EIDS: Embedded Intrusion Detection System using Machine Learning to detect attack over the CAN-BUS

Marco Lombardi
Department of Industrial Engineering, University of Salerno, Italy. E-mail: malombardi@unisa.it

Francesco Pascale
Department of Industrial Engineering, University of Salerno, Italy. E-mail: fpascale@unisa.it

Domenico Santaniello
Department of Industrial Engineering, University of Salerno, Italy. E-mail: dsantaniello@unisa.it

Nowadays, with the rapidly increasing of connected vehicles more and more cyber-attacks, which could compromise the driving experience, are possible. Indeed, connected vehicles are not able to guarantee the information security and shown their vulnerabilities in terms of Confidentiality, Integrity and Availability (CIA). These vulnerabilities advise the inefficiency of modern vehicles which, due to the advent of the Internet of Things paradigm, are equipped with many Internet access points.
In this paper, we propose an approach based on Intrusion Detection System (IDS), which use a Machine Learning technique through Bayesian Networks approach to detect possible attacks on Controller Area Network Bus (CAN-Bus). In this way, a framework, which takes advantage of an embedded system able to discover a non-linear messages flow on CAN-Bus, is presented.

Keywords: Internet of Things, Cybersecurity, Automotive, Intrusion Detection System, Bayesian Network.

1. Introduction

Today, being able to process data to extrapolate useful information in order to predict events is the subject of much scientific research (Clarizia et al. 2019) (Casillo et al. 2017). In the last few years, more and more vehicles are connected to the Internet through various systems, such as On-Board Units (OBU), Roadside Unit (RSU), Vehicular Ad Hoc Networks (VANET), and Infotainment Systems. More generally, modern vehicles can be considered as objects of the Internet of Things (IoT) paradigm. An example of vehicles that currently make the most of the tight integration between themselves and connection to the internet are found in self-driving cars. In this scenario, new frontiers are opened for scientific research both in terms of development of new features and in terms of safety, understood both as safety for the driver and passengers and as safety of the vehicle itself. The modern vehicles are equipped with multiple connections to the internet and this allows them to have numerous new features but also critical issues and vulnerabilities due to the increased possibility of remote access to the vehicle. In fact, these new technologies have led to the development of connected vehicles not only to the Network (V2R) but more generally to any entity that can influence the vehicle and vice versa (V2X) through the use of communication channels such as cellular networks, Wi-Fi and Bluetooth and platforms like Android Auto or proprietary platforms developed by various manufacturers, such as Sync for Ford, Uconnect for FCA, and OnStar for General Motors. However, these new technologies have exposed numerous security vulnerabilities. The presence of these vulnerabilities meant that security became a priority and that it was integrated into the vehicle development process.
A critical challenge in vehicle cybersecurity is that the various electronic control unit of a car (ECU) are connected via internal network. For example: hackers could access ECU if there is Bluetooth or infotainment system vulnerability, in this case they could control safety critical ECUs and could sabotage brakes. In general, the approach that is used in order to implement a cybersecurity system to guarantee the security can be divided into 3 phases (Chhawri et al. 2017):

- Risk and threat analysis
- Software and hardware design
- Test

In the case of the automotive industry, it is important to analyze well what are the risks and threats that can negatively affect the vehicle.

Proceedings of the 30th European Safety and Reliability Conference and the 15th Probabilistic Safety Assessment and Management Conference
Edited by Piero Baraldi, Francesco Di Maio and Enrico Zio
Published by Research Publishing, Singapore.
ISBN: 978-981-14-8593-0; doi:10.3850/978-981-14-8593-0

Evaluating the possible attack possibilities, intended as points of access to the vehicle, there are connection points at a short distance (Bluetooth, Wi-Fi), long distance (4G, 5G, LTE, DSRC) and physical access points (OBD II, ECU, USB) (Craig 2016). Once understand what access points are, it is important to understand how an attacker can negatively affect the vehicle. Some attacks are aimed at stealing information on target; others are aimed at taking control or tampering the vehicle. The riskiest for vehicles is the latter as undermine the integrity and safety of the vehicle and its passengers. In modern vehicles, all the information inside them, whether are control messages or diagnostic messages, travel through the internal communication channel, the Controller Area Network Bus (CAN-Bus). This allows the interconnection between all the ECUs present on the vehicle (Al-Jarrah et al. 2019).

To ensure the security on CAN-Bus, which is implemented in modern vehicles and used to handle communications among vehicle internal components, is very important. An attack on CAN-Bus can be very dangerous: hacker can have vehicle total control. For this reason, one important purpose of vehicle cybersecurity is to make CAN-bus safe, adopting classical safety techniques.

It is evident as all possible attacks, both that ones originated from outside the vehicle and that ones from inside, are designed to control the CAN-Bus, through which potentially dangerous messages can be sent.

Once the threat model and the risks associated with it have been identified, a system is devised to mitigate these dangers and to test them. In this work we aim to create an Embedded System for the control of intrusions on the CAN-bus based on Bayesian Networks in order to identify possible attacks and properly treat them. This research work wants to investigate on what is the current state of the art of IoT and Cybersecurity in order to evaluate which are the possible solutions applied to a real case of study such as Automotive.

2. Related Works

In recent years many research works have treated the application of machine learning methodologies as their main topic for the detection of cyber-attacks on the network. In IoT systems, the focus has been on understanding what the countermeasures may be to be applied. In fact, the devices have a reduce computational capacity and that must take into account these limitations and find solutions that are effective and at the same time supported by the infrastructures. In particular, the problem of security on CAN-bus has been addressed in various ways, all rather effective but few efficient (Casillo et al. 2019). In literature, there are many works on that aim, in fact, several of its have highlighted how it is possible to use intrusion detection systems (IDS) to prevent a malicious attack. One of the most critical aspects highlighted is the computing power of the hardware used: In fact, the IDS that using Machine Learning is efficient if the device performances are high. In many contexts, such as the IoT, we have components with reduced computational capacity and therefore it becomes essential to arrive at an efficient design (Xin et al. 2018) (Azwar et al. 2018). As it is possible to see in (Abbott-McCune and Shay 2016) (Song et al. 2016), it is used IDS for the control of intrusions in IoT: however, the required computational capacity far exceeds that with which the ECU microcontrollers are typically equipped. In addition to IDS, other methods used to ensure security onboard of vehicles. An approach can be illustrated by Lukasiewycz et al. in which a possible randomization of the CAN-Bus frame identification field is generated (Lukasiewycz et al. 2016). A set of random identifiers are generated, and a new mapping of the various nodes is performed with the new calculated values. The values that make up the generated list must however change over time: it can be calculated when the vehicle is started or modified at clearly defined intervals. Other methods are based on cryptation. Xie et al. shown how is possible to see a CAN-Bus security approach based on a message authentication code (MAC), this method helps protect the vehicle from masked attacks (Xie et al. 2015). However, the resulting bandwidth makes it necessary to find the compromise between security, real-time usage and bandwidth. Siddiqui et al. shown a safe and reliable framework based on hardware has been proposed which implements mutual authentication based on a non-cloneable physical function (PUF) and secure encryption on the non-secure communication channel (Siddiqui et al. 2017). To avoid the problem of delay, Zhang and Masrur proposed a technique that consists in assigning different priorities to encrypted CAN frames so as to compensate for increased delay. The idea is that when the first frame wins arbitration, the second one always wins arbitration within a specific domain and it will be sent with less delay (Zhang and Masrur 2019).

3. Backgrounds on Cybersecurity in automotive and Bayesian Networks

With the advent of IoT in daily life, as regards cyber security, the number of problems to be taken into account has improved and the chances of a cyber-attack have worsened. In literature the attention was paid in particular to the principles and models on which IoT applications are based,

while issues relating to privacy and security were treated only in a generic way.

3.1 *Information Security in Automotive*

The introduction of drive-by-wire technologies in the cars meant that the mechanical and hydraulic connections, which were previously used to control the machine, were replaced by electronic control units (Electronic Control Unit or ECU) which, based on the input of various sensors, they control the various mechanical parts of the car through actuators. More recently, the electronics on board the cars have been exploited even more effectively with the addition of various network interfaces such as Wi-Fi, cellular network, Dedicated Short Range Communication (DSRC), etc. This has allowed manufacturers to send over-the-air (OTA) updates, receive diagnostic information and offer various types of multimedia services, also thanks to these interfaces the car can send and receive signals so as to perceive the surrounding reality and interact with other vehicles (Vehicle to Vehicle or V2V communication) or with specific infrastructures such as those implemented in the context of smart cities (Vehicle to Infrastructure or V2I communication). More generally, at present, all this is indicated by the term "Vehicle to Everything" (V2X) which indicates a system of communication between a vehicle to any entity that may influence the vehicle and vice versa (Levi et al. 2018) (Huang et al. 2019). All this, as already said, offers numerous benefits in terms of comfort, efficiency and safety but also represents a springboard for new cyber-attacks. To explain these problems, the architecture of a car's internal network and the various protocols used (paying particular attention to the CAN protocol) will be described in a general way, and finally the automotive problems already exposed for a generic IoT application will be specified. The problems previously exposed for a generic IoT application are also valid for the automotive world, in fact, as has already been said that cars have become in all respect of smart objects. Also, in this case there is the problem of the heterogeneity of the devices. To be able to implement all the services in the V2V and V2I area, many network interfaces are required. These interfaces can be divided according to the range of action:

- Physical access points: allow to have direct or indirect physical access to the car's internal network (USB, OBD, etc.)
- Short range access points: allow to communicate with the vehicle at a distance that generally varies from 5 to 300 meters. Interfaces such as Wi-Fi, Bluetooth, Remote Keyless Entry (RKE), Tire Pressure Monitoring System (TPMS), etc. are part of this class.
- Long range access points: allow to communicate with the vehicle at a distance greater than 1 Km. Its groups interfaces such as cellular network (4G, 5G), Global Positioning System (GPS), etc.

All this involve into an increase in the attack surface and represents a very serious problem also because the many ECUs on board the car that offer certain services must necessarily dialogue. For example, infotainment systems must dialogue with the systems that manage the powertrain to provide the user with information on consumption, engine status, etc. This problem is therefore very recent and at present there are no (or incomplete) standards that indicate how to approach it. For example, one of the most used safety standards in the automotive sector is ISO26262 which describes a series of methodologies and guidelines to ensure the functional safety of the vehicle, but malicious threats are not taken into consideration (Macher et al. 2016). Even the attacks and countermeasures described for the IoT world can be particularized for the automotive sector. Jamming attacks to disturb communications between the car and the infrastructure (V2I) or between the car and the car (V2V) and methods for accessing the vehicle by exploiting vulnerabilities in the Remote system are also described, considering the car as a smart object. Keyless Entry. Carsten et al. explore the possible vulnerabilities are described (therefore the network level is considered) from which the CAN protocol suffers, in particular (Carsten et al. 2015):

- It is a broadcast protocol, so anyone can write and read messages on the bus to inject malicious messages or listen to information in transit;
- Authentication protocols are missing, so once access to the bus it is easy to impersonate an ECU and send bogus messages in its place;
- Absence of encryption, the traffic passes unencrypted on the bus so, after having access to the bus, it is easy to intercept and reproduce messages.

These vulnerabilities allow Dos attacks (to make the vehicle unusable), spoofing (for example to override the commands given by a legitimate user

and obtain control of the vehicle), etc. To access the bus, it is possible take advantage of the access points described in the previous paragraph. Application layer attacks target application front ends and are similar to those described for generic IoT applications. Even the countermeasures adopted in the automotive sector and the problems that make their implementation difficult are similar to those already described for the IoT world. For example, to solve the security problem in the CAN protocol, it is possible proceed in various ways:

- Use encryption in order not to pass unencrypted messages on the bus, in which case, however, it is necessary not to use algorithms that require too many resources to avoid introducing latencies that could compromise the functionality of the vehicle;
- Use authentication systems to prevent illegitimate ECUs from writing messages on the bus, so the same considerations on computational complexity apply.

3.2 *Bayesian Networks*

A Bayesian network is a probabilistic model that predicts the dependency relationships between a set of random variables and an effect a probabilistic inference process (using the unit of Bayes' theorem) directly related to each other. A Bayesian network can be represented graphically through a direct acyclic graph (direct acyclic graph or DAG), i.e. a graph with oriented arcs and without direct cycles. Each node of the graph is associated with a random variable that can take on various states. The latter, which must be mutually exclusive, are associated with a probability value. The arcs that connect two nodes, on the other hand, indicate a relationship of conditional dependence between the latter. In this case, the node from which the bow starts are called "parent node" while the node to which the bow points is called "child node". If two nodes are not connected, they are conditionally independent. Nodes that do not have parents are associated with a priori probability tables that express previous knowledge on the value that the random variable associated with the node can assume. The nodes that have at least one parent, on the other hand, are associated with a conditional probability table (CPT) which contains the probabilities that the states of the node can assume conditioned by the possible combinations of the states assumed by the parent nodes. The application areas are innumerable and range from decision support systems to monitoring and diagnostics systems. As seen above, many research works focus on the importance of Bayesian networks in critical systems as they give the opportunity to understand how our network has "reasoned" to get a result. his turns out to be very important for Explainable AI as unlike neuronal networks and machine learning and deep learning algorithms these are able to provide the modalities that have been used to obtain a result.

4. The Proposed Approach and Methodology

First of all, we have to define the possible attacks. Starting from the EURO-NCAP[a] specification for the safety of cars, provides from European new car assessment programme, we decide to take into account with only these cases of possible attacks:

- The car goes straight at a constant speed between 90 and 130 km/h; attacker sends a command to turn right or left, at an angle of over 30°, for a time of at least 0.4s, without the presence of an obstacle
- The car goes straight with constant speed between 90 and 130 km/h; attacker send a command to brake, more than 80% of the total possible pressure, for a time at least 0.4 s, without the presence of an obstacle
- The car goes straight with constant speed; attacker switch off the cooling system

In order to obtain the actual conditions of possible attack on the vehicle, the three cases indicated above have been studied in detail and conditions have been defined in which one can be in the presence of a specific attack.
In order to decode a possible attack, a two-step classification algorithm has been developed. The algorithm thus conceived works as follows:

- In the first step, it analyzes 10 state frames (containing each frame the exact values of each car parameter considered for our case study) and verifies through the use of masks obtained from the empirical analysis of the problem if in that sequence of values may or may not be a possible attack. Each status frame is recorded with a unique timestamp and its recording takes place every 4ms.
- In the second step, through the use of a Bayesian network, previously trained through a pre-established dataset during the simulation phase, it is able to decode

[a] New Car Assessment Programme - https://www.euroncap.com/it

if we are in the presence or not of an attack, keeping in mind both the parameters that make up the frame values status, both the parameters obtained as information from these parameters

The figure 1 shows the operating framework of the algorithm proposed as follows:

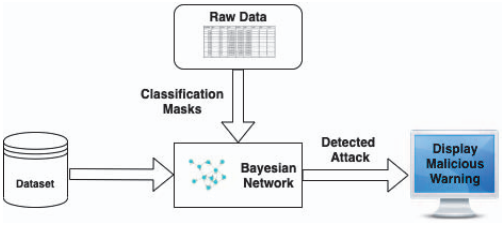

Fig. 1. The proposed Framework.

Let's now analyze in detail the two steps of the proposed algorithm. In the first, as mentioned, the raw information that comes from the state of the system is analyzed. To do this, all the values of the various ECUs that have been considered are recorded frame by frame. In groups of 10 frames at a time, the values of the individual parameters are averaged, and the highest and lowest values are excluded from the calculation:

$$\frac{\sum_{F_i}^{F_{i+9}}[(P_{ji}+P_{ji+1}+\cdots+P_{ji+9})-\min(P_j)-\max(P_j)]}{N-2} \quad (1)$$

Where Fi represents the i th Frame, Pji, .. Pji + 9 are the values that the parameter assumes at each frame i, min (Pj) and max (Pj) represent the minimum and maximum values that the considered parameter can assume in the interval of frames considered, N represents the number of frames considered. At this point a vector of averaged values will be obtained for each parameter which will constitute the state of the system in a period of time equal to 40 ms. at this point, in order to understand if we are in the presence or not of a possible attack, the state vector is compared with pre-established masks that identify us of the possible attack conditions. These masks were obtained starting from the attack conditions considered (see previous chapter) and indicate possible cases of non-normal operation. The masks thus obtained are shown in the table 1:

Table 1. Attack Masks.

Parameter	Lidar	Lines	Throttle	Brake	Steer	Speedometer	Radiator	RPM	Gear
Mask 1	0	any value	any value	any value	-1/0,5 or 0,5/1	> 0,6 and < 0,8	any value	any value	any value
Mask 2	0	any value	any value	> 0,8	any value	> 0,6 and < 0,8	any value	any value	any value
Mask 3	any value	any value	any value	any value	any value	any value	off	any value	any value

As it can possible to see there are 3 masks. Each parameter has been normalized between 0 and 1 except for Steer which has been normalized between -1 and 1 to identify the left and the right. The Lidar and Lines parameters can have a value of 1 or 0 if there is an obstacle less than 20 meters away and if you are crossing a line on the road surface or not. The Radiator parameter indicates whether the radiator is working or not. As you can see, the RPM and Gear values were not taken into consideration during the mask creation process as they did not affect the attack conditions considered. By applying the masks described above it is possible to detect if there is a possible attack. At this point, to understand if we are really under attack, the values of these parameters plus those of the information obtained from them will be passed to the Bayesian network which will indicate to us with a certain probability whether we are under attack or not. In the event that none of the masks had been activated, the vehicle status would have been considered normal and no action would have been taken. In the next phase a Bayesian network was generated starting from a pre-established dataset during the simulation phase with the following parameters listed above:

- Steer: CAN message related to steering, 7 Classes (-1:1 norm., step variable, very left – middle left - left – center – right – middle right – very right);
- Throttle: CAN message related to acceleration, 4 Classes (0:1 norm., step variable, pedal not pressed - low – medium - high);
- Brake: CAN message related to braking, 4 Classes (0:1 norm., pedal not pressed - low – medium - high);
- RPM: CAN message related to rotations per minute, 5 classes (0:1, step variable, stop, slow, normal, medium, high);
- Gear: CAN message related to gear of car, 5 Classes (0,1,2,3,4,5);
- Radiator: State of ignition of the cooling system, 2 Classes (on, off);
- Lidar: Presence or absence of obstacles, 2 Classes (0, 1);

- Lines: Crossing a road line or not, 2 Classes (0, 1);
- Speedometer: Speed in absolute value, 6 classes (0:1 norm., stop – very slowly – slowly – medium – fast – very fast);
- Acceleration: Car acceleration, 5 classes (-1:1 norm., step variable, deceleration high – deceleration low - no acceleration – acceleration low – acceleration high);
- Speed: Car current speed, 6 classes (0:1 norm., step variable, stop[0 km/h] – very slowly[0-30 km/h] – slowly[30-50 km/h] – medium[50-90 km/h] – fast[90-130 km/h] – very fast[130-150km/h]);
- Engine Temperature: Car engine temperature, 4 classes (0:150, step variable, normal operation - low overheating – medium overheating – high overheating);
- Swerve: Car swerve, 7 classes (-1:1 norm., step 0,285, very left[-60° to -45°] – middle left[-45° to -30°] – left[-30° to -5°] – center[-5° to 5°] – right[5° to 30°] – middle right[30° to 45°]– very right[45° to 60°]);
- Obstacle: Presence or not of generic obstacle within a radius of 20 meters, 2 classes (true, false);
- Attack: Presence or not of attack, 2 classes (true, false);

In the case of the rpm, acceleration, speed and engine temperature parameters, a normalization was carried out with respect to constant values greater than the maximum values reached within the simulation, for the parameters consisting of numerical values, the classes were constructed by dividing the equal parts whole range considered. These parameters constitute the nodes of our Bayesian network. The arches were obtained taking into account the ontology made by the Automotive Ontology Community Group[b], the W3C working group and by the Toyota Computational Intelligence Laboratory[c] and according to Colace et al. (Colace and De Santo 2010) (Colace et al. 2010). The net obtained is shown in the figure 2:

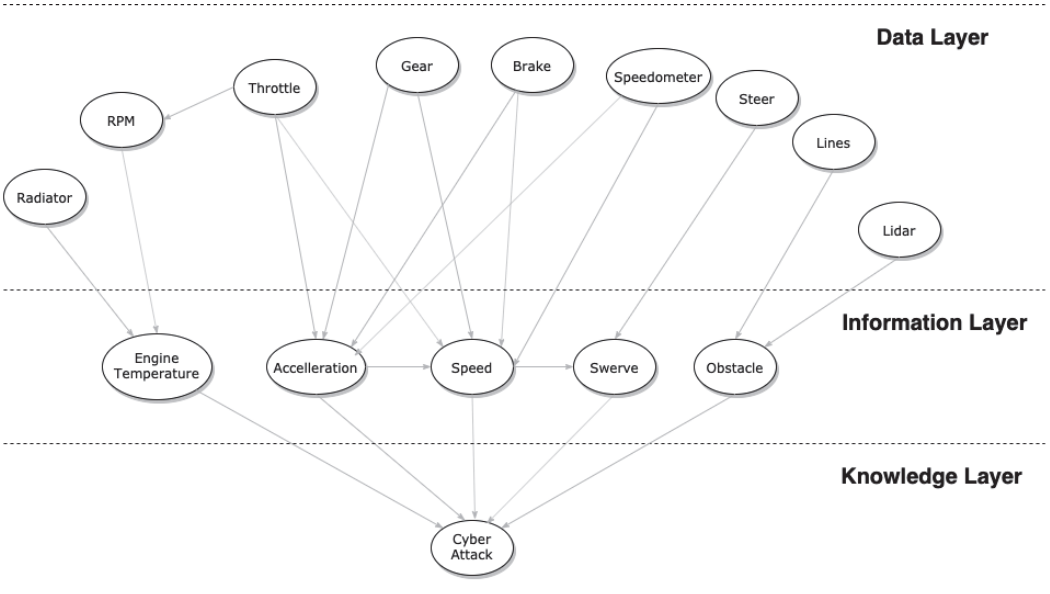

Fig. 2. Obtained Bayesian Network.

[b] Automotive Ontology Community Group - https://www.w3.org/community/gao

[c] Computational Intelligence Laboratory, Toyota Technological Institute, Nagoya, Japan - https://www.toyota-ti.ac.jp/Lab/Denshi/COIN/Ontology/TTICore-0.03/TTICarOnto.owl

As can be seen from the figure 2, the Bayesian Network is constituted in a hierarchical manner with three different levels: Data Layer, Information Layer and Knowledge Layer. The Data Layer level refers to the raw data coming from the vehicle, the Information Layer level refers to the processed information coming from the Data Layer and finally the Knowledge Layer level refers to the knowledge starting from the information in our possession. The network thus obtained is able to decode the presence or absence of an attack with a certain probability. Through a training process before the network and then the inference one, it was possible to evaluate the effectiveness of the method presented as it will see in the next section.

5. Experimental Results

For procedures in the testing phase, it is first necessary to decide which Hardware and Software components to use in order to test the proposed approach and then the classification algorithm and the trained Bayesian network must be implemented. The proposed solution consists in a simulator that emulates a real vehicle and its interaction with the environment, CARLA (Dosovitskiy et al. 2017). This is an open-source software used to carry out research to make a simulation test for connected vehicles and autonomous driving. In addition to the simulator, the architecture includes a steering wheel and pedals that allow to control the vehicle connected to the CAN-Bus through an emulated CAN-Bus; a server that simulates the external environment; an infotainment system that ensures to an access point to the CAN-bus, and a board equipped with a SoC that implements the intrusion detection system. To carry out the experimental phase, a dataset was created containing 50233 frames. Each frame contains all the status parameters for a specific timestamp interval. To do this, was simulated through a city track with CARLA environment the driving of a car. It was realized with a python script, executed for about 24 hours. During driving the vehicle was attacked to simulate a possible intrusion based on the use case. Furthermore, assuming that the channel is ideal and therefore without losses, only the ID and Data Frame fields of the CAN Frame have been considered. In this scenario, the attack node uniquely identifies when a frame has been labeled as an attack. Once the dataset was obtained, the Bayesian Network was then implemented. The bayesian network presented in previous section was created using Weka software (Mhetre and Nagar 2017). In order to test the network thus obtained, it then moved on to translate the xml obtained from the Weka into Python code and through the use of the TensorFlow libraries the Xilinx/PYNQ-Z1 board was then programmed which in our case will act as IDS of our system. The Simple Estimator was used as an algorithm to calculate a priori probabilities and Conditional Probability Tables (CPT). In order to evaluate the system 8158 frame were simulated, where at regular intervals the vehicle was attacked with a malicious message. In order to be considered an attack, the evidence on the attack node must be greater than 0.9. To evaluate the goodness of the proposed algorithm, the experimentation was carried out take in place the value of Precision, Recall and F1 Score, as shown in table 2:

Table 2. Confusion Matrix and Precision, Recall and F1 Score.

		Predicted	
		NEGATIVE	**POSITIVE**
Actual	**NEGATIVE**	6666	74
	POSITIVE	96	1322
		Precision	0.9469
		Recall	0.9322
		F1-Score	0.9394

As it is possible to see, the application of the proposed methodology has good results of the experimentation. The values of Precision Recall and F1 Score in table 2 settle around 0.94. This was possible because the first step that works as pre-filtering with respect to the application of the Bayesian network prevents spurious cases in which the presence of the attack event is not contemplated from being discarded a priori. Furthermore, as can be seen in table 2, the application of the proposed method for the generation of the Bayesian network through the use of the of taxonomy it is possible to find those dependencies that it would be difficult to find with self-training algorithms. A very important factor for the generation of the Bayesian network remains having entered the environmental values within the dataset which give us a clear indication of what is happening around the vehicle.

6. Conclusions

This work showed an embedded system intrusion detection system for automotive that exploiting the potential of machine learning algorithms and in particular Bayesian networks is able to decode possible cyber-attacks on the vehicle CAN-Bas. The whole experimental phase was carried out through the use of an automotive simulator, CARLA, which provides the possibility of performing hardware and software tests in a loop to verify the correct operation of new components for cars. From the first experimental tests conducted we saw how this IDS system can work within the described context.

References

Clarizia, F., Colace, F., De Santo, M., Lombardi, M., Pascale, F., Santaniello, D., & Tuker, A. (2019). A multilevel graph approach for rainfall forecasting: A preliminary study case on london area. Concurrency Computation.

Casillo, M., Colace, F., Pascale, F., Lemma, S., & Lombardi, M. (2017). A tailor made system for providing personalized services. Paper presented at the Proceedings of the International Conference on Software Engineering and Knowledge Engineering, SEKE, 495-500.

Chhawri, S. Tarnutzer, S. Tasky, T. and Lane, G. R. Smart Vehicles, Automotive Cyber Security & Software safety applied to Leader-Follower (LF) and Autonomous Convoy Operations. Proc. of the *Ground Vehicle Systems Engineering and Technology Symposium (GVSETS)*, 2017.

Craig, S. (2016) The car hacker's handbook: a guide for the penetration tester. In *No Starch Press*.

Al-Jarrah, O. Y. Maple, C. Dianati, M. Oxtoby, D. and Mouzakitis, A. (2019) Intrusion Detection Systems for Intra-Vehicle Networks: A Review. In *IEEE Access*, vol. 7, pp. 21266-21289.

M. Casillo, S. Coppola, M. De Santo, F. Pascale and E. Santonicola, "Embedded Intrusion Detection System for Detecting Attacks over CAN-BUS," *2019 4th International Conference on System Reliability and Safety (ICSRS)*, Rome, Italy, 2019, pp. 136-141.

Y. Xin et al., "Machine Learning and Deep Learning Methods for Cybersecurity," in IEEE Access, vol. 6, pp. 35365-35381, 2018.

H. Azwar, M. Murtaz, M. Siddique and S. Rehman, "Intrusion Detection in secure network for Cybersecurity systems using Machine Learning and Data Mining," 2018 IEEE 5th International Conference on Engineering Technologies and Applied Sciences (ICETAS), Bangkok, Thailand, 2018, pp. 1-9.

S. Abbott-McCune and L. A. Shay, "Intrusion prevention system of automotive network CAN bus," 2016 IEEE International Carnahan Conference on Security Technology (ICCST), Orlando, FL, 2016, pp. 1-8.

H. M. Song, H. R. Kim and H. K. Kim, "Intrusion detection system based on the analysis of time intervals of CAN messages for in-vehicle network," 2016 International Conference on Information Networking (ICOIN), Kota Kinabalu, 2016, pp. 63-68.

Lukasiewycz, M., Mundhenk, P., & Steinhorst, S. "Security-aware obfuscated priority assignment for automotive can platforms", ACM Transactions on Design Automation of Electronic Systems (TODAES), 21(2), 32, 2016.

Y. Xie, L. Liu, R. Li, J. Hu, Y. Han and X. Peng, "Security-aware signal packing algorithm for CAN-based automotive cyber-physical systems," in IEEE/CAA Journal of Automatica Sinica, vol. 2, no. 4, pp. 422-430, 10 October 2015.

A. S. Siddiqui, Y. Gui, J. Plusquellic and F. Saqib, "Secure communication over CANBus," 2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS), Boston, MA, 2017, pp. 1264-1267.

M. Zhang and A. Masrur, "Improving Timing Behavior on Encrypted CAN Buses," 2019 IEEE 25th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA), Hangzhou, China, 2019, pp. 1-6.

M. Levi, Y. Allouche, and A. Kontorovich, "Advanced Analytics for Connected Car Cybersecurity," IEEE Veh. Technol. Conf., vol. 2018-June, pp. 1–7, 2018.

J. Huang, M. Zhao, Y. Zhou, and C. Xing, "In-Vehicle Networking: Protocols, Challenges, and Solutions," no. February, pp. 92–98, 2019.

G. Macher, E. Armengaud, E. Brenner, and C. Kreiner, "Threat and Risk Assessment Methodologies in the Automotive Domain," Procedia Comput. Sci., vol. 83, pp. 1288–1294, 2016.

P. Carsten, T. R. Andel, M. Yampolskiy, and J. T. McDonald, "In-vehicle networks: Attacks, vulnerabilities, and proposed solutions," ACM Int. Conf. Proceeding Ser., vol. 06-08-Apri, no. October 2018, 2015.

F. Colace and M. De Santo, "Ontology for E-Learning: A Bayesian Approach," in *IEEE Transactions on Education*, vol. 53, no. 2, pp. 223-233, May 2010

F. Colace, M. De Santo and M. Vento, "A MultiExpert Approach for Bayesian Network Structural Learning," *2010 43rd Hawaii International Conference on System Sciences*, Honolulu, HI, 2010, pp. 1-11.

Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., & Koltun, V., "CARLA: An open urban driving simulator", arXiv preprint arXiv:1711.03938, 2017.

V. Mhetre and M. Nagar, "Classification based data mining algorithms to predict slow, average and fast learners in educational system using WEKA," 2017 International Conference on Computing Methodologies and Communication (ICCMC), Erode, 2017, pp. 475-479.