Vehicular Communications ••• (••••) ••••••

Contents lists available at ScienceDirect

Vehicular Communications

www.elsevier.com/locate/vehcom

Detection, control and mitigation system for secure vehicular communication ☆

Carlos Hidalgo [a,*], Myriam Vaca [a], Mateusz P. Nowak [b], Piotr Frölich [b], Martin Reed [c], Mays Al-Naday [c], Asterios Mpatziakas [d], Aikaterini Protogerou [d,e], Anastasios Drosou [d], Dimitrios Tzovaras [d]

[a] Tecnalia Research and Innovation, Parque Científico y Tecnológico de Bizkaia, Geldo Auzoa, Edif. 700, 48160 Derio, Bizkaia, Spain
[b] Institute of Theoretical and Applied Informatics, PAS, Baltycka 5, 44-100 Gliwice, Poland
[c] Electronic Systems Engineering Department, University of Essex, Colchester, C04 3SQ, Essex, United Kingdom
[d] Information Technology Institute, Centre for Research and Technology Hellas, P.O.Box 60361, 6th km Harilaou, Thermi, 57001, Thessaloniki, Greece
[e] Applied Informatics Department, University of Macedonia, 156 Egnatia str., Thessaloniki, Greece

ARTICLE INFO

Article history:
Received 29 April 2021
Received in revised form 16 September 2021
Accepted 12 October 2021
Available online xxxx

Keywords:
C-ITS
Secure communications
SerIoT system
Fleet management
Smart intersection

ABSTRACT

The increase in the safety and privacy of automated vehicle drivers against hazardous cyber-attacks will lead to a considerable reduction in the number of global deaths and injuries. In this sense, the European Commission has focused attention on the security of communications in high-risk systems when receiving a cyber-attack such as automated vehicles. The project SerIoT comes up as an possible solution, providing a useful open and reference framework for real-time monitoring of the traffic exchanged through heterogeneous IoT platforms. This system is capable of recognize suspicious patterns, evaluate them and finally take mitigate actions. The paper presents a use case of the SerIoT project related to rerouting tests in vehicular communication. The goal is to ensure secure and reliable communication among Connected Intelligent Transportation Systems (C-ITS) components (vehicles, infrastructures, etc) using the SerIoT's system capabilities to detect and mitigate possible network attacks. Therefore, fleet management and smart intersection scenarios were chosen, where vehicles equipped with On Board Units (OBU) interact with each other and Road Side Units (RSU) to accomplish an optimal flow of traffic. These equipments use the SerIoT systems to deal with cyber-attacks such as Denial of Service (DoS). Tests have been validated in different scenarios under threats situations. It shows the great performance of the SerIoT system taking the corresponding actions to ensure a continuous and safety traffic flow.

1. Introduction

Cyber threats impact every area of life that depends on exchange of information. The threat is increasing to critical levels, particularly with the rapid adoption of the Internet of Things (IoT), leading to a 100% increase in IoT threats from 2019 to 2020 [38]. The increase in cyber-physical systems means that a cyber-attack not only affects the realm of pure information, but can also have effects in physical environments possibly endangering human life and causing material damage. Such a danger poses a severe threat within the intensely developing field of Connected Autonomous Vehicles (CAV). For example, a malfunction in the steering systems of a vehicle, controlled by digital devices, can lead to severe injuries or even death. Hence, there is a strong focus on cyber-security in autonomous vehicle projects focusing on V2X communications where the vehicle can communicate with everything i.e. vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I), vehicle-to-network (V2N), vehicle-to-cloud (V2C), vehicle-to-device (V2D) and vehicle-to-pedestrian (V2P).

☆ This document is the results of the research project funded by the European Commission under Grant Agreement no. 780139.
* Corresponding author.
E-mail addresses: carlos.hidalgo@tecnalia.com (C. Hidalgo), myriam.vaca@tecnalia.com (M. Vaca), mateusz@iitis.pl (M.P. Nowak), pfrohlich@iitis.pl (P. Frölich), mjreed@essex.ac.uk (M. Reed), mfhaln@essex.ac.uk (M. Al-Naday), ampatziakas@iti.gr (A. Mpatziakas), k.protogerou@iti.gr (A. Protogerou), drosou@iti.gr (A. Drosou), Dimitrios.Tzovaras@iti.gr (D. Tzovaras).

https://doi.org/10.1016/j.vehcom.2021.100425
2214-2096/

Please cite this article as: C. Hidalgo, M. Vaca, M.P. Nowak et al., Detection, control and mitigation system for secure vehicular communication, Vehicular Communications, https://doi.org/10.1016/j.vehcom.2021.100425

Vehicular communication systems bring the vehicles closer to IoT devices leading to a number of different threats [9]; example attacks include: remotely controlling the vehicles [35]; accessing and changing the CAN bus messages [4]; denying message exchange by simultaneously sending messages that affects the network [3]; and others that have been carried out in the past.

This article presents the SerIoT system, developed within the H2020 project of the same name [50], as a means to mitigate cyber attacks against CAV environments. The SerIoT project provides a number of security solutions including: protection of the core networks interconnecting IoT devices; IoT-specific solutions for anomaly detection (AD); early mitigation through re-routing to specialized analytic devices such as honeypots; and, automated mitigation for autonomous vehicles. The following article takes a closer look at these solutions.

Specifically, this work utilizes the secure network architecture provided by the SerIoT project to facilitate resilient vehicular communications in non-time critical scenarios. The work exploits the capabilities of anomaly detection and mitigation combined with intelligent routing to provide dynamic re-routing around attacked resources, within a time-frame that ensures seamless vehicular operation. We evaluate the system performance using realistic simulation environments, an international testbed and experimental settings, using real vehicles operated on the premises of Tecnalia. Our evaluation shows the system's ability to detect anomalous behavior and act upon it within seconds, sufficiently fast to support non-time critical vehicle maneuvers. The article is structured as follows. In section 2, it presents the state of the art in cyber-security for communication with autonomous cars. The next, section 3, is devoted to the system architecture of SerIoT and describes the basic features that distinguish a SerIoT network from other types of networks, with particular attention paid to multi-agent anomaly detection. Then, sections 4 and 5 describe the details of the automated driving framework and specific Use Cases in which the SerIoT project solutions were tested. The article finally ends by presenting the results and conclusions.

2. State of the art

Connected autonomous vehicles generate, exchange and process data at intensive rates by virtue of incorporating a large number of IoT sensors and actuators. This section reviews the state of the art literature in vehicular communications and cybersecurity, including related work in anomaly detection and mitigation solutions.

2.1. Vehicular communication

Vehicular communications are generally developed as part of Cooperative Intelligent Transportation Systems (C-ITS) as communications between vehicles and other agents improving the efficiency and security of the control systems. Vehicular communication networks are flexible communication systems that guarantee optimal real-time transmissions between devices in different environments (i.e. cities, roads, ports, etc.), exchanging information, such as safety warnings, traffic information, orders and movements to be carried out, etc [43]. As a cooperative approach, vehicle communication systems can be more effective in avoiding accidents and traffic congestion than if each vehicle tries to solve these problems individually [52].

The vehicular communication standard defined in the United States is Dedicated Short Range Communications (DSRC) and Europe defines its standard as Intelligent Transport System (ITS-G5). Colloquially and globally, this technology can be defined as vehicular Wi-Fi, since it is based on IEEE 802.11.p, but with the requirement for low latency communication, essential for safety applications. The final draft of the standard was approved in 2009 [23]. It is a short/medium-range wireless communication channel that works at 5.9 GHz, with a bandwidth of 75 MHz and an approximate range of 1000 m.

The vehicular communication systems that are commonly used in the connection between vehicles, infrastructure, and operators are made up of two main components: the OBU (On-Board Unit) which is in the vehicle, and the RSU (Road-Side Unit) which is in the infrastructure. This technology has been validated after numerous field tests with commercial equipment. For example, one of the largest pilots has been funded by the U.S. Department of Transportation (USDOT) for 45 million USD [58]. Various applications have been implemented at different sites (Tampa, Wyoming, and New York). In Tampa, in collaboration with Hyundai, data was collected during 18 months of testing, during which the system warned drivers of the following hazards: accessing an oncoming highway (14 cases); potential collisions with the tram (9 cases); speeding (1,500 cases per month) [56].

The largest pilot is in New York, currently having more than 3,000 cars, buses, and fleet vehicles. However, due to the conditions presented in the city (i.e. large buildings), the signal from satellite navigation systems is not reliable. Therefore, these scenarios emerge as a good example to show the benefits of the deployment of RSUs on the road, improving the positioning of the vehicles [34]. Other activities carried out by the USDOT on automated vehicles can be found at [59].

On the other hand, the CAR2CAR Communication Consortium (C2C-CC) is led by European and international vehicle manufacturers, equipment suppliers, engineering companies, road operators and research institutions. This initiative contributes to the development and specification of robust and reliable solutions that allow technologies driven by innovation, thereby fostering concepts of cooperation between the road users and with the road infrastructure sharing information (V2X communication). This is based on the development of a European industry standard for vehicular communication systems, active safety application prototyping and demonstrations [45]. Likewise, in Europe, several projects have been developed based on vehicle communications research [52] such as MARSS-5G [12] and PRESERVE [13]. An analysis of the EU's competitive positioning related to challenges and opportunities of IoT for connected and autonomous vehicles and the key actors is presented in [62], where it presents relevant stakeholders and a clear overview of the current and future landscape of these technologies in the EU.

Therefore, as modern vehicles are capable of connecting to an external infrastructure and V2X communication matures, the necessity to secure communications is increasing due to the real risk that vehicles are subjected to cyber-attacks that target vehicular communications and compromise the security on the road. Thus, it is necessary from a design point of view, to profile threats and structure a mitigation plan based on best practices for cybersecurity. An overview of this relationship between vehicle safety and its functions, and cybersecurity, has been previously studied by Nilsson et al. [37].

2.2. Cybersecurity

Cybersecurity itself is a huge topic and there are many published works in the area. Specifically for IoT, useful survey papers report the relevant literature in the general IoT area [24] and more specifically for Vehicular systems [10]. Additionally, IoT systems are increasingly reliant on Cloud and Fog computing so that this area of cybersecurity is also an important consideration [54].

While there are specific cybersecurity solutions discussed in this paper and many more in the wider literature, it is vital that the cybersecurity solution itself remains secure. This is particularly important as security infrastructure often has access to critical systems and sensitive data as these are the most important to protect. Indeed, there have been recent cases of security systems themselves being the cause of serious vulnerabilities [55]. Traditional network security architectures have relied upon a so called "moats-and-castles" approach with trusted zones kept separate by systems such as firewalls. However, more recent approaches have proposed using a *zero-trust* architecture [48] which moves to a model where every system is treated as untrusted until it has been authenticated and all connectivity is authenticated, audited and encrypted. For a security system, such a zero-trust approach is to be preferred and increasingly becoming the architecture of choice [40]. Within such a zero-trust architecture, as selected for use in SerIoT, the auditing of connectivity is often used as a mechanism for anomaly detection which leads to automated mitigation strategies as discussed in the following subsections.

2.3. Anomaly detection

In recent years, Machine Learning (ML), Neural Networks (NN) and Deep Learning (DL) algorithms have been widely used to detect malicious traffic and classify network attacks. Various examples that use Deep NN can be found in the literature such as the intrusion detection system (IDS) presented in [25] or the IDS method presented in [61], which used low-level spatial features of network traffic.

Multiple studies have proposed graph-based solutions to detect abnormalities, i.e. representing the IoT entities and communication links as nodes and edges of a graph network. Such a graph-based research for anomaly detection was presented in [57], where the web traffic (i.e. web requests and connections to web servers) was modeled using graphs to point out probable malicious-clients (e.g. botnet members). An Attention-based temporal Graph Convolutional Network (GCN) model was developed in [66], where temporal features were used to identify anomalous edges of the graph within dynamic graphs. Another dynamic graph anomaly detection was developed in [65], where deep auto-encoders in conjunction with clustering techniques were employed on the nodes of the network to identify anomalies in real-time.

Another scheme utilizing anomaly detection using a Graph Neural Network (GNN) was proposed in [5]. In this study, the interconnections of the involved nodes were taken into account in the computed adjacency matrix, along with various topological characteristics of the graph. Edge-level anomaly detection methods were proposed in [11] and [51]. In these cases, the malicious edges are determined using the density of sub-graphs along with structural, temporal, and content feature extraction and a greedy search mechanism, respectively. An anomaly node detection model was proposed in [2] applying a probabilistic approach of detection.

A detailed overview of Recurrent, Convolutional, Auto-encoders and Spatial-Temporal Graph neural network approaches was presented in [63] resulting in a taxonomy of GNNs mechanisms applied in several IoT application domains. In [17] and [46] an attack detection scheme was proposed where agents implementing a GNN model offered both localized monitoring in IoT networks and feature exchange in a distributed synergistic detection mechanism. This approach will be used in this paper and is briefly presented in section 3.4.

Enhancing security mechanisms in the modern Internet of Vehicles field is a challenging task researched extensively in several studies. Most approaches utilize techniques based on ML and DL: An example of DL use is presented in [22], where the authors propose a long short-term memory-based (LSTM) detection system against in-vehicle CAN bus network attacks such as DoS, Fuzzing and Spoofing, whereas in [29] a DL approach is deployed on the side of the Cloud to apply the detection process, while offloading the burden of DL heavy computation to a server with more computational resources. They deal with DoS, command injection and malware attack examples, using time-series data as input to their proposed neural network. An example of ML use is presented in [64], a multi-tier algorithm is developed, combining signature-based and zero-day attack detection. The developed schema uses two ML stages for data pre-processing and feature engineering and four learning tiers. The algorithm is evaluated against offline datasets. An extensive review of the subject of anomaly detection in V2X is available in [47].

2.4. Mitigation approaches

Threat mitigation of cyber-systems, including IoT systems, can be classified in one of the following two approaches: 1) design tools to mitigate a single attack type e.g. a Distributed Denial of Service (DDoS) [68] which can be handled using Software Defined Networking (SDN), examples of such approaches can be found in various review studies such as [1] and [31]; 2) select mitigation actions or countermeasures by employing mechanisms that optimize one or more metrics related to the system.

Solutions that belong to the second category can additionally be divided into three separate classes as described below. The first includes approaches that measure the values of one or more measures but offer no automated mechanism or response to the threats faced by the system. In this case the operator must make a manual choice. Such a mechanism is presented in [20].

The second class involves the automated mitigation of attacks using thresholds based on the values of the metrics. Based on predefined scenarios and values, the system chooses predefined actions to react to the threats. An example of a threshold-based tool is available in [27]. Such approaches can be difficult to scale up.

The last class includes approaches where the selection of mitigation actions is based on the optimization of the value of one or more metrics. One example is [67] where the authors select mitigation actions by trying to minimize the cost required to deploy. Often, Evolutionary Algorithms are used to solve these optimization problems as in [6].

Concerning the mitigation of attacks against V2X communication networks, the majority of past works focused on approaches that countered a single attack type. In [53], the authors present an exhaustive list of threats against V2X networks along with countermeasures against them, all based on utilizing an SDN based network architecture, similar to the solutions proposed in this work. In [39], researchers present a comprehensive presentation of attacks against vehicular communication networks along with schemes to mitigate them. They distinguish between three broad approaches, in which the mitigation schemes depend on the attack detection method: 1) intrusion detection systems (anomaly or signature based) such as the one presented in Section 3.4, 2) secure routing which in the case presented are handled by the SerIoT decision system presented in 3.2 and finally 3) Cryptography based methods.

3. Security SerIoT framework

The SerIoT system aims to provide a useful and reference framework for real-time monitoring of the traffic exchanged through IoT platforms within the IoT network to recognize suspicious patterns, to evaluate them, and finally to decide on the detection while offering parallel mitigation actions.

3.1. SerIoT system architecture

The secure network communication system for IoT devices developed within the SerIoT project (referred to as the SerIoT network hereafter) consists of a number of components. Its general architecture is shown in Fig. 5 of the document [26], where several basic parts are presented:

1. Network domain, based on SDN technology, equipped with path optimization mechanisms according to different criteria that can be applied simultaneously; the network domain includes SerIoT Fog components.
2. Management domain and functions, including Anomaly Detection modules, monitors network traffic to ensure the security of network elements and clients.
3. User domain or network edge elements, including Honeypots and Autopolicy modules.

Components belonging to domains 1 and 2 are included in the solution proposed in the paper, and the architecture of the pilot system presented in the paper is shown in Fig. 1. The components are described in the subsequent sections.

3.2. The SerIoT decision system

The SerIoT network is based on SDN technology. The use of SDN, in which the data plane and the control plane are separated, made it possible to combine the concept of Cognitive Packets, developed in [18], with a network based on a standard four-layer architecture using the TCP/IP protocol stack. The combination allows the advantages of Cognitive Packet Networks to be used in industry-standard networks. Cognitive Packets provide the network with real time measurements of the state of the network without the need of estimation or statistical evaluation. Cognitive Packets simply traverse the network and, by sampling the performance as the progress, report the actual state to the Cognitive Network Map in the controller.

The SerIoT network, using SDN and the Openflow management protocol [32], extends its mechanisms with two essential, closely related elements. SDN is managed centrally, by an SDN controller located outside the data plane of the network. The controller, controlling the data plane devices (called switches or forwarders), decides on the paths along which the data belonging to the individual connections (flows) are routed in the network, and on the admission of devices to communicate in the network. To make the right decisions, SDN needs a decision-making module and data on the state of the network. The smart decision module referred to as the SerIoT Routing Engine (SRE) is based on the Random Neural Network approach. The SRE is supported by a data-gathering solution – a cognitive packets (CP) mechanism, integrated into the SerIoT network to provide data for the SRE [16].

Cognitive Packets are special-purpose packets, designed to measure link latency in the network. CP works in two modes. The classic approach uses time-synchronized devices, and the timestamp of every device is included in the payload of the CP. The packet wanders through the network, along the actual paths for given flows and also alternative paths, measuring end-to-end delay on the path, and sending the data to the SRE. When time-synchronization between neighboring nodes in the network cannot achieve the desired accuracy an alternative way of measurement is used by measuring RTT to the neighboring node and sending this directly to the controller [15] [16].

As a SerIoT SDN Controller, we use standard SDN software, ONOS [41]. On top of it, the SerIoT Routing Engine (SRE) is built, using Random Neural Networks for routing decisions. For each forwarder in the data plane, an individual neural network is created, consisting of the number of neurons equal to the number of forwarder outputs. The networks are then trained with the use of the Reinforcement Learning algorithm. The training is continuous, as new incoming data from the cognitive packets train the Neural Networks making them reacting relevantly to the current situation in the network without losing the history. Thanks to the training, the most stimulated neuron in each RNN indicates the output by which a given packet should be sent.

Reinforcement learning is based on the Goal Function, $G()$, formulated in (1), which the neural network tries to minimize [18]. Also, the RNNs take decisions aimed at minimizing the Goal function. $G()$ may include one or many components. Typical usage includes QoS optimization and specifically reduction in average latency. In SerIoT we added additional components for security, energy, and privacy. Thus, the Goal function for a given flow and given path has the form:

$$G(f,p) = \alpha Q(f,p) + \beta S(f,p) + \gamma E(f,p) + \pi R(f,p), \quad (1)$$

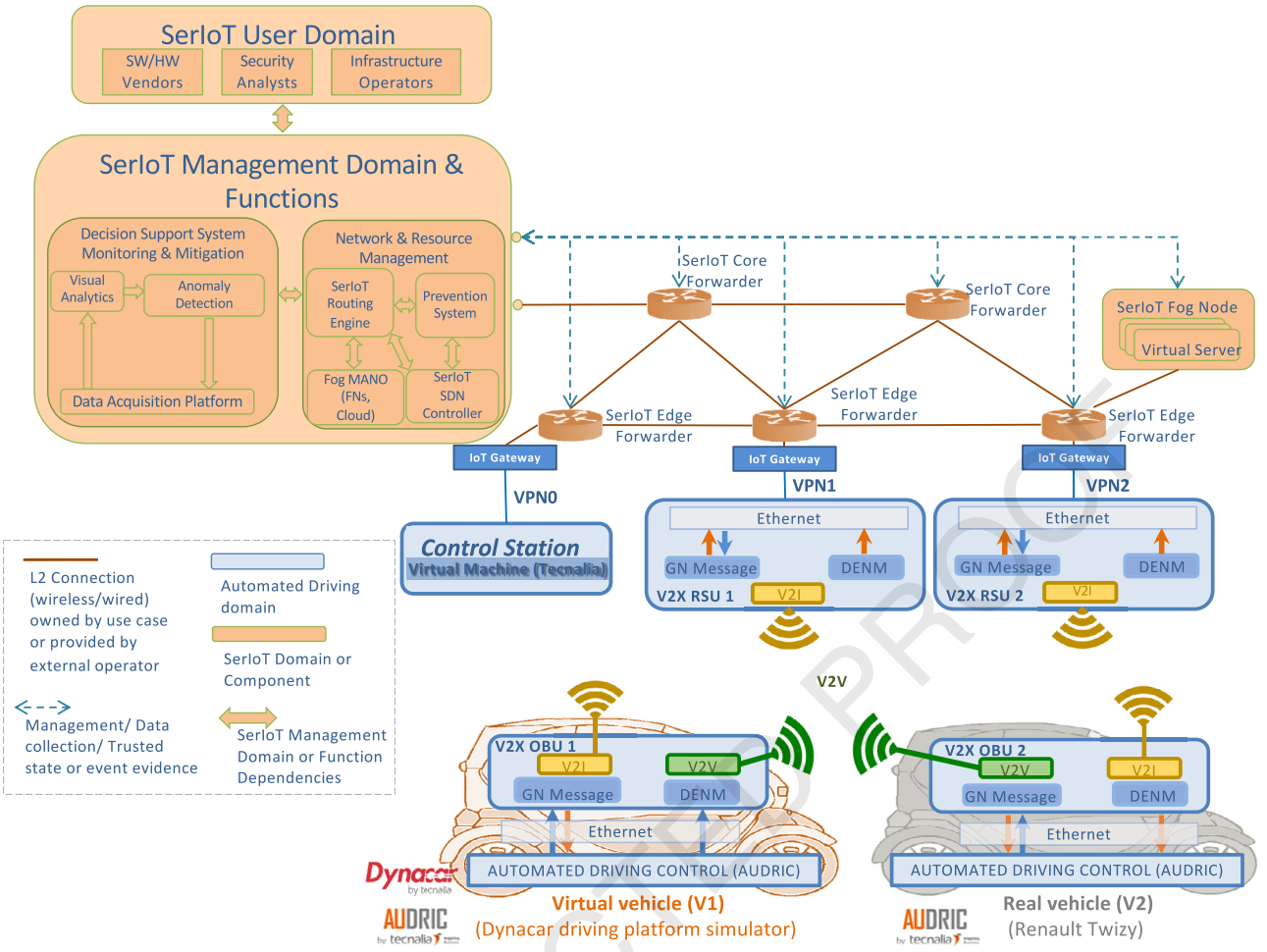


Fig. 1. Fleet management S-UC scheme.

where $Q(f,p)$ denotes QoS value - latency of flow f on the path p. $S(f,p)$ is a security function or trust factor; the trust level is given by external Anomaly Detection modules. $E(f,p)$ is the energy used by devices in the path p when transporting the flow f. Last is $R(f,p)$, which takes the value 0 when the path p matches the privacy policy for the flow f, and 1 — otherwise.

It should be noted, that not all four components of (1) need to be implemented in every SerIoT installation. These components, which are included in the implemented Goal function, are taken into account in the routing decisions of the SerIoT SDN Controller.

3.3. External mitigation

Except for decisions of the SRE, which are based on a mixture of weighted factors, where security assessment is only one of them, the SerIoT controller is able to accept decisions from external Monitoring & Mitigation components. The controller's API allows external, authorized entities to send mitigation orders to change the routing of the traffic; for example: adding the sender to a block-list in the case of malicious traffic; or deflecting the traffic to a backup node in case of failure or denial of service; or, deflecting traffic to an analytic module e.g. a honeypot. This configuration requires an external entity to not only judge the anomaly level but also take decisions (automatically or by a human operator) that are implemented by the SerIoT controller. Such an entity is described in Section 3.5.

3.4. Multi-agent anomaly detection

Supporting co-operation among the interconnected network devices, the proposed Anomaly Detection method (AD) offers several of the advantages generally provided by multi-agent systems. To address multiple cyber-threats, such as DOS attacks, Port Scan and SSL attacks. The AD module engages a network of agents, from which it determines the network of involved devices, their inter-communication links, and significant traffic characteristics. Each agent is installed on a monitoring node of the network to implement a Graph Neural Network (GNN) algorithm.

The algorithm takes as input traffic statistics and classifies the node as normal or abnormal reporting a probability value. Traffic information deriving from neighboring agents is also fed to the algorithm, which in turn results in a probability score describing the neighbor's status. To this end, each node and edge is associated with a feature vector. The collection of feature vector parameters is captured for a successive time slot (denoting a time-window) thus each feature vector's values captures traffic statistics within such time (see Table 1).

Table 1
Features used for the Anomaly Detection.

Anomaly Detector Network Features	
Time-related entities	Start time, Duration
Flow-related identifiers	Src IP, Dst IP, Dir, Label
Payload-related entities	Tot Fwd Pkts, Tot Fwd Bytes

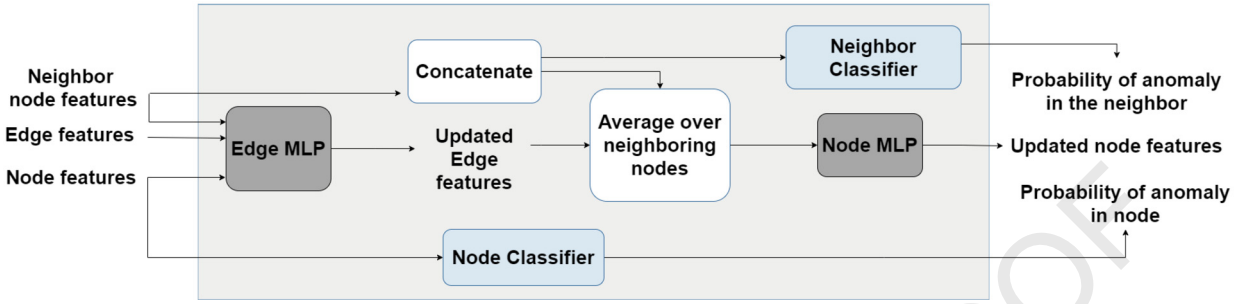


Fig. 2. The Graph Neural Network architecture comprising two Multi-Layer Perceptrons both for Edge and Node Deep Neural Network implementation. Edge Deep Neural Network takes as input the features of neighboring nodes and updates the edge feature vector, whereas the Node Deep Neural Network is fed with the adjacent updated edge features so as to update the feature vector of each particular node.

Specifically, to meet the needs of the Smart Intersection scenario of this article, a network monitoring tool has been developed based on the pmacct and nDPI [8] open-source monitoring tools to acquire Netflow v5 traffic characteristics and assist in the classification of the monitoring nodes.

3.4.1. Node and edge feature vectors

Assigning attributes to the network nodes requires the information to be captured in a flow-wise manner. Afterward, the detection algorithm averages the connection's duration, sent/received packets and sent/received bytes information per node and connection link, thus generating the GNN input node and edge feature vector accordingly.

3.4.2. GNN model architecture

The Graph Neural Network architecture comprises two Multi-Layer Perceptrons for Edge and Node Deep Neural Network implementations. (Fig. 2.) The Edge Deep Neural Network takes as input the features of neighboring nodes and updates the edge feature vector, whereas the Node Deep Neural Network is fed with the adjacent updated edge features to update the feature vector of each particular node. Probability scores are reported to spread the knowledge regarding the status of the environment (neighboring nodes) and the status of a node itself. The most significant functions denoting the feature vectors of the edges and nodes are explained below.

Given a network represented as a Graph structure $G(V, E)$ with nodes V interconnected by edges E the edges interconnecting them, we define a node feature vector for every node and an edge feature vector to associate with each edge on the graph. We assume neighboring nodes denoted as i and j. Traffic statistic measurements that populate their feature vectors are collected in successive time slots $[(t-1)T, tT]$, where t is the slot's index and T its length.

Therefore we denote equation (2) below in which an edge's current feature vector $e_{ij}^{k,t}$ along with these of the nodes it interconnects, are used to update its features, as $x_j^{k,t}$ is the t-th value of x_j (feature vector of node j) in the k-th time slot and similarly $x_i^{k,t}$ is the t-th value of x_i in the k-th time-slot (feature vector of node i).

$$e_{ij}^{k+1,t} \rightarrow EDNN(x_j^{k,t}, x_i^{k,t}, e_{ij}^{k,t}) \tag{2}$$

Consequently, a node's feature values along with the feature values of the nodes it communicates with and the values of the associated edges, are used to update its vector as follows

$$x_j^{k+1,t} \rightarrow NDNN(e_j^{t-1}, x_j^{k,t}) \tag{3}$$

Following this procedure, the average value of the features, associated with the edges between a node and its neighbors as well as the average value of features associated with the nodes neighboring a specific node is calculated. The probability score of node i and j is then reported.

3.5. Mitigation engine

The scope of the Mitigation Engine module is to provide mitigation actions against threats or when an attack to the network is detected, based on AI. The main functionality of this component is the automated decision of the mitigation action based on the output of

Table 2
Abstract OpenFlow rules for redirect action.

Source	Destination	Instructions
Host A	Host B	IP destination = host C IP, MAC destination = host C MAC
Host C	Host A	IP source = host B IP, MAC source = host B MAC

the Anomaly Detection module. The module was implemented in Python, while the Pytorch framework is used for the Pointer Networks [60].

For each of the attacks or threats detected by the system, multiple mitigation actions might be available but a single action must be chosen. An AI solution was developed, using reinforcement learning via a DNN architecture called Pointer Networks [60]. We modified the architecture to select a set of countermeasures to be applied, based on optimizing multiple security-related KPIs while taking into account constraints. Separate DNNs are used to solve the problem separately for each KPI. A decomposition method called Normalized Normal Constraint [33] uses the initial solutions to transform the problem to a single objective, also solved by a Pointer network. This solution is transformed back to result in the Pareto solution set for the entire multi-objective optimization problem.

An additional mechanism is implemented for the RSU units. When an attack is detected on them, the system checks a predefined list of all RSUs: if an uncompromised unit is available, all traffic is redirected to it. The mitigation actions are finally sent to the SerIoT SDN Controller (see Section 3.6) to be applied to the network.

3.6. Mitigation enforcement in the SerIoT controller

Upon receiving a mitigation query, or making an autonomous decision, the SerIoT Routing Engine (SRE) utilizes the Mitigation Enforcement to provide the requested mitigation for the network. Since the SerIoT core network is an SDN network the SRE utilizes the OpenFlow protocol (version >= 1.3) to enable mitigation actions. The SRE can enforce several mitigation actions – only those used in experiments and use cases reported in this paper will be described here in detail.

Block - this mitigation action is performed on-forwarder or many forwarders to which specified endpoints (source and destination) are connected to. This mitigation action blocks all traffic on a flow described by source and destination (protocol IP address and port). If there are no such devices in the network this mitigation action is enforced on all of the forwarders in the network. This mitigation action is enforced until time t which is specified by the administrator of the network. After time t, if the device has stopped being malicious, it can connect to the network again. The identified flow is translated into the OpenFlow rules with the action specified to drop any incoming packets that are fulfilling that rule. The OpenFlow **match fields** contain the source and destination of specified endpoints - e.g. (OFPXMT_OFB_ETH_TYPE = IPv4, OFPXMT_OFB_IPV4_SRC = 10.0.1.2). For all possible fields for the OpenFlow rule, see page 42 of [42]. The **priority** is set to highest possible value (59999 in ONOS implementation). The **instructions** field is set to DROP and the **timeout** field is set to t. An OpenFlow rule prepared in such a way guarantees that traffic matching the specified flow will be dropped on the first forwarder it reaches. All fields of the OpenFlow rules are based on the documentation for the OpenFlow Switch in version 1.3.0 [42].

Block-list - this mitigation action is very similar to **Block** but the connection for described flow is severed until the flow is allow-listed unlike the **Block** action where the connection is resumed after the time t.

Block-list MAC - this mitigation action forbids certain MAC addresses from entering the SerIoT network. It's installed on all of the forwarders and is valid until the MAC address is allow-listed. This is used to mitigate the attacks based on spoofing the source IP address.

Deflect (reroute) - this mitigation action utilizes OpenFlow version 1.3 and higher. This action is designed to reroute traffic between **host A** and **host B** to **host C** in that way that **host A** thinks it is connected to the **host B**. This action can be used in several different ways (e.g. to redirect an attacker to a honeypot). The mitigation action is translated into the OpenFlow rules in such a way that the **match fields** are the same as host A, B, and C. The **instruction** field is created in such a way that the rule is masking the real destination by assigning and reassigning the IP and MAC addresses (see Table 2). This mitigation action can transparently swap the destination host of the flow. The **priority** must be set to be of value $max(prioritiesinthenetwork) + 1$ - otherwise, there can be a flow rule with such a priority that the deflection action would be out-prioritized.

4. Automated driving framework

This section presents the test platforms used to validate the SerIoT system under C-ITS environments. In order to safely represent the scenarios described in Section 5, a mixed environment test was chosen. In this type of environment, a real vehicle is used to execute a specific maneuver whereas virtual components such as vehicles, traffic lights, among others help to complement the scenario [21].

The real platform consists of a Renault Twizy 80, capable of reaching 80 Km/h, and this is equipped with different sensors such as LiDAR, Laser, D-GPS, among others. Moreover, it is instrumented with two servo-motors, one for steering control, and another for brake control; and an Electronic Control Unit (ECU) for throttle control. The virtual vehicle is simulated through the Dynacar® environment [44], which uses a multi-body model to virtually represent the vehicles [7], and a 3D interface to monitor them. The traffic lights, as well as the control station in charge of supervising the scenario, are simulated through the Matlab/Simulink software.

Both platforms run the AUDRIC© architecture, which consists of the 6 blocks distribution proposed by [19]: acquisition, perception, communication, decision, control, and actuation (see Fig. 3). The acquisition block is in charge of collecting the data coming from the on-board sensors (real platform) or the multi-body model (virtual platform). The perception block generates the virtual vehicle and the representations of the obstacles. The communication block is where all the information of other ITS components is gathered using V2X technologies and where the main contribution of this work is located. The decision module generates the trajectory and the actions to be taken during the driving process. The control block is related to the longitudinal and lateral controllers. The actuation module interprets the control signals to generate the values that can be used by the multi-body model or the actuators, depending on the case.

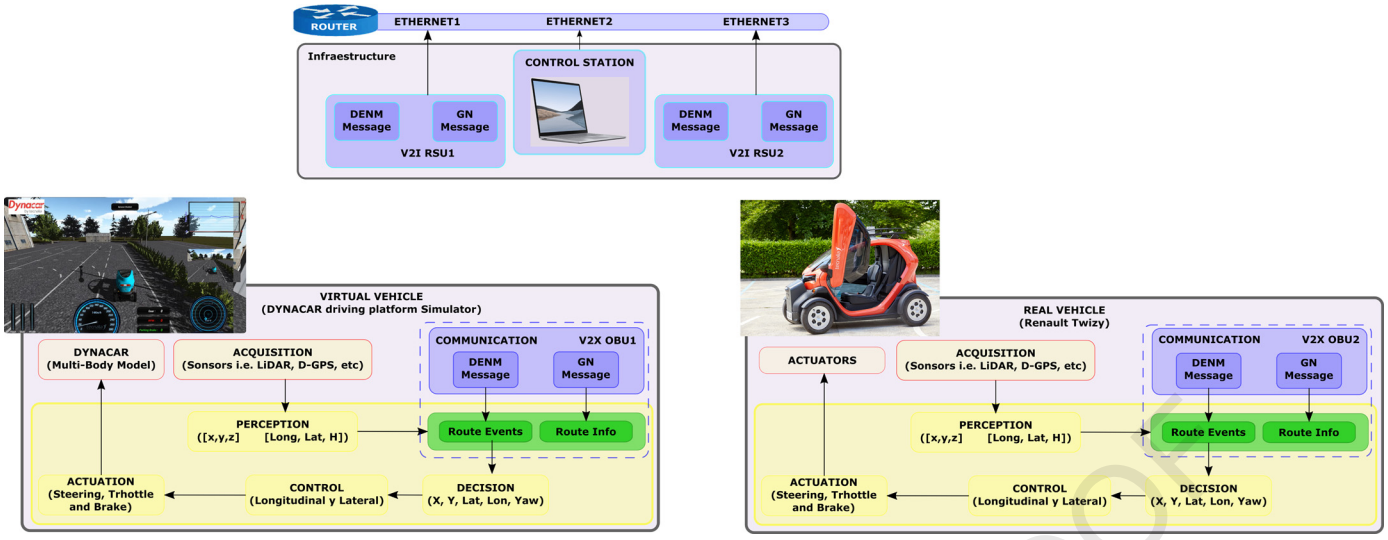


Fig. 3. AUDRIC© architecture.

In order to communicate to each component, V2X communication devices are used; an OBU communicates with the real and virtual vehicles; and, an RSU to communicate with the control station and the traffic lights (see Fig. 3). These devices are equipped with a dual 5.9 GHz antenna with GPS/GLONASS, Wi-Fi and Wi-Max capabilities. Furthermore, the devices have a security module that ensures the confidentiality of the data content by using symmetric or asymmetric cryptography and keys known only by trusted participants, following the normative IEEE 1609.2-2016. This, ensures each component can transmit their corresponding information according to the ETSI-G5 standards [30]. Specifically, in this work, two types of message protocols are used: 1) Geo-Network (GN) [28] and 2) Decentralized Environmental Notification Message (DENM) [49].

Geo-Networking supports heterogeneous application requirements, including applications for road safety, traffic efficiency and infotainment. More specifically, it enables periodic transmission of safety status messages at a high rate, rapid multi-hop dissemination of packets in geographical regions for emergency warnings [14]. Therefore, this protocol is selected to transmit two types of information:

- The identification of the route followed by the vehicle. With this information, the control station can supervise the trajectory followed by the vehicles.
- The status of each traffic light. Due to the flexibility of the GN messages, a shorter version of the SPAT protocol with only the specific fields needed in the test can be sent fulfilling the requirements needed.

The DENM is a facilities layer message that is mainly used by ITS applications in order to alert road users of a detected event using ITS communication technologies [14]. Regarding the present work, the messages transmitted correspond to *unexpected road works* warnings.

5. Use case

In this section, two sub-use cases of the SerIoT project are presented. The goal is to show SerIoT system capabilities to detect and mitigate the attack of a network link, under complex driving scenarios with non-critical communication requirements. To do so, two different detection and mitigation methods were implemented, one in each scenario.

The analysis presented here is based on Research Questions, as defined in SerIoT project report [36], namely:

1. Will SerIoT efficiently detect and mitigate against a fake operation in cooperative maneuvers? (RQ-SE-21-00)
2. Will SerIoT operation affect latency in Automated Driving non-critical comms.? (RQ-QS-21-001)
3. Will SerIoT operation affect loss of packets in Automated Driving non-critical comms.? (RQ-QS-21-003)
4. Do potential Automated Driving users perceive SerIoT as a useful system? (RQ-UA-21-001)

5.1. Sub use case 1: fleet management

The first scenario consists of a fleet of vehicles under a traffic jam problem (see Fig. 4). In this scenario, the route of Vehicle 2 (V2) is modified via V2I communications when a traffic jam is located by Vehicle 1 (V1). This route change is instigated through communication with RSU 2 and the Control Station (CS), which is in charge of monitoring the vehicles. More specifically, during this process, the vehicles are indicating to the Control Station, through GN messages, the route that they follow. When a traffic jam is encountered, the vehicle sends a warning message to the Control Station, using the DENM protocol. Once the warning message is detected, the Control Station sends a GN message indicating the new route to follow.

In order to test the SerIoT system, a Denial of Service (DoS) attack is introduced in the infrastructure. The goal of the attack is to produce a bottleneck in the road by overloading the RSU close to the conflict point (K1), thus, producing its shutdown. By doing so, the

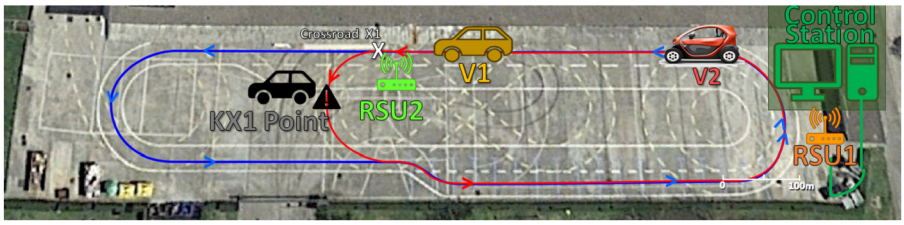


Fig. 4. Fleet management scenario description.

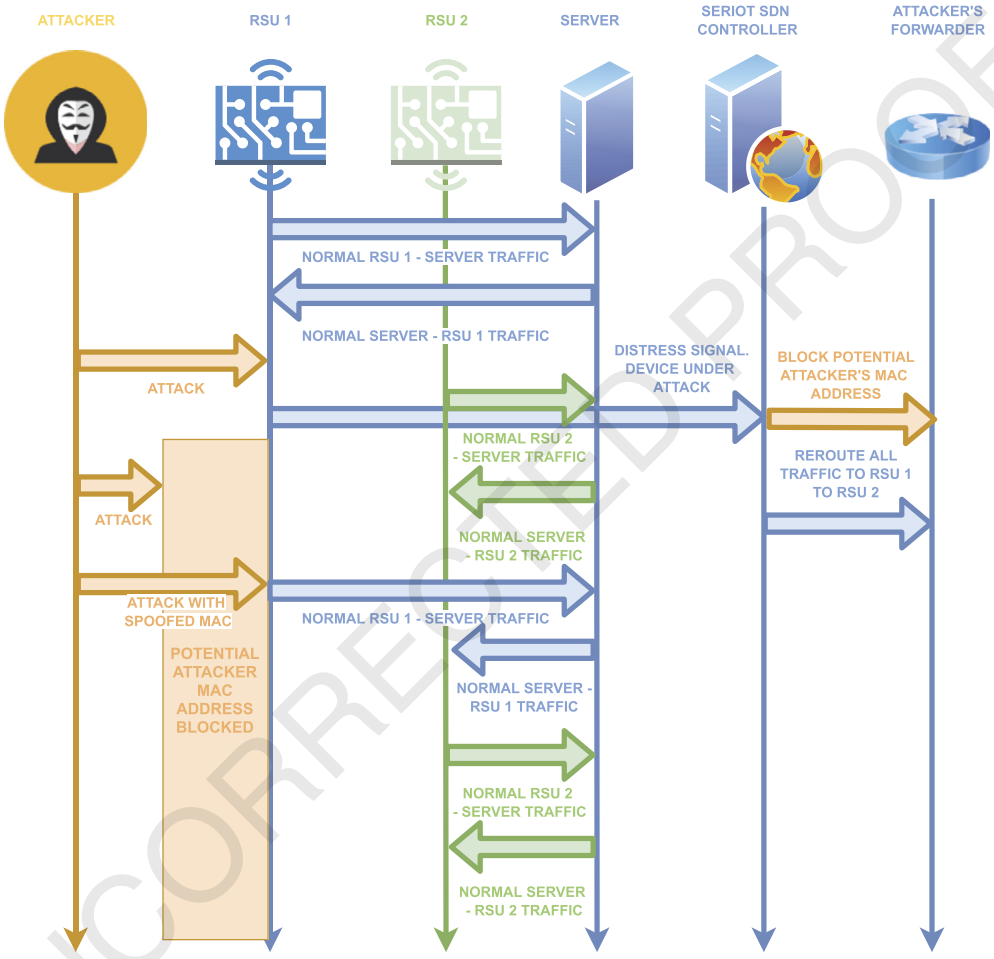


Fig. 5. Data flow of the fleet management use case scenario.

vehicle (V2) cannot change the route, despite the message sent by the vehicle (V1). As a mitigation action, a reroute of the packet traffic from one RSU2 to RSU1 is proposed, so the message can be transmitted normally, and the vehicle can change the route. We can observe this scenario in greater detail in the presented data flow of the components in the network (see Fig. 5). Note that two mitigation actions for the attack are proposed – first, it attempts to block the attacker but due to limited knowledge of the lightweight detector deployed on the RSU1 and RSU2 only the MAC address of the attacker can be obtained. Then it proceeds to ensure the communication between RSU1, the server and the vehicle by rerouting packets for the attacked RSU.

In Fig. 1, the connectivity scheme of the UC is presented. In this figure, it can be seen that the SerIoT system is located between the RSUs and the Control Station. To enable connectivity that emulates a network provider configuration, the RSUs were connected to carrier class switches that were controlled through SDN. The switches were interconnected through 10 Gb/s optical links as would be expected in a metropolitan area network. However, because the different infrastructures (UC and network/SerIoT infrastructure) were hosted at different partner sites, inter-connectivity was back-hauled to the SDN switches and SerIoT components through a virtual private network (VPN) connection. The VPN connection was closely monitored and found to provide consistent delay (30 ms $\pm$ 0.35 ms) with no packet loss at the bit-rates used by the UC.

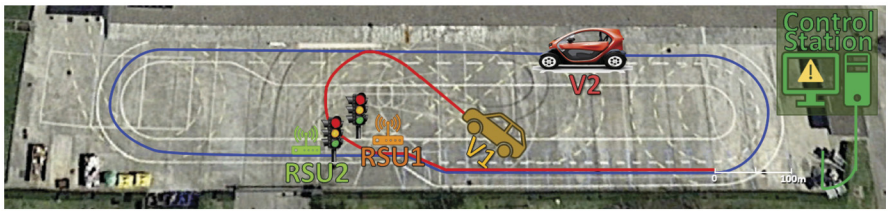


Fig. 6. Smart intersection scenario description. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

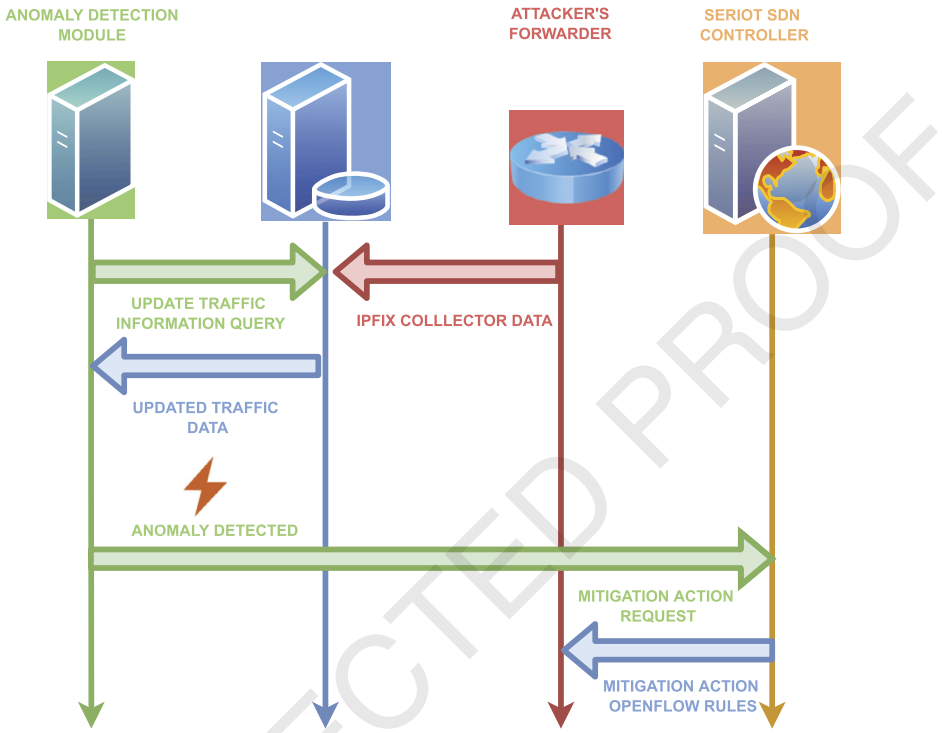


Fig. 7. The data flow between components in the smart intersection use case.

5.2. Sub use case 2: smart intersection

The second Sub Use Case consists of a Smart Intersection scenario where an algorithm manages the control of the traffic light, in order to coordinate the vehicles around the intersection. This coordination is mainly performed through V2I communication. In Fig. 6 a description of the scenario is presented. Vehicle 1 (V1) follows a predefined route, arriving towards the intersection. Vehicle 2 (V2) also follows a predefined path and arrives towards the same intersection in another road segment. The traffic lights (TL1 and TL2) broadcast their information (state, time to change, etc.) via Road Side Units (RSU1 and RSU2). V1 receives TL1 information through its OBU, and continues its trajectory when TL1 is green, or stops when is red, while V2 on the other side of the intersection does the same with TL2 and its OBU. The control station is in charge of controlling both traffic lights.

As with UC1, to test the SerIoT system a DoS attack was introduced in the infrastructure. In this case, the goal of the attack is to produce an accident in the intersection by disabling the RSU2 in charge of transmitting the information of the traffic light 2. By doing so, if there is no human intervention, the vehicles crash at the intersection point. As a mitigation action, the same rerouting option is taken into consideration, nevertheless, a different method to detect and execute the rerouting is implemented. The exact data flow between components in this scenario is presented in Fig. 7.

The connectivity scheme of this UC is similar to the one presented in Fig. 1. The same connection between the SerIoT system and the infrastructure is used, adding two traffic lights, one for each RSU. The main differences are the SerIoT modules used with the addition of Fog nodes at the network edge to detect the DoS attack.

6. Experimental results and discussion

The tests were developed in the Tecnalia test tracks, according to the methodology explained in Section 5. The vehicle V1 is virtually represented with the Dynacar® simulator, whereas the vehicle V2 is the Renault Twizy 80. During the tests, both vehicles run at 16 Km/h.

Each maneuver will be represented by a sequence of images of both vehicles. Each image has three components:

- The right figure with the plot of the vehicles positioning over time. Being the blue rectangle, V1, and the orange rectangle, V2. In the case of Smart intersection, intersection traffic lights are also displayed.

- The lower left figure corresponds to the Dynacar® visualization of the virtual vehicle.
- The upper left figure corresponds to the real vehicle.

6.1. Fleet management

Regarding the fleet management use case, Fig. 8 shows a sequence of images of both vehicles during the maneuver. Fig. 8a shows at the beginning both vehicles follow the same path, and once V1 arrives at the KX1 point in the second lap (Fig. 8b) it sends a DENM message to the control station, which orders V2 to change the route. This change can be seen in Fig. 8c. Finally, Fig. 8d shows V2 driving through route 2.

Performance results of the SerIoT system for the fleet management UC are shown in Table 3. The average time of detection of the DoS attack by the lightweight anomaly detector was 4.34 s. This was a major component of the system's response time to an emergency situation. Additional components with a total value well below 200 ms are the delay introduced by the VPN in the experimental setup and the controller response time to the mitigation command.

6.2. Smart intersection

Regarding the smart intersection use case, Fig. 9 shows a image sequence of the maneuver where it can be seen the vehicles crossing the intersection without difficulties, even though the DoS attack has occurred. Fig. 9a presents the beginning of the scenario, where initially T1 is in red and the T2 in green. Fig. 9b shows when V1 stops at the traffic light T1, whereas V2 approaches T2. In Fig. 9c, it can be observed that V2 is stopping at T2, whereas V1 accelerates due to T1 changing to green. Finally, 9d shows V2 accelerating once T2 changes to green.

In Table 4 the average reaction times of the SerIoT system in this UC are presented. The metrics used for the evaluation of the distributed Anomaly Detection module were Accuracy of Detection, and Detection time. Violation of the normal traffic pattern, brought by the DoS attack, is easily recognized within the analyzed data chunks in every abnormal scenario with 100% accuracy. Detection delay results are fast enough to allow for adequate time for the network to heal and restore to normal operation. More specifically, the multi-agent AD module requires a mean time of 3.27 seconds with a standard deviation of 2.98 seconds, to alert the system for the security breach. This is depicted in the CDF of the detection latency in seconds as shown in Fig. 10. The results incorporate the time needed for the traffic collector to retrieve the abnormal flows together with the agent's time to process the flows and send the anomaly probability report. The solution used in Section 5.2 requires a lot of additional infrastructure for it to be deployed but, its strength is that it is independent of the hardware and software limitations. The anomalous event was then passed to the Mitigation Engine in order for the mitigation rule to be applied.

The mean E2E time for the Mitigation Engine to receive data from the Anomaly Detection system and make a decision i.e. send the mitigation decision to the SDN and receive the answer that the mitigation have been enforced, is 1.58 seconds with a standard deviation of 0.199 seconds. Fig. 11 shows the Cumulative Distribution Function for the E2E mitigation time in seconds. These results include the VPN transport time which is adding 120 ms.

Finally, Fig. 12 shows the latency in the network between: an RSU and a Fog mitigation node; RSU to the Control Node; and, RSU to the cloud. This clearly demonstrates the benefit of using a fog implementation which brings a mitigation element very close to the end system to reduce overall latency.[1]

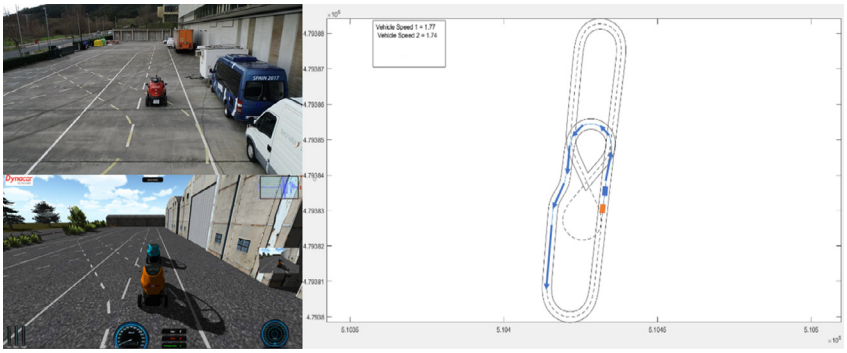
Table 3
Performance of the SerIoT system in the Fleet Management UC.

Detection Time	Mitigation Time	Detected Packet Loss
4.34 [s]	120 [ms] VPN connection time + 46.6 [ms] pure mitigation action	0%

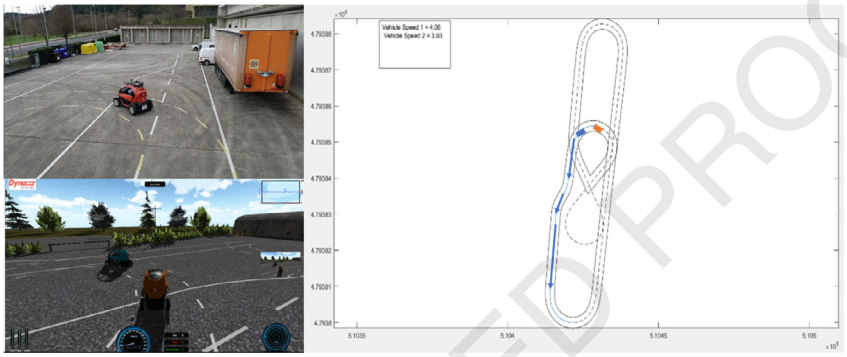
Table 4
Performance of the SerIoT system in Smart Intersection UC.

Detection Time	Mitigation Time	Detected Packet Loss
3.27 [s]	31.1 [ms] VPN connection time + 1,576 [ms] pure mitigation action	0%

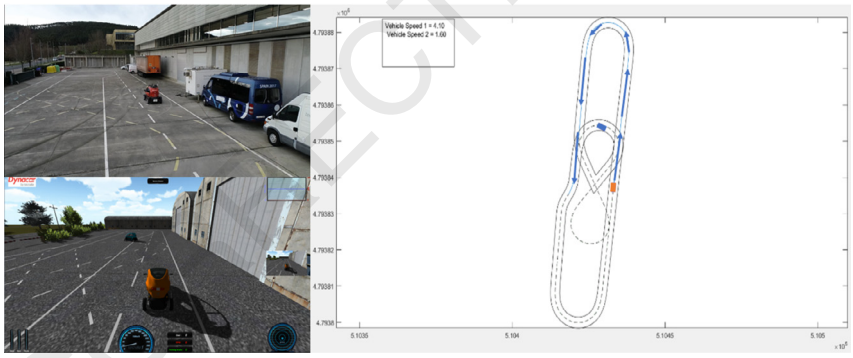
[1] This removes any latency caused by a VPN to interconnect the testbed components.



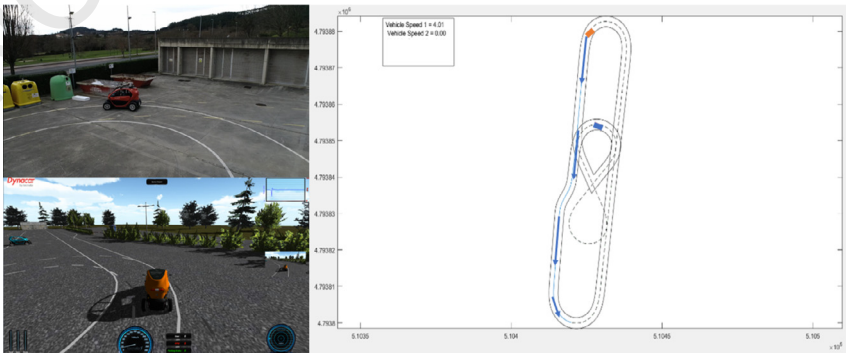
(a) Maneuver Start.



(b) Following first route and continuing for a second lap.

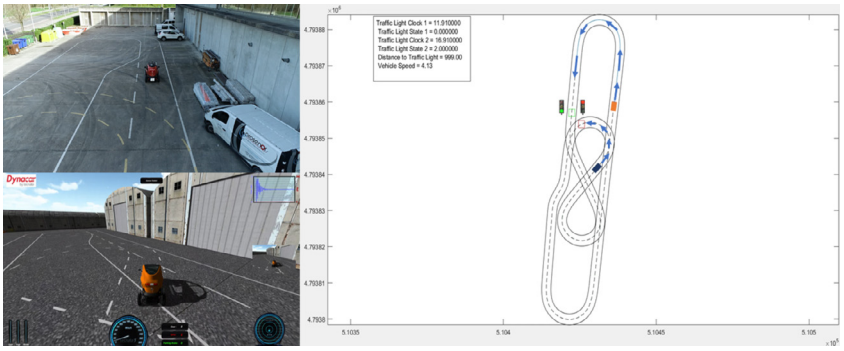


(c) On second lap: orange car changing to second route due to traffic stopping blue car.

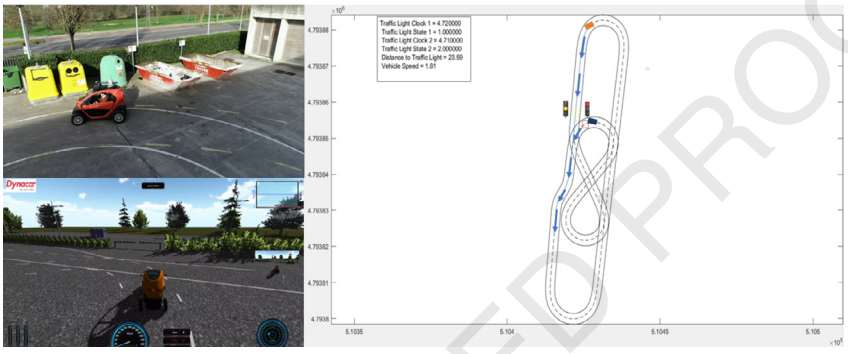


(d) Following second route.

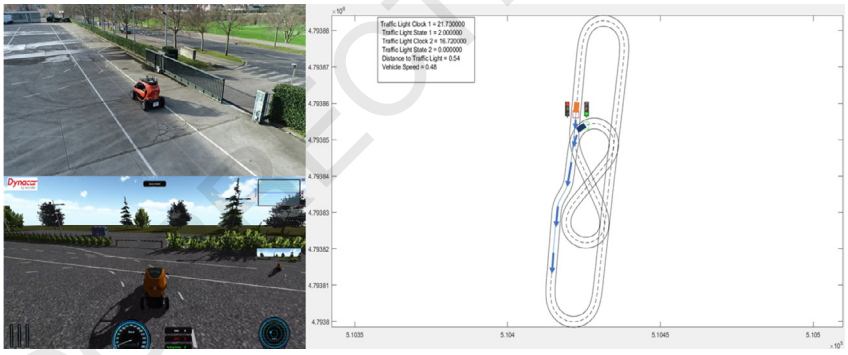
Fig. 8. Fleet management sequence.



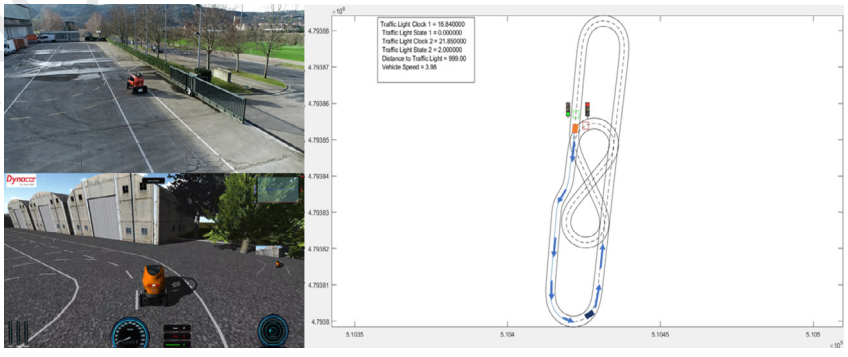
(a) Maneuver Start.



(b) V1 stopping at T1.



(c) V1 accelerating and V2 stopping at T2.



(d) V2 accelerating.

Fig. 9. Smart intersection sequence.

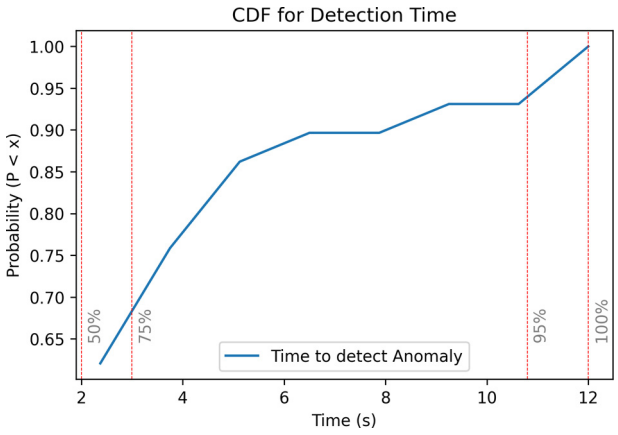


Fig. 10. Cumulative distribution of attack detection latency for UC2, measured for the Multi-agent detection module. Latency measures the time required for the detection to report results from launching timestamp.

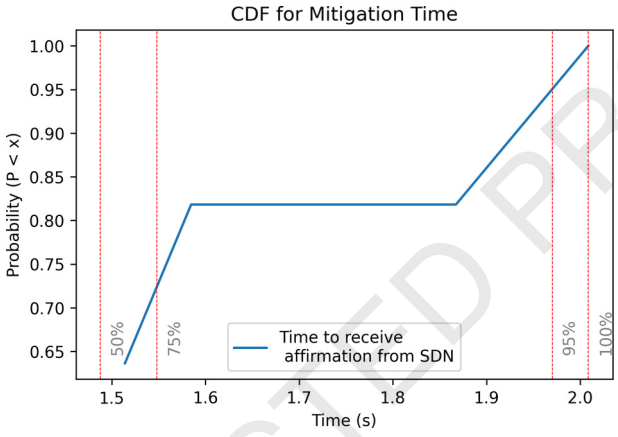


Fig. 11. Cumulative Distribution Function for the E2E mitigation time in seconds for UC2, i.e. the time required to send the Mitigation Decision to the SDN system and receive confirmation.

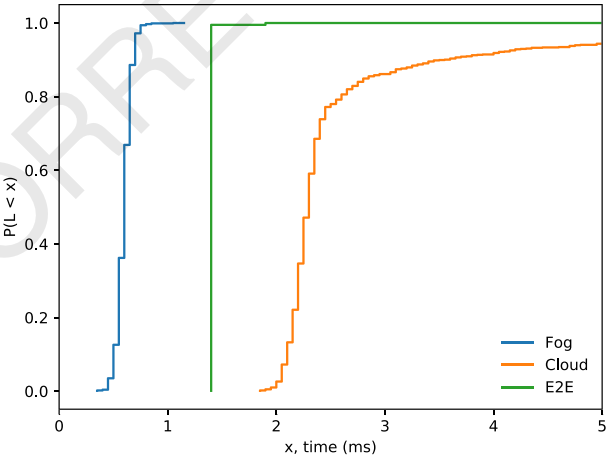


Fig. 12. Network latency (in ms) for end-to-end, end-to-cloud and end-to-fog.

6.3. Discussion

In both Table 3 and Table 4, we can see that both distributed and local detectors are working similarly well with the fastest response seen in the distributed approach.

The response time of a few seconds (on average less than 5 s) is acceptable for car traffic control, taking also into account that commercial implementation will allow for stronger time optimization of both hardware and software side. The solution shown in the Fig. 5 requires an adequate choice of hardware because it must be able to run the respective attack detector code.

On the other hand, the local solution presented in Section 5.1 does not require previous training of the model with anomalous and normal traffic. We can see the trade-off between the flexibility and performance – combining those two approaches would increase the applicability of the solution – even as an off-the-shelf solution. Of course, the solution used in Section 5.2 requires a lot of additional infrastructure for it to be deployed but, its strength is that it is independent of the hardware and software limitations. We also can see that the execution time of sole process of the enforcement of the mitigation action by the SDN controller can be neglected – in the scope of the detection time, it can be treated as instantaneous (with the exclusion of the network-related delays due to the nature of the testbed used for the experiments).

7. Conclusions

Attacks on content and quality of service of the IoT platforms can have economic, energy, and physical security consequences that go way beyond the traditional Internet's claimed lack of security, and beyond the threats posed by attacks to mobile telephony. The SerIoT platform is a system that can be used to secure IoT platforms and networks anywhere and everywhere because it implements and tests a generic IoT framework based on an adaptive smart Software Defined Network with verified components that can spearhead Europe's success in the IoT.

Based on this flexibility, the proposed work presents the validation of the SerIoT system through cooperative maneuvers in an automated driving framework. The performance of these maneuvers is based on V2I communication taking into account the policies and developed technologies for a secure IoT developed in the SerIoT project.

The two use cases raised with their respective attack, detection, and mitigation systems were successfully implemented, validating each of the modules that make up the SerIoT system. Moreover, having two different methods to detect and mitigate the DoS attack, shows robustness in the system. In both cases, the malicious information was detected and the RSU affected was successfully replaced, re-routing its information to the other RSU. Therefore, the maneuvers can be carried out without any inconveniences from the vehicle's perspective.

As we saw in Table 3 and Table 4, both systems perform well in non-critical areas of the autonomous driving, since neither systems produced any significant latency or packet loss. The trade-off is between the infrastructure and technical payload to deploy both solutions - we have a very hardware constrained solution presented in 5.1 and an infrastructure dependent solution in 5.2. From the results, the conclusion is that the hybrid approach would be most favorable - it mixes the hardware self-sufficiency of the second solution with the low deployment costs of the first.

Overall, the work demonstrates that security mitigation actions are both successful and fast enough for non-critical communication such as those between vehicles and road side infrastructure that is communicating general information about the state of the road in some distance ahead. However, the work also concludes that general purpose mitigation strategies are not yet fast enough to protect critical communication, for example a highly sloping road with non-line-of-sight to an obstacle where incorrect performance of the communications can lead to vehicle collisions. Therefore, in future research faster detection and mitigation strategies are needed, allowing complex driving scenarios that involve more vehicles and other types of intersections, where each component is provided with a secure and reliable communication.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] M.M. Ahemd, M.A. Shah, A. Wahid, Iot security: a layered approach for attacks & defenses, in: 2017 International Conference on Communication Technologies (ComTech), 2017, pp. 104–110.
[2] B.L. Bars, A. Kalogeratos, A probabilistic framework to node-level anomaly detection in communication networks, arXiv preprint, arXiv:1902.04521, 2019.
[3] S. Biswas, J. Mišić, V. Mišić, DDoS attack on WAVE-enabled VANET through synchronization, in: 2012 IEEE Global Communications Conference (GLOBECOM), IEEE, 2012, pp. 1079–1084.
[4] Z. Cai, A. Wang, W. Zhang, M. Gruffke, H. Schweppe, 0-days & mitigations: roadways to exploit and secure connected BMW cars, Black Hat USA (2019) 39.
[5] A. Chaudhary, H. Mittal, A. Arora, Anomaly detection using graph neural networks, in: 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), IEEE, 2019, pp. 346–350.
[6] S. Chehida, A. Baouya, M. Bozga, S. Bensalem, Exploration of impactful countermeasures on IoT attacks, in: 2020 9th Mediterranean Conference on Embedded Computing (MECO), 2020, pp. 1–4.
[7] J. Cuadrado, D. Vilela, I. Iglesias, A. Martín, A. Peña, A multibody model to assess the effect of automotive motor in-wheel configuration on vehicle stability and comfort, in: ECCOMAS Multibody Dynamics, 2013.
[8] L. Deri, M. Martinelli, T. Bujlow, A. Cardigliano, nDPI: open-source high-speed deep packet inspection, in: 2014 International Wireless Communications and Mobile Computing Conference (IWCMC), IEEE, 2014, pp. 617–622.
[9] Z. El-Rewini, K. Sadatsharan, D.F. Selvaraj, S.J. Plathottam, P. Ranganathan, Cybersecurity challenges in vehicular communications, Veh. Commun. 23 (2020) 100214.
[10] Z. El-Rewini, K. Sadatsharan, D.F. Selvaraj, S.J. Plathottam, P. Ranganathan, Cybersecurity challenges in vehicular communications, Veh. Commun. 23 (2020) 100214, https://doi.org/10.1016/j.vehcom.2019.100214, https://www.sciencedirect.com/science/article/pii/S221420961930261X.
[11] D. Eswaran, C. Faloutsos, S. Guha, N. Mishra, Spotlight: detecting anomalies in streaming graphs, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018, pp. 1378–1386.
[12] European Commission, a. Modeling and Analysis of Random Spatial Systems for 5G Networks | MARSS-5G Project | H2020 | CORDIS | European Commission, https://cordis.europa.eu/project/id/659933.
[13] European Commission, b. Preparing Secure Vehicle-to-X Communication Systems | PRESERVE Project | FP7 | CORDIS | European Commission, https://cordis.europa.eu/project/id/269994/es.
[14] European Telecommunications Standards Institute, EN 302 636-4-1 Intelligent Transport Systems (ITS), https://www.etsi.org/deliver/etsi_en/302600_302699/3026360401/01.03.01_60/en_3026360401v010301p.pdf.
[15] F. François, E. Gelenbe, Towards a cognitive routing engine for software defined networks, in: 2016 IEEE International Conference on Communications, ICC, Kuala Lumpur, May 22-27, 2016, pp. 1–6.
[16] E. Gelenbe, J. Domanska, P. Fröhlich, M.P. Nowak, S. Nowak, Self-aware networks that optimize security, QoS, and energy, Proc. IEEE 108 (2020) 1150–1167, https://doi.org/10.1109/JPROC.2020.2992559.

[17] E. Gelenbe, P. Fröhlich, M. Nowak, S. Papadopoulos, A. Protogerou, A. Drosou, D. Tzovaras, IoT network attack detection and mitigation, in: 9th Mediterranean Conference on Embedded Computing, MECO 2020, Budva, Montenegro, June 8-11, 2020, IEEE, 2020, pp. 1–6.
[18] E. Gelenbe, Z. Xu, E. Seref, b. Cognitive packet networks, in: Proceedings of the 11th IEEE International Conference on Tools with Artificial Intelligence, IEEE Computer Society, p. 47.
[19] D. González, J. Pérez, V. Milanés, F. Nashashibi, A review of motion planning techniques for automated vehicles, IEEE Trans. Intell. Transp. Syst. 17 (2016) 1135–1145, https://doi.org/10.1109/TITS.2015.2498841.
[20] G. Gonzalez-Granadillo, J. Garcia-Alfaro, E. Alvarez, M. El-Barbori, H. Debar, Selecting optimal countermeasures for attacks against critical systems using the attack volume model and the RORI index, Comput. Electr. Eng. 47 (2015) 13–34, https://doi.org/10.1016/j.compeleceng.2015.07.023.
[21] C. Hidalgo, M. Marcano, G. Fernández, J. Pérez, Maniobras cooperativas aplicadas a vehículos automatizados en entornos virtuales y reales, Rev. Iberoam. Autom. Inform. Ind. 17 (2020) 56–65.
[22] M.D. Hossain, H. Inoue, H. Ochiai, D. Fall, Y. Kadobayashi, Lstm-based intrusion detection system for in-vehicle can bus communications, IEEE Access 8 (2020) 185489–185502.
[23] IEEE, IEEE Standard for Information Technology– Local and Metropolitan Area Networks– Specific Requirements– Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6: Wireless Access in Vehicular Environments, 2010, pp. 1–51, https://standards.ieee.org/standard/802_11p-2010.html, http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5514475'escapeXml='false'/%3E.
[24] A. Jurcut, T. Niculcea, P. Ranaweera, N.A. Le-Khac, Security considerations for Internet of things: a survey, SN Comput. Sci. 1 (2020) 193–211, https://doi.org/10.1007/s42979-020-00201-3.
[25] J. Kim, J. Kim, H.L.T. Thu, H. Kim, Long short term memory recurrent neural network classifier for intrusion detection, in: 2016 International Conference on Platform Technology and Service (PlatCon), IEEE, 2016, pp. 1–5.
[26] H. Koshutanski, M. Al-Naday, I. Cuevas, A. Marcelli, G. Molina, M. Nowak, S. Papadopoulos, P. Sampatokas, A. Shaik, D2.5, Seriot architecture & specifications (r2). Deliverable of the EU Project SerIoT, available online, https://seriot-project.eu/download/1202/.
[27] I. Kotenko, E. Doynikova, Dynamical calculation of security metrics for countermeasure selection in computer networks, in: Proceedings - 24th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing, PDP 2016, 2016, pp. 558–565.
[28] S. Kuhlmorgen, I. Llatser, A. Festag, G. Fettweis, Performance evaluation of etsi geonetworking for vehicular ad hoc networks, in: 2015 IEEE 81st Vehicular Technology Conference (VTC Spring), IEEE, 2015, pp. 1–6.
[29] G. Loukas, T. Vuong, R. Heartfield, G. Sakellari, Y. Yoon, D. Gan, Cloud-based cyber-physical intrusion detection for vehicles using deep learning, IEEE Access 6 (2017) 3491–3508.
[30] N. Lyamin, A. Vinel, M. Jonsson, B. Bellalta, Cooperative awareness in VANETs: on ETSI EN 302 637-2 performance, IEEE Trans. Veh. Technol. 67 (2017) 17–28.
[31] R. Mahmoud, T. Yousuf, F. Aloul, I. Zualkernan, Internet of things (IoT) security: current status, challenges and prospective measures, in: 2015 10th International Conference for Internet Technology and Secured Transactions (ICITST), 2015, pp. 336–341.
[32] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, J. Turner, OpenFlow: Enabling Innovation in Campus Networks, ACM SIGCOMM Computer Communication Review, vol. 38, ACM New York, NY, USA, 2008, pp. 69–74.
[33] A. Messac, A. Ismail-Yahaya, C. Mattson, The normalized normal constraint method for generating the Pareto frontier, Struct. Multidiscip. Optim. 25 (2003) 86–98.
[34] New York City DOT, Connected Vehicle technology is coming to the streets of New York City! https://cvp.nyc/, 2018.
[35] S. Nie, L. Liu, Y. Du, Free-fall: hacking tesla from wireless to can bus, Briefing, Black Hat USA 25 (2017) 1–16.
[36] O. Nikolis, T. Rodrıguez, A. Drosou, I. Cuevas, F. Sánchez, I. Anastasiadis, A. Golfi, B. Monschiebl, P. Hofmann, D8.1. Pilots installation methodology, evaluation plans & assessment framework, SerIoT project report, http://www.seriot-project.eu, 2019.
[37] J. Nilsson, C. Bergenhem, J. Jacobson, R. Johansson, J. Vinter, Functional safety for cooperative systems, in: SAE Technical Papers, SAE International, 2013, https://www.sae.org/publications/technical-papers/content/2013-01-0197/.
[38] NOKIA, Nokia threat intelligence report 2020. Nokia document CID210088, available online, https://onestore.nokia.com/asset/210088, 2020.
[39] M. Obaidat, M. Khodjaeva, J. Holst, M. Ben Zid, Security and privacy challenges in vehicular ad hoc networks, in: Z. Mahmood (Ed.), Connected Vehicles in the Internet of Things: Concepts, Technologies and Frameworks for the IoV, Springer International Publishing, Cham, 2020, pp. 223–251.
[40] Okta, Inc., The state of zero trust security in global organizations, Survey results published online by Okta, https://www.okta.com/resources/reports/state-of-zero-trust-security-in-global-organizations/, 2021.
[41] ONOS Community, Open network operating system (ONOS®), https://opennetworking.org/onos/.
[42] Open Networking Foundation, OpenFlow Switch Specification, https://opennetworking.org/wp-content/uploads/2014/10/openflow-spec-v1.3.0.pdf, 2014.
[43] P. Papadimitratos, A. La Fortelle, K. Evenssen, R. Brignolo, S. Cosenza, Vehicular communication systems: enabling technologies, applications, and future outlook on intelligent transportation, IEEE Commun. Mag. 47 (2009) 84–95, https://doi.org/10.1109/MCOM.2009.5307471.
[44] A. Parra, A.J. Rodrıguez, A. Zubizarreta, J. Pérez, Validation of a real-time capable multibody vehicle dynamics formulation for automotive testing frameworks based on simulation, IEEE Access 8 (2020) 213253–213265, https://doi.org/10.1109/ACCESS.2020.3040232.
[45] K.O. Proskawetz, Car 2 Car Communication Consortium, 2017, pp. 1–15, https://www.car-2-car.org/, https://www.car-2-car.org/fileadmin/downloads/PDFs/car-2-car-journal/Journal_19_C2C-CC_Oct_2017.pdf.
[46] A. Protogerou, S. Papadopoulos, A. Drosou, D. Tzovaras, I. Refanidis, A graph neural network method for distributed anomaly detection in IoT, Evolv. Syst. (2020) 1–18.
[47] G.K. Rajbahadur, A.J. Malton, A. Walenstein, A.E. Hassan, A survey of anomaly detection for connected vehicle cybersecurity and safety, in: 2018 IEEE Intelligent Vehicles Symposium (IV), IEEE, 2018, pp. 421–426.
[48] S.W. Rose, O. Borchert, S. Mitchell, S. Connelly, Zero Trust Architecture, Technical Report NIST Special Publication 800-207, U.S. Department of Commerce, Washington, D.C., 2020.
[49] J. Santa, F. Pereñíguez, A. Moragón, A.F. Skarmeta, Experimental evaluation of cam and denm messaging services in vehicular communications, Transp. Res., Part C, Emerg. Technol. 46 (2014) 98–120.
[50] SerIoT Project, European Commission's H2020 programme (RIA), Grant agreement Nr: 780139, http://www.seriot-project.eu, 2018–2021.
[51] K. Shin, B. Hooi, C. Faloutsos, M-zoom: fast dense-block detection in tensors with quality guarantees, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2016, pp. 264–280.
[52] P.K. Singh, S.K. Nandi, S. Nandi, A tutorial survey on vehicular communication state of the art, and future research directions, Veh. Commun. 18 (2019) 100164, https://doi.org/10.1016/j.vehcom.2019.100164, https://www.sciencedirect.com/science/article/pii/S2214209618300901.
[53] R. Sultana, J. Grover, M. Tripathi, Security of SDN-based vehicular ad hoc networks: state-of-the-art and challenges, Veh. Commun. 27 (2021) 100284, https://doi.org/10.1016/j.vehcom.2020.100284, https://www.sciencedirect.com/science/article/pii/S2214209620300553.
[54] P.J. Sun, Privacy protection and data security in cloud computing: a survey, challenges, and solutions, IEEE Access 7 (2019) 147420–147452, https://doi.org/10.1109/ACCESS.2019.2946185.
[55] The MITRE Corporation, 2021. CVE-2021-22986. Available from MITRE, CVE-ID CVE-2021-22986, http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-22986, 2021.
[56] THEA, THEA Connected Vehicle Pilot, https://theacvprogram.com/, 2018.
[57] M.C. Tran, L. Heejeong, Y. Nakamura, Abnormal web traffic detection using connection graph, Bull. Netw. Comput. Syst. Softw. 3 (2014) 57–62.
[58] U.S. Department of Transportation, Intelligent transportation systems - connected vehicle pilot deployment program, https://www.its.dot.gov/pilots/pilots_overview.htm, 2020.
[59] U.S. Department of Transportation, Automated Vehicles Comprehensive Plan | US Department of Transportation, https://www.transportation.gov/av/avcp, 2021.
[60] O. Vinyals, M. Fortunato, N. Jaitly, Pointer networks, in: Advances in Neural Information Processing Systems, vol. 28, 2015, pp. 2692–2700.
[61] W. Wang, Y. Sheng, J. Wang, X. Zeng, X. Ye, Y. Huang, M. Zhu, HAST-IDS: learning hierarchical spatial-temporal features using deep neural networks to improve intrusion detection, IEEE Access 6 (2017) 1792–1806.
[62] P. Watch, Advanced technologies for industry – product watch IoT components in connected and autonomous vehicles, https://ati.ec.europa.eu/reports/product-watch/iot-components-connected-and-autonomous-vehicles, 2020.
[63] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, P.S. Yu, A comprehensive survey on graph neural networks, arXiv preprint, arXiv:1901.00596, 2019.

[64] L. Yang, A. Moubayed, A. Shami, MTH-IDS: a multi-tiered hybrid intrusion detection system for internet of vehicles, IEEE Int. Things J. (2021), https://doi.org/10.1109/JIOT.2021.3084796.
[65] W. Yu, W. Cheng, C.C. Aggarwal, K. Zhang, H. Chen, W. Wang, Netwalk: a flexible deep embedding approach for anomaly detection in dynamic networks, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018, pp. 2672–2681.
[66] L. Zheng, Z. Li, J. Li, Z. Li, J. Gao, AddGraph: anomaly detection in dynamic graph using attention-based temporal GCN, in: Proceedings of the 28th International Joint Conference on Artificial Intelligence, AAAI Press, 2019, pp. 4419–4425.
[67] S. Zonouz, P. Haghani, Cyber-physical security metric inference in smart grid critical infrastructures based on system administrators' responsive behavior, Comput. Secur. 39 (2013) 190–200, https://doi.org/10.1016/j.cose.2013.07.003.
[68] M. Özçelik, N. Chalabianloo, G. Gür, Software-defined edge defense against iot-based DDoS, in: 2017 IEEE International Conference on Computer and Information Technology (CIT), 2017, pp. 308–313.