electronics

MDPI

Article

Cybersecurity in Automotive: An Intrusion Detection System in Connected Vehicles

Francesco Pascale [1,*], Ennio Andrea Adinolfi [2], Simone Coppola [2] and Emanuele Santonicola [2]

[1] Department of Energy, Polytechnic of Milan, 20156 Milan, Italy
[2] Department of Industrial Engineering, University of Salerno, 84084 Fisciano, Italy; eadinolfi@unisa.it (E.A.A.); simo992simo@gmail.com (S.C.); santonicolaemanuele@gmail.com (E.S.)
* Correspondence: francesco.pascale@polimi.it

Abstract: Today's modern vehicles are connected to a network and are considered smart objects of IoT, thanks to the capability to send and receive data from the network. One of the greatest challenges in the automotive sector is to make the vehicle secure and reliable. In fact, there are more connected instruments on a vehicle, such as the infotainment system and/or data interchange systems. Indeed, with the advent of new paradigms, such as Smart City and Smart Road, the vision of Internet of Things has evolved substantially. Today, we talk about the V2X systems in which the vehicle is strongly connected with the rest of the world. In this scenario, the main aim of all connected vehicles vendors is to provide a secure system to guarantee the safety of the drive and persons against a possible cyber-attack. So, in this paper, an embedded Intrusion Detection System (IDS) for the automotive sector is introduced. It works by adopting a two-step algorithm that provides detection of a possible cyber-attack. In the first step, the methodology provides a filter of all the messages on the Controller Area Network (CAN-Bus) thanks to the use of a spatial and temporal analysis; if a set of messages are possibly malicious, these are analyzed by a Bayesian network, which gives the probability that a given event can be classified as an attack. To evaluate the efficiency and effectiveness of our method, an experimental campaign was conducted to evaluate them, according to the classic evaluation parameters for a test's accuracy. These results were compared with a common data set on cyber-attacks present in the literature. The first experimental results, obtained in a test scenario, seem to be interesting. The results show that our method has good correspondence in the presence of the most common cyber-attacks (DDoS, Fuzzy, Impersonating), obtaining a good score relative to the classic evaluation parameters for a test's accuracy. These results have decreased performance when we test the system on a Free State Attack.

Keywords: cybersecurity; automotive; Bayesian network; intrusion detection system; CAN-bus; Internet of Things; embedded systems

Citation: Pascale, F.; Adinolfi, E.A.; Coppola, S.; Santonicola, E. Cybersecurity in Automotive: An Intrusion Detection System in Connected Vehicles. *Electronics* **2021**, *10*, 1765. https://doi.org/10.3390/electronics10151765

Academic Editor: Krzysztof Szczypiorski

Received: 18 June 2021
Accepted: 21 July 2021
Published: 23 July 2021

1. Introduction

Modern vehicles are considered smart objects of an IoT ecosystem [1]. Automated and connected vehicles have a complex architecture, as they integrate multiple automated driving functions and a wide variety of communication interfaces [2,3]. An external attack can compromise these functions, not only endangering the safety of motorists, but can have repercussions in the privacy, financial and operational aspects of companies and passengers. Consequently, increased vehicle connectivity increases the potential risk of cyber-attacks [4].

To integrate a safety assessment into connected and automated vehicle prototypes, it is necessary to ensure that threats to the security and privacy of drivers, business models and the operator's intellectual property are well countered [5–7]. An IoT security assessment of automated vehicles allows manufacturers to do the following:

- Strengthen interest in automated vehicles, demonstrating that security risks have been mitigated, the concept of cyber security has been validated and verified, and systems have been systematically tested.
- Protect motorists and manufacturers by ensuring that cybersecurity threats are handled following state-of-the-art standards and best practices.
- Develop safe and state-of-the-art AV technologies by ensuring that the automated guidance systems adopted are developed with security-by-design and defense-in-depth in mind.
- Gain a competitive advantage by collaborating with international experts who have up-to-date knowledge on information security, vulnerabilities, and applicable standards.

Today, after various attempts to analyze the problem and find a remedy, the ISO 21434 standard has been introduced: this new standard represents an effort aimed at strengthening the culture and presence of cybersecurity within companies involved in automotive product development. It also integrates the cybersecurity process into existing safety processes, especially in the impact assessment and software development process. ISO 21434 sets the clear objective of ensuring that all major players in the automotive sector, be they vehicle manufacturers (so-called OEMs) or component suppliers (so-called TIERs), are aware of the importance of cybersecurity in the development process of products, creating what is called the "security by design" approach [8].

Taking into account the issues outlined above, a framework aimed at cybersecurity should, therefore, foresee different aspects of the life cycle of a connected vehicle, focusing in particular on the following [9,10]:

- Continuous vulnerability management: defining authorized channels for firmware and application updates that restrict the perimeter of attack.
- Security maintainability: if we want to refer, for example, to the cryptographic protection of data, it is unlikely that the keys and algorithms adopted in the initial phase will guarantee the same level of protection over time. For this reason, Security-by-design must be associated with a modular development approach that allows the creation of products capable of adapting to emerging threats.
- Cybersecurity evolution: from this point of view, it is useful to refer to the experience gained by the aeronautical industry, where the use of partitioned embedded systems and domain segregation have made it possible to achieve particularly high security standards.
- The definition of a chain-of-trust, from the prototyping of the individual components of a vehicle, and the system that drives it, to the cloud infrastructure used for data exchange and communications. Solutions based on distributed technologies and blockchain can provide a fundamental contribution in the certification of the phases that participate in the production chain and in the dynamics of the supply chain.
- The implementation of interfaces dedicated to the sector that refer to specialized security policies. The need to develop such countermeasures is accentuated by the frequent use of technologies borrowed from other sectors, such as OTA and bluetooth connections.

In this work, it is proposed an intrusion detection system capable of analyzing traffic over the CAN-Bus and of understanding whether the messages that transmit over the communication channel are malicious or not. After extracting this information status, a two-step algorithm for identifying possible attacks is used: in the first phase, the parameters of the various ECUs of interest are analyzed, comparing them with spatial and temporal analyses that identify possible anomalies in the values. If positive, through the use of Bayesian networks, it is possible to calculate, through a process of inference, the probability that the combination of messages present over the bus represents an anomalous state given by a possible cyber-attack.

In this article, we want to analyze a subsystem of the onboard network as a case study. In particular, we focus on some units highlighted by experts as critical to the vehicle's correct functioning. In fact, the aim is to have a preliminary analysis of the possible use

of these methodologies inside the connected vehicles [11]. The paper is organized in the following way: In the next section, some related works are presented. After, we discuss the backgrounds of cybersecurity in IoT, machine learning and Bayesian networks, and finally, the automotive sector and CAN-Bus. The next section presents our case of study and follows the proposed approach. Finally, some experimental results are discussed with conclusions.

2. Related Works

With the advent of technology, the automotive sector is progressively equipping vehicles with new features, which were unimaginable a few years ago. In the immediate future, the main news will be linked to the connectivity of such vehicles and the raising of autonomous driving levels based on them. It is estimated that by 2022, new vehicles will be connected and capable of communicating with each other [12]. Nowadays, many connected vehicles exchange information via APIs, Wi-Fi, or ad hoc cloud systems. However, each new communication channel opens up new vulnerabilities, particularly toward what underlies the entire vehicle functioning, that is, the internal network [13,14]. In fact, internet access exposes vehicles to greater possibilities of cyberattacks, increasing the entire system's vulnerability. The reasons behind a cybersecurity attack could be numerous and different from each other. An attacker could, for example, smuggle personal information, monitor a person's movements, and, in the worst case, take remote control of the vehicle. In detail, the internal network of modern vehicles, called CAN-Bus, is composed of about 70 nodes. Each node corresponds to an ECU responsible for controlling a specific vehicle component, such as windows, ventilation system, or engine [15,16]. The ECUs communicate in broadcast mode through an unencrypted communication channel called CAN-bus. If an attacker could access them, the entire vehicle's security would be compromised [17,18]. Securing the CAN-bus problem has been known for some time and has already been addressed in various ways, all somewhat effective but not efficient in terms of performance. Many proposals in the literature aim, in fact, to redesign the CAN standard, making a sort of evolution, both on the hardware and the software side. This type of solution, especially regarding the hardware side, does not ensure the vehicles' already marketed safety since it would be necessary to update all the components involved in communications within the vehicle, from the ECU to the cables [19]. Other approaches aim to create intrusion detection systems by potential attackers. In this case, however, the required computational power exceeds the capacity of the microcontroller of today's vehicles. According to the need to keep the amount of information exchanged unchanged, the low computational capacity does not consider MAC-based solutions [20]. Another exciting work introduces a novel algorithm to extract the CAN bus's real-time model parameters and develop SAIDuCANT, a specification-based intrusion detection system (IDS), using anomaly-based supervised learning with the real-time model as input [21]. Some recent studies have analyzed cyber risk from IoT and existing cyber risk assessment approaches and advancements in IoT cyber risk assessment with artificial intelligence and machine learning [22,23]. Other interesting works are related to vehicle mobility and its management [24,25].

3. Backgrounds on Cybersecurity in IoT, Bayesian Networks and Information Security in Automotive

In this section, the principles of security in IoT are illustrated and then specified in the automotive world, with particular attention to intrusion control systems based on probabilistic approaches.

3.1. Cybersecurity in IoT

With the advent of IoT in daily life, the number of vulnerabilities has increased, as has the risk of cyberattacks. In the literature, attention has been paid to the principles and models on which IoT applications are based. At the same time, issues relating to privacy and security have only been treated in a generic way [26–28]. Firstly, the problem of the heterogeneity of devices must be considered. In fact, with the advent of many smart objects connected to the network, we move from the protection of individual computers to the protection of several different devices, built by various manufacturers and each with its level of security. Even if most devices are designed with particular attention paid to safety (which, in reality, often does not happen), few with a few vulnerabilities would be enough to bring down the entire structure. In addition, in many cases, devices must always remain connected, so a possible attacker could launch an attack at any time [29]. Since the IoT security problem has been considered relatively recently, there is no valuable empirical data repository or signatures on the attacks already made.

Furthermore, in IoT applications, the difference between safety and security is very blurred. In general, the concept of "security" indicates the security of data and IT systems; the concept of "safety" indicates physical objects and people's safety. Until now, these two concepts have been separated. Another critical problem concerns user privacy; privacy threats could come from possible attackers and the companies that manage the IoT market. IoT devices, in fact, have become part of daily life in a too pervasive way and collect a large amount of information about people, such as habits, tastes, health, etc. This information can be used both by companies (for example, to profile users) and by hackers with malicious purposes [30]. In order to study effective countermeasures, it is necessary to create a taxonomy of possible attacks. It is possible to group attacks based on possible vulnerabilities present in the three levels of the generic architecture of an IoT application with a perception layer (the bottom layer of the IoT architecture; it interacts with physical devices through smart devices, for example, RFID, sensors, actuators, etc.), network layer (the layer in the middle that is responsible of information transmission) and application layer (the layer on the top of the architecture of IoT. It receives data transmitted from the network layer and uses these data to provide required services or operations) [31–34]:

- Theft or damage of device: (perception layer) physical damage to the device.
- Side-channel attacks: (perception layer) collect information on the running time, power consumption, electromagnetic radiation, or sounds produced by a device during the execution of a particular task to deduce information contained in the device memory;
- Fake Node attacks: (perception layer) inserting into the network nodes created by the attacker in order to transmit bogus information or consume the resources, in terms of energy, of the legitimate nodes;
- Replay attacks: (perception layer) after having intercepted authentic credentials of a node; an attacker then sends them back to the recipient simulating the identity of the issuer;
- Node Tampering attacks: (perception layer) replace part of the node hardware or firmware with components created by the attacker and equipped with malicious functions;
- Jamming attacks: (perception layer) consists, if the nodes communicate via wireless protocols, of disturbing the frequencies used by the protocol;
- Denial-of-Service (DoS): (network layer) its purpose is to prevent reaching the nodes via the network. To achieve this goal, it is possible to use many techniques, such as sending a large number of bogus packets on the network to make sure that the various nodes have more information in input than they can process (flooding), compromising a node in the network in order to modify its topology and degrade its performance (sinkhole attack);
- Man-in-the-middle: (network layer) consists of intercepting the data transmitted by the various nodes before they arrive at the recipient to steal them or retransmit a modified version;

- Storage attacks: (network layer) consist of modifying user information in the device memory or in the cloud;
- Routing attacks: (network layer) a class of attacks (the sinkhole attack is an example) in which an attacker tries to alter the information that the devices use to route packets to create loops, to send error messages, or to lose packets;
- Cross-Site-Scripting (XSS): (application layer), which uses client-side scripting languages (for example, JavaScript) to execute malicious code through a browser that shows a specific web page. This type of attack is also exploited in the IoT field because embedded devices often use web interfaces for configuration, more particularly in this case, we speak of Cross-Channel-Scripting (XCS);
- Malicious Code: (application layer) inject malicious code (malware) into the application for it to execute it;
- Credential theft: (application layer) in order to impersonate legitimate users. This layer can be accomplished through eavesdropping, man-in-the-middle attacks, brute force or dictionary attacks (to try to guess credentials), etc.

Just as the attacks are combined to achieve a goal, the various countermeasures must be combined to hinder the attacker in order to attack as lengthily and expensively as possible. To avoid device damages or violations, a series of actions, which do not necessarily include the use of advanced technologies, can be used [35,36].

3.2. Bayesian Network

A Bayesian network is a probabilistic graph that predicts the dependency relationships between a set of random variables through a probabilistic inference process (using the unit of Bayes' theorem). 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, is 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 the "parent node", while the node to which the bow points is called the "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 to be 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 allow understanding how our network has "reasoned" to obtain a result. It turns out to be crucial for Explainable AI, as, unlike neuronal networks and machine learning and deep learning algorithms, these can provide the modalities that were used to obtain a result.

3.3. Information Security in Automotive

The problems previously exposed for a generic IoT application are also valid for the automotive world, in fact, as it has already been said that cars have become, in all respect, smart objects. 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 areas, many network interfaces are required [37,38]. These interfaces can be divided, according to the range of action:

- Physical access points: allow direct or indirect physical access to the car's internal network (USB, OBD, etc.).
- Short range access points: allow communication with the vehicle at a distance that generally varies from 5 to 300 m. 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 communication with the vehicle at a distance greater than 1 km. Its groups interfaces include cellular networks (4G, 5G), global positioning system (GPS), etc.

All this involves an increase in the attack surface and represents a severe problem also because the many ECUs on board the car that offers specific services must necessarily dialogue. 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 [39–41]. For example, to solve the CAN protocol's security problem, it is possible to proceed in various ways: with the use of encryption, access control, an authentication system, or an intrusion detection system.

3.3.1. Encryption, Access Control and Authentication Systems

Encryption allows to "obfuscate" data to only be read by the user for whom it is intended. This allows the implementation of systems that guarantee the confidentiality and integrity of data to counter attacks, such as Man-in-the-Middle, Storage Attack, Node Tampering, etc. Two types of cryptography can be distinguished: symmetric and asymmetric. In the first case, the key used to encrypt the message is the same as that used to decrypt it; in the second case, different keys are used to encrypt and decrypt the message. However, many of the cryptographic methods currently used require many resources that, as already mentioned, are not available in IoT nodes. In applications with particularly stringent real-time requirements, the use of cryptography would introduce too-high delays. For these reasons, light cryptographic algorithms have been implemented (elliptical cryptography, a type of asymmetric cryptography), which are less robust but still able to hinder possible attackers. Asymmetric encryption also allows signing a message since the data encrypted with one of the two keys can only be decrypted with the other; thus, it is possible to be sure that the sender has the key used to encrypt. However, there is no information on the sender's identity; this identity is ascertained (always through mechanisms based on asymmetric encryption) by a third party called the certification authority (CA). Implementing authentication and access control systems is particularly important to ensure that only legitimate users or nodes can interact with each other. It is used to face attacks, such as Fake Node, Sybil Attack, etc. As for the authentication of users in the literature, it is proposed to use multi-factor authentication, using, in a combined way, a password, biometric characteristics, smart cards, or physical keys. The authentication between nodes is proposed to use systems based on digital signatures, pre-shared keys between devices, and devices created specifically to act as a certification authority [42,43]. In the literature, there are many works on this subject [44–48]. Another important countermeasure is the message authentication code, a critical cybersecurity measure in the current automotive industries. This type of data authentication method is already used as some industrial standards, such as AUTOSAR.

3.3.2. Intrusion Detection System

An intrusion detection system (IDS) is a hardware and/or software system that aims to detect any set of actions that aim to compromise the confidentiality, integrity, or availability of a resource. These systems can be classified into two types, based on the strategy adopted to detect intrusions:

- Signature detection based in which the collected data are compared with traces of already known attacks looking for a correspondence that confirms the fact that there is an attack in progress;
- Anomaly detection based, by which the system's behavior is monitored by checking that its methods of use do not deviate from the regular use.

The first class is very effective against attacks that are already known and well modeled, but it is difficult to recognize new attacks unless it receives costly updates. The IDS belonging to this class comprises rules-based systems, such as expert systems, systems based on fuzzy logic, etc. The second class of IDS, on the other hand, is more likely to detect attacks that are not known a priori but generally have less accuracy. Neural networks, systems based on genetic programming, Bayesian networks, etc., can be used for implementation. In general, to obtain good performance, it can be useful to use a hybrid approach of those described. In the IoT field, the use of IDS is problematic due to the considerable resources required by these systems and the absence of a good number of well-modeled attacks, making it challenging to implement systems belonging to the first class. To solve this latter problem, however, in recent years, many researchers have tried to implement honeypots to obtain and analyze more data [49].

4. The Proposed Approach and Methodology

In this section, the proposed approach is shown. Starting from the case study, in this part are shown what the particular cases deal with in the chosen context.

4.1. Case of Study

Once the threat model and the risks associated with it have been identified, a system is devised to mitigate these dangers and test them. Using a two-step detection algorithm that exploits both the variation of the status parameters of the various ECUs over time and the Bayesian networks, it can identify a possible attack. First of all, we have to analyze the domain to understand the parameters and their related ECUs that must be taken into consideration to map the vehicle and all the possible cyber risks associated with it. To obtain the actual conditions of a possible attack on the vehicle, the conditions were defined in which one can be in the presence of a specific attack. In particular, as we will see in the next section, the parameters that, combined, can identify a possible attack situation were identified. We must define the application domain through ontologies. Subsequently, the reference system's architecture was defined as well as the parameters considered for the definition of our case study. Therefore, to identify our application domain, we referred to various reference ontologies in the automotive sector. Among the various ontologies to which we referred, we considered those made available by the Automotive Ontology Community Group, the W3C working group and a group of domain experts of University of Salerno. Other reference ontologies taken into consideration were those present in the research in [50–52]. Once the context was identified, it was necessary to define the taxonomy of a car to identify the characteristics of the system and the identification of the parameters to be monitored. Domain ontologies were then analyzed in order to identify the characteristic parameters and the relationships between them. Figure 1 shows the obtained taxonomy:

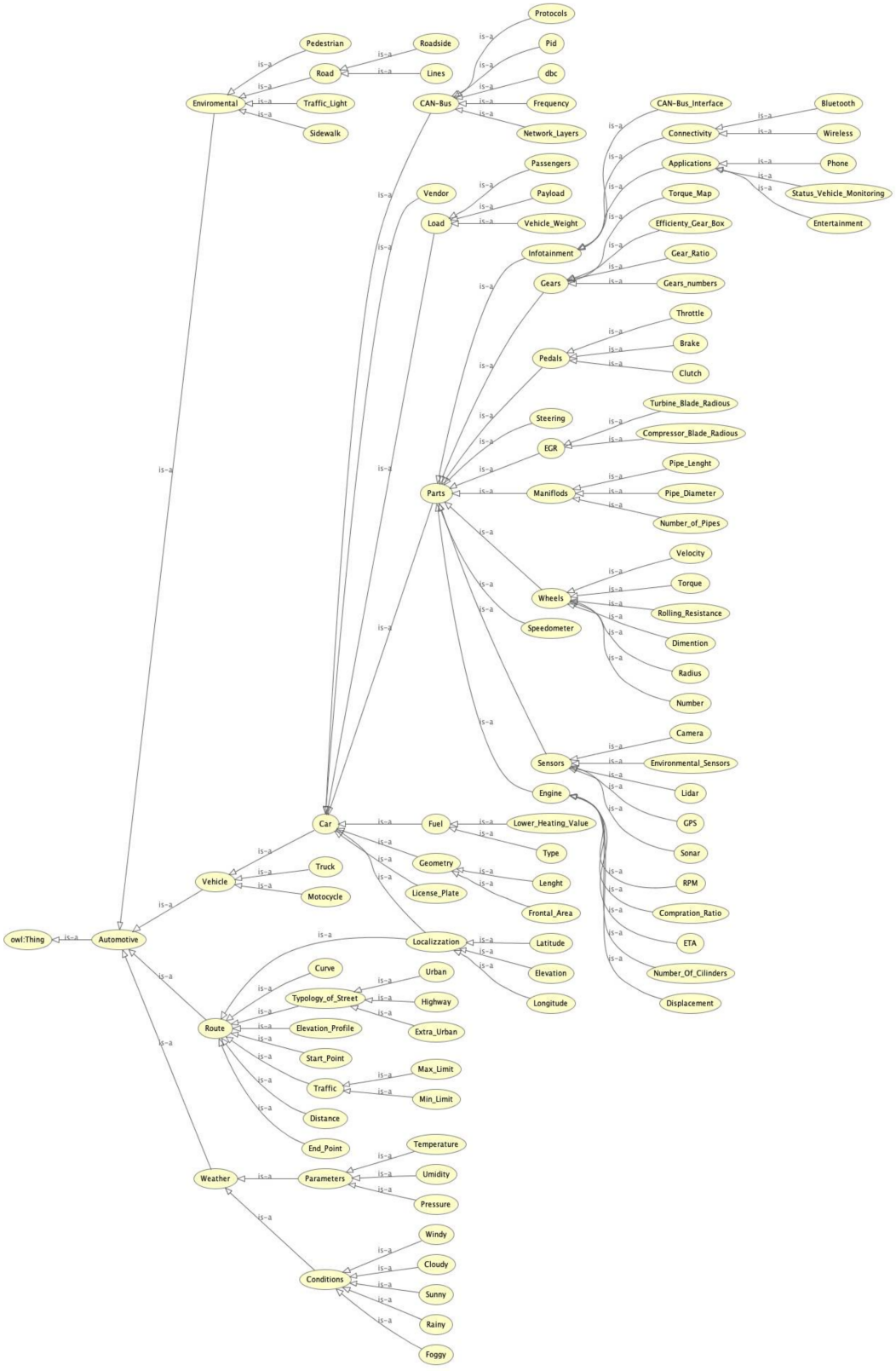

Figure 1. Obtained taxonomy from domain analysis.

Starting from the analysis of the taxonomy, we have taken into consideration these parameters as being of interest for our case study: RPM, throttle, brake, steering, gears, speedometer, radiator, lidar, and lines. These parameters give us the possibility to evaluate the vehicle dynamics and the conditions of a possible attack. From there, it is possible

to trace the values of speed, acceleration, engine temperature, steering, and presence of obstacles that allow us to understand whether or not we are in the presence of one of the possible attack conditions contemplated. At this point, it is necessary to define the architecture of the reference system. Figure 2 shows the architecture.

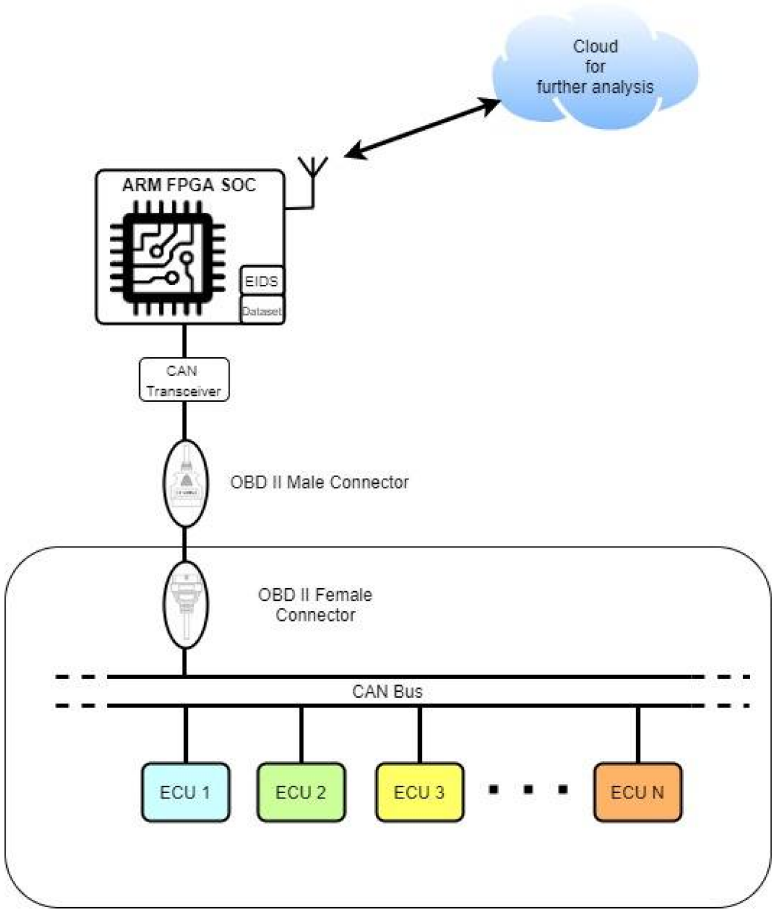

Figure 2. Proposed architecture.

Through the use of a system on a chip (SoC) connected through a transceiver can to the OBDII port, it is possible to create an embedded intrusion control system (EIDS) capable of analyzing the flow of data present on the vehicle and detecting if there are any cyber-attacks. The system, as we will see later, uses a two-step algorithm where there is first a temporal–spatial analysis and then a probabilistic analysis carried out starting from a reference data set. The architecture then provides for the possibility of any future subsequent analysis through a connectivity module that allows interaction with, for example, an external cloud. For this analysis, as we will see later, an experimental data set was implemented starting from a simulated environment and this was then compared with other data sets present in the literature.

4.2. Two-Steps Algorithm

In order to decode a possible attack, a two-step classification algorithm was developed. The algorithm thus conceived works as follows:

- The first step, called pre-processing, analyzes ten state frames (containing each frame the exact values of each car parameter considered for our case study, tab). Moreover, it verifies through spatial and temporal analysis obtained from an analysis of the problem whether it may or may not be a possible attack in that sequence of values. Each status frame is recorded with a unique timestamp, and its recording takes place every 4 ms.

- In the second step, through the use of a Bayesian network, previously trained through a pre-established data set during the simulation phase, it can decide whether we are in the presence or not of an attack, keeping in mind both the parameters that make up the frame values status, and the parameters obtained as information from these parameters.

Figure 3 shows the operating framework of the algorithm proposed. Let us now analyze in detail the two steps of the proposed algorithm.

Figure 3. The matching process of two-step algorithm.

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 were considered are recorded frame by frame. In groups of 10 frames (N) at a time, the values of the individual parameters are averaged, and the highest and lowest values are excluded from the calculation:

$$\frac{\sum_{Fi}^{Fi+N-1}\left[\left(P_{ji}+P_{ji+1}+\cdots+P_{ji+N-1}\right)-\min(P)-\max(P)\right]}{N-2} \quad (1)$$

where F_i represents the i th frame, P_{ji}, ... $P_{ji} + N - 1$ are the values that the parameter assumes at each frame i, min(P_j) and max(P_j) represent the minimum and maximum values that the considered parameter can assume in the interval of frames considered, and N represents the number of frames considered. At this point, a vector of averaged values are obtained for each parameter which constitute the system's state in a period equal to 40 ms. At this point, in order to understand whether or not we are in the presence of a possible attack, the values of these parameters plus those of the information obtained from them are passed to the Bayesian network, which indicates to us with a certain probability as to whether we are under attack or not. If none of the masks are activated, the vehicle status is considered normal, and no action is taken. In the next phase, a Bayesian network is generated, starting from a pre-established data set during the simulation phase with the following parameters (Figure 3 shows the matching process):

- **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 [130–150 km/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 absence of generic obstacle within a radius of 20 m, 2 classes (true, false);
- **Attack**: Presence or absence of attack, 2 classes (true, false).

In the RPM, acceleration, speed, and engine temperature parameters, a normalization was carried out concerning 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 whole parts range considered. These parameters constitute the nodes of our Bayesian network. The arches were obtained, taking into account the obtained taxonomy of car (Figure 1) created for this case study according to Colace et al. [53–55] and with Casillo et al. [56]. The net obtained is shown in Figure 4:

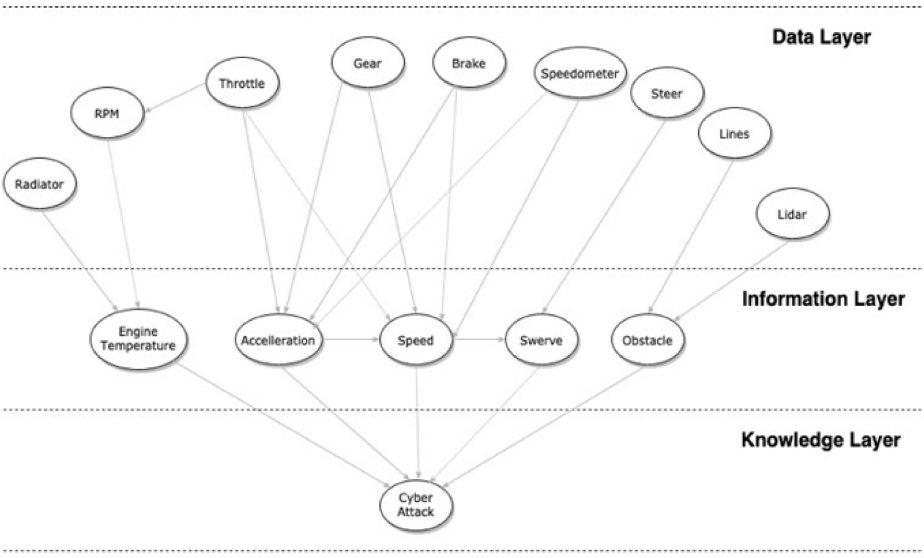

Figure 4. The obtained Bayesian network.

As can be seen from Figure 4, the Bayesian network presents 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. Thus, the network can decode the presence or absence of an attack with a certain probability [57–59]. Through a training process before the network and then the inference one, it was possible to evaluate the method's effectiveness as shown in the next section.

5. Experimental Results

To evaluate our method, we have to define the type of attack that we can take into account for experimental phase. We consider this kind of attack:

1. DoS attack: injecting messages of '0 × 000' CAN ID in a short cycle.
2. Fuzzy attack: injecting messages of spoofed random CAN ID and DATA values.
3. Impersonation attack: injecting messages of Impersonating node, arbitration ID = '0 × 164'.
4. Attack Free State: normal CAN messages.

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; then, the classification algorithm and the trained Bayesian network must be implemented. The proposed solution consists of a simulator that emulates a real vehicle and its interaction with the environment, CARLA [60]. This is an open-source software used to carry out research to make a simu-

lation test for connected vehicles and autonomous driving. In addition to the simulator, the architecture includes a steering wheel and pedals that allow controlling the vehicle connected to the CAN-Bus through an emulated CAN-Bus; a server that simulates the external environment; an infotainment system that ensures 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, 4 data sets, one for each kind of attack, were created, containing about 8158 frames, where, at regular intervals, the vehicle was attacked for a total of about 1000 malicious messages. Each frame contains all the status parameters for a specific timestamp interval. To do this, it was simulated through a city track, with the CARLA environment, the driving of a car. It was realized with a Python script, executed for about 24 h. 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 were considered. In this scenario, the attack node uniquely identifies when a frame is labeled as an attack. Once the data set was obtained, the Bayesian network was then implemented. The Bayesian network presented in the previous section was created using Weka software [61]. 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 [62], the Xilinx/PYNQ-Z1 board was then programmed, which, in our case, acted as the IDS of our system. The simple estimator was used as an algorithm to calculate a priori probabilities and conditional probability tables (CPT). To classify the results obtained, the following cases were distinguished:

- True Positives (TP): attack present and correct classification;
- True Negatives (TN): attack not present and correct classification;
- False Positives (FP): attack not present and incorrect classification;
- False Negatives (FN): attack present and incorrect classification.

This classification of merit can be schematized in a confusion matrix observable in Table 1.

Table 1. Confusion matrix.

Detected Attack	YES	NOT
YES	True Positives (TP)	False Negatives (FN)
NOT	False Positives (FP)	True Negatives (TN)

A confusion matrix is a table used to estimate a classifier's goodness. There are the events considered in the rows, while in the columns, their classification is present. The data on the main diagonal represent correct classifications. From this table are also derived three merit factors that contribute to the analysis of a classifier's performance: the precision (P) (2) merit factor takes into account the number of correct attack identifications concerning the total number of detections. It is obtained with the following formula:

$$P = TP/TP + FP \tag{2}$$

The pecall (R) (3) factor of merit takes into account the number of correct attack identifications compared to the total number of attacks made:

$$R = TP/TP + FN \tag{3}$$

Finally, the F1-Score factor (F1) (4) is given by the harmonic average of precision and recall and measures the accuracy of the classification of events:

$$F1 = 2 \cdots R \cdots P/P + R \tag{4}$$

To evaluate the system performance, we have to decide to compare our solution and the created data set with a common data set present in the literature [63]. In this way, it is possible to see the effectiveness of the proposed methodology. As can be seen in Figure 5, the obtained data sets from the Carla simulation are compared with KIA SOUL data sets presented in the literature. In Figure 5, it is possible to see the experimental results of the second test carried out.

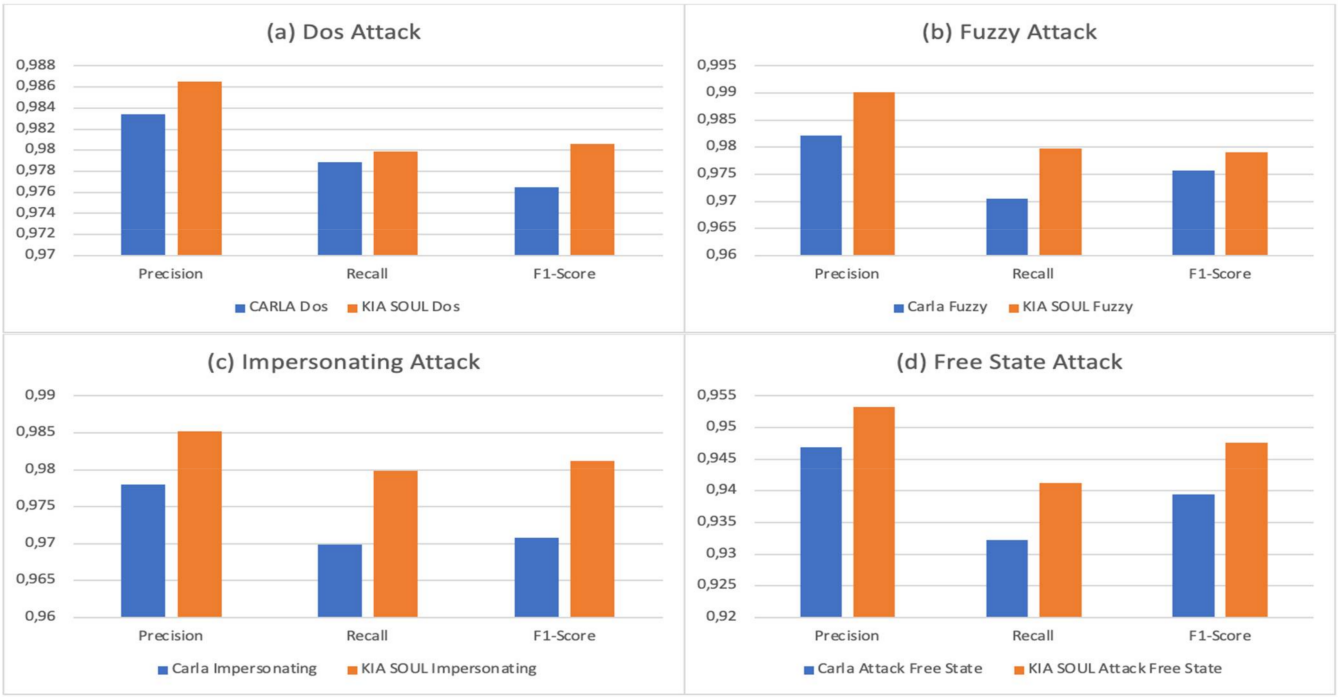

Figure 5. Precision, recall and F1-Score of experimental results: (**a**) Dos Attack, (**b**) Fuzzy Attack, (**c**) Impersonating Attack, (**d**) Free State Attack.

As it is possible to see in Figure 5a–c, we have very high values of precision, recall and F1-score, which gives us a good response from the system. As regards Figure 5d, we have fewer performing results but this is correct, as the last case is more difficult to identify, as it could be easily labeled as a malfunction or other. The thing that encourages us is that our system responds well with real data sets rather than simulated data sets; this bodes well for a possible future test on real simulated environments or vehicles [64–66].

6. Conclusions

This article shows an embedded intrusion control system capable of verifying the presence or absence of cyber attacks on connected vehicles; in practice, by probabilistically analyzing the data traveling in the subsystem of the ECUs connected to each other via the CAN protocol, it is able to identify possible attacks. This system uses a two-step algorithm capable of carrying out a temporal–spatial analysis and a probabilistic analysis through Bayesian networks. Thanks to the ontological study domain and the elaboration of a reference taxonomy, it was possible to identify the critical systems that must be considered in the analysis phase.

The purpose of this research work is to analyze the vulnerabilities inside the connected vehicles and try to find a custom solution in order to limit the vulnerabilities due to the entry into the network of modern vehicles. An embedded intrusion detection system was developed, which is able to analyze the data traffic inside the CAN-Bus and identify those flows that can be labeled as malicious. The study took into consideration what was reported in the literature and was compared with the most common cyber attacks in use

today. In order to evaluate the effectiveness of the method, this was compared with various data sets present in the literature.

The first simulated experimental results compared with data sets present in the literature give us the vision and perception of the effectiveness of this method. Other in-depth studies will have to be conducted in real cases to ascertain their validity.

7. Patents

This research work is a preparatory part of the work carried out within the Italian patent pending No. 102021000009548 registered on 14 April 2021.

Author Contributions: Conceptualization, F.P., E.A.A., S.C. and E.S.; methodology, F.P., E.A.A., S.C. and E.S.; formal analysis, E.A.A.; investigation, F.P.; resources, E.S.; data curation, F.P.; writing—original draft preparation, E.A.A. and E.S.; writing—review and editing. F.P. and S.C.; visualization, E.A.A., S.C. and E.S.; supervision, F.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Lombardi, M.; Pascale, F.; Santaniello, D. Internet of Things: A General Overview between Architectures, Protocols and Applications. *Information* **2021**, *12*, 87. [CrossRef]
2. Lu, Y.; Xu, L.D. Internet of Things (IoT) Cybersecurity Research: A Review of Current Research Topics. *IEEE Internet Things J.* **2019**, *6*, 2103–2115. [CrossRef]
3. Botte, M.; Pariota, L.; D'Acierno, L.; Bifulco, G.N. C-ITS communication: An insight on the current research activities in the European Union. *Int. J. Transp. Syst.* **2018**, *3*, 52–63.
4. Nzabahimana, J.P. Analysis of security and privacy challenges in Internet of Things. In Proceedings of the 2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies (DESSERT), Kyiv, Ukraine, 24–27 May 2018; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2018; pp. 175–178.
5. Whitman, M.E.; Mattord, H.J. *Principles of Information Security*, 4th ed.; Course Technology: Boston, MA, USA, 2011.
6. Chhawri, S.; Tarnutzer, S.; Tasky, T.; Lane, G.R. Smart Vehicles, Automotive Cyber Security & Software safety applied to Leader-Follower (LF) and Autonomous Convoy Operations. In Proceedings of the 2017 Ground Vehicle Systems Engineering and Technology Symposium (GVSETS), Novi, MI, USA, 8–10 August 2017.
7. Haus, M.; Waqas, M.; Ding, A.Y.; Li, Y.; Tarkoma, S.; Ott, J. Security and Privacy in Device-to-Device (D 2 D) Communication: A Review. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 1054–1079. [CrossRef]
8. Buczak, A.L.; Guven, E. A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 1153–1176. [CrossRef]
9. Xin, Y.; Kong, L.; Liu, Z.; Chen, Y.; Li, Y.; Zhu, H.; Gao, M.; Hou, H.; Wang, C. Machine Learning and Deep Learning Methods for Cybersecurity. *IEEE Access* **2018**, *6*, 35365–35381. [CrossRef]
10. Azwar, H.; Murtaz, M.; Siddique, M.; Rehman, S. Intrusion Detection in secure network for Cybersecurity systems using Machine Learning and Data Mining. In Proceedings of the 2018 IEEE 5th International Conference on Engineering Technologies and Applied Sciences (ICETAS), Bangkok, Thailand, 22–23 November 2018; pp. 1–9.
11. Lokman, S.-F.; Othman, A.T.; Abu Bakar, M.H. Intrusion detection system for automotive Controller Area Network (CAN) bus system: A review. *EURASIP J. Wirel. Commun. Netw.* **2019**, *2019*, 184. [CrossRef]
12. Kulandaivel, S.; Goyal, T.; Agrawal, A.K.; Sekar, V. Canvas: Fast and inexpensive automotive network mapping. In Proceedings of the 28th USENIX Security Symposium, Santa Clara, CA, USA, 14–16 August 2019; pp. 389–405.
13. Lin, C.-W.; Sangiovanni-Vincentelli, A. Cyber-Security for the Controller Area Network (CAN) Communication Protocol. In Proceedings of the 2012 International Conference on Cyber Security, Alexandria, VI, USA, 14–16 December 2012; pp. 1–7.
14. Fowler, D.S.; Cheah, M.; Shaikh, S.A.; Bryans, J. Towards a Testbed for Automotive Cybersecurity. In Proceedings of the 2017 IEEE International Conference on Software Testing, Verification and Validation, Tokyo, Japan, 13–17 March 2017; pp. 540–541.
15. Hoppe, T.; Kiltz, S.; Dittmann, J. Security threats to automotive can networks—Practical examples and selected short-term countermeasures. *Reliab. Eng. Syst. Saf.* **2011**, *96*, 11–25. [CrossRef]
16. Koscher, K.; Czeskis, A.; Roesner, F.; Patel, S.; Kohno, T.; Checkoway, S.; McCoy, D.; Kantor, B.; Anderson, D.; Shacham, H.; et al. Experimental Security Analysis of a Modern Automobile. In Proceedings of the 2010 IEEE Symposium on Security and Privacy, Barkeley, CA, USA, 22–25 May 2010; pp. 447–462.
17. Onishi, H. Paradigm change of vehicle cyber security. In Proceedings of the 2012 4th International Conference on Cyber Conflict (CYCON 2012), Tallinn, Estonia, 5–8 June 2012; pp. 1–11.

18. Reilly, J.; Martin, S.; Payer, M.; Bayen, A. On cybersecurity of freeway control systems: Analysis of coordinated ramp metering attacks. *Transp. Res.* **2014**, 1–20.
19. Li, R.; Liu, C.; Luo, F. A design for automotive CAN bus monitoring system. In Proceedings of the 2008 IEEE Vehicle Power and Propulsion Conference, Harbin, China, 3–5 September 2008; pp. 1–5.
20. Zalman, R.; Mayer, A. A secure but still safe and low cost auto- motive communication technique. In Proceedings of the 51st Annual Design Automation Conference, San Francisco, CA, USA, 1–5 June 2014; pp. 1–5.
21. Olufowobi, H.; Young, C.; Zambreno, J.; Bloom, G. SAIDuCANT: Specification-Based Automotive Intrusion Detection Using Controller Area Network (CAN) Timing. *IEEE Trans. Veh. Technol.* **2019**, *69*, 1484–1494. [CrossRef]
22. Radanliev, P.; De Roure, D.; Walton, R.; Van Kleek, M.; Montalvo, R.M.; Maddox, L.; Santos, O.; Burnap, P.; Anthi, E. Artificial intelligence and machine learning in dynamic cyber risk analytics at the edge. *SN Appl. Sci.* **2020**, *2*, 1–8. [CrossRef]
23. Radanliev, P.; De Roure, D.C.; Nurse, J.R.C.; Montalvo, R.M.; Cannady, S.; Santos, O.; Maddox, L.; Burnap, P.; Maple, C. Future developments in standardisation of cyber risk in the Internet of Things (IoT). *SN Appl. Sci.* **2020**, *2*, 1–16. [CrossRef]
24. Waqas, M.; Niu, Y.; Li, Y.; Ahmed, M.; Jin, D.; Chen, S.; Han, Z. A Comprehensive Survey on Mobility-Aware D2D Communications: Principles, Practice and Challenges. *IEEE Commun. Surv. Tutor.* **2019**, *22*, 1863–1886. [CrossRef]
25. Ullah, S.; Abbas, G.; Waqas, M.; Abbas, Z.; Tu, S.; Hameed, I. EEMDS: An Effective Emergency Message Dissemination Scheme for Urban VANETs. *Sensors* **2021**, *21*, 1588. [CrossRef]
26. Vijayalakshmi, A.V.; Arockiam, L. A Study on Security Issues and Challenges in IoT. *Int. J. Eng. Sci. Manag. Res.* **2016**, *3*, 34–43.
27. Sfar, A.R.; Natalizio, E.; Challal, Y.; Chtourou, Z. A roadmap for security challenges in the Internet of Things. *Digit. Commun. Netw.* **2018**, *4*, 118–137. [CrossRef]
28. Neshenko, N.; Bou-Harb, E.; Crichigno, J.; Kaddoum, G.; Ghani, N. Demystifying IoT Security: An Exhaustive Survey on IoT Vulnerabilities and a First Empirical Look on Internet-Scale IoT Exploitations. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 2702–2733. [CrossRef]
29. Frustaci, M.; Pace, P.; Aloi, G.; Fortino, G. Evaluating critical security issues of the IoT world: Present and future challenges. *IEEE Internet Things J.* **2018**, *5*, 2483–2495. [CrossRef]
30. Wang, H.; Te Lai, T.T.; Choudhury, R.R. MoLe: Motion leaks through smartwatch sensors. In Proceedings of the 21st Annual International Conference on Mobile Computing and Networking, Paris, France, 7–11 September 2015; pp. 155–166.
31. Burhan, M.; Rehman, R.A.; Khan, B.; Kim, B.-S. IoT Elements, Layered Architectures and Security Issues: A Comprehensive Survey. *Sensors* **2018**, *18*, 2796. [CrossRef]
32. Rizvi, S.; Kurtz, A.; Pfeffer, J.; Rizvi, M. Securing the Internet of Things (IoT): A Security Taxonomy for IoT. In Proceedings of the 2018 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/12th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE), New York, NY, USA, 1–3 August 2018; pp. 163–168.
33. Ali, I.; Sabir, S.; Ullah, Z. Internet of Things Security, Device Authentication and Access Control: A Review. *arXiv* **2019**, arXiv:1901.07309.
34. Zhang, T.; Antunes, H.; Aggarwal, S. Defending Connected Vehicles against Malware: Challenges and a Solution Framework. *IEEE Internet Things J.* **2014**, *1*, 10–21. [CrossRef]
35. Lea, P. *Internet of Things for Architects: Architecting IoT Solutions by Implementing Sensors, Communication Infrastructure, Edge Computing, Analytics, and Security*, 1st ed.; Packt Publishing: Birmingham, UK, 2018.
36. Sidhu, S.; Mohd, B.J.; Hayajneh, T. Hardware Security in IoT Devices with Emphasis on Hardware Trojans. *J. Sens. Actuator Netw.* **2019**, *8*, 42. [CrossRef]
37. Levi, M.; Allouche, Y.; Kontorovich, A. Advanced Analytics for Connected Car Cybersecurity. In Proceedings of the 2018 IEEE 87th Vehicular Technology Conference (VTC Spring), Porto, Portugal, 3–6 June 2018.
38. Huang, J.; Zhao, M.; Zhou, Y.; Xing, C.-C. In-Vehicle Networking: Protocols, Challenges, and Solutions. *IEEE Netw.* **2018**, *33*, 92–98. [CrossRef]
39. Macher, G.; Armengaud, E.; Brenner, E.; Kreiner, C. Threat and Risk Assessment Methodologies in the Automotive Domain. *Procedia Comput. Sci.* **2016**, *83*, 1288–1294. [CrossRef]
40. Smith, C. *The Car Hacker's Handbook: A Guide for the Penetration Tester*; No Starch Press: San Francisco, CA, USA, 2016.
41. Carsten, P.; Andel, T.R.; Yampolskiy, M.; McDonald, J.T. In-vehicle networks: Attacks, vulnerabilities, and proposed solutions. In Proceedings of the 10th Annual Cyber and Information Security Research Conference, Oak Ridge, TN, USA, 6–8 April 2015; pp. 1–8.
42. Checkoway, S.; McCoy, D.; Kantor, B.; Anderson, D.; Shacham, H.; Savage, S.; Koscher, K.; Czeskis, A.; Roesner, F.; Kohno, T. Comprehensive Experimental Analyses of Automotive Attack Surfaces. In Proceedings of the USENIX Security Symposium, San Francisco, CA, USA, 8–12 August 2011; p. 2021.
43. Li, S. *Securing the Internet of Things*; Syngress: Maryland Heights, MO, USA, 2017; pp. 69–95.
44. Tu, S.; Waqas, M.; Huang, F.; Abbas, G.; Abbas, Z.H. A revocable and outsourced multi-authority attribute-based encryption scheme in fog computing. *Comput. Netw.* **2021**, *195*, 108196. [CrossRef]
45. Tu, S.; Waqas, M.; Rehman, S.U.; Mir, T.; Abbas, G.; Abbas, Z.H.; Halim, Z.; Ahmad, I. Reinforcement Learning Assisted Impersonation Attack Detection in Device-to-Device Communications. *IEEE Trans. Veh. Technol.* **2021**, *70*, 1474–1479. [CrossRef]

46. Tu, S.; Waqas, M.; Meng, Y.; Rehman, S.U.; Ahmad, I.; Koubaa, A.; Halim, Z.; Hanif, M.; Chang, C.-C.; Shi, C. Mobile fog computing security: A user-oriented smart attack defense strategy based on DQL. *Comput. Commun.* **2020**, *160*, 790–798. [CrossRef]
47. Tanveer, M.; Abbas, G.; Abbas, Z.H.; Waqas, M.; Muhammad, F.; Kim, S. S6AE: Securing 6LoWPAN Using Authenticated Encryption Scheme. *Sensors* **2020**, *20*, 2707. [CrossRef] [PubMed]
48. Waqas, M.; Tu, S.; Rehman, S.U.; Halim, Z.; Anwar, S.; Abbas, G.; Abbas, Z.H.; Rehman, O.U. Authentication of Vehicles and Road Side Units in Intelligent Transportation System. *Comput. Mater. Contin.* **2020**, *64*, 359–371. [CrossRef]
49. Di Mauro, M.; Galatro, G.; Liotta, A. Experimental Review of Neural-Based Approaches for Network Intrusion Management. *IEEE Trans. Netw. Serv. Manag.* **2020**, *17*, 2480–2495. [CrossRef]
50. Syzdykbayev, M.; Hajari, H.; Karimi, H.A. An Ontology for Collaborative Navigation Among Autonomous Cars, Drivers, and Pedestrians in Smart Cities. In Proceedings of the 2019 4th International Conference on Smart and Sustainable Technologies (SpliTech), Split, Croatia, 18–21 June 2019; pp. 1–6.
51. Klotz, B.; Datta, S.K.; Wilms, D.; Troncy, R.; Bonnet, C. A Car as a Semantic Web Thing: Motivation and Demonstration. In Proceedings of the 2018 Global Internet of Things Summit (GIoTS), Bilbao, Spain, 4–7 June 2018; pp. 1–6.
52. Klotz, B.; Troncy, R.; Wilms, D.; Bonnet, C. VSSo: A Vehicle Signal and Attribute Ontology. In Proceedings of the 9th International Semantic Sensor Networks Workshop, Monterey, CA, USA, 9 October 2018.
53. Colace, F.; De Santo, M. Ontology for E-Learning: A Bayesian Approach. *IEEE Trans. Educ.* **2009**, *53*, 223–233. [CrossRef]
54. Colace, F.; De Santo, M.; Vento, M. A MultiExpert Approach for Bayesian Network Structural Learning. In Proceedings of the 2010 43rd Hawaii International Conference on System Sciences, Honolulu, HI, USA, 5–8 January 2010; pp. 1–11.
55. Colace, F.; De Santo, M.; Lombardi, M.; Pascale, F.; Santaniello, D.; Tucker, A. A Multilevel Graph Approach for Predicting Bicycle Usage in London Area. In Proceedings of the 4th International Congress on Information and Communication Technology. Advances in Intelligent Systems and Computing, London, UK, 25–26 February 2019; Volume 1027. [CrossRef]
56. Casillo, M.; Coppola, S.; De Santo, M.; Pascale, F.; Santonicola, E. Embedded Intrusion Detection System for Detecting Attacks over CAN-BUS. In Proceedings of the 2019 4th International Conference on System Reliability and Safety (ICSRS), Rome, Italy, 20–22 November 2019; pp. 136–141.
57. Lombardi, M.; Pascale, F.; Santaniello, D. EIDS: Embedded Intrusion Detection System using Machine Learning to Detect Attack over the CAN-BUS. In Proceedings of the 30th European Safety and Reliability Conference and the 15th Probabilistic Safety Assessment and Management Conference, Venice, Italy, 21–26 June 2020. [CrossRef]
58. Colace, F.; Khan, M.; Lombardi, M.; Santaniello, D. A Multigraph Approach for Supporting Computer Network Monitoring Systems. In Proceedings of the 5th International Congress on Information and Communication Technology, London, UK, 20–21 February 2021; pp. 470–477.
59. Castiglione, A.; Palmieri, F.; Colace, F.; Lombardi, M.; Santaniello, D.; D'Aniello, G. Securing the internet of vehicles through lightweight block ciphers. *Pattern Recognit. Lett.* **2020**, *135*, 264–270. [CrossRef]
60. Dosovitskiy, A.; Ros, G.; Codevilla, F.; Lopez, A.; Koltun, V. CARLA: An open urban driving simulator. *arXiv* **2017**, arXiv:1711.03938, preprint.
61. Mhetre, V.; Nagar, M. Classification based data mining algorithms to predict slow, average and fast learners in educational system using WEKA. In Proceedings of the 2017 International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 18–19 July 2017; pp. 475–479.
62. Ertam, F.; Aydin, G. Data classification with deep learning using Tensorflow. In Proceedings of the 2017 International Conference on Computer Science and Engineering (UBMK), Antalya, Turkey, 5–7 October 2017; pp. 755–758.
63. Lee, H.; Jeong, S.H.; Kim, H.K. OTIDS: A Novel Intrusion Detection System for In-vehicle Network by Using Remote Frame. In Proceedings of the 2017 15th Annual Conference on Privacy, Security and Trust (PST), Calgary, AB, Canada, 28–30 August 2017; pp. 57–5709. [CrossRef]
64. Di Mauro, M.; Galatro, G.; Fortino, G.; Liotta, A. Supervised feature selection techniques in network intrusion detection: A critical review. *Eng. Appl. Artif. Intell.* **2021**, *101*, 104216. [CrossRef]
65. Erhan, L.; Ndubuaku, M.; Di Mauro, M.; Song, W.; Chen, M.; Fortino, G.; Bagdasar, O.; Liotta, A. Smart anomaly detection in sensor systems: A multi-perspective review. *Inf. Fusion* **2021**, *67*, 64–79. [CrossRef]
66. Pascale, F.; Adinolfi, E.A.; Avagliano, M.; Giannella, V.; Salas, A. A Low Energy IoT Application Using Beacon for Indoor Localization. *Appl. Sci.* **2021**, *11*, 4902. [CrossRef]