Car Hacking Identification through Fuzzy Logic Algorithms

Fabio Martinelli*, Francesco Mercaldo*, Vittoria Nardone‡, Antonella Santone‡
*Institute for Informatics and Telematics, National Research Council of Italy (CNR), Pisa, Italy
{fabio.martinelli, francesco.mercaldo}@iit.cnr.it
‡Department of Engineering, University of Sannio, Benevento, Italy
{vnardone, santone}@unisannio.it

***Abstract*—Modern vehicles have lots of connectivity, this is the reason why protect in-vehicle network from cyber-attacks becomes an important issue. The Controller Area Network is a de facto standard for the in-vehicle network. However, lack of security features of CAN protocol makes vehicles vulnerable to attacks. The message injection attack is a representative attack type which injects fabricated messages to deceive original Electronic Control Units or to cause malfunctions. In this paper we propose a method able to detect four different type of attacks targeting the CAN protocol adopting fuzzy algorithms. We obtain encouraging results with a precision ranging from 0.85 to 1 using the fuzzy NN algorithm in the identification of attacks targeting CAN protocol.**

I. Introduction

Cars are no longer only mechanical vehicle. Actually, cars contain a plethora of different components networked together that as a whole are responsible for monitoring and controlling the state of the vehicle. Each components is able to communicate with neighboring components generating a large amounts of real-time data [1]. Modern automobiles contain upwards of 50 electronic control units (i.e., the so-called ECUs) networked together [2]. The overall safety of the vehicle relies on near real time communication between the various ECUs. While communicating with each other, ECUs are responsible for predicting crashes, detecting skids, performing anti-lock braking and other safety-critical tasks.

Typically ECUS are networked together on one or more buses based on the Controller Area Network (CAN) standard. CAN[1] is a high-integrity serial data communication technology developed in the early 1980s by Robert Bosch GmbH to provide efficient communication between automotive applications. The ecosystem of the ECUs communicate with one another by sending CAN packets. These kinds of packets are broadcast to all components on the bus and each component decides whether it is intended for them, although segmented CAN networks do exist. In CAN packets there is not built-in source identified or authentication. Because of these two facts, it is easy for components to both sniff the CAN network as well as masquerade as other ECUs and send CAN packets. It also makes reverse engineering traffic more difficult because it is impossible, a priori, to know which ECU is sending or receiving a particular packet.

Drivers and passengers are strictly at the mercy of the code running in their automobiles and, unlike when their web browser crashes or is compromised, the threat to their physical well-being is real [3], [4], [2].

Starting from these considerations in this paper we propose a method, based on fuzzy techniques, able to discriminate attacks targeting the CAN protocol. As explained in [5], [2], these kind of attacks towards the in-vehicle network can be performed by misusing well chosen mechanisms in the protocols [6].

We analyze a real-world dataset composed by CAN packets in order to discover when a message injected by an attacker happens. Considering as feature vector the data belonging to CAN packets, we demonstrate that using fuzzy logic techniques our method is able to discriminate between CAN packets generated by normal behaviour and by attackers.

The paper poses the following research question:

- RQ: is it possible to identify car attacks using CAN packets as a feature vector?

The rest of the paper is organized as follows: the next section describes the background of the study; the third section provides an overview of related work; the following section illustrates the proposed features and the detection technique; the fifth section presents the results of the evaluation, and, finally, conclusion and future works are given in the last section.

II. Background

In this section we explain a brief background about the CAN protocol and the fuzzy classification algorithms we employed.

A. CAN protocol

The need to allow the communication between the plethora of electronic devices presents inside modern automobiles (as the centralized locking system, the air conditioning control, the traction control and the antilock braking system for instances) and their growing complexity would lead to an unsustainable increase in dedicated connections and to a duplication of the necessary sensors to more devices, with the consequent increase of the costs related to production and above the growing physical footprint.

[1] www.can.bosch.com

978-1-5090-6034-4/17/$31.00 ©2017 IEEE

These are the reason why the CAN protocol was introduced, in order to allow to controllers, sensors and actuators to communicate with one another at a speed of up to 1 Mbit/sec with several benefits: (i)lower design and costs of implementation, (ii)it works in hostile environments and (iii)ease of change and configuration.

The CAN protocol is part of the ISO/OSI stack in levels 1 and 2 physical data link, leaving total freedom for the layer 7 application: as a matter of fact the core of the protocol is in the data link level.

CAN packets contain an identifier and data. The identifier may be either 11 or 29 bits long, although for cars usually only 11 bit identifier are seen. After the identifier, there are from 0 to 8 bytes of data. The data may contain checksums or other information within the 8 bytes of application layer, but this is not part related to the CAN specification. For instance, in FORD cars, almost all CAN packets contain 8 bytes of data, while in Toyota cars the number of bytes varies greatly and often the last byte contains a checksum of the data.

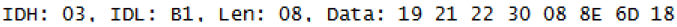
In CAN automotive networks, there are two main types of CAN packets i.e., normal and diagnostic. Normal packets are sent from ECUs and can be seen on the network at any given time. They may be broadcast messages sent with information for other ECUs to consume or may be interpreted as commands for other ECUs to act on. There are many of these packets being sent at any given time, typically every few milliseconds.

The other type of CAN packets seen in automotive systems are diagnostic packets. These packets are sent by diagnostic tools used by mechanics to communicate with and interrogate an ECU. These packets will typically not be seen during normal operation of the vehicle.

The CAN packets are contained in a message: each message is composed by following values:

- Timestamp : recorded time (s);
- CAN ID : identifier of CAN message in HEX (i.e., 03B1);
- DLC : number of data bytes, from 0 to 8;
- DATA[0 7] : data value (byte);

An example of such a normal message with identifier 03B1 from the Ford Escape MS bus is shown in Figure 1:



```
IDH: 03, IDL: B1, Len: 08, Data: 19 21 22 30 08 8E 6D 18
```

Fig. 1: An example of CAN packet

As shown in Figure 1, each CAN packets contains eight bytes codified in hexadecimal (i.e., the data value).

B. Fuzzy logic techniques

We adopt fuzzy classification algorithms in order to discriminate between injected real-world CAN packets and normal ones.

We describe the classification algorithms we employed in order to evaluate the effectiveness of feature vector:

- *FuzzyRoughNN*: the rationale behind the Fuzzy-rough K-nearest neighbours [7] algorithm is that the lower and the upper approximation of a decision class, calculated by means of the nearest neighbours of a test object, provide good clues to predict the membership of the test object to that class;
- *NN*: the Fuzzy-rough K-nearest neighbors (II) [7] algorithms is based on the same principle of the previous one;
- *DiscernibilityClassifier*: the Discernibility NN classifier [7] algorithm in order to distinguish the instances computes the Index of Discernibility (ID), a metric developed for assessing how easily distinguishable the classes are from the dataset [8]. This algorithm assumes a fixed radius around each element of the dataset, which corresponds to the average distance between this and the rest of the elements of that class. The radius depends on the class structure, so elements belonging to different classes may have different radii. The ID of the full dataset is computed as the number of elements having discernibility higher than 0.5 divided by the total number of elements, considering the distance of each neighbour.
- *FURIA*: the Fuzzy Unordered Rule Induction Algorithm [9] algorithm extends the RIPPER algorithm [10], a state-of-the-art rule learner. The specificity of this algorithm is represented by the fact that it is able to learns fuzzy rules instead of conventional rules and unordered rule sets rather than rule lists. Moreover, to deal with uncovered examples, the algorithm use an efficient rule stretching method. Each individual rule is learned in two steps. The training data, which has not yet been covered by any rule, is thus split into a growing and a pruning set. In the first step, the rule will be specialized by adding antecedents which were learned using the growing set. Afterward, the rule will be generalized by removing antecedents using the pruning set.

III. Related Work

In this section we review current literature about the issue of the identification of car-related attacks.

Checjoway et colleagues [11] discover that remote exploitation is feasible through broad range of attack vectors (i.e., mechanics tools, CD players, Bluetooth and cellular radio), and in addition, that wireless communications channels enable long distance vehicle control, location tracking, in-cabin audio exfiltration and theft.

Researchers in [12] show that a long-range wireless attack is possible using a real-world car and dangerous smartphone application in a connected car environment. In addition authors propose a security protocol working for CAN as a countermeasure designed in accordance with current CAN specifications, evaluating the proposed security protocol using CANoe software and a DSP-F28335 microcontroller.

Authors in [13] propose an anomaly detector considering a Long ShortTerm Memory neural network in order to detect CAN bus attacks. Their proposal works by learning to predict the next data word originating from each bus sender. Their detector is able to recognize the synthesized anomalies with low false positive rates.

Yan et al. [14] discuss a method to gather data from three resources: radar, traffic and neighboring with the aim to safeguard cars from attacks. Basically, the system computed the similarity between these data.

Liu et al. [15] propose a system to secure the communications system for the vehicles depending on roadside units. The proposed infrastructure contains a Certification Authority (CA) based cluster distributed in different regions. The aim is to show that IDS using a CA database provided more protection against malicious vehicles with legal certificates.

Lyamin et al. [16] design an algorithm to detect denial of service attacks in real-time based on performance metrics such as the percentage of false alarms for any jamming channel and the average beacon time for vehicular networks.

Alheeti et colleagues [17] propose a system to secure external communication for self-driving and semi-self-driving vehicles intelligent IDS-based. The security system considers the data collected from the network. They created a VANET environment on the NS2 simulator, generating two types of behavior: normal and malicious (a vehicle is marked as malicious when it drops the packets).

In order to identify user behaviour by analyzing driving style, Kwak et al. [18] analyse CAN packet demonstrating that the collected data are useful to discriminate different owner drivers and impostors.

Considering that several fuzzy logic approaches have been proposed to model human behaviour [19], [20], in this paper we apply fuzzy classification algorithms in order to discriminate between normal CAN messages (i.e., generated by the human driver) and injected ones (i.e. messages generated by attacker).

At the best of authors knowledge, this is the first method that exploits fuzzy techniques to identify attacks targeting CAN protocol using real-world data extracted by cars.

IV. The Method

In this section we describe our method to identify car attacks targeting CAN protocol.

In order to discriminate message injected by an attacker by normal ones, we consider these bytes as the feature vector composed in the following way:

- 1st byte: F1 feature (value 19 in Figure 1);
- 2nd byte: F2 feature (value 21 in Figure 1);
- 3rd byte: F3 feature (value 22 in Figure 1);
- 4th byte: F4 feature (value 30 in Figure 1);
- 5th byte: F5 feature (value 08 in Figure 1);
- 6th byte: F6 feature (value 8E in Figure 1);
- 7th byte: F7 feature (value 6D in Figure 1);
- 8th byte: F8 feature (value 18 in Figure 1).

We extracted the feature vector from four dataset freely available for research purposes[2] including normal real-world CAN messages and four different kinds of injected messages caused by following attacks: dos attack (*dos*), fuzzy attack (*fuzzy*), spoofing the drive gear (*gear*) and spoofing the RPM gauge (*rpm*). Dataset were constructed by logging CAN traffic through the OBD-II (On-Board Diagnostics) port from a real vehicle while message injection attacks were performing. Dataset contain each 300 intrusions of message injection. Each intrusion performed for 3 to 5 seconds, and each dataset has total 30 to 40 minutes of CAN traffic.

We describe in details the four type of attacks:

- *dos* : it represent the denial of service attack, performed by injecting messages of 0000 CAN ID every 0.3 milliseconds. The 0000 CAN ID is the most dominant;
- *fuzzy*: injecting messages of totally random CAN ID and DATA values every 0.5 milliseconds;
- *gear/rpm* : injecting messages of certain CAN ID related to gear/rpm information every 1 millisecond. The *rpm* (i.e., revolutions per minute) measures the number of revolutions completed in one minute around a fixed axis. Running an engine at a high RPM may cause damage to the engine and reduce its expected lifespan.

Table I shows the overall number of messages for the four dataset with the detail related to the injected and the normal messages.

Attack	# messages	# normal messages	# injected messages
dos	3,665,771	3,078,250	587,521
fuzzy	3,838,860	2,759,492	1,079,368
gear	4,443,142	2,766,522	1,676,620
rpm	4,621,702	2,290,185	2,331,517

TABLE I: Number of (total, normal and injected) messages in the four dataset.

We designed an experiment in order to evaluate the effectiveness of the feature vector we propose, expressed through the research question RQ stated in the introduction.

More specifically, our aim is to verify if the eight features are able to discriminate the four type of attacks by the normal CAN messages.

We learn several state of the art fuzzy classifiers with the eight features.

The evaluation consists of three different stages: (i) we provide a comparison of descriptive statistics of the normal and injected messages populations; (ii) hypotheses testing, to verify whether the features vector exhibit different distributions for attacks and normal messages populations; and (iii) classification analysis in order to assess if the eight features are able to discriminate between attacks and normal messages.

With regards to the descriptive statistics, we report the box plot of the distribution of attacks and normal messages in order to demonstrate that the distribution are different and the features we consider are good candidate for the discrimination between attack and normal messages.

Relating to the hypotheses testing, the null hypothesis to be tested is:

H_0 : ‘injected and normal messages have similar values of the considered features’.

The null hypothesis was tested with Mann-Whitney (with the p-level fixed to 0.05) and with Kolmogorov-Smirnov Test

[2] https://sites.google.com/a/hksecurity.net/ocslab/Datasets/car-hacking-dataset

(with the p-level fixed to 0.05). We run two tests in order to enforce the conclusion validity.

The goal of the tests is to determine the level of significance, i.e., the probability that erroneous conclusions be drawn: in this case, we consider the significance level equal to .05 i.e., we accept to make mistakes 5 times out of 100.

The classification analysis goal is to verify if the considered features are able to correctly classify between attacks and normal messages. As explained in details in the background section, four fuzzy algorithms of classification were used: FuzzyRoughNN, NN, DiscernibilityClassifier and FURIA. These algorithms were applied to the eight features (i.e., to the feature vector).

The classification analysis is performed using the Weka[3] tool, a suite of machine learning software, employed in data mining for scientific research with the fuzzy algorithms package[4].

V. The Evaluation

The results of the evaluation will be discussed reflecting the data analysis' division in three phases explained in previous section: descriptive statistics, hypotheses testing and classification.

A. Descriptive statistics

Figures 2, 3, 4, 5 show the box plots related to F1 and F2 features with the four types of attacks. We indicate in the box plots each features as follows: 'attacks_feature_distribution', where *attacks* indicates the type of attacks (i.e., *dos*, *fuzzy*, *rpm* and *gear*), *feature* the feature under analysis and with *distribution* we indicate whether the box plot is referred to the injected messages distribution (IM) or to the normal one (NM). For instance, with dos_F1_IM we indicate the distribution related to the dos attack injected messages of F1 feature. For spaces reasons, we report only the box plot related to F1 and F2 features, but similar trends are also observed for F3, F4, F5, F6, F7 and F8 features.

Relating to Figure 2, the differences between the box plots of the injected and normal messages for the F1 and F2 features suggest that the two populations could belong to different distributions. As matter of fact, the distribution belonging to normal messages (i.e., dos_F1_NM and dos_F2_NM) appears to be wider if compared with the injected ones (i.e., dos_F1_IM and dos_F2_IM).

The distributions of F1 and F2 features, related to the fuzzy attack, shown in Figure 3 seem to show a difference in the two samples. This evidence strengthens the starting assumption of the paper, that will be confirmed by the results of the hypotheses testing explained in the next section.

Figure 4 shows the distribution related to the RPM attack. Also in this case the injected and normal messages box plot do not overlap. This is symptomatic that the features involved are good candidate to correctly discriminate between injected and normal messages.

Finally, the differences between the normal and injected message box plots for the F1 and the F2 features related to the gear attack are much more pronounced as in the previous cases.

B. Hypothesis testing

The hypothesis testing aims at evaluating if the features present different distributions for the populations of injected and normal messages with statistical evidence.

We assume valid the results when the null hypothesis is rejected by both the tests performed.

Table II shows the results of hypothesis testing: the null hypothesis H_0 can be rejected for all the eight features. This means that there is statistical evidence that the feature vector is a potential candidate for correctly classifying between injected and normal messages.

Variable	Mann-Whitney	Kolmogorov-Smirnov
F1	0,00000	$p < .001$
F2	0,00000	$p < .001$
F3	0,00024	$p < .001$
F4	0,00000	$p < .001$
F5	0,00000	$p < .001$
F6	0,00000	$p < .001$
F7	0,00000	$p < .001$
F8	0,00000	$p < .001$

TABLE II: Results of the null hypothesis H_0 test.

This result will provide an evaluation of the risk to generalize the fact that the selected features produce values which belong to two different distributions (i.e., the one related of the four types of injected messages and the normal messages): the null hypothesis H_0 test confirms that the features can distinguish those observations. With the classification analysis we will be able to establish the accuracy of the features in associating any message to the injected or to the normal distribution.

C. Classification analysis

We used five metrics in order to evaluate the results of the classification: FP rate, Precision, Recall, F-Measure and ROC Area.

The false positive rate is calculated as the ratio between the number of messages wrongly categorized as belonging to the normal distribution (i.e., the false positives) and the total number of injected messages (i.e., the true negatives):

$$FP\ rate = \frac{fp}{fp+tn}$$

where *fp* indicates the number of false positives and *tn* the number of true negatives.

The precision has been computed as the proportion of the examples that truly belong to class X among all those which were assigned to the class. It is the ratio of the number of relevant records retrieved to the total number of irrelevant and relevant records retrieved:

[3] http://www.cs.waikato.ac.nz/ml/weka/

[4] http://users.aber.ac.uk/rkj/book/wekafull.jar

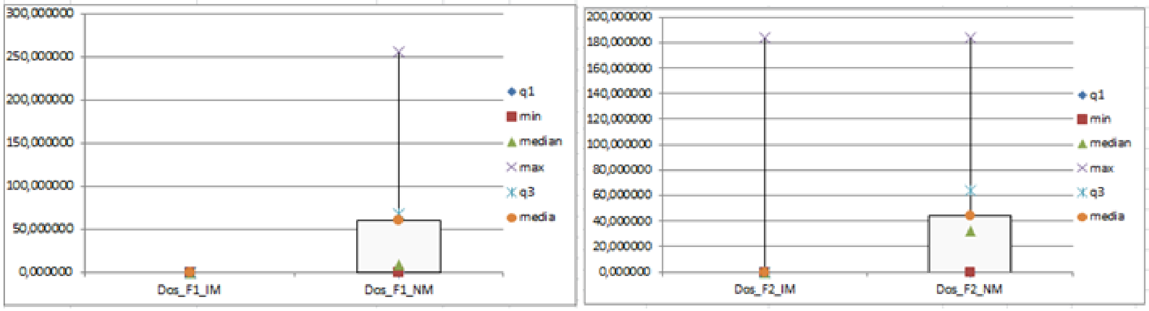


Fig. 2: Box plots related to the F1 and F2 features when dos type attack happens.

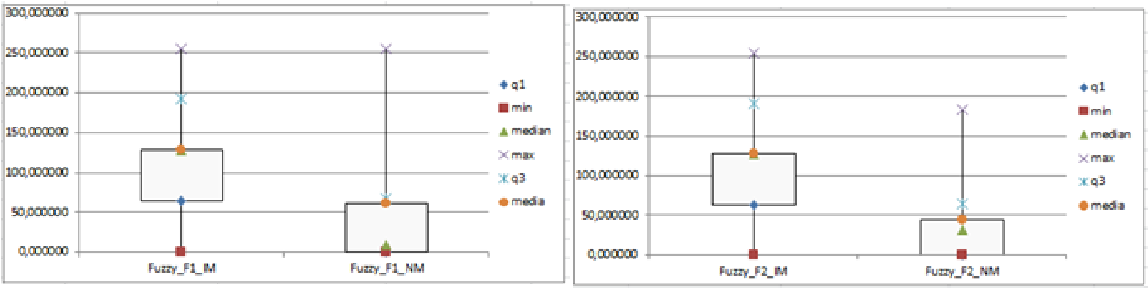


Fig. 3: Box plots related to the F1 and F2 features when fuzzy type attack happens.

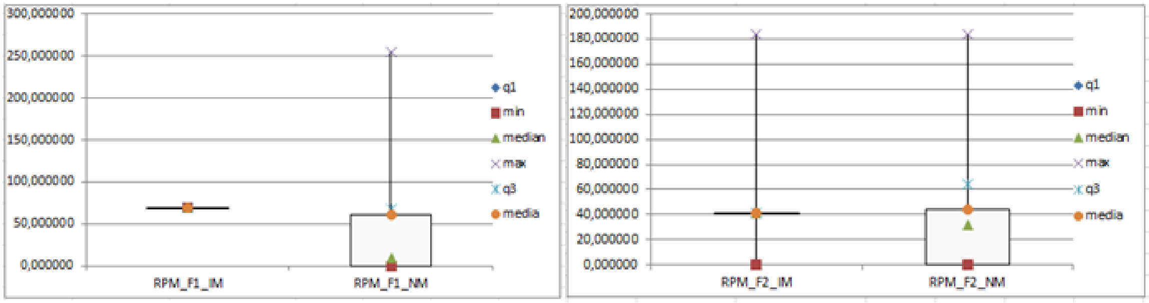


Fig. 4: Box plots related to the F1 and F2 features when rpm type attack happens.

$$\textit{Precision} = \frac{tp}{tp+fp}$$

where *tp* indicates the number of true positives and *fp* indicates the number of false positives.

The recall has been computed as the proportion of examples that were assigned to class X, among all the examples that truly belong to the class, i.e., how much part of the class was captured. It is the ratio of the number of relevant records retrieved to the total number of relevant records:

$$\textit{Recall} = \frac{tp}{tp+fn}$$

where *tp* indicates the number of true positives and *fn* indicates the number of false negatives.

The F-Measure is a measure of a test's accuracy. This score can be interpreted as a weighted average of the precision and recall:

$$\textit{F-Measure} = 2 * \frac{Precision*Recall}{Precision+Recall}$$

The Roc Area is defined as the probability that a positive instance randomly chosen is classified above a negative randomly chosen.

The classification analysis consisted of building classifiers in order to evaluate the feature vector accuracy to distinguish between injected and normal messages.

For training the classifier, we defined *T* as a set of labeled messages *(M, l)*, where each *M* is associated to a label $l \in \{IM,\ NM\}$. For each *M* we built a feature vector F $\in R_y$, where y is the number of the features used in training phase ($y = 8$).

For the learning phase, we use a k-fold cross-validation: the dataset is randomly partitioned into k subsets. A single subset is retained as the validation dataset for testing the model, while the remaining $k-1$ subsets of the original dataset are used as training data. We repeated the process for $k = 10$ times; each one of the k subsets has been used once as the validation dataset. To obtain a single estimate, we computed the average

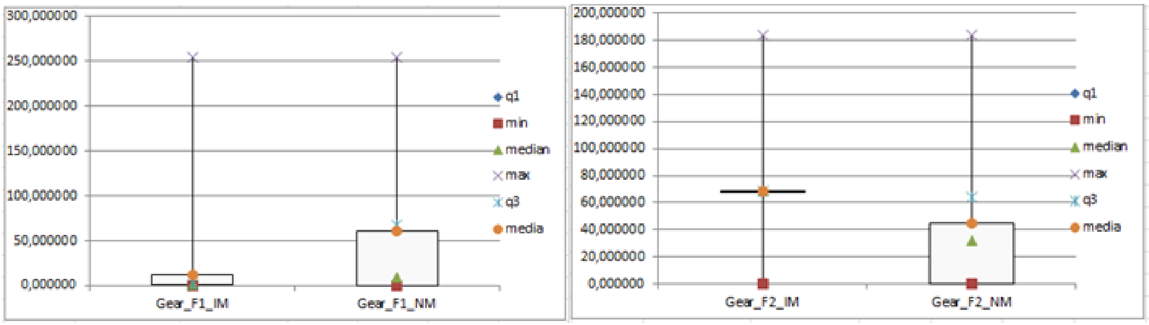


Fig. 5: Box plots related to the F1 and F2 features when gear type attack happens.

of the k results from the folds.

We evaluated the effectiveness of the classification method with the following procedure:

1) build a training set $T \subset D$;
2) build a testing set $T' = D \div$T;
3) run the training phase on T;
4) apply the learned classifier to each element of *T'*.

Each classification was performed using 20% of the dataset as training dataset and 80% as testing dataset employing the full feature set.

The results that we obtained with this procedure are shown in table III.

As Table III explains, the fuzzy algorithm able to obtain the best results is the NN one. As matter of fact, classifying the feature set with this algorithm we obtain following results:

- a precision equal to 0.963 and a recall equal to 1 in the identification of *dos* attack;
- a precision equal to 0.85 and a recall equal to 1 in the identification of *fuzzy* attack;
- a precision equal to 1 and a recall equal to 1 in the identification of *gear* and *rpm* attacks.

Relating to FP Rate, we obtain using the NN algorithm, a value ranging between 0.038 (with *dos* attack) to 0 (with *gear* and *rpm* attacks).

Moreover, even with slightly lower performance, the FuzzyRoughNN algorithm obtains good performance in discriminating injected messages from normal ones, especially in case of *gear* and *rpm* attacks where the precision and the recall are equal to 1, while for the *dos* and the *fuzzy* attacks the precision performance is ranging between 0.974 and 1 and the recall is ranging between 0.7 and 0.747.

As in the case of previous algorithms, also the DiscernibilityClassifier and the FURIA algorithms are able to classify all the instances related to the injected messages generated by *rpm* and *gear* attacks (with a precision and a recall equal to 1). When analyzing *dos* injected messages the precision obtained is equal to 0.974 and 0.969, while the recall is equal to 1 with both the algorithm.

The DiscernibilityClassifier algorithm performance dramatically decreases in discriminating *fuzzy* injected messages, as matter of fact the precision and the recall are both equal to 0.

RQ response: from the evaluation results it appears that the fuzzy NN algorithm exhibits high values for the analyzed metrics to discriminate between all types of injected attack messages and the normal ones. In particular following results are obtained for the attacks identification:

- FP rate ranging from 0 to 0.038;
- Precision ranging from 0.963 to 1;
- Recall ranging from 0.823 to 1;
- F-Measure ranging from 0.981 to 1;
- Roc Area ranging from 0.986 to 1.

VI. Conclusion and Future Work

Modern cars are evolved in connected vehicles. As matter of fact today automobiles have a lot of components that need to exchange messages using network resources. Considering that drivers and passengers safety are at the mercy of the code running in their vehicle, this scenario calls for mechanisms able to assure the safety of the information exchanged by these network connected components. For these reasons, in this paper we propose a method able to detect attacks targeting the CAN protocol. We use real-world data embedded in CAN packets as feature vector and we demonstrate that using fuzzy classification algorithms our method is able obtain a precision ranging from 0.85 to 1 in attacks targeting CAN protocol identification. As future work, we plan to extend our experiment to other real-world attacks and to investigate whether the adoption of formal verification techniques is useful to improve the results [21].

Acknowledgements

This work has been partially supported by H2020 EU-funded projects NeCS and C3ISP and EIT-Digital Project HII and PRIN "Governing Adaptive and Unplanned Systems of Systems".

References

[1] E. Massaro, C. Ahn, C. Ratti, P. Santi, R. Stahlmann, A. Lamprecht, M. Roehder, and M. Huber, "The car as an ambient sensing platform," *Proceedings of the IEEE*, vol. 105, no. 1, pp. 3–7, 2017.

[2] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham *et al.*, "Experimental security analysis of a modern automobile," in *Security and Privacy (SP), 2010 IEEE Symposium on*. IEEE, 2010, pp. 447–462.

Attack	Algorithm	FP Rate	Precision	Recall	F-Measure	Roc Area	Class
dos	FuzzyRoughNN	0.3	0.766	0.981	0.86	0.986	NM
		0.019	0.974	0.7	0.814	0.986	IM
	NN	0	1	0.962	0.98	0.986	NM
		0.038	0.963	1	0.981	0.986	IM
	DiscernibilityClassifier	0	1	0.973	0.987	0.987	NM
		0.027	0.974	1	0.987	0.987	IM
	FURIA	0	1	0.968	0.984	0.984	NM
		0.032	0.969	1	0.984	0.984	IM
fuzzy	FuzzyRoughNN	0.253	0.798	1	0.888	0.988	NM
		0	1	0.747	0.855	0.988	IM
	NN	0.177	0.85	1	0.919	0.986	NM
		0	1	0.823	0.903	0.986	IM
	DiscernibilityClassifier	1	0.5	1	0.667	0.905	NM
		0	0	0	0	0.905	IM
	FURIA	0.715	0.583	1	0.737	0.983	NM
		0	1	0.285	0.444	0.983	IM
gear	FuzzyRoughNN	0	1	1	1	1	NM
		0	1	1	1	1	IM
	NN	0	1	1	1	1	NM
		0	1	1	1	1	IM
	DiscernibilityClassifier	0	1	1	1	1	NM
		0	1	1	1	1	IM
	FURIA	0	1	1	1	1	NM
		0	1	1	1	1	IM
rpm	FuzzyRoughNN	0	1	1	1	1	NM
		0	1	1	1	1	IM
	NN	0	1	1	1	1	NM
		0	1	1	1	1	IM
	DiscernibilityClassifier	0	1	1	1	1	NM
		0	1	1	1	1	IM
	FURIA	0	1	1	1	1	NM
		0	1	1	1	1	IM

TABLE III: Classification results: FP Rate, Precision, Recall, F-Measure and RocArea for classifying the full feature set related to the four attacks (i.e. *dos*, *fuzzy*, *gear* and *rpm*) and the normal messages computed with four different algorithms, with the IM label we referred to the injected message class while with the NM label to the normal messages one.

[3] M. S. Al-Kahtani, "Survey on security attacks in vehicular ad hoc networks (vanets)," in *Signal Processing and Communication Systems (ICSPCS), 2012 6th International Conference on*. IEEE, 2012, pp. 1–9.

[4] G. Samara, W. A. Al-Salihy, and R. Sures, "Security issues and challenges of vehicular ad hoc networks (vanet)," in *New Trends in Information Science and Service Science (NISS), 2010 4th International Conference on*. IEEE, 2010, pp. 393–398.

[5] P. Kleberger, T. Olovsson, and E. Jonsson, "Security aspects of the in-vehicle network in the connected car," in *Intelligent Vehicles Symposium (IV), 2011 IEEE*. IEEE, 2011, pp. 528–533.

[6] M. Wolf, A. Weimerskirch, and C. Paar, "Security in automotive bus systems," in *Workshop on Embedded Security in Cars*, 2004.

[7] R. Jensen and C. Cornelis, "A new approach to fuzzy-rough nearest neighbour classification," in *International Conference on Rough Sets and Current Trends in Computing*. Springer, 2008, pp. 310–319.

[8] Z. Voulgaris and G. D. Magoulas, "Extensions of the k nearest neighbour methods for classification problems," in *Proc. of the 26th IASTED International Conference on Artificial Intelligence and Applications (AIA), Innsbruck, Austria, February 11*, vol. 13, 2008, pp. 23–28.

[9] J. Hühn and E. Hüllermeier, "Furia: an algorithm for unordered fuzzy rule induction," *Data Mining and Knowledge Discovery*, vol. 19, no. 3, pp. 293–319, 2009.

[10] W. W. Cohen, "Fast effective rule induction," in *Proceedings of the twelfth international conference on machine learning*, 1995, pp. 115–123.

[11] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, T. Kohno *et al.*, "Comprehensive experimental analyses of automotive attack surfaces." in *USENIX Security Symposium*. San Francisco, 2011.

[12] S. Woo, H. J. Jo, and D. H. Lee, "A practical wireless attack on the connected car and security protocol for in-vehicle can," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 993–1006, 2015.

[13] A. Taylor, S. Leblanc, and N. Japkowicz, "Anomaly detection in automobile control network data with long short-term memory networks," in *Data Science and Advanced Analytics (DSAA), 2016 IEEE International Conference on*. IEEE, 2016, pp. 130–139.

[14] G. Yan, S. Olariu, and M. C. Weigle, "Providing vanet security through active position detection," *Computer Communications*, vol. 31, no. 12, pp. 2883–2897, 2008.

[15] W. Liu, H. Zhang, and W. Zhang, "An autonomous road side infrastructure based system in secure vanets," in *2009 5th International Conference on Wireless Communications, Networking and Mobile Computing*. IEEE, 2009, pp. 1–6.

[16] N. Lyamin, A. V. Vinel, M. Jonsson, and J. Loo, "Real-time detection of denial-of-service attacks in ieee 802.11 p vehicular networks." *IEEE Communications letters*, vol. 18, no. 1, pp. 110–113, 2014.

[17] K. M. A. Alheeti, A. Gruebler, and K. D. McDonald-Maier, "An intrusion detection system against malicious attacks on the communication network of driverless cars," in *2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC)*. IEEE, 2015, pp. 916–921.

[18] B. I. Kwak, J. Woo, and H. K. Kim, "Know your master: Driver profiling-based anti-theft method," in *PST 2016*, 2016.

[19] B. Yao, H. Hagras, D. Alghazzawi, and M. J. Alhaddad, "A type-2 fuzzy logic machine vision based approach for human behaviour recognition in intelligent environments," in *Fuzzy Systems (FUZZ), 2013 IEEE International Conference on*. IEEE, 2013, pp. 1–8.

[20] K. Ishii and M. Sugeno, "A model of human evaluation process using fuzzy measure," *International Journal of Man-Machine Studies*, vol. 22, no. 1, pp. 19–38, 1985.

[21] N. D. Francesco, G. Lettieri, A. Santone, and G. Vaglini, "Heuristic search for equivalence checking," *Software and System Modeling*, vol. 15, no. 2, pp. 513–530, 2016. [Online]. Available: http://dx.doi.org/10.1007/s10270-014-0416-2