perform unintended actions. Flooding of messages can also result in Denial of Service (DoS), disrupting in-vehicle communication systems.

Data in vehicular networks include data and signals that are dynamic and non-linear. The change in data is very much dependent on factors such as environmental conditions [3], other networks and vehicles in the vicinity and even attacks from malicious entities. This dynamic nature of the signal makes it challenging to detect anomalies using conventional methods. SVM creates a robust hyperplane to capture high-level features. This defines a clear margin between classes of normal and any abnormalities in the signals. The definition of this margin enables identifying outliers and unexpected patterns, which are characteristic of zero-day attacks. The ability of SVM to accurately handle non-linear and high-dimensional data enables identifying unforeseen threats.

The contributions in this work are as follows: ***a)*** in-depth comparison of some of the state-of-the-art approaches to CAN bus Intrusion Detection Systems (IDS), and examining their adaptability to various vehicular environments. ***b)*** Introduction of "ZeroCAN", an SVM-based IDS that focuses on capturing complex patterns and behaviours within the vehicular network. The proposed method improves the detection of zero-day attacks and improves the reliability of the vehicular network. ***c)*** Comprehensive evaluation of "ZeroCAN" performance on embedded hardware that includes individual ECU modeling using data sets from four different car models to validate the efficiency of our proposed method.

The rest of the paper is structured as follows; In Section II, we present the state-of-the-art research work in the field of conventional IDS and ML-based IDS methods. Followed by Section III, where we discuss preliminary information essential for understanding the proposed work. Then, Section IV details the main contribution. Finally, Sections V and VI show the experimental setup and evaluation results.

II. Related work

This section presents the state-of-the-art solutions for IDS, which covers anomaly detection in CAN bus and general anomaly detection in time series data from both the conventional and Machine Learning (ML) perspectives. Table I presents some of the key aspects of both these methods in line with accuracy, adaptability, complexity and maintenance.

979-8-3315-2493-7/25/$31.00 ©2025 IEEE
DOI 10.1109/PDP66500.2025.00025

TABLE I: Aspects of Conventional and ML-based IDS

Aspect	Conventional IDS	ML- based IDS
Detection Method	Signature-based, Rule based	SVM, LSTM
Accuracy	High for known Low for unknown threats	High for both known and unknown
Adaptability	Low, relies on predefined rules/signatures	High, adapts to new patterns
Complexity	Lower, simpler rules and signatures	Higher, requires model training
False Positives	High due to rigid rules	Can be reduced with training
Scalability	Limited, need updates	High, can handle large datasets
Maintenance	Regular updates needed for signatures/rules	Less frequent, needs periodic training

A. Conventional IDS

IDS has been done through a Graph-based Intrusion Detection system for CAN [4]. This method analyses a window of messages to create a directed graph. The IDS uses features extracted from the graph to classify message windows as containing an anomaly or not. This is done by utilizing chi-squared distribution to compare the distributions of different features.

Features derived from the Page Rank algorithm and Gaussian naive Bayes have also been used for IDS [5]. The work shows the relative importance of the features used, for the different types of attacks separately. The accuracy of their model for different window sizes was analysed in this work. However, in this work, they do not consider the data content of the CAN packets but only the order in which the different ECUs send their packages. This opens the vulnerability of an unknown zero-day attack without altering the transmission sequence of the ECUs.

Researchers in [6] propose a system that measures the timing of different CAN packages and can detect anomalies in this timing with a One-Class Support Vector Machine. Inserting new messages that alter the timing in the network is a common zero-day attack. In this work, time-dependent features such as interpacket timing are utilized to detect malicious inserted messages. Connected vehicles incur cost problems and lack reliable communication when additional modules are added or when cloud computing is utilized. A signature-based lightweight intrusion detection system is proposed in [7] that can be applied directly to the vehicle's ECUs. In this work, the authors detect the CAN-based anomalies caused by real-world scenarios.

A rule-based approach is also an efficient approach for IDS. However, it has some limitations in the detection accuracy. On the other hand, machine learning-based detection provides a higher detection accuracy but lacks in the area of effective computation costs. Researchers in [8] propose a hybrid IDS that combines the benefits of both rule-based and machine learning-based approaches. In this approach, low computational requirements are maintained by offsetting the detection with a rule-based component.

B. ML-based IDS

Ergen et al. [9] propose a model that combines a Long Short-Term Memory (LSTM) network with an SVM for anomaly detection in time-dependent data of variable length. This approach avoids the need for a probabilistic error prediction model, as required in conventional LSTM methods. While the model was tested on various time series datasets, it was not applied to CAN-bus or network traffic, which is the focus of this paper. However, the solution could potentially be relevant to CAN-bus traffic analysis. Hossain et al. [10] introduced the first effective LSTM-based model for detecting well-known network attacks in CAN-bus communication, classifying traffic as normal or one of three predefined attacks. However, the system cannot detect unknown zero-day attacks, and the authors did not explore creating a system for detecting any anomaly, which is a key difference from this paper's goal.

Taylor et al. [11] developed a CAN IDS using multiple LSTM networks to model the behaviour of individual ECUs, an approach not seen in previous research. They achieved over 90% true positive rate and eliminated false alarms, with near-perfect detection allowing a minimal false alarm rate (0.01%). The system uses semi-supervised learning and can detect unknown attacks, but the authors did not address real-time requirements, model complexity, or runtime. A potential drawback is the independent modelling of ECUs, which overlooks dependencies between them.

Kang et al. [12] proposed a Deep Neural Network model for detecting intrusions in a CAN-bus network using supervised learning with a labelled dataset, limiting its ability to detect zero-day attacks. Despite this, the model achieved a true positive rate of 98.39% and a false positive rate of 2.79%. Although the authors evaluated the model's real-time performance, they did not test it on actual ECU hardware.

Aziz et al. [13] utilized Explainable Neural Networks to detect known attacks with a high accuracy of 99.7%. However, the system was not tested on zero-day attacks, which limits its relevance to this thesis. While the authors mentioned real-time performance, they did not explain how it should be evaluated, and it was not included in the results.

Khan et al. [14] introduced DivaCAN, a machine learning-based IDS that uses an ensemble of classifiers including LGBM, ET, RF, Bagging, kNN, DNN, and MLP—to detect intrusions with high accuracy. The system achieved a precision of 94.93%, recall of 94.98%, and an F1 score of 94.97% on a real-world dataset, outperforming other classifiers. However, DivaCAN had the highest training time and the second-highest testing time due to its complex ensemble approach. The authors did not compare its execution time to real-time requirements.

Purohit et al. [15] proposed a hybrid rule-based and machine learning anomaly detection model that shows promising results with high accuracy and low computational cost. However, the system focuses on known security threats and relies on supervised learning with labelled datasets, making it unsuitable for detecting zero-day attacks. Although the model's execution time was evaluated, it was not clearly compared to real-time requirements.

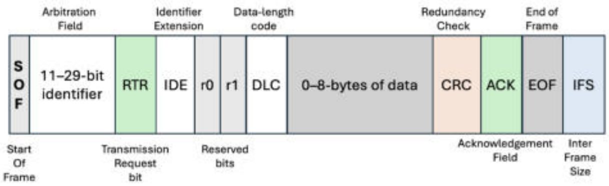

Fig. 1: Standard CAN frame format.

III. Preliminaries

This section provides the necessary context for understanding the research problem, its significance, and the work presented in this paper. This includes contextualising the field of study and clarifying relevant concepts and terms.

A. *Controller Area Network (CAN bus)*

CAN bus is a message protocol commonly used in the vehicle industry designed to enable communication between various ECUs within modern cars [16], due to its ability to handle the complex and real-time communication needs of automotive systems. Unlike traditional point-to-point wiring systems, CAN bus allows multiple ECUs to communicate over a single network, significantly reducing wiring complexity and improving reliability. This network architecture is essential for managing the increasingly sophisticated systems in vehicles, ranging from engine control and braking systems to infotainment and safety features. In a CAN bus network, multiple nodes can initiate communication and broadcast it to every node on the network. Every connected node in the bus receives the message. However, each node decides whether to process the message individually. The messaging in CAN bus is facilitated through a rigid structure called the standard CAN frame (Figure 1). This process is initiated based on the message's identifier and allows functions such as message prioritisation and filtering. The rigid structure of the CAN frame comprises the following components, RTR (Remote Transmission Request) is a single bit that differentiates data frames (RTR=0) and remote frames (RTR=1) requesting data from another node. The IDE (Identifier Extension): is also a single bit indicates whether the frame uses a standard 11-bit identifier (IDE=0) or an extended 29-bit identifier (IDE=1). The DLC (Data Length Code) is a 4-bit field that specifies the number of data bytes in the data field, ranging from 0 to 8 bytes. The CRC (Cyclic Redundancy Check) is a 15-bit field that is used for error detection and ensures the integrity of the transmitted data. The Acknowledgement (ACK) field comprises of 2-bits and acknowledges the successful receipt of a message. End of Frame (EOF) is a 7-bit field that signifies the end of the CAN frame. Finally, Inter Frame Space (IFS) consists of 3 bits and provides a separation between consecutive frames, allowing the bus to be idle. Though the CAN bus is robust for communication in vehicular networks, it was designed with no security measures against malicious intrusions. This protocol lacks encryption mechanisms, thus the broadcasted data is vulnerable to interception and potential exposure. This raises a serious concern as there is no inherent safety measure against attacks like injection of unauthorised messages, spoofing and unauthorised access. Also, a compromised high-priority node can monopolise the entire bus by continuously transmitting data resulting in denial of service for other nodes in the network.

B. *Common types of attacks in CAN bus*

Due to the lack of built-in security features such as encryption, authentication, and access control, the CAN bus is vulnerable to adversarial attacks (Figure 2) the exploit its inherent limitations as follows:

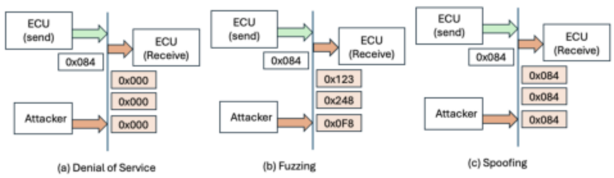

Fig. 2: Common types of attacks on CAN bus

1) Denial of Service (DoS): CAN bus operates on a priority-based arbitration system. Hence, when a malicious node transmits high-priority messages (ID: 0x000) at a rapid rate. This action overwhelms the bus and prevents the legitimate nodes from communicating with each other, as they think the bus is reserved for high priority messages [17].

2) Fuzzy: Fuzzing attack represents the action of sending random data to the CAN bus to trigger unintended behaviours in the system. By sending unexpected messages, an attacker can disrupt the system and induce crashes, thus leading to potential exploitation. The use-case stated in [17] presents a situation where critical actions such as the steering wheel, turn signal lamps, instrument, and even the gear shift were compromised.

3) Spoofing: Spoofing represents the injection of forged messages into the CAN bus that masquerades as messages from a legitimate node. Thus gaining unauthorized access to critical devices of the ECU. The study in [18] shows the disruption of speed and the gear change through spoofing.

4) Data Alteration: When an attacker intercepts and modifies the data during the transit on the CAN bus. Therefore, the legitimate nodes receive malicious information resulting in erroneous system failures. As there are authentication mechanisms to prevent such middle-man attacks, the CAN messages are very vulnerable to this type of attack.

C. *Anomaly detection using ML techniques*

Anomaly detection using ML is done through the identification of patterns in data that deviate from expected behaviour [19]. The data itself can be classified into normal and abnormal, which indicates any malicious activity or even system faults. Unlike conventional systems discussed in the related work, ML techniques have the capability to adapt to evolving threats and unseen anomalies like zero-day attacks.

Anomaly detection through ML can generally be divided into the following categories:

Algorithm 1 ZeroCAN-SVM-based IDS

Input: Packet stream $P = \{p_1, p_2, \ldots, p_n\}$
Output: Feature set F, TPR, FPR, RT
Initialize $F \leftarrow \emptyset$, $TP \leftarrow 0$, $FP \leftarrow 0$, $TN \leftarrow 0$, $FN \leftarrow 0$
$start\_time \leftarrow$ current time
for $i \leftarrow 2$ to n **do**
  $p_{current} \leftarrow p_i$, $p_{previous} \leftarrow p_{i-1}$
  Extract ID, $time_{current}$, $time_{previous}$, and $data_{current}$
  $dT \leftarrow time_{current} - time_{previous}$
  if $ID = p_{previous}.ID$ **then**
    $ID\_dT \leftarrow dT$, $dData \leftarrow data_{current} - p_{previous}.data$
    if $i > 2$ **then**
      $ID\_ddData \leftarrow dData - F[i-2].ID\_dData$
    end if
  end if
  $F[i] \leftarrow \{dT, ID\_dT, data_{current}, dData, ID\_ddData\}$
  if is_attack($p_{current}$) **then**
    Update TP or FN based on prediction
  else
    Update FP or TN based on prediction
  end if
end for
$RT \leftarrow$ current time $-start\_time$
$TPR \leftarrow \frac{TP}{TP+FN}$, $FPR \leftarrow \frac{FP}{FP+TN}$
return F, TPR, FPR, RT

1) Supervised anomaly detection: Supervised anomaly detection requires a labelled dataset with both normal and abnormal data. This technique has several limitations. It can be difficult to obtain this dataset mainly because the system is only optimised to detect anomalies present in the dataset.

SVM based systems can be used to create a one-class classifier [20]. The SVM learns what is normal data by defining a boundary. If data lies outside of this boundary, it is considered to be an anomaly. This technique is effective in some applications, but it is less suitable for anomaly detection in time series data since it is unable to capture time dependencies on its own [9].

An LSTM network is a type of neural network that uses historical information from its own output as input, it is appropriate for detecting anomalies in time series data [21]. The architecture is based on RNN, but unlike conventional RNN:s, there is no need to define the time window length to be classified. Instead, LSTM can be used to learn long-term patterns in time series data. The model does not require training on data of known anomalies. Only data known to be normal is needed for the training. The anomaly detection works by training the model to predict what should happen in the next time frame based on what happened before. An anomaly has been detected if what actually happens deviates from the prediction more than expected. Zheng et al. [22] describe feature engineering. An ML model takes features as input. These features are a representation of the data. Feature engineering is the process of creating useful features from the original data to increase the performance of the ML model. The introduction of manually created high-level features can reduce the required complexity of the model since it no longer has to learn these features.

2) Unsupervised anomaly detection: Unsupervised anomaly detection does not require a labelled dataset at all. Instead, the technique assumes that most of the instances in the test data are considered normal.

A standard k-Nearest-Neighbour classifier is used for supervised learning multi-class applications [23]. In its simplest form, the classification works by selecting the majority class label among the k-number of nearest neighbours from a new observation. A distance metric is used to determine the nearest neighbours [23]. A nearest neighbour-based method can also be used for semi-supervised anomaly detection [19]. This approach works under the assumption that normal data will be grouped close to each other. The distance to the nearest normal neighbour is then compared to a threshold to determine if a new observation is normal or an anomaly [19]. One major drawback of Nearest Neighbour-based anomaly detection is that the need for distance computations for every new classification can be computationally expensive and the method might therefore not be suitable for real-time applications [19].

3) semi-supervised anomaly detection: Semi-supervised anomaly detection only requires a dataset containing normal behaviour. This data is used to design a model that can be used to identify behaviour that deviates from normal behaviour.

D. Feature Engineering

Proper feature engineering can allow the use of a simple model like SVM instead of more complex models like a neural network. This is a very relevant possibility to investigate since a less complex model will make it easier to fulfil the requirement of real-time operation on low-powered hardware. The use of feature engineering can bypass the limitations of traditional SVM, such as the lack of time-dependency modelling. The time dependencies of the CAN bus network traffic can be partially represented with high-level features. A high-level feature is derived from the raw data and contains more meaningful information than the raw CAN bus data on its own. This information could reduce the required model complexity since the model does not have to extract this information on its own. Examples of possible high-level CAN bus features that could be used are the time difference between the current and previous packet, the amount of time that has passed between two packet transmissions from the same ECU, or how much the packet data differs from the last transmission, just to name a few. It is also important to consider how to convert the CAN frame data field to features. The data is 8 bytes, and it is possible to create many different configurations of features. For example, the entire 64-bit number could be considered as one feature or every individual byte could become a feature.

IV. Methodology

ZeroCAN is an intrusion detection algorithm that leverages SVM to identify anomalies and potential security threats, particularly focusing on detecting zero-day attacks that conventional methods miss. It involves capturing high-level features from CAN data, that include time-based metrics, packet patterns, and data integrity checks. Sections IV-A and IV-B present the ZeroCAN model and the high-level features used by it in detail, respectively.

A. ZeroCAN Algorithm

A standard CAN frame supports limited unique identifiers; therefore, it is intuitive to use separate models for learning the behaviour of the individual ECUs instead of having a single model learn the behaviour of the entire network. Similar approaches using LSTM have been proposed in the literature [11] and showed promise that these methods outperform when compared to the singular model approach.

In ZeroCAN approach, the training is done by splitting the whole dataset into subsets based on ECU ID. Then a separate SVM is trained for every data subset. The modeling of individual ECUs also allows for customized features and manually selected conditions. For example, an ECU sending sensor values within a known range could be checked with a manually selected condition in addition to the SVM. It is also possible to model each ECU separately, which can eventually allow us to have separate models for different driving modes such as driving, parking, etc.

In Algorithm 1, we present ZeroCAN approach to detect potential attacks in vehicular networks using SVM. As the first step, the algorithm identifies features such as time differences and data differences between consecutive packets. Following this, the characteristics, such as packet ID and data fields, of previous packets calculating first-order and second-order changes in time and data are compared. The extracted features are then stored in a feature set that is used to classify whether a packet represents normal or malicious activity. We evaluate the performance of ZeroCAN using performance metrics such as True Positive Rate (TPR), False Positive Rate (FPR), and Runtime (RT). For every packet, ZeroCAN verifies whether the packet was predicted correctly as an attack or non-attack by SVM and updates the TP, FP, TN, and FN values. Finally, the TPR and FPR are computed to assess the accuracy of the intrusion detection system and return these metrics along with the feature set and the runtime of the analysis. Thus, anomalies in vehicular communication can be identified while the IDS's efficiency is evaluated in real-time.

B. High-level features

High-level features represent several aspects of the vehicular data. These features can encapsulate multiple aspects of data transmission, such as packet timing, data integrity, and frequency of communication between entities. These usually represent the temporal relationships between the data. Sudden changes in the time intervals between packets and unexpected variations in the data payload can identify attacks like spoofing, data injection, or signal tampering. Increasing the number of features improves the chance of successful zero-day attack detection. The high-level time-dependent features that are evaluated are: ***a)*** **dT** is the time difference between the current and previous packets. This feature captures the average packet-sending frequency on the network. ***b)*** **ID_dT** is the time difference between the current packet and the previous packet sent from the same ECU. This feature captures the average packet-sending frequency of the individual ECUs. ***c)*** **Data** is the contents of the data field in the current packet. This feature allows the model to learn the normal range of packets that every ECU transmits and to detect anomalies outside of this range. ***d)*** **ID_dData** is the difference between the current data field and the data field from the most recent packet sent by the same ECU. Even if the data payload itself is within range, its rate of change might not be. ***e)*** **ID_ddData** is the change in ID_dData from the most recent packet sent by the same ECU. In the same way, as the data could be within the normal range while its rate of change is not, it could also be possible for the rate of change to be normal while the second-order rate of change is not.

Dynamic environments like vehicular networks include constantly evolving communication where the attack vectors are complex and unpredictable. In this line, there is a need for increasing the features to enhance ZeroCAN's ability to identify deviations from normal behaviour, thus detecting zero-day attacks. Our experiments focus on evaluating variant sets of feature sets while ensuring consistency by keeping all other model parameters fixed, allowing for a clear comparison between feature sets that contribute to the overall detection performance. Once an optimal feature set is defined, the model will be optimized for the final evaluation.

V. Experimental Evaluation

This section discusses the experimental setup, methods, and techniques used to evaluate the proposed methodology (ZeroCAN). The performance evaluation is an important test since it determines if a method meets the real-time requirements necessary for zero-day attack detection. First, we introduce the datasets used in our experiments in Section V-A. Second, we explain our technique in performance evaluation of anomaly detection methods in Section V-B.

A. Datasets

The datasets used for training and evaluation of the models are provided by CAN-train-and-test (CT&T) [18]. The dataset was created "[...] in order to address the limitations of existing open-access datasets." [18]. Each message in the dataset contains the timestamp, ECU ID, data payload of the message, and a label. Normal data traffic is included from four different car models: a) *2011 Chevrolet Impala*, b) *2011 Chevrolet Traverse*, c) *2016 Chevrolet Silverado*, and d) *2017 Subaru Forester*. Each car has a separate dataset that allows for training and evaluation with four different datasets. This work introduces traditional types of attacks, such as DoS, Fuzzy, and various Spoofing attacks. Also introduces an additional attack type referred to as a data-altering attack. This attack dataset is created by modifying the attack-free validation data from each car model. The attack does not insert any new packets into the network traffic. Instead, it takes control over one ECU and alters its data payload whenever it sends a packet. The fact that no extra packets are inserted and the timing remains unaltered means that this attack could require different information from the packets compared to the attacks already included in the dataset. The purpose of introducing this attack is to increase the diversity of attacks that are used to validate model performance.

In addition to the data-altering attack, we explore the size of variations in the payload that the model can detect. Different values of multipliers are tested on an ECU that sends different data payloads in every packet, and an ECU that always sends a payload of zeros has its data altered in small increments.

TABLE II: Feature comparison for ZeroCAN.

		Fuzzing	DoS	Spoof	Data altering	Normal
ID-dT	TPR	99.31%	99.99%	100.00%	0.02%	0.00%
	FPR	4.83%	5.62%	4.36%	0.04%	0.04%
	RT %	25.63%	30.14%	25.55%	25.49%	25.46%
ID-dT, Data	TPR	99.83%	100.00%	100.00%	98.93%	0.00%
	FPR	9.04%	9.05%	9.30%	4.73%	4.47%
	RT %	25.66%	29.47%	25.6%	25.34%	25.62%
ID-dT, Data, ID-dData	TPR	99.87%	100.00%	100.00%	99.997%	0.00%
	FPR	8.75%	8.78%	8.95%	4.68%	4.42%
	RT %	25.80%	29.21%	25.93%	25.65%	25.24%
ID-dT, Data, ID-dData, dT	TPR	99.90%	100.00%	100.00%	99.996%	0.00%
	FPR	37.86%	37.22%	39.24%	4.60%	4.34%
	RT %	25.65%	29.76%	25.91%	25.50%	26.28%
ID-dT, Data, ID-dData, ID-ddData	TPR	99.87%	100.00%	100.00%	100.00%	0.00%
	FPR	8.50%	8.53%	8.64%	4.62%	4.37%
	RT %	25.50%	29.32%	25.91%	25.67%	25.37%

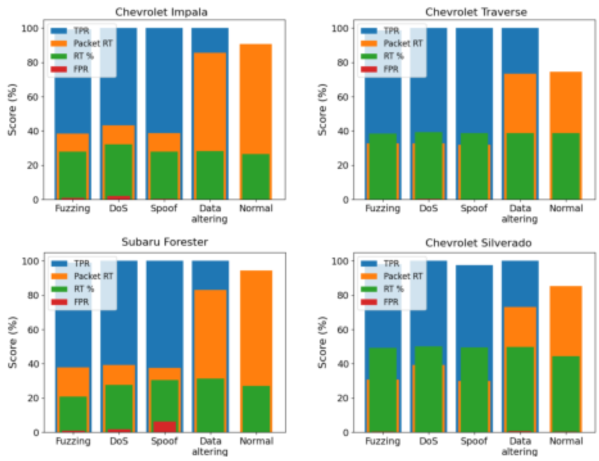

Fig. 3: ZeroCAN achieves a true positive rate over 98% for all attacks

The result of this experiment is not considered in relation to the requirements since there are no previous results for comparison; instead, they are simply included as a benchmark for future research.

B. Anomaly detection performance evaluation

The performance of the system is evaluated by exposing it to several known attacks not seen during training: DoS, Fuzzing, Spoofing and Data altering. These types of attacks are considered zero-day since they have not been used to train the model. It is important to realise that it is not possible to verify the model for all possible attacks, but the chance of successful detection is increased by testing a wide range of different attacks targeting different aspects of the CAN bus data. These commonly known attacks have been used to evaluate similar systems [10]. The datasets described in Section V-A contain data recorded from these attacks.

Four base metrics are considered while performing the tests. These are True Positive (TP), False Positive (FP), True Negative (TN) and False Negative (FN). Positive means that a packet is anomalous, and negative means that it is normal. It is important to keep the false positive rate as low as possible while testing the model on normal traffic. When testing with anomalous data, it is important that the false negative rate is kept low. The anomalous data could affect the characteristics of the normal packets, for example, delaying a packet with lower priority as described in Section III-A. This means that normal traffic being sent during an attack is likely to be misclassified as anomalous, hence increasing the false positive rate. The goal is to identify anomalies accurately while not raising any unnecessary alarms during normal operations. This means that normal packets falsely classified as anomalous during an attack are not a big issue as long as the anomalous packets are classified correctly.

In addition to the TPR and FPR used to evaluate the requirements, we provide the unprocessed confusion matrices and ROC curves for the final model evaluation to be used as a benchmark for future research. The ROC curve measures the model performance at several different decision thresholds where the TPR is plotted against the FPR [24]. A straight diagonal line would suggest that the model is giving random predictions, and a line that follows the upper left corner suggests that it is a perfect model. The Area Under the Curve (AUC) is also included as a performance metric for future research. An AUC of 1.0 is the best possible result.

VI. Results

This section presents the results in line with ZeroCAN, which is the individual ECU modelling with the SVM approach. In this regard, we present the overall performance under several attack patterns. Furthermore, we evaluate ZeroCAN through real-time datasets from four car models by comparing the TPR, FPR, RT and packet RT.

ZeroCAN is an individual ECU modelling with an SVM approach, which enables identifying and extracting features that are relevant to the underlying IDS system. It transforms raw time-series data and logs into a suitable format for SVM input. The goal of ZeroCAN is to reduce FPRs, thus avoiding unnecessary alerts.

A. Feature set evaluation

Table II shows the performance comparison of the features **dT**, **ID-dT**, **Data**, **ID-dData** and **ID-ddData** for different attacks, i.e. fuzzing, DoS, Spoofing and data altering. The results show that the TPR is consistent at 99-100%. Only the **dT** feature did not improve the TPR and instead made the FPR higher during the attack. The reason is that some normal driving scenarios, e.g. engaging the brakes, may increase the normal packet traffic rate on the CAN bus, which means variations in **dT** that may lead to a FP.

We evaluate our feature set with four different car models to evaluate the decision threshold, as shown in Figure 3. The requirement of a false positive rate under 0.01% during attack-free data is satisfied for all car models except the Chevrolet Silverado. The model achieves a true positive rate over 99% for all attacks except Fuzzing against Chevrolet Traverse (98.64%) and Fuzzing and Spoof against Chevrolet Silverado (98.09% and 97.54% respectively). Different car models have unique ECUs resulting in variance in communication patterns, thus affecting how certain attack types impact data communication. This shows the need for fine-tuning car models to enhance detection and maintain a low false positive rate.

TABLE III: ECU data payload altering in Chevrolet Impala.

Data altering	TPR	Increment	TPR
50%	100.00%	3	100.00%
40%	99.92%	2	100.00%
30%	86.54%	2	66.49%

B. Condition-based evaluation

The Chevrolet Silverado is tested with a model where some conditions are added. This is because it is the car model with the worst overall performance and therefore it could be possible to see an improvement. Some ECUs on the network always send the exact same data payload and therefore the data of these ECUs can be checked with a fixed condition instead of relying on the SVM. The result of this experiment is presented in Figure 4 and shows that the true positive rates are increased slightly by this approach, but some of them are still below the goals.

Table III shows the results from an experiment on data payload alteration in a Chevrolet Impala's ECU. The left section shows the results when the payload of an ECU that sends varying data in each packet is altered by a specific percentage. The right section displays results when the payload of an ECU that consistently sends zeros is altered incrementally, with 1 as the smallest possible change. The result shows that the model is able to reliably detect a change of 40% in the data of an ECU that sends different values every packet and an increment of 2 for an ECU that always sends zeros.

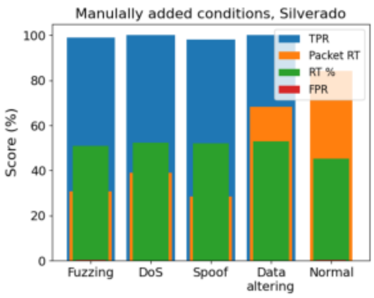

Fig. 4: Manually added conditions with train-test combinations

Figure 5 shows the ROC curves for the different car models during the final model evaluation together with the Areas Under the Curves (AUCs) shown in Table IV. The table reports the AUC scores for each ROC, rounded to 5 decimals, for each car and type of attack. In each plot in Figure 5, all types of attacks have a separate curve. In the left columns are the regular plots, and in the right column are the same ROCs, but zoomed in on the upper left corner. For all cars and attacks, the ROCs show excellent performance in the model. The curve's proximity to the top-left corner indicates high True Positive Rates (TPR) with low False Positive Rates (FPR), demonstrating effective attack detection with minimal false alarms. This balance confirms ZeroCAN's capability for accurate, real-time intrusion detection in automotive environments, where limiting false positives is essential.

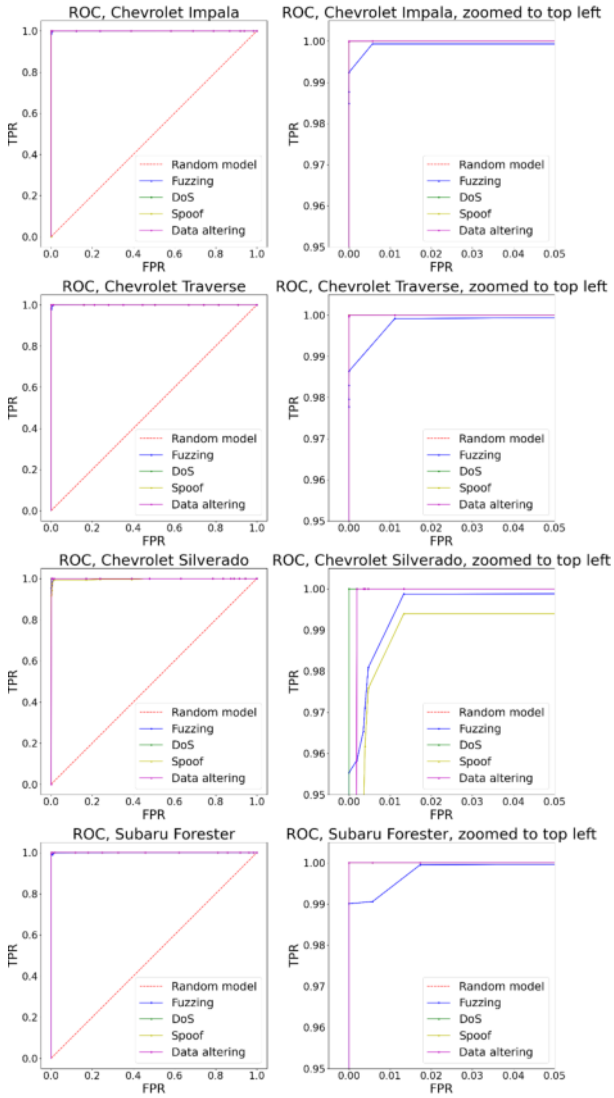

Fig. 5: ROC curves for all car models.

TABLE IV: The AUC scores for each ROC.

	Fuzzing	DoS	Spoof	Data altering
Chevrolet Impala	0.9996	1.0	1.0	1.0
Chevrolet Traverse	0.9998	1.0	1.0	1.0
Chevrolet Silverado	0.99921	1.0	0.99749	0.99903
Subaru Forester	0.99983	1.0	1.0	1.0

C. Cross-model evaluation

To further verify the claim that variations in the car models affect the performance of detection, we do a cross-model evaluation. We train ZeroCAN with normal data from one car and test with data from another car model from the same manufacturer (Chevrolet). This is important because it highlights how model-specific characteristics can impact

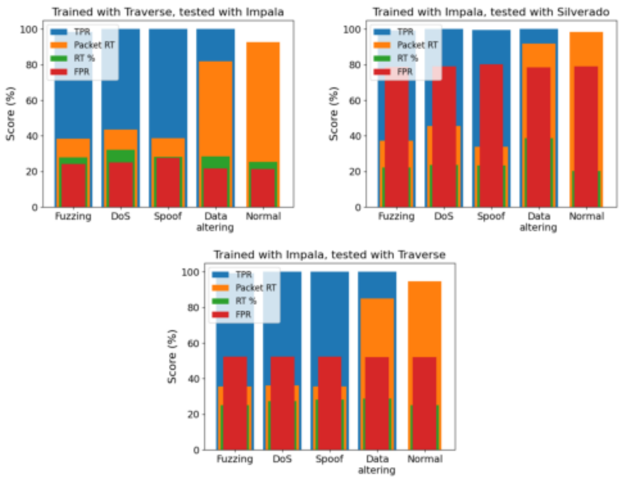

Fig. 6: Comparison of TPR and FPR when the model is trained on data from the same car model as during evaluation.

detection performance, revealing that even within the same manufacturer, differing car models require tailored training data to maintain both high detection accuracy and low false positive rates. Each image in Figure 6 shows the results for a combination of train and test car models. Each attack has a column of stacked pillars presenting the score for each of the four metrics. They are true positive rate (TPR), false positive rate (FPR), a portion of total dataset time used for classification (RT %) and the number of packets classified in time before the next one arrives (packet RT%). The result shows that the detection rate (TPR) remains high, but the false positive rate increases to unreasonable levels compared to when the model is trained on data from the same car model as during evaluation.

VII. Conclusions

In conclusion, this paper introduces ZeroCAN, an effective anomaly detection method to mitigate zero-day attacks on vehicular networks. ZeroCAN employs Support Vector Machines (SVMs) to model the normal behavior of each electronic control unit on the CAN bus, utilizing high-level features that capture both the timing and data payload aspects of network traffic. By distinguishing abnormal patterns in complex high-dimensional data, ZeroCAN achieves a detection rate of more than 99% while maintaining a false positive rate below 0.01% under normal conditions. This work underscores the potential of SVM-based models in protecting vehicular networks against unknown security threats, offering a robust and scalable solution for real-time deployment. Future directions could include further refinement of feature extraction techniques and adaptation to evolving automotive network standards.

Acknowledgment

This work was supported by the CISTER Research Unit (UIDP/UIDB/04234/2020), financed by National Funds through FCT/MCTES (Portuguese Foundation for Science and Technology); by project Route 25 (ref. TRB/2022/00061 - C645463824-00000063), funded by the EU/Next Generation, within call n.º 02/C05-i01/2022 of the Recovery and Resilience Plan (RRP) and HAL4SDV funding within the Chips Joint Undertaking (Chips JU) - the Public-Private Partnership for research, development and innovation under Horizon Europe – and National Authorities under grant agreement n° 101139789.

References

[1] X. Li *et al.*, "Connected vehicles' security from the perspective of the in-vehicle network," *IEEE Network*, 2018.

[2] H. Ali *et al.*, "Navigating the challenges and opportunities of securing internet of autonomous vehicles with lightweight authentication," *IEEE Access*, Feb. 2024.

[3] H. Kurunathan *et al.*, "Drl-keyagree: An intelligent combinatorial deep reinforcement learning-based vehicular platooning secret key generation," *IEEE Transactions on Intelligent Transportation Systems*, Jun. 2024.

[4] R. Islam *et al.*, "Graph-based intrusion detection system for controller area networks," *IEEE Transactions on Intelligent Transportation Systems*, 2022.

[5] R. Islam, M. K. Devnath, M. D. Samad, and S. M. J. Al Kadry, "Ggnb: Graph-based gaussian naive bayes intrusion detection system for can bus," *Vehicular Communications*, vol. 33, p. 100442, 2022.

[6] A. Taylor *et al.*, "Frequency-based anomaly detection for the automotive can bus," in *2015 World Congress on Industrial Control Systems Security (WCICSS)*. IEEE, Dec. 2015.

[7] S. Jin *et al.*, "Signature-based intrusion detection system (ids) for in-vehicle can bus network," in *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2021.

[8] L. Zhang and D. Ma, "A hybrid approach toward efficient and accurate intrusion detection for in-vehicle networks," *IEEE Access*, 2022.

[9] T. Ergen and S. S. Kozat, "Unsupervised anomaly detection with lstm neural networks," *IEEE transactions on neural networks and learning systems*, vol. 31, no. 8, pp. 3127–3141, 2019.

[10] M. D. Hossain *et al.*, "Lstm-based intrusion detection system for in-vehicle can bus communications," *IEEE Access*, 2020.

[11] A. Taylor, S. Leblanc, and N. Japkowicz, "Anomaly detection in automobile control network data with long short-term memory networks," in *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, 2016.

[12] M.-J. Kang and J.-W. Kang, "Intrusion detection system using deep neural network for in-vehicle network security," *PloS one*, vol. 11, no. 6, p. e0155781, 2016.

[13] S. Aziz *et al.*, "Anomaly detection in the internet of vehicular networks using explainable neural networks (xnn)," *Mathematics*, Apr. 2022.

[14] M. H. Khan *et al.*, "Divacan: Detecting in-vehicle intrusion attacks on a controller area network using ensemble learning," *Computers and Security*, 2024.

[15] S. Purohit and M. Govindarasu, "Ml-based anomaly detection for intra-vehicular can-bus networks," in *2022 IEEE International Conference on Cyber Security and Resilience (CSR)*. IEEE, Jul. 2022.

[16] M. Bozdal *et al.*, "Evaluation of can bus security challenges," *Sensors*, vol. 20, no. 8, p. 2364, 2020.

[17] H. Lee *et al.*, "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)*, 2017.

[18] B. Lampe and W. Meng, "can-train-and-test: A new can intrusion detection dataset," in *2023 IEEE 98th Vehicular Technology Conference (VTC2023-Fall)*. IEEE, Oct. 2023.

[19] V. Chandola *et al.*, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 3, pp. 1–58, 2009.

[20] G. Ratsch *et al.*, "Constructing boosting algorithms from svms: An application to one-class classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2002.

[21] P. Malhotra *et al.*, "Long short term memory networks for anomaly detection in time series." in *Esann*, vol. 2015, 2015, p. 89.

[22] A. C. Alice Zheng, *Feature Engineering for Machine Learning, Preface*. O'reilly Media, 2018.

[23] G. Guo *et al.*, "Knn model-based approach in classification," in *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE*. Springer Berlin Heidelberg, 2003.

[24] A. P. Bradley, "The use of the area under the roc curve in the evaluation of machine learning algorithms," *Pattern recognition*, vol. 30, no. 7, pp. 1145–1159, 1997.