Computer Communications 191 (2022) 486–499

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Detection of cybersecurity spoofing attacks in vehicular networks with recurrence quantification analysis

Gianmarco Baldini

European Commission, Joint Research Centre, Ispra, 21027, Italy

A R T I C L E I N F O

Keywords:
Recurrence quantification analysis
Automotive
Cybersecurity
Machine learning
Non-linear analysis
In-vehicle networks
Computer communications

A B S T R A C T

Cybersecurity aspects in modern automotive vehicles are becoming increasingly important due to the recent demonstration of successful cybersecurity attacks. Intrusion Detection System (IDS) is one of the approaches proposed in the research literature to detect such attacks. This paper proposes a novel IDS approach based on the application of Recurrence Quantification Analysis (RQA), in combination with a sliding window, to the information of the CAN-bus message arrival time, which has the benefit of not requiring the processing of the arbitration field or the payload data of the CAN-bus message. The rationale for the application of RQA to this problem is to consider the in-vehicle network as a dynamic system where the CAN-bus message time is used as an observable. This approach is evaluated with various machine learning algorithms on two public data sets recently published by the research community with a focus on spoofing attacks since they are the most difficult to detect. The proposed approach is compared with the application of entropy measures for attack detection, which are commonly adopted in the literature and with the results from the literature on the same data sets. The results show that the RQA based approach provides a better detection accuracy than entropy measures in a consistent way across different sliding window sizes in both data sets and it is competitive against other approaches in the literature. This paper provides also an extensive evaluation of the impact on detection accuracy of the sliding window size and the hyper-parameters present in the definition of RQA and the machine learning algorithms.

1. Introduction

With the evolution of the automotive industry to increased levels of connectivity and automation, the potential for cybersecurity attacks is growing as the vehicle is more exposed to digital attacks. A modern vehicle today is implemented with various electronic components including sensors, actuators, Electronic Control Unit (ECU)s and communication devices, which are connected to different types of in-vehicle networks. One of the most common in-vehicle network standards in the automotive industry is the Controller Area Network-bus (CAN-bus), which has a widespread adoption in the automotive industry and whose security has been significantly investigated in recent times [1,2]. CAN-bus protocol was invented by Robert Bosch GmbH and officially released in 1991. It is a message-based protocol, which was designed to meet the specific requirements of in-vehicle environment, such as real-time processing, strong robustness, and cost effectiveness. The CAN-bus protocol uses broadcast communication to transmit messages among the ECUs accessing the in-vehicle network. A detailed description of the CAN-bus protocol with its frame structure is presented in Section 3.

The remote exploitation of a passenger car was demonstrated in the seminal paper by [3], where the authors managed to demonstrate a successful cybersecurity attack against an unaltered Jeep Cherokee. The weakness exploited by the authors of [3] and other vulnerabilities identified by other authors [2] prompted the research community to investigate more in detail techniques, which could address such vulnerability and mitigate possible attacks.

One of the most common techniques in cybersecurity is Intrusion Detection System (IDS), which has a long history in cybersecurity literature in general (beyond the specific field of automotive cybersecurity) as described in one of the initial surveys on this topic in 1993 [4], where it is mentioned that IDS performs the essential function to detect unauthorized intruders and attacks to the network infrastructure (i.e., the automotive in-vehicle network in this study). The survey [4] mentions two main IDS categories: offline IDS where the analysis of logs and audit records is performed some time after the traffic network operation (e.g., the analysis is executed the day after the network or computer system activity) and the online (or real-time), IDS where the analysis is performed directly on the traffic or immediately after the traffic features are calculated. In the automotive domain, it should be considered that a commercial vehicle has limited computing capabilities, which can be used to implement the IDS. A potential approach could be to outsource the IDS implementation to cloud-based systems with powerful computing capabilities but this approach would require

E-mail address: gianmarco.baldini@ec.europa.com.

https://doi.org/10.1016/j.comcom.2022.05.021
Received 12 October 2021; Received in revised form 5 April 2022; Accepted 19 May 2022
Available online 27 May 2022

significant connectivity means to transmit the traffic logs from the in-vehicle network [5,6]. For example, Deep Learning (DL) has been recently introduced with success in IDS for in-vehicle networks where it achieved excellent detection results in comparison to shallow machine learning [7,8] but the application of DL requires significant computing resources which may not be available in a vehicle but they could be implemented in a cloud system. For this reason, this paper focuses on the online (or real-time) IDS where the analysis is performed directly on the in-vehicle traffic and only 'shallow' (as opposed to 'deep' machine learning) machine learning algorithms are used. Then, the problem addressed by the study presented in this paper is to design a IDS approach, which could be time efficient, minimize the processing of in-vehicle traffic data (e.g., CAN-bus payload data) but still able to achieve a high detection accuracy.

Recent surveys on the design and implementation of IDS in in-vehicle networks [1,5,6,9] have classified IDSs in different categories with each category with specific advantages and disadvantages. One classification is related to the specific observable in the CAN-bus traffic (a brief description of the CAN-bus traffic standard is provided in Section 3.1): the time of arrival of the CAN-bus message, the CAN-ID/arbitration field or the payload data. While the IDSs based on CAN-ID or payload data requires the processing of the CAN-bus message, the time of arrival can be more easily extracted (thus requiring less computing power) from the CAN-bus traffic. Another classification is related to the processing of the in-vehicle traffic in batches or sliding windows where the analysis is performed on features extracted from a set (i.e., the sliding window) of CAN-bus messages rather than the single ones. The sliding window approaches are more time efficient because of the dimensionality reduction introduced by the window [10–13] but they may suffer from a degraded detection performance because information may be lost in the window processing [1,5]. Finally, different types of attacks have been investigated in the literature including the Distributed Denial of Service (DDoS) (where CAN-bus messages of the same type and content are used to overload the in-vehicle network and ECUs), the Fuzzy attack (where CAN-bus messages are randomly selected and injected in the in-vehicle network) and the spoofing attacks where injected traffic by a malicious attacker tries to replicate traffic related to a specific function (e.g., Gear). A detailed description of these attacks is provided in Section 3. From the literature, it is known that spoofing attacks are more difficult to detect [8,12,14] and for this reason, this approach focuses on spoofing attacks even if an analysis and comparison with the literature results is done for other attacks as well (e.g., DoS, Fuzzy) for completeness.

To summarize, with reference to the classification described in the previous paragraphs, the approach proposed in this paper focuses on online real-time IDS for in-vehicle networks with optimal computing efficiency while trying to preserve a good detection accuracy. Then, it is based on the analysis of CAN-bus message arrival time, it adopts a sliding window approach and it uses shallow machine learning algorithm rather than DL. Finally, it focuses only on spoofing attacks as they are more challenging to detect.

Regarding the application of RQA, the proposed approach is based on the assumption that in-vehicle network and the traffic they are carrying can be considered as a system, whose dynamics may give indications on the presence of attacks. The main idea is that the injection of an attack in the in-vehicle network traffic can change its dynamics. In particular, it can impact the arrival time of the CAN-bus messages because of the way the CAN-bus network is designed (see Section 3.1) and it has been proven in the literature that arrival time and inter-arrival time (the time difference between two CAN-bus messages) can be used to detect attacks [15]. Then, non-linear methods designed for monitoring and analysis of dynamic systems could be applied to detect the attack. This assumption was empirically evaluated in this paper with the application of a specific approach called RQA to two recent public data sets of in-vehicle network traffic where spoofing attacks were implemented.

Table 1
Abbreviations used in this paper.

Abbreviation	Explanation
AMI	Average Mutual Information
AuC	Area Under Curve
CAN	Controller Area Network
DDoS	Distributed Denial of Service
DoS	Denial of Service
DT	Decision Tree
ECU	Electronic Control Unit
ER	Error Rate
FNR	False Negative Rate
FPR	False Positive Rate
ICT	Information and Communication Technology
KNN	K Nearest Neighbor
IDS	Intrusion Detection System
MaxDL	Maximum Diagonal Line
MeDL	Mean of the lengths of the Diagonal Lines
OBD-II	On-board diagnostics version 2
ROC	Receiver Operating Characteristics
RP	Recurrence Plot
RPM	Revolutions Per Minute
RQA	Recurrence Quantification Analysis
RTE	Recurrence Time Entropy
RR	Recurrence Rate
SVM	Support Vector Machine
TPR	True Positive Rate

The two data sets were chosen because they were created using real vehicles in both stationary and real driving conditions, are fully labeled to support a supervised ML method and they are significantly large in size (i.e., contain millions of CAN-bus messages).

Regarding the main algorithm used in this study, the RQA is a method of nonlinear data analysis (e.g., chaos theory) for the investigation of dynamical systems. It quantifies the number and duration of recurrences of a dynamical system presented by its phase space trajectory [16]. RQA has been used for traffic classification in [17] but it has never been used (to the knowledge of the author) for the detection of cybersecurity attacks in the automotive in-vehicle network traffic and even more specifically for spoofing attacks.

Table 1 presents all the abbreviations used in this paper.

This paper is organized as follows: Section 2 provides a state of art of related work for IDS in in-vehicle networks with a specific focus on approaches based on the sliding window and statistical or information theory features. It also gives an overview of the application of RQA in cybersecurity or traffic analysis. Section 3 describes the CAN-bus protocol and the data sets used in this study, how they were generated and the related attacks scenarios. Section 4 describes the overall methodology used in the proposed approach including the machine learning algorithms and the metrics of evaluation. This section also describes the RQA used in this paper and the related hyper-parameters present in the design of RQA and the adopted shallow machine learning algorithms and metrics to perform the classification and detection of the attacks. Section 5 provides the results of the evaluation on the two different data sets and an analysis of the impact of the hyperparameters. Section 6 draws conclusions and describes future developments.

2. Related work

The field of IDS, in general, is quite vast and even the specific field of IDS in in-vehicle networks has many references. For reasons of space, this section describes the studies in the literature, which are pertinent to the proposed approach and the data sets used in this paper. In particular, this section focuses on three different areas: (a) intrusion detection systems in in-vehicle networks with a specific focus on sliding windows methods, (b) the application of RQA in the field of cybersecurity and traffic analysis, and (c) studies using the same data sets used in this paper.

As it was already described in the introduction, this paper proposes a time efficient IDS based on a sliding window approach combined with machine learning and applied to the CAN-bus in-vehicle traffic observable, which is the time of arrival of the CAN-bus message. Various studies have adopted the time of arrival as observable. In [18], the authors have used the time intervals between sequential CAN-bus messages (as in this paper) obtaining detection accuracy up to 100% for the DDoS attack. On the other side, the DDoS attack is relatively easier to detect than spoofing attacks addressed in this paper as shown in various studies [8,12,14]. In addition, the approach proposed in [18] is based on the analysis of each CAN-bus message while this paper proposes a more time efficient approach based on a sliding window. Another study exploiting the differences in the distributions of the inter-arrival times of the CAN-bus messages (like this paper) is [19], which demonstrates that the analysis of the distribution of the inter-arrival times has discriminating power in detecting attacks (in particular the DDoS attacks). A recent study on the application of the sliding windows to vehicular IDS is also presented in [11] where a similarity-based IDS approach is presented and compared to Shannon entropy based approach for DoS attacks. The results show that the similarity-based IDS approach has low computing complexity but it is still able to achieve very high accuracy for DoS attacks. On the other side, other attacks are not explored.

Statistical analysis of the traffic in the in-vehicle network uses various methods including the application of information theory measures (e.g., entropy measures). A common approach is to use a sliding window of the in-vehicle network traffic where the measures are calculated on the window data. The rationale is that the value of the calculated measures can change significantly between the normal traffic and the traffic where the attacks are present.

A paper that adopts a similar approach to ours is [13], where the authors have used Shannon entropy to detect two types of attacks: a replay attack (comparable with spoofing attacks) and a fuzzy attack. A sliding window where the entropy is calculated and evaluated against a threshold k was used. This initial study was based on the timing of the messages and the detection accuracy suffered when the rate of attacks is relatively low. This study uses [13] for comparison in Section 5 because it adopts the same observable (i.e., the timing of the CAN-bus message), it adopts a sliding window approach as in this paper, it also addresses spoofing attacks and it extracts features from the window of the in-vehicle network traffic. The difference between this paper with [13] is that RQA is used instead of Shannon entropy. A sliding window approach to detect intrusions in in-vehicle networks is also proposed in [10] where it is used to detect two different types of attack: a DoS Attack and injection attacks including spoofing attacks. In [10], the impact of different window sizes is evaluated as well as the threshold used to determine when an attack is implemented or not. In comparison to this paper, the authors of [10] use the arbitration field (also called CAN-ID), which requires the processing of the CAN-bus messages. The authors of [10] use Shannon entropy as a detection feature. Both [10,13] show that spoofing attacks are more difficult to evaluate in comparison to DDoS attack and this is the reason why this paper focuses on spoofing attacks.

Another study based on the sliding window approach is [20], which also uses the time interval of the CAN-bus messages to detect spoofing attacks. The authors of [20] used the time interval information and the correlation coefficient between offsets and time intervals to detect fuzzy, DoS, and impersonation attacks (thus including spoofing attacks).

Other papers used additional entropy measures beyond Shannon Entropy together with the sliding window approach. For example the authors in [21] use the Renyi entropy of orders 2, 3 and 4 to detect a DoS and Fuzzy attack. The approach is evaluated for different sizes of the sliding window. Contrary to this paper, the authors of [21] have used the observable of the arbitration field for their analysis.

Other papers have used alternative approaches. For example, the recent paper [22] applied an improved isolation forest method with data mass (MS-iForest) for data tampering attack detection where data mass is used as the base function to group traffic sample in regions for further processing by the isolation forest algorithm.

Finally, other studies have used the sliding window in combination with deep learning. In [8], the authors have used a fixed sliding window to transform the in-vehicle traffic to images, which are given as in input to a convolutional neural network for classification. On the other side, the application of deep learning algorithms requires considerable computing resources, which is the reason why this paper uses conventional ('shallow' in opposition to 'deep') machine learning algorithms.

The application of RQA in IDS has not been investigated to the best knowledge of the author in in-vehicle networks, but there are few examples in the IDS applied to ICT infrastructures. For example, in [23], the authors have used RQA based features (named non linear analysis) for intrusion detection in conventional (non automotive related) networks achieving high accuracy. In a similar way, recurrent plots (which form the basis of RQA) are used in [24] for the implementation of intrusion detection systems. The approach is similar to ours in the definition of a sliding window where the recurrent plots are calculated and the features are extracted.

In a similar way, another paper, which investigated the application of RQA for network traffic analysis is [25] where the natural complexity of wireless mobile network traffic dynamics has been assessed by tracing the presence of non-linearity and chaos in the profile of daily peak hour call arrival and daily call drop of a sub-urban local mobile switching center.

Finally, we identify the studies, which use the same data sets used in this study. Apart from the Refs. [8,14] created by the same authors of the CarHack2018 data set, other authors have used the same data sets and published the results of their evaluation. In [26], the authors have focused on the family of Tree-based machine learning algorithms (e.g., Decision Tree, Random Forest) on the CarHack2018 data set achieving very good accuracy, but a sliding window approach was not used. In [27], the authors have focused only on the DoS and Fuzzy attacks of the CarHack2018 data set using KNN and SVM machine learning algorithms. As in this paper, the IDS design is based on the analysis of the offset ratio and time interval between the messages request and the response in the CAN-bus protocol, but no sliding window approach was used and the spoofing attacks were not considered. A very recent study using a sliding windows approach on the CarHack2018 data set is presented in [28] where histograms are created on the CAN-bus payload. When a malicious attack is detected, a filtering method is applied to filter normal traffic. Finally, the KNN algorithm and one-class SVM is used to implement the ML classification. Then, it is a two step based approach. The proposed approach manages to achieve 100% accuracy for the spoofing attacks and less for the other attacks in the CarHack2018 data set. In comparison to this study, the advantage of the approach proposed in this paper is that the payload data does not need to be processed.

To the best knowledge of the author, only two studies have been published (at the time of drafting this study) which use the CarHack2020 data set. The first study [29] is by the authors of the data set itself but it provides a description of the hackathon challenge related to the data set and it does not provide the specific details on the used approach. Instead, one recent and quite extensive study [30] applies deep learning algorithms (RNN, CNN, LSTM) to the CarHack2020 data set and compares the results with 'shallow' machine learning algorithms like DT, SVM and KNN. The DL algorithms manage to achieve a higher detection accuracy. The sliding window method is not used. In addition, the study in [30] uses both the CAN-ID and the CAN-bus payload data.

To summarize, the study presented in this paper advances the state of art in intrusion detection in in-vehicle networks by applying RQA in combination to the sliding window concept and the inter-time of arrival of the CAN-bus messages, which are known (from the references identified above) to be more time efficient than other observables in the CAN-bus messages, which require the processing of the payload data or the CAN-ID (i.e., arbitration field).

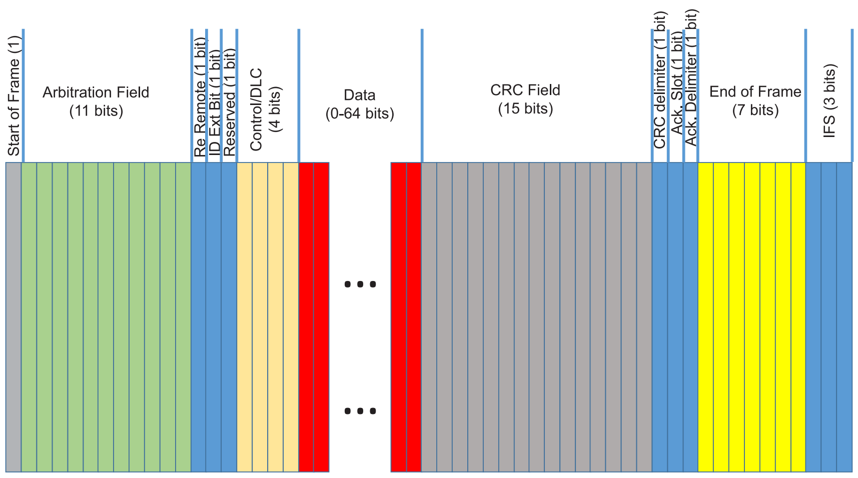

Fig. 1. Standard CAN-bus message frame.

3. Data sets and attack scenarios

To evaluate the proposed approach, two different public data sets are used, which have been recently created and published: the first one (called CarHack2018 in the rest of this paper) was generated by the Hacking and Countermeasures Research Lab and it is described in [8,14], the second data set (called CarHack2020 in the rest of this paper) is the Car Hacking: Attack and Defense Challenge 2020 Data set, which is described in [29,31].

These data sets have been chosen because: (1) they are based on the collection of extensive (i.e., millions of CAN-bus messages) and real data (i.e., not simulated) from vehicles where cybersecurity attacks are implemented by the authors of the data set, (2) both data sets are fully labeled which is essential, because the approach proposed in this paper is based on supervised learning, (3) both data sets included spoofing attacks (other public data sets only provide DoS or Fuzzy attacks) and (4) both data sets are relatively recent (2018 and 2020).

This section is structured in three sub-sections: Section 3.1 describes the CAN protocol, Section 3.2 describes the CarHack2018 data set and Section 3.3 describes the CarHack2020 data set.

3.1. Description of the controller area network protocol

CAN-bus protocol was invented by Robert Bosch GmbH and officially released in 1991. It is a message-based protocol, which was designed to allow robust communication among microcontrollers in a vehicle and meet the specific requirements of in-vehicle environment, such as real-time processing, strong robustness, and cost effectiveness. CAN-bus protocol uses broadcast communication to transmit messages.

A description of the standard CAN-bus (CAN 2.0) frame structure with the identification of the specific fields is provided in Fig. 1.

As mentioned before, the focus of this paper is not on the use of the CAN-bus fields of the CAN messages like the arbitration field/ CAN-ID used in [8,10,14] or the payload data used in [12] but on the arrival time stamp of the CAN-bus message, which does not require specific processing of the CAN-bus messages and it is more time efficient.

3.2. Description of the CarHack2018 data set

The data set CarHack2018 is based on data extracted from a Hyundai YF Sonata through a Y-cable plugged into the OBD-II port through a Raspberry Pi3 as described in [8,14]. The recorded CAN-bus traffic matches the specification of CAN 2.0 with a CAN-bus message interpretation based on the Hyundai YF Sonata model.

The data sets contain each 300 intrusions of message injection. Each intrusion is performed for a time ranging from 3 to 5 s, and each data set has total 30 to 40 min of CAN-bus traffic. Then, the data sets are quite extensive and they contain millions of messages as described in the following Table 2:

Table 2
Data set used in this paper from [8,14].

Attack type	Number of messages	Number of normal messages	Number of injected messages
DoS attack	3,665,771	3,078,250	587,521
Fuzzy attack	3,838,860	3,347,013	491,847
Spoofing the drive gear	4,443,142	3,845,890	597,252
Spoofing the RPM gauge	4,621,702	3,966,805	654,897

The four attack scenarios are described below:

- In the Denial of Service (DoS) attack, messages of '0000' CAN-bus ID were inserted in the in-vehicle network every 0.3 ms.
- In the Fuzzy attack, totally random CAN-bus ID and payload data values of the CAN-bus messages were injected every 0.5 ms.
- In the Spoofing attack of type RPM, messages related to the RPM information were injected every 1 ms.
- In the Spoofing attack of type Gear, messages related to the Gear information were injected every 1 ms.

The data sets were created by logging CAN-bus traffic (from 30 to 40 min of CAN-bus traffic) via the OBD-II port from a real vehicle while message injection attacks were performed. As described in [8,14] and other papers, which used this data like [12], the spoofing attacks (RPM and Gear) are the ones most difficult to detect because these attacks replicate traffic naturally generated in the vehicle. For this reason, this paper focuses mostly on spoofing attacks even if a comparison with results in the literature is also presented for the other two attacks (e.g., DoS, Fuzzy) in the Results Section 5.

3.3. Description of the CarHack2020 data set

The second data set CarHack2020 was recently created in the context of a competition aimed to develop attack and detection techniques of CAN-bus [29,31]. The target vehicle of competition was of model Hyundai Avante CN7. The data set is CAN-bus network traffic of Avante CN7 including normal messages and attack messages. Various data sets were created for the competition: a preliminary training set, a preliminary submission set, and the final submission data set without labels. In this paper, we have used only the preliminary training set because it is fully labeled and this paper is focused only on supervised learning. Since the data set is created both for a stationary scenario (i.e., vehicle switched on) and a driving scenario (i.e., vehicle driving)

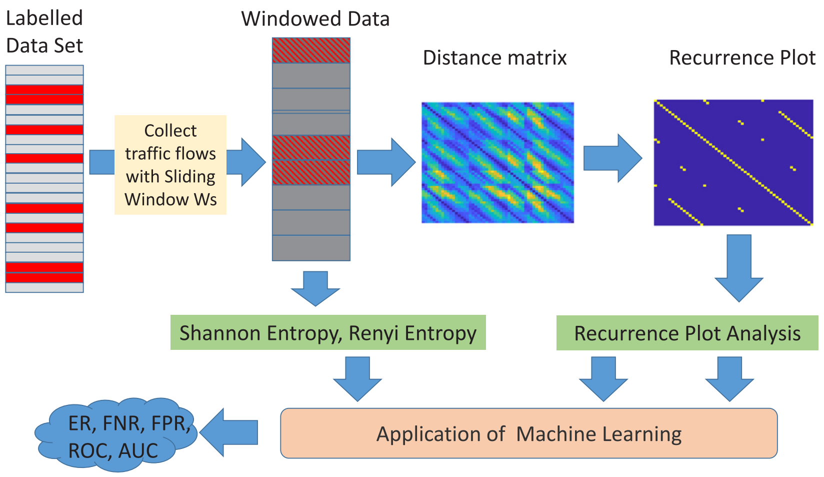

Fig. 2. Overall workflow for the analysis of in vehicle traffic data and the application of Recurrence Plot Analysis.

our approach was evaluated in both cases. The stationary scenario data set is called CarHack2020Sta or CarHack2020Dri in the rest of the paper. In a similar way to the previous data set CarHack2018, different attacks are present: (1) Flooding attack which aims to consume CAN-bus bandwidth by sending a massive number of messages. This attack disrupts normal driving and limits the communication between ECU nodes by sending high frequency and high priority messages (e.g., 0 000). This is equivalent to the DoS attack of CarHack2018. (2) Spoofing where CAN-bus messages are injected to control certain desired functions. (3) Fuzzing (equivalent to the Fuzzy attack of CarHack2018) where random messages are injected to cause unexpected behavior of the vehicle. In the fuzzy attack, a malicious ECU transmits random frames with spoofed CAN IDs with arbitrary data values, which caused the vehicle function to be unavailable (e.g., 0 4CC, 0 7C6). As reported in [31], the data set is quite extensive with 3,372,743 Normal CAN-bus messages, 299,408 Attack CAN-bus messages for a total of 3,672,151 CAN-bus messages. As stated before, the focus of this paper is on the spoofing attacks and most of the analysis is implemented on these attacks, but an evaluation of the approach presented in this paper on the other attacks is also presented in the Results Section 5.

4. Materials and methods

4.1. Workflow

The description of the workflow for the processing of the data is described in this section and it is pictorially described in Fig. 2. As described in the introduction, a sliding window approach is implemented where a set of CAN-bus messages is processed and used to generate a sample for the subsequent data analysis process. The number of CAN-bus messages (i.e., the window size of the sliding window) used to create the sample is defined by the parameter W_s in the rest of this paper. In a similar way to what has been done in the literature, a sample of size W_s is considered normal/legitimate (Note: the terms legitimate traffic and normal traffic have the same meaning in the rest of this paper.) if the sample contains only normal CAN-bus messages. The sample is considered malicious (e.g., an attack is being implemented) if it contains at least a single CAN-bus message labeled as malicious in the data set. The use of a moving window allows faster detection of the attack as the CAN-bus messages are processed in 'batches' (i.e., the samples) rather than a single CAN-bus message at the time. In this paper, the choice is to avoid overlapping among samples: no CAN-bus message belongs to two samples at the same time. The reason for this choice is to foster time efficiency as overlap would obviously increase the processing time.

As mentioned before, the observable used in the approach is the inter-arrival time between two consecutive CAN-bus messages. The arrival time of the CAN-bus messages is information, which is present in the data sets. Then, in the initial pre-processing phase, the difference in arrival time between subsequent messages is calculated. The initial CAN-bus traffic data set is transformed (for all the data sets) to a time series x_i (i = $1, \ldots, N_T$-1, where N_T is the number of CAN-bus messages in the data set) on which the sliding window is applied. The entire time series T_i is segmented in non-overlapping segments of size W_S, where W_S itself is an hyper-parameter in the data analysis. In other words, the proposed approach is evaluated against different values of W_S. For each segment, the distance matrix and the related recurrence plot is calculated.

A visual presentation of the recurrence plots for the normal and the spoofing attack traffic is shown in Figs. 3 and 4 respectively for the normal and spoofing related traffic in the CarHack2020Sta data set. Even a visual check shows the difference between normal traffic and the traffic where the spoofing attack is present.

As described in the definition of recurrence plots and RQA in Section 4.2, the threshold T_{hr} to generate the recurrence plot is another hyper-parameter in the approach. While it is possible to adaptively calculate the T_{hr} on the basis of the distribution of the Recurrence Plots (RP) elements, the final goal is the detection accuracy which may not be fully related to the adaptive criteria to define T_{hr} and a more extensive grid approach is preferred. Then, an extensive number of RQA features are calculated from the recurrence plot to generate a feature matrix, which is fed to machine learning algorithms. The delay parameter is set to 1 in this study, because delay values greater than 1 may generate the risk to miss CAN-bus messages including attacks.

To compare the approach proposed in this paper with other approaches proposed in the literature, the Shannon entropy and Renyi entropy of order 2,3,4 are also calculated to generate other feature matrices. After the entropy measures are extracted, machine learning algorithms are applied on the feature matrices to estimate the detection accuracy on the basis of different metrics like Error Rate (ER), False Positive Rate (FPR) and False Negative Rate (FNR), Receiver Operative Characteristics (ROC), and Area Under Curve (AUC) as described in Section 4.3.

4.2. Recurrence quantitative analysis

The classification strategy to identify the spoofing attacks in the in-vehicle traffic starts from the identification of nonlinear characteristics, such as recurrence phenomena and non-stationary transition patterns in the time series created from the segmentation of the in-vehicle network traffic with the sliding window and the extraction of the inter arrival time. The main idea is to reconstruct the (unknown) system dynamics in the phase space by using time-delay embedding, then computing

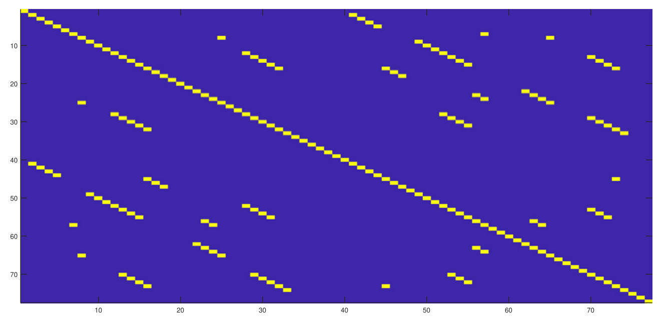

Fig. 3. Example of Recurrence Plot created from in-vehicle traffic in presence of the normal traffic (CarHack2020Sta data set).

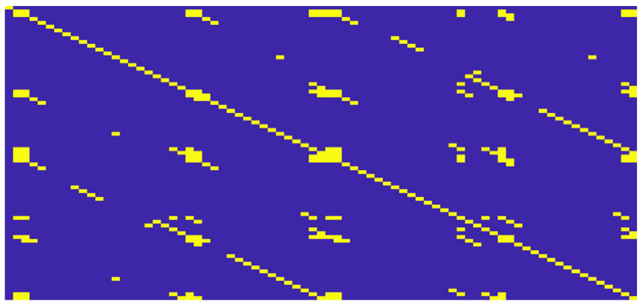

Fig. 4. Example of Recurrence Plot created from in-vehicle traffic in presence of a spoofing attack (CarHack2020Sta data set).

Table 3
RQA features used in the study described in this paper.

Id	Acronym	Description
1	RR	Recurrence rate, which is the measure of the density of recurrence points in the RP. It is also called correlation sum.
2	DET	Determinism, which is the ratio of recurrence points that form diagonal structures to all recurrence points.
3	MeDL	Mean of the lengths of the diagonal lines found in the RP.
4	MaxDL	Longest diagonal line found in the RP.
5	Entr	Shannon entropy-based on the probability p(l) to find a diagonal line of exactly length l in the RP.
6	LAM	Laminarity. Ratio between the recurrence points forming the vertical structures and the entire set of recurrence points
7	TT	Average length of the vertical structures in the RP.
8	Vmax	Maximal length of the vertical lines in the RP.
9	RTmax	Maximal white vertical line length in the RP.
10	T2	Recurrence time of 2nd type. T2 contains information about the time distance between the beginning of subsequent recurrence structures in the RP.
11	RTE	Recurrence Time Entropy. It contains information about the periodicity characteristics of a signal related to the RP in the context of dynamic systems.
12	Clust	Clustering coefficient. Average of the clustering coefficients for the states in the RP.
13	Trans	Transitivity. Transitivity of a complex network is related to the probability that two neighbors of any state in the RP are also neighbors, and this measure indicates how much a network is locally clustered.

the distances between all pairs of embedded vectors, generating a symmetric two-dimensional square matrix (the distance matrix) on which the recurrence plots are created and the RQA features are generated (see [16,32] for a description of RQA and the definition of the related lemmas and proofs). This section describes how this is achieved, which hyper-parameters are defined in the process and which RQA features are calculated.

To help to determine the non statistical behavior of the vehicle as a system from the in-vehicle network traffic, we can lean on the Takens theorem [33], which states that a topologically equivalent picture (i.e., the phase space trajectory) of the behavior of the original system can be generated from the time series of a single observable variable, by means of the method of time delays.

Regarding notation, in the rest of this section, the time series x_i with $i = 1, \dots, N$ is the value of the inter-arrival time of the in-vehicle network traffic. N is equal to the size of the sliding window is W_S.

Then, we can create a set of all embedded vectors y(i) in the space $\mathfrak{R}^M$ defined by:

$$y(i) = (x_i, x_{(i+\tau)}, x_{(i+(2*\tau))}, x_{(i+(3*\tau))}, x_{(i+((M-1)*\tau))}) \tag{1}$$

where M is the embedding dimension and τ is the time delay. In this context, $1 + ((M-1) * \tau)$ must be less than W_S. Considering that the value of τ is set to 1 to ensure that all relevant CAN-bus messages are used to detect the attack, this condition is satisfied for the values of M considered in this analysis (see Section 5 for the range of M values). The set of all embedded vectors y(i), constitutes a trajectory in space $\mathfrak{R}^M$. Then, each unknown point of the phase space at time i is reconstructed by the delayed vector y(i) in the M-dimensional space called the reconstructed phase space. The lemmas and proofs related to the definition of RQA are described in the Refs. [16,33], and [32].

While there are different methods to determine the value of m like the Average Mutual Information (AMI) used in [23], in this paper the values are evaluated empirically to evaluate their impact on the detection accuracy of the spoofing attack as shown in the Section Results 5. In other words, they are hyper-parameters of the proposed approach together with the size of the sliding window W_S and the machine learning parameters. An additional hyper-parameter is the threshold T_{hr}, which is described below. As mentioned before, the delay parameter is set to 1.

The next step of the approach is to build the RPs for each sliding window. The RP is a two dimensional representation created by calculating the mutual distances between embedded vectors of the phase space and comparing it to a threshold. In other words, the RP is a two-dimensional graphical representation of the distances matrix $D = d_{i,j}$, where the pixel located at coordinates (i, j) is shaded according to the distance between the i_{th} and j_{th} vectors and a fixed cutoff value T_{hr}. The meaning of the RP is to provide information about the temporal correlation of phase space points since each horizontal coordinate i in RP refers to the state of the system at i and each vertical coordinate j refers to the state in j. Such temporal correlation indicates the system dynamic status in each sliding window. The assumption of the approach is that a spoofing attack would modify the sequence of the inter-arrival times which in turn represents the dynamic status of the in-vehicle traffic. Then, features extracted from the RP may have discriminating power to distinguish traffic windows that contain spoofing attacks from normal traffic conditions.

The final step is to implement the RQA and extract features from the generated recurrence plots. The RQA features described in Table 3 are defined from [16,32] and they are used in the approach proposed in this paper. For the reproducibility of the results, the MATLAB code provided by the authors of [16,32] is used in this study. In the table, the first column is a numeric identifier used to identify the feature in the subsequent sections of this paper and especially in Section 5. The second column is an acronym of the feature and the third column provides a brief description of the feature. A more detailed explanation of these features can be found in [16].

4.3. Machine learning algorithms and metrics of evaluation

Three different machine learning algorithms were used for the analysis. The algorithms were selected on the basis of their usages in the literature, on the need to address a heavy unbalanced data set where the traffic samples related to traffic are much less than the traffic samples related to normal traffic, and on the need for computing efficiency. Both the Naive Bayes algorithm and the Decision Tree (DT) algorithms are known to be robust against unbalanced data sets and quite scalable for large data sets (as in this case where the data sets are composed of millions of records). The DT algorithm was used in [26] for the implementation of the IDS on the same CarHack2018 data set used in this paper. The K Nearest Neighbor (KNN) algorithm was also used in the evaluation because it is relatively simple and it is commonly used as a baseline. It was used in [27] for the implementation of the IDS algorithm in the CarHack2018 data set.

The hyper-parameters for the DT algorithm were defined as following: the maximal number of decision splits per tree (this hyper-parameter is called N_B in the rest of this paper) and the split criterion among the choice of Gini's diversity index, twoing rule and cross-entropy. The kernel type was the hyper-parameter identified for the Naive Bayes algorithm. The hyper-parameter for the KNN is the parameter K. Euclidean distance was used.

A 3-fold approach was used for classification, where 1/3 of the data set was used for test, and 2/3 was used for training and validation. The overall classification process was then repeated 10 times, each time with different training and test sets. The final results were averaged. The metrics of evaluations are: the Error Rate (ER) expressed as $1 - (TP+TN)/(TP+TN+FP+FN)$, the FPR expressed as $FP/(FP+TN)$, the FNR expressed as $FN/(FN+TN)$, the ROC, and the AUC. The ROC curve is created by plotting the True Positive Rate (TPR) against the FPR at various threshold settings and the AUC is the area under the ROC curve. A higher value of AUC means a higher detection performance. TP is the number of true legitimate traffic samples correctly identified as legitimate traffic. FP is the number of traffic samples falsely predicted as legitimate traffic but actually related to an attack. The design of the IDS should minimize FP to ensure that attacks are correctly identified. FN is the number of traffic samples falsely predicted as attack traffic but actually related to legitimate traffic. The design of the IDS should minimize FN because it increases the operational burden of an administrator: s(he) has to react to the IDS notification to see if there was really an attack even if it was not the case. Finally, TN represents the number of traffic attack related samples, which are correctly identified as attacks.

Regarding the simulation environment, the analysis on the data sets was performed using MATLAB language on a laptop with Intel processor i7-8550 CPU with a 1.8 GHz clock and 16 GB of memory.

5. Results

This section provides the results on the application of RQA for the two data sets used in the analysis. This section is divided into three main sub-sections. Section 5.1 provides an analysis of the impact of the hyper-parameters on the detection accuracy. Section 5.2 compares the results obtained with this approach based on RQA with the results obtained with the entropy measures commonly used in the research literature. Finally, Section 5.3 provides the results obtained with this approach based on the RQA with the results obtained in the literature on the same data sets CarHack2018 and CarHack2020 used in this paper, taking into consideration that completely different approaches and data (e.g., CAN-ID instead of time differences) might be used.

5.1. Analysis of the impact of the hyper-parameters

An initial analysis was conducted on the optimal choice of the W_S window size and the M and T_{hr} parameters defined in RQA. The delay parameter was set to 1 to consider all the elements in the time series because an attack may be present even on a single CAN-bus message. A grid-search approach was used with a range of parameters of $W_S = [100, \dots, 1500]$, a range of $M = [2, 3, 4, 6, 8, 12, 14]$ and a range of $T_{hr} = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.95]$.

The following figures presented in this section show the results for a specific set of hyper-parameters while keeping fixed one or more hyper-parameters.

Fig. 5 are color maps, which show the impact of the choice of the parameters M and T_{hr} for the specific value of $W_S = 500$. The Decision Tree ML algorithm was used to implement the detection process (a performance analysis of the different ML algorithms is presented later in this section). The optimal values of M and T_{hr} are indicated in the title of the figures. It can be seen from these figures that the range of values adopted for the optimization phase is adequate since the optimal values of M and T_{hr} are located well within the border of the map. From these initial results, it can be seen that the detection accuracy of the proposed method is higher for the CarHack2018 data sets rather than the CarHack2020 data sets. This is reasonable since they have been created in different conditions. These results can be used as an indication of the optimal values of the hyper parameters for a practical deployment of the IDS based on RQA presented in this paper. On the other side, different vehicle models or driving scenarios may have different values of the hyper-parameters since they are dependent on the architecture of the vehicle and its characteristics (e.g., type of engine). From a practical point of view, the optimal values should be identified in a training phase of the IDS algorithm and eventually repeated in different times/conditions of the vehicle operation. Such practical considerations are not specific to the approach presented in this paper, but they are common to vehicular IDSs based on machine learning [1,5,6].

Then, the detection performance of the proposed approach was evaluated for a wide range of values of W_S. The trade-off in defining the range is that smaller values of W_S increase the time needed for the data analysis because the time series is segmented into a large number of segments while larger values of W_S can make more challenging the identification of the specific time of the attack. In addition, since a supervised ML approach was adopted, larger values of W_S decrease the samples space since the overall duration of the data set is constant. The range of values of W_S was chosen on the basis of these considerations and the adoption of similar values in the literature.

5.2. Comparison with entropy measures

The performance of the proposed approach based on the use of RQA was also compared in the following set of figures with the entropy-base approach where Renyi and Shannon entropy measures were used.

Figs. 6 and 7 show such comparison for the CarHack2020 data set (respectively for driving and stationary vehicle) while Figs. 8 and 9 show the comparison for the CarHack2018 data set (Gear and RPM spoofing attacks). The figures show the results for the metrics of accuracy, FPR, and FNR. As for the previous results, the DT ML algorithm was used to implement the detection process (a performance analysis of the different ML algorithms is presented later in this section)

It can be seen that the proposed approach outperforms significantly the use of entropy-based measures for both data sets. This is a significant result, because it shows that RQA can successfully replace entropy-based measures in in-vehicle intrusion attack systems based on the use of CAN-bus message time-interval information because of its higher detection performance.

On the other side, it can be seen that the RQA based approach performs very well in absolute terms (detection accuracy higher than 95% for most of the values of W_S) in the CarHack2018 data set while it performs less (but still better than entropy-based measures) in the CarHack2020 data set. In particular, the FPR is quite poor for the entropy measures (almost total failure in recognizing normal traffic), while it is significant even for RQA. As a first consideration, it is noted that it is the first time (to the knowledge of the author) that a time-interval approach was used on the CarHack2018 and it may not be

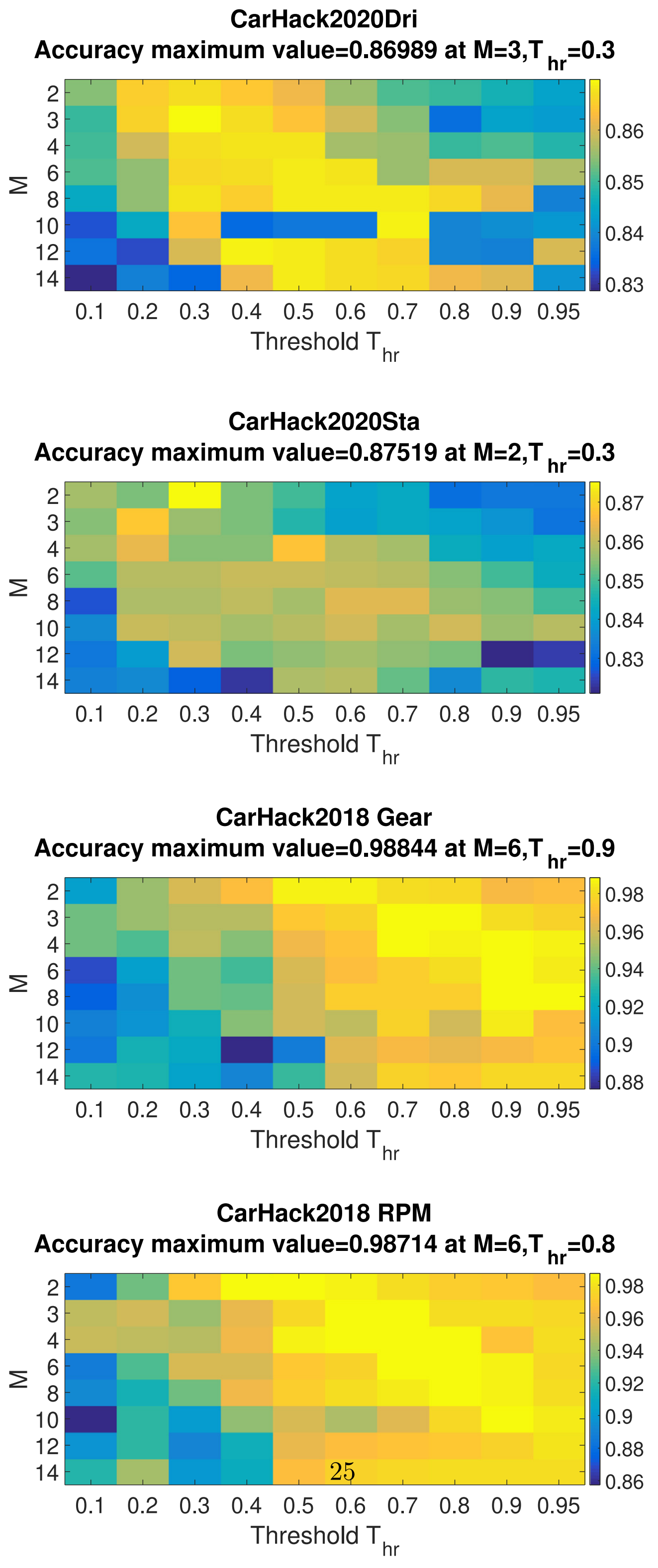

Fig. 5. Impact of M and T_{hr} hyperparameters on detection accuracy for $W_S = 500$. The color bar indicates the accuracy values. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

particularly suitable for this data set (other fields in the CAN-bus messages like the arbitration field may have higher discriminating power). On the other side, the results show that the RQA based approach is more stable than the entropy-based approach since the FPR and FNR are more balanced. As shown in the accuracy subplot, the RQA based approach has an overall detection performance, which is better than the one obtained with the entropy-based measures (both Renyi and Shannon entropy measures). It is also noted for both data sets that there is no need to select high values of W_S because the optimal results are obtained with W_S in the 400–800 range. The results obtained in the CarHack2018 data set are more stable than the CarHack2020 data set, which could also point to a possible explanation of the poor relative performance of the time-interval based approach. The performance of the proposed approach based on the sliding window and the extraction of features (either RQA or entropy-based) may be lower if the data set is not uniformly distributed and it is dispersed. Then, a potential reason for the reported results is that the CarHack2018 data set may be less dispersed than the CarHack2020 data set for the time interval information of the CAN-bus message. To prove this assumption, a simple check was implemented by calculating the variance of the entire time series for both data sets. The variance of the stationary time series of the CarHack2020 data set was calculated to be $Var = 1.7030$. The variance of the driving time series of the CarHack2020 data set was calculated to be $Var = 3.5531$. The variance of the gear spoofing attack time series of the CarHack2018 data set was calculated to be $Var = 0.2514$ and the variance of the RPM spoofing attack time series of the CarHack2018 data set was calculated to be $Var = 0.1799$. These values seem to confirm the initial assumption that the approach proposed in this study is more applicable (i.e., provides better accuracy and more stable results) when the variance of the time series is in the range of the CarHack2018 data set.

All the previous results have been obtained using the DT algorithm, which is one of the three selected ML algorithms. While it is not the intention of this study to conduct a detailed analysis of the impact of the choice of the ML algorithm since the focus of this study is on the use of RQA in IDS, a comparison of DT with Naive Bayes and KNN is presented in Fig. 10 and the related subplots. For space reasons, only the accuracy metric is used for comparison. All the three machine learning algorithms have been optimized on the basis of the method described in Section 4.3 and the range of ML hyper-parameters described in that section. We note that the KNN algorithm has a slightly better performance than the DT algorithm for the CarHack2020 data sets. As a consequence, it is used in the subsequent results in this section.

On the basis of the previous results, which show that KNN has a superior performance than DT for the CarHack2020 data set, the classification process was executed again for the different values of hyper-parameters.

Table 4 provides a summary of the optimal results obtained across the range of the hyper-parameters M, T_{hr} and W_S for the RQA based approach and the entropy measures for all the considered four time series of the two data sets CarHack2018 and CarHack2020.

The improvement in accuracy of the proposed approach based on RQA can come at the cost of the higher computing complexity in comparison to the entropy measures. An empirical indication of the computing costs on the application of RQA to the data sets shows that the calculation of the RQA features and the related classification time is from 15 times to 18 times the time requested to calculate the entropy measures and perform the related classification. While this higher computing time can affect both the training phase and testing phase, it should be considered that it is still a limited time in absolute terms. For example, the calculation of a sliding window of 1500 CAN-bus messages (the longest window considered in the study) using RQA still executes in 0.4 s on a laptop with an Intel main processor i7-8550 with a 1.8 GHz clock.

To provide a more detailed assessment of the performance of the proposed approach based on the RQA features, the ROC figures and the related AUC values were calculated using the optimal values identified in the second column of Table 4. Two sets of ROCs are provided in the following figures for both data sets. The first set of ROCs shown in

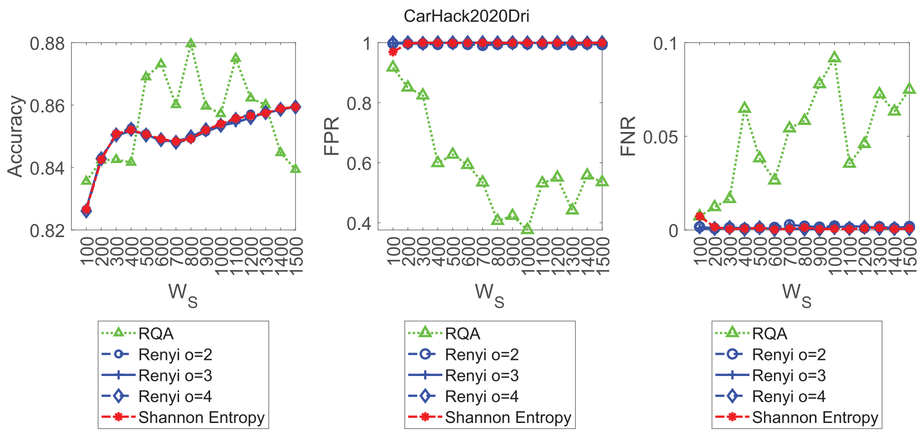

Fig. 6. CarHack2020Dri Driving data set. Accuracy, FPR and FNR for optimal values of M and T_{hr} for RQA features, Renyi entropy of different orders, and Shannon entropy features for different values of W_S (size of the sliding window).

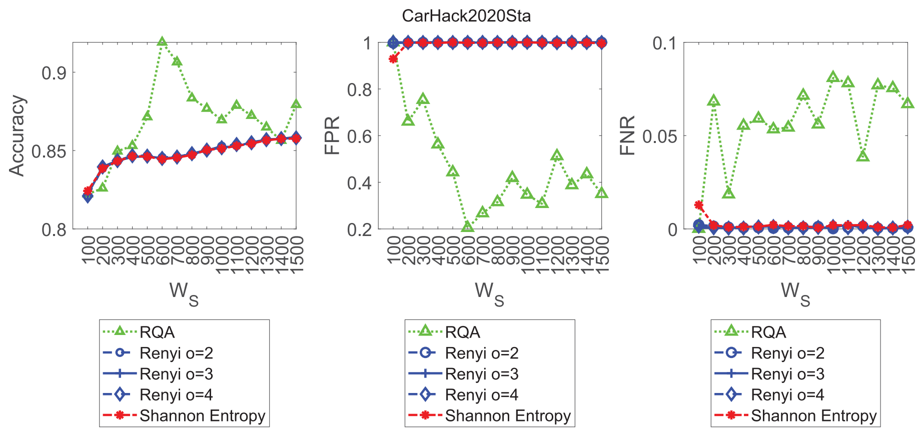

Fig. 7. CarHack2020Sta stationary data set. Accuracy, FPR and FNR for optimal values of M and T_{hr} for RQA features, Renyi entropy of different orders and Shannon entropy features for different values of W_S (size of the sliding window).

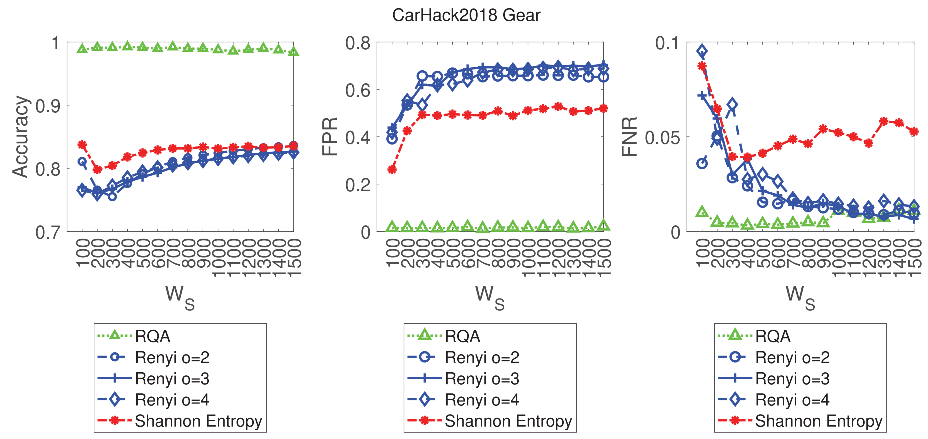

Fig. 8. CarHack2018Gear data set. Accuracy, FPR and FNR for optimal values of M and T_{hr} for RQA features, Renyi entropy of different orders and Shannon entropy features for different values of W_S (size of the sliding window).

Fig. 11 compares the performance across sliding windows of different sizes W_S while the second set of ROCs shown in Fig. 12 compares the performance of the RQA based approach against the approach based on entropy features. The legends in both figures show the related AUC value for each ROC. Only a subset of W_S values, which are more relevant to the analysis are shown ($W_S = 400 : 900$) are used to create the figures to avoid a cluttering effect in the figures. The results presented in Fig. 11 confirm the similar results presented in the previous figures: the parameter W_S has an impact on the detection of the attack. While the ROCs for the CarHack2018 data set demonstrates the excellent capability of the proposed approach to detect spoofing attacks, the ROCs for CarHack2020 data set show a slighter worst

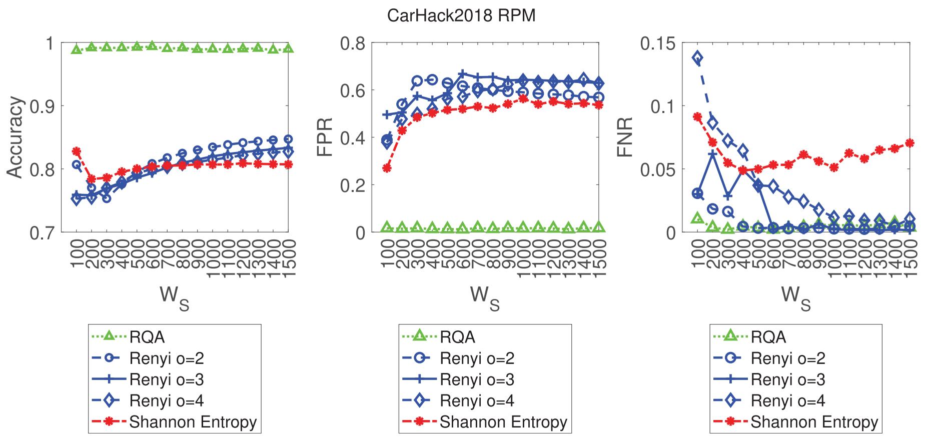

Fig. 9. CarHack2018RPM data set. Accuracy, FPR and FNR for optimal values of M and T_{hr} for RQA, Renyi entropy of different orders and Shannon entropy features for different values of W_S (size of the sliding window).

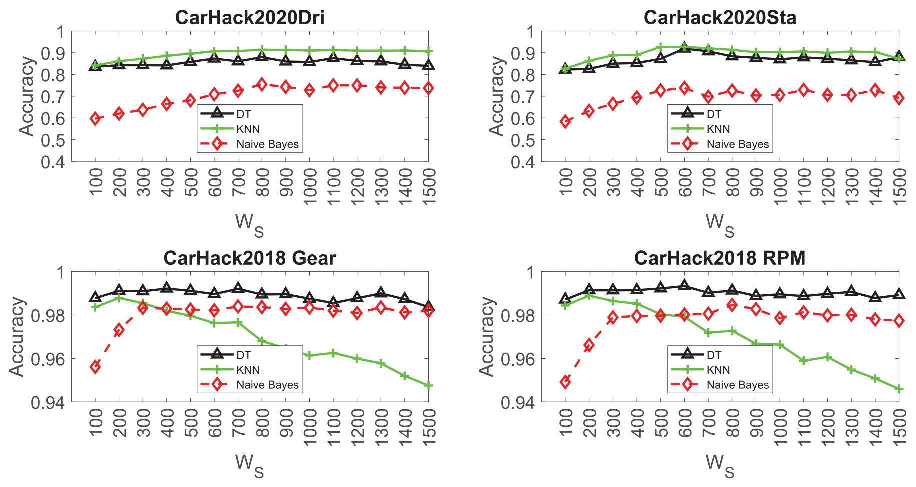

Fig. 10. Comparison of the application of different machine learning algorithms for the two different data sets.

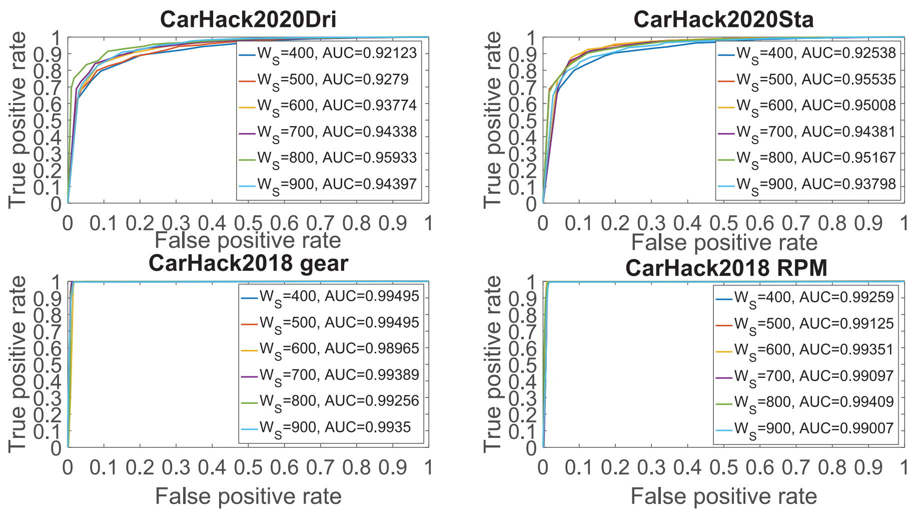

Fig. 11. ROCs and related values of AUC for different sizes of the sliding windows (different values of W_S). The optimal hyper-parameters values and ML algorithms used to generate the ROCs are identified in Table 4.

Table 4
Summary of the optimal detection results with the related values of the hyper-parameters and choice of ML.

Data set and features	Optimal parameters	Acc.	FPR	FNR
CarHack2020Dri				
RQA	KNN, $K = 8$, $M = 2$, $T_{hr} = 0.3$, $W_S = 800$	0.914	0.0337	0.0316
Renyi entropy $o = 2$ [21]	DT, Gini's diversity index, $N_B = 12$, $W_S = 1300$	0.86	0.9926	0.0017
Renyi entropy $o = 3$ [21]	DT, Gini's diversity index, $N_B = 12$, $W_S = 1200$	0.853	0.991	0.0013
Renyi entropy $o = 4$ [21]	DT, Gini's diversity index, $N_B = 10$, $W_S = 1300$	0.853	0.993	0.0009
Shannon entropy [10,13]	DT, Gini's diversity index, $N_B = 14$, $W_S = 1400$	0.853	0.999	0.0013
CarHack2020Sta				
RQA	KNN, $K = 7$, $M = 3$, $T_{hr} = 0.3$, $W_S = 600$	0.9271	0.02575	0.0479
Renyi entropy o = 2 [21]	DT, Gini's diversity index, $N_B = 10$, $W_S = 1400$	0.8582	0.999	0.0009
Renyi entropy o = 3 [21]	DT, Gini's diversity index, $N_B = 10$, $W_S = 1400$	0.8587	0.999	0.0007
Renyi entropy o = 4 [21]	DT, Gini's diversity index, $N_B = 10$, $W_S = 1400$	0.858	0.999	0.0006
Shannon entropy [10,13]	DT, Gini's diversity index, $N_B = 8$, $W_S = 1400$	0.876	0.999	0.0016
CarHack2018 Gear				
RQA	DT, $N_b = 10$, $M = 6$, $T_{hr} = 0.9$, $W_S = 400$	0.9992	0.0135	0.0032
Renyi entropy o = 2 [21]	DT, Gini's diversity index, $N_b = 12$ $W_S = 1400$	0.8341	0.6528	0.0109
Renyi entropy o = 3 [21]	DT, Gini's diversity index, $N_b = 12$ $W_S = 1400$	0.8252	0.6968	0.0087
Renyi entropy o = 4 [21]	DT, Gini's diversity index, $N_b = 12$ $W_S = 1400$	0.8234	0.6869	0.0142
Shannon entropy [10,13]	DT, Gini's diversity index, $N_b = 10$ $W_S = 1400$	0.8334	0.5099	0.0574
CarHack2018 RPM				
RQA	DT, $N_b = 12$, $M = 6$, $T_{hr} = 0.8$, $W_S = 600$	0.9933	0.0116	0.0020
Renyi entropy o = 2 [21]	DT, Gini's diversity index, $N_b = 10$ $W_S = 1400$	0.8436	0.5777	0.0026
Renyi entropy o = 3 [21]	DT, Gini's diversity index, $N_b = 12$ $W_S = 1400$	0.8289	0.6356	0.0015
Renyi entropy o = 4 [21]	DT, Gini's diversity index, $N_b = 12$ $W_S = 1300$	0.8239	0.6332	0.0092
Shannon entropy [10,13]	DT, Gini's diversity index, $N_b = 8$ $W_S = 1300$	0.8078	0.5405	0.0649

detection capability. The results presented in Fig. 12 do also confirm the superior performance of the RQA based approach in comparison to the entropy-based approaches by comparing the values of the AUCs.

Table 5
Comparison with the literature of this approach on the CarHack2018 data set.

Reference	Acc.	FPR	FNR	Sliding windows	CAN-bus field
CarHack2018 DoS					
This study	0.9836	0.0386	0.0063	Yes	Time
[14]	0.979	NA	NA	No	CAN-ID
[11]	1.00	NA	NA	Yes	CAN-ID
[28]	0.9728	NA	0.038	Yes	Payload
[8]	0.997	NA	0.0011	No	CAN-ID
CarHack2018 Gear					
This study	0.9992	0.0135	0.0032	Yes	Time
[14]	0.962	NA	NA	No	CAN-ID
[28]	1.00	0	0	Yes	Payload
[8]	0.995	NA	0.0011	No	CAN-ID
CarHack2018 RPM					
This study	0.9933	0.337	0.0316	Yes	Time
[14]	0.98	NA	NA	No	CAN-ID
[28]	1.00	0	0	Yes	Payload
[8]	0.997	NA	0.0006	No	CAN-ID
CarHack2018 Fuzzy					
This study	0.9849	0.0297	0.0073	Yes	Time
[14]	0.98	NA	NA	No	CAN-ID
[28]	0.9517	NA	0.05	Yes	Payload
[8]	0.9982	NA	0.0035	No	CAN-ID

The AUCs of the RQA based approach are higher than the ones obtained with the entropy measures.

Then, it is also evaluated what is the performance of each RQA feature across the two data sets. Fig. 13 provides the results of such an evaluation for the optimal parameters in Table 4 and for $W_S = 600$. It can be seen from Fig. 13 that for the CarHack2020 data set, the performance of each feature is rather similar with the feature id = 8 and id = 9 slightly better than others. The results for the CarHack2018 data set are less uniform but with the feature id = 8 still providing a higher accuracy than the other features. As described in Table 3, feature id = 8 is the maximal length of the vertical lines in the RP and the maximal white vertical line length in the RP. A potential explanation for the superior performance of the feature id = 8 is that the implementation of the spoofing attacks modifies more significantly the vertical lines of the RPs than the other characteristics of the RP.

5.3. Comparison with the literature results on the data sets CarHack2018 and CarHack2020

Finally, the results on the data sets used in this paper are compared with the results in the literature on the same data sets and they are presented in Tables 5 and 6 respectively for the data sets CarHack2018 and CarHack2020. For clarity, it is also reported the type of CAN-bus data used by the approach from the literature (e.g., CAN-ID) and if it uses a sliding window.

It is noted that the comparison of results in Table 5 is only indicative because most of the papers using these data sets do not use a sliding window approach (which may decrease detection accuracy with the benefit of dimensionality reduction) or they use different information (the CAN-ID data or the CAN-bus payload data). As stated in the abstract and introduction, this paper focuses on spoofing attacks but for completeness, we also report in this table the application of the proposed approach to the other attacks: in particular, the DOS and Fuzzy attacks in CarHack2018. The reason is also that these attacks are often investigated in the CarHack2018 data set used by other researchers.

Table 5 shows that the approach proposed in this paper is quite competitive in comparison to other approaches presented in the literature, taking into consideration that this approach does not require the processing of the CAN-ID data or CAN-bus payload data and it only relies on the time differences of the CAN-bus messages. It is interesting

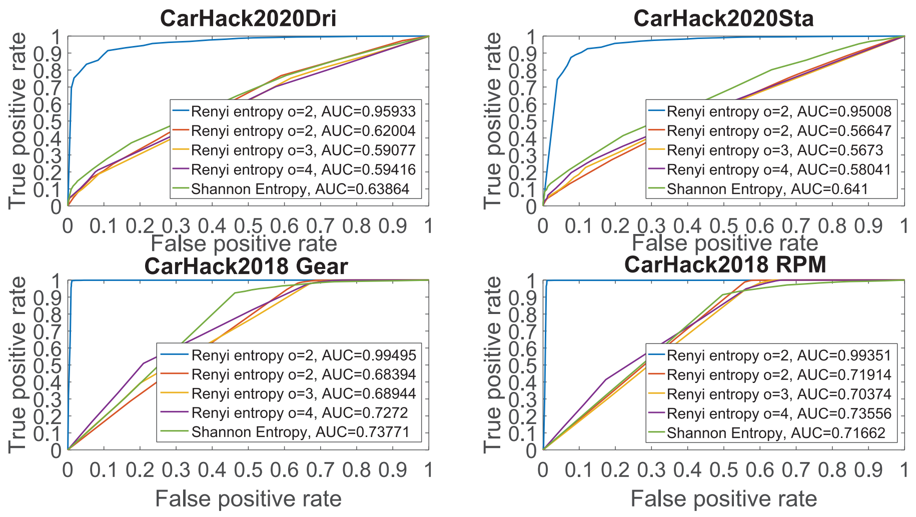

Fig. 12. ROCs and related values of AUC for the different approaches based on RQA and entropy-based features. The optimal hyper-parameters values and ML algorithms used to generate the ROCs are identified in Table 4.

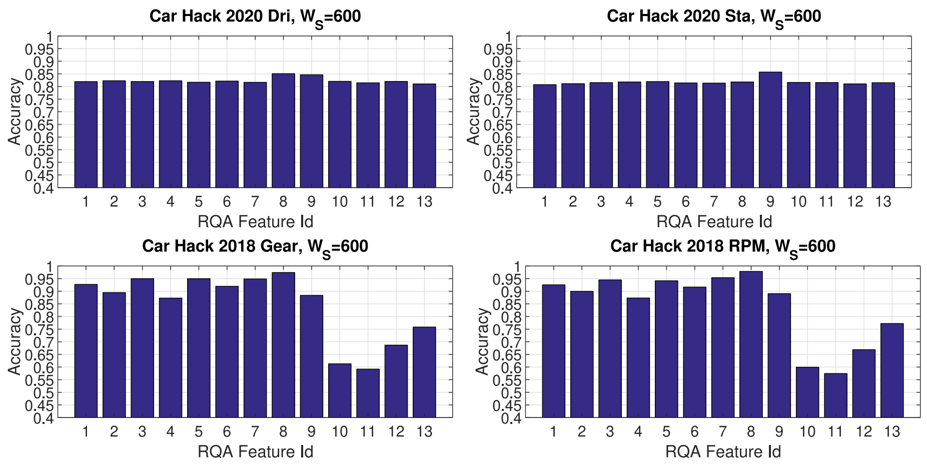

Fig. 13. Accuracy of each RQA feature across the two data set and the four time series. The optimal values of the hyper-parameters from Table 4 were used with $W_S = 600$.

to see that each approach has its own strength and weakness against different attacks. In particular, [28] is particularly effective for the RPM and GEAR spoofing attacks where it achieves perfect detection, but it is less performing in comparison to this approach for the DOS and Fuzzy attacks. This result seems to indicate that a combination of approaches could be an optimal strategy for the implementation of IDS in in-vehicle networks, where the strengths of the different approaches could be combined. The overall best performance (i.e., if all the attacks are used) is obtained by [8], where a sophisticated Deep Learning (DL) method was used. In a way, this confirms the superior performance of DL approaches used in many fields (i.e., primarily in image analysis), but it must be taken in consideration the considerable computing resources, which must be deployed in the vehicle for this purpose. In the automotive sector, this deployment may not be very practical and a Cloud DL could be used, but this may require the transmission of the CAN-bus data information from the vehicle to the cloud which can require significant connectivity resources.

Table 6 summarizes the results for the CarHack2020 data set where also the DoS (i.e., Flooding) and Fuzzy (i.e., Fuzzing attacks) were considered for the driving and stationary case (i.e., CarHack2020Sta and CarHack2020Dri).

The comparison with the only published study [30] (at the time of writing this paper) is somewhat difficult because the authors of [30] have used a different approach for using the data set by merging the two CarHack2020Dri and CarHack2020Sta data sets and merging all the attacks together. In addition, the authors of [30] use the CAN-ID and the CAN-bus payload rather than the time difference and they do not use a sliding window approach. Overall, it is a radically different approach from this study and the results are not directly comparable. Table 6 shows that the flooding and the fuzzing attacks can be detected with higher accuracy than the spoofing attacks using the approach proposed in this paper.

6. Conclusions

In this paper, we have investigated the application of RQA to the problem of spoofing attack detection in in-vehicle networks using a sliding window attack and the inter-arrival time between CAN-bus messages. This approach is efficient because it does not require the processing of the CAN-bus messages and the sliding window provides a dimensionality reduction proportional to the size of the window. The approach is compared with other popular sliding window approaches in the literature for the analysis of the traffic in in-vehicle networks based on the application of entropy measures. The experimental results on two recent (2018 and 2020) public data sets show that the RQA based approach is able to significantly outperform the entropy-based approaches and it is also more stable (i.e., better balance between FPR and FNR) in the CarHack2020 data set. In addition, it is competitive

Table 6
Comparison with the literature of this approach on the CarHack2020 data set.

Reference	Acc.	FPR	FNR	Sliding windows	CAN-bus field
This study for CarHack2020 Driving DoS (Flooding)	0.9988	0.0078	0	Yes	Time
This study for CarHack2020 Driving Spoofing	0.914	0.0337	0.0316	Yes	Time
This study for CarHack2020 Driving Fuzzy (Fuzzing)	0.9952	0.0278	0.0020	Yes	Time
This study for CarHack2020 Stationary DOS (Flooding)	0.9989	0.0066	0	Yes	Time
This study for CarHack2020 Stationary Spoofing	0.9271	0.0386	0.0063	Yes	Time
This study for CarHack2020 Stationary Fuzzy (Fuzzing)	0.9919	0.0457	0.0019	Yes	Time
Approach from [30] using SVM with combined attacks and data sets	0.9532	NA	0.0442	No	CAN-ID and Payload
Approach from [30] using Deep Learning with combined attacks and data sets	0.9810	NA	0.0196	No	CAN-ID and Payload

against other approaches (based or not based on the sliding windows) on the same public data sets for spoofing attacks but also for other attacks commonly investigated by the research community (e.g., DoS and Fuzzy attacks). The proposed approach based on the time interval information and the sliding window with RQA manages to obtain a very high detection accuracy (more than 99%) in the CarHack2018 data set for both spoofing attacks while the same approach manages to obtain an accuracy higher than 91% in the CarHack2020 data set. A potential reason for this discrepancy has been linked to the distribution of the time intervals in the time series and a simple estimate metric for the applicability of the proposed approach has been identified.

To the best knowledge of the author, this is the first time that RQA is applied to the problem of intrusion detection for spoofing attacks in in-vehicle networks. These promising results suggest that RQA related features should be taken into consideration for the implementation of efficient in-vehicle intrusion detection systems as they provide significantly more discriminative power than entropy-based measures which are commonly used in the literature.

Future developments can be implemented in different directions. One direction is to investigate the extension of the basic RQA used in this study to more sophisticated implementations like the fuzzy RQA. Another development could try to solve the need to find the optimal values of the hyper-parameters by applying meta-heuristics algorithms or adaptive methods (e.g., adaptive sliding window).

CRediT authorship contribution statement

Gianmarco Baldini: Conceptualization, Software, Methodology, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Gianmarco Baldini reports financial support was provided by European Commission Joint Research Centre Ispra Sector. Gianmarco baldini reports a relationship with European Commission Joint Research Centre Ispra Sector that includes: employment.

Acknowledgment

The author acknowledges the project smart adaptive remote DIagnostic Antitampering Systems (DIAS) funded by the European Union H2020 Programme under Grant Agreement No. 814951-2.

References

[1] S.-F. Lokman, A.T. Othman, M.-H. Abu-Bakar, Intrusion detection system for automotive controller area network (CAN) bus system: a review, EURASIP J. Wireless Commun. Networking 2019 (1) (2019) 1–17.
[2] J. Petit, S.E. Shladover, Potential cyberattacks on automated vehicles, IEEE Trans. Intell. Transp. Syst. 16 (2) (2014) 546–556.
[3] C. Miller, C. Valasek, Remote exploitation of an unaltered passenger vehicle, Black Hat USA 2015 (S 91) (2015).
[4] T.F. Lunt, A survey of intrusion detection techniques, Comput. Secur. 12 (4) (1993) 405–418.
[5] W. Wu, R. Li, G. Xie, J. An, Y. Bai, J. Zhou, K. Li, A survey of intrusion detection for in-vehicle networks, IEEE Trans. Intell. Transp. Syst. 21 (3) (2019) 919–933.
[6] C. Young, J. Zambreno, H. Olufowobi, G. Bloom, Survey of automotive controller area network intrusion detection systems, IEEE Des. Test 36 (6) (2019) 48–55.
[7] M.-J. Kang, J.-W. Kang, Intrusion detection system using deep neural network for in-vehicle network security, PLoS One 11 (6) (2016) e0155781.
[8] H.M. Song, J. Woo, H.K. Kim, In-vehicle network intrusion detection using deep convolutional neural network, Veh. Commun. 21 (2020) 100198.
[9] G. Karopoulos, G. Kambourakis, E. Chatzoglou, J.L. Hernández-Ramos, V. Kouliaridis, Demystifying in-vehicle intrusion detection systems: A survey of surveys and a meta-taxonomy, Electronics 11 (7) (2022) 1072.
[10] W. Wu, Y. Huang, R. Kurachi, G. Zeng, G. Xie, R. Li, K. Li, Sliding window optimized information entropy analysis method for intrusion detection on in-vehicle networks, IEEE Access 6 (2018) 45233–45245.
[11] S. Ohira, A.K. Desta, I. Arai, H. Inoue, K. Fujikawa, Normal and malicious sliding windows similarity analysis method for fast and accurate IDS against DoS attacks on in-vehicle networks, IEEE Access 8 (2020) 42422–42435.
[12] G. Baldini, On the application of entropy measures with sliding window for intrusion detection in automotive in-vehicle networks, Entropy 22 (9) (2020) 1044.
[13] M. Marchetti, D. Stabili, A. Guido, M. Colajanni, Evaluation of anomaly detection for in-vehicle networks through information-theoretic algorithms, in: 2016 IEEE 2nd International Forum on Research and Technologies for Society and Industry Leveraging a Better Tomorrow (RTSI), IEEE, 2016, pp. 1–6.
[14] E. Seo, H.M. Song, H.K. Kim, GIDS: GAN based intrusion detection system for in-vehicle network, in: 2018 16th Annual Conference on Privacy, Security and Trust (PST), IEEE, 2018, pp. 1–6.
[15] M. Gmiden, M.H. Gmiden, H. Trabelsi, An intrusion detection method for securing in-vehicle CAN bus, in: 2016 17th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA), IEEE, 2016, pp. 176–180.
[16] N. Marwan, M.C. Romano, M. Thiel, J. Kurths, Recurrence plots for the analysis of complex systems, Phys. Rep. 438 (5–6) (2007) 237–329.
[17] F. Palmieri, U. Fiore, A nonlinear, recurrence-based approach to traffic classification, Comput. Netw. 53 (6) (2009) 761–773.
[18] H.M. Song, H.R. Kim, H.K. Kim, Intrusion detection system based on the analysis of time intervals of CAN messages for in-vehicle network, in: 2016 International Conference on Information Networking (ICOIN), IEEE, 2016, pp. 63–68.
[19] M.R. Moore, R.A. Bridges, F.L. Combs, M.S. Starr, S.J. Prowell, 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, 2017, pp. 1–4.
[20] H. Lee, S.H. Jeong, H.K. Kim, 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), IEEE, 2017, pp. 57–5709.

[21] K.-S.Y.S.-H. Kim, D.-W. Lim, Y.-S. Kim, A multiple Rényi entropy based intrusion detection system for connected vehicles, Entropy 22 (2) (2020) 186.
[22] X. Duan, H. Yan, D. Tian, J. Zhou, J. Su, W. Hao, In-vehicle CAN bus tampering attacks detection for connected and autonomous vehicles using an improved isolation forest method, IEEE Trans. Intell. Transp. Syst. (2021).
[23] F. Palmieri, U. Fiore, Network anomaly detection through nonlinear analysis, Comput. Secur. 29 (7) (2010) 737–755.
[24] M.A. Righi, R.C. Nunes, Combining recurrence quantification analysis and adaptive clustering to detect ddos attacks, Cyber Defense Rev. (2019) 15–30.
[25] S. Mukherjee, R. Ray, R. Samanta, M.H. Khondekar, G. Sanyal, Nonlinearity and chaos in wireless network traffic, Chaos Solitons Fractals 96 (2017) 23–29.
[26] L. Yang, A. Moubayed, I. Hamieh, A. Shami, Tree-based intelligent intrusion detection system in internet of vehicles, in: 2019 IEEE Global Communications Conference (GLOBECOM), IEEE, 2019, pp. 1–6.
[27] A. Alshammari, M.A. Zohdy, D. Debnath, G. Corser, Classification approach for intrusion detection in vehicle systems, Wirel. Eng. Technol. 9 (4) (2018) 79–94.
[28] A. Derhab, M. Belaoued, I. Mohiuddin, F. Kurniawan, M.K. Khan, Histogram-based intrusion detection and filtering framework for secure and safe in-vehicle networks, IEEE Trans. Intell. Transp. Syst. (2021).
[29] H. Kang, B.I. Kwak, Y.H. Lee, H. Lee, H. Lee, H.K. Kim, Car hacking and defense competition on in-vehicle network, in: Workshop on Automotive and Autonomous Vehicle Security (AutoSec), Vol. 2021, 2021, p. 25.
[30] S. Mehedi, A. Anwar, Z. Rahman, K. Ahmed, et al., Deep transfer learning based intrusion detection system for electric vehicular networks, Sensors 21 (14) (2021) 4736.
[31] H. Kang, B.I. Kwak, Y.H. Lee, H. Lee, H. Lee, H.K. Kim, Car hacking: Attack and defense challenge 2020 dataset, 2021, http://dx.doi.org/10.21227/qvr7-n418.
[32] N. Marwan, J.F. Donges, Y. Zou, R.V. Donner, J. Kurths, Complex network approach for recurrence analysis of time series, Phys. Lett. A 373 (46) (2009) 4246–4254.
[33] F. Takens, Detecting strange attractors in fluid turbulence, in: D. Rand, L.-S. Young (Eds.), Dynamical Systems and Turbulence, Springer-Verlag, Berlin, 1981.