A taxonomy and survey of cyber-physical intrusion detection approaches for vehicles

George Loukas, Eirini Karapistoli, Emmanouil Panaousis, Panagiotis Sarigiannidis, Anatolij Bezemskij, and Tuan Vuong

***Abstract*—With the growing threat of cyber and cyber-physical attacks against automobiles, drones, ships, driverless pods and other vehicles, there is also a growing need for intrusion detection approaches that can facilitate defence against such threats. Vehicles tend to have limited processing resources and are energy-constrained. So, any security provision needs to abide by these limitations. At the same time, attacks against vehicles are very rare, often making knowledge-based intrusion detection systems less practical than behaviour-based ones, which is the reverse of what is seen in conventional computing systems. Furthermore, vehicle design and implementation can differ wildly between different types or different manufacturers, which can lead to intrusion detection designs that are vehicle-specific. Equally importantly, vehicles are practically defined by their ability to move, autonomously or not. Movement, as well as other physical manifestations of their operation may allow cyber security breaches to lead to physical damage, but can also be an opportunity for detection. For example, physical sensing can contribute to more accurate or more rapid intrusion detection through observation and analysis of physical manifestations of a security breach. This paper presents a classification and survey of intrusion detection systems designed and evaluated specifically on vehicles and networks of vehicles. Its aim is to help identify existing techniques that can be adopted in the industry, along with their advantages and disadvantages, as well as to identify gaps in the literature, which are attractive and highly meaningful areas of future research.**

***Index Terms*—Vehicles, cyber-physical systems, intrusion detection, vehicular networks, VANET, cyber security, aircraft, unmanned aerial vehicles, robotic land vehicles, automobiles, driverless pods.**

I. Introduction

Cyber-physical attacks are breaches in cyber space that have an adverse effect in physical space [1]. Vehicles constitute attractive targets for such attacks primarily because of their mobility in physical space. Beyond the impact on physical privacy or driver inconvenience through fraudulent warnings, in extreme cases a remotely hijacked car can be steered off the road, a drone can be flown into a crowd, and a driverless military vehicle can be directed to enemy lines to be captured. Examples of attacks documented in the literature range from compromising a car's in-vehicle network via malware-infected audio files [2], and hijacking the navigation of surface vessels via Global Positioning System (GPS) spoofing [3], to overwhelming the lidar sensors of driverless vehicles [4]. Traditional approaches designed for conventional computing systems, enterprise networks and the Internet at large are not always the most appropriate in this context. Research on cyber security of vehicles has focused primarily on cryptography as a means for preventing integrity and confidentiality threats, such as unauthorised unlocking of a vehicle or eavesdropping on the video streamed by unmanned aerial vehicles (UAVs). As the attack surface for vehicles becomes larger and more diverse, it is becoming less practical to assume that prevention mechanisms are sufficient, and researchers are turning towards intrusion detection systems (IDSs) designed specifically for vehicles.

In this paper, we make the following contributions:

- We present the first taxonomy of IDS characteristics and architectures designed for vehicles
- We produce the first systematic review of the broad landscape of IDS techniques designed for vehicles, with 66 techniques reviewed in total
- We identify open issues in developing IDS for vehicles where further research can have considerable impact

The adoption of computing in a diverse range of applications has led to a similarly diverse range of related surveys, with Modi *et al.* [5] specialising in threats to cloud computing, Mitchell and Chen in wireless networks [6], and Butun *et al.* [7] in sensor networks. Other surveys have addressed different IDS from the angle of the technique used, with recent examples focusing on the use of machine learning and data mining [8] and deep learning [9]. These surveys have not looked at techniques designed for vehicles, but rather generalist computer networks. An exception is the 2014 survey and taxonomy of IDS for cyber-physical systems by Mitchell and Chen [10], which, however, was published before attacks against vehicles became the vibrant area for research that it now is, and as a result, before the recent influx of IDS techniques proposed specifically for vehicles. So, it included only three relevant examples. Two more recent surveys are the works of Sakiz and Sen [11], who have focused specifically on vehicular ad hoc networks comprising smart vehicles and roadside units, and Thing and Wu [12], who have included intrusion detection in their taxonomy of attacks and defences for autonomous vehicles.

Here, we expand the scope beyond a specific type of vehicle or existence of a supporting network infrastructure, and produce a comprehensive taxonomy of IDS for vehicles, whether they operate individually or as parts of groups, and whether in land, sea or air. We place particular emphasis on the practicality of each proposed IDS, not only from the perspective of the technique used and the types of attacks it has been tested on, but also regarding the conceptual IDS architecture it can support and crucially how ready it is for adoption or further development. In the next sections, we start with a

brief description of the key aspects of an IDS and then the factors affecting IDS design before presenting our taxonomy of vehicle IDS characteristics and design architectures. We continue with a brief description of the different cyber attacks considered to date and with the main body of this work, which is the survey of different techniques for single vehicles and networks of vehicles, classified based on the taxonomy criteria. This is followed by lessons learned and open issues that can be attractive areas of research.

II. Factors affecting IDS design in vehicles

An IDS is a software or a physical device monitoring a system with the purpose to detect signs of attempts to compromise the integrity, confidentiality or availability of one or more of its resources, which may be important vehicle data, a vehicle's subsystem or an internal or external network. The assumption of its existence is that intrusion prevention measures are not always successful and as such, some attacks against a system (here, a vehicle) do go through. The job of the IDS is to detect them when this is the case and accordingly inform an administrator or trigger an appropriate countermeasure. In its simplest form, an IDS should include data collection and aggregation components for monitoring a variety of often heterogeneous sources of data (referred to as the "audit features") that are relevant to the security of the vehicle at hand, and a reasoning component for determining whether the vehicle is currently under attack. The latter is typically a binary classification problem (attack *Vs.* normal) and more rarely a multiclass classification problem when the aim is not only to detect the existence of an attack, but also to identify its type. In the vast majority of IDS solutions found in the literature, the aim has been to achieve correct detection as evaluated by the accuracy and precision of the binary classification and, in more detail, by the confusion matrix of true positive, true negative, false positive and false negative rates. Contrary to conventional computer networks, for cyber-physical systems and especially vehicles, an additional meaningful metric is detection latency, which is the time it takes the IDS to correctly detect an attack. The specific design of an IDS for vehicles depends on a number of factors, which are detailed in the following subsections.

A. Vehicle application

Vehicle architectures tend to differ as much as vehicle applications. For some, the differences are only in the name. Others differ dramatically in terms of communication, sensing and actuation technologies. The degree and nature of automation also plays an increasingly significant role, especially in differentiating between the intrusion detection needs and possible architectures for remote control vehicles, connected cars, driverless cars, robotic cars, robocabs, robotrucks, podcars, deliverbots, driverless platoons, remote-controlled UAV, fully autonomous UAV and other highly overlapping vehicle types. For example, the IDS of a driverless car may have to rely on data collected on board or through interaction with a smart infrastructure or other driverless cars in the vicinity, while a driverless platoon may also have the opportunity to distribute the processing load or share threat data between its vehicles. A fully autonomous UAV may need to take defence decisions completely on its own, while for a remote-controlled one, it may be sufficient to collect data and visualise the threat picture to the user piloting it. So, what is meaningful in terms of detection depends first on the type of vehicle application, as defined by the degree of automation, its proximity to other vehicles or infrastructure, and whether there are human users involved as passengers or as drivers/pilots.

B. Processing and energy constraints

For a severely resource-constrained vehicle, such as a small UAV, collecting security-relevant data may be prohibitive altogether, and even if the data can be collected, there may not be sufficient power to perform meaningful processing of that data locally. For most vehicles, energy efficiency is a priority, whether because it can otherwise not achieve its mission (a reconnaissance UAV will not loiter long enough over its target area) or because its potential buyer wishes to reduce the cost of fuel or damage to the environment (today's car commercials almost invariably emphasise on the miles per gallon achieved). For a security measure to be integrated in a vehicle, it is often a requirement that it will not noticeably affect the energy consumption.

C. Nature of cyber risk

Hijacking a deliverbot may cause inconvenience and may have financial cost, but is unlikely to cause mass physical damage. A hijacked driverless platoon, on the other hand, would. The perceived risk in terms of the likelihood and potential impact of different attacks on a vehicle influences the configuration of its IDS. In this example, the deliverbot might not need an IDS at all, especially considering the increase in financial, energy and processing cost, or may have one that is lightweight and prioritises having a low false positive rate even if that meant missing a few attacks. In contrast, a driverless platoon would certainly need an IDS, in addition with other security measures, and would tolerate a few false positives if that meant achieving a very high true positive rate. Minimising detection latency would also be a very important target, because delaying the detection of an attack that would hijack a critical system, such as steering or braking, by a few seconds could be disastrous.

III. A taxonomy of vehicle IDS characteristics

Different vehicular systems tend to be most vulnerable to different types of attacks, which in turn may lead to different audit approaches, types and features for these attacks to be detected. Note that the taxonomy presented here (Figure 1) is not exhaustive of all the possible approaches, but rather a taxonomy of the approaches that have been proposed in the literature.

A key characteristic of a vehicular IDS is the **deployment location**, i.e., whether it is deployed onboard the vehicle or externally. Local onboard deployment means that the vehicle can only use the information collected on that vehicle

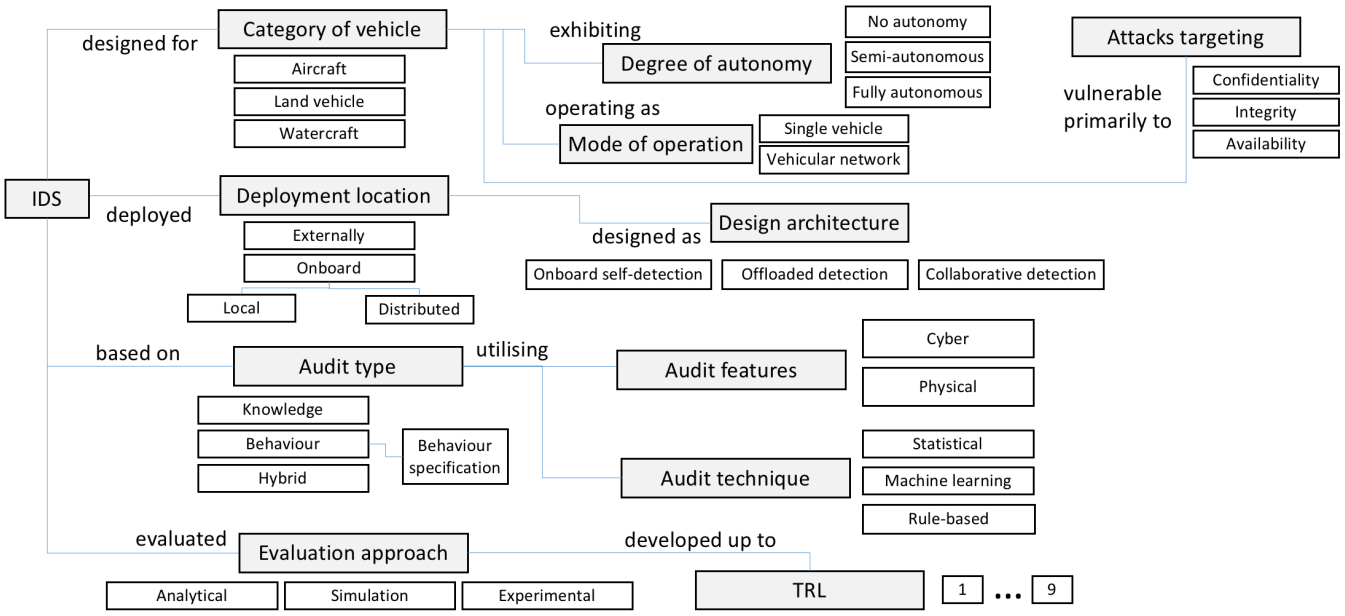

Fig. 1. Taxonomy of IDS for vehicles

and is limited to its own processing power. The process of continuously trying to self-detect attacks against itself can noticeably affect its performance and energy consumption. For this reason, most local onboard IDS approaches tend to be lightweight. In vehicular networks, where there is the opportunity for multiple vehicles to collaborate with each other, the data collection and processing can be shared between them and the detection decisions can be taken in a distributed manner. Where the vehicle itself is not powerful enough to perform meaningful IDS onboard and there is no opportunity for collaboration with other vehicles, then the IDS can be run externally, for example by the computing system of a human operator controlling the vehicle remotely or by offloading the IDS processing to a remote cloud infrastructure [13].

Naturally, it is the area of application in the form of **category of vehicle** that dictates the IDS requirements. The majority of the published research focuses on aircraft, robotic vehicles and automobiles, exhibiting varying **degree of autonomy**, from conventional aircraft and automobiles, to semi-autonomous robotic rescue vehicles, and fully autonomous UAVs and driverless cars, which operate in a **mode of operation** that can be as single vehicles or in vehicular networks. Note that by the latter we refer to a network of any type of vehicle, not only automobiles in the context of intelligent transportation.

The next key characteristic is whether it is more important to detect attacks that have been seen before or attacks that are completely new (zero-day threats). Here, the research community has largely settled in terms of *knowledge-based* versus *behaviour-based* approaches [10]. We refer to this category in the taxonomy as **audit type**. Knowledge-based approaches assume that a vehicle is likely to be attacked in a manner that has been seen before and as such, it makes sense to look for signatures of known attacks, perhaps in the patterns of network traffic received or the impact on the operation of the vehicle. This works very well in IDSs designed for computer networks, as vast dictionaries of attack signatures exist, but is not necessarily the case for vehicles, which can differ considerably between them and attacks against them are still extremely rare. Importantly, knowledge-based IDSs cannot naturally detect zero-day threats. Here, behaviour-based approaches have an advantage. Instead of knowing what an attack looks like, they know what the normal state looks like, and assume that significant deviation from this normal state is sign of an intrusion. The problem here is that what is normal cannot often be determined accurately and also that not all deviations are of malicious nature. As a result, behaviour-based approaches can exhibit high false positive rates [10]. A particular subtype is **behaviour specification** [10], where what is normal is determined by identifying the complete set of normal states of a vehicle based on its specification and checking whether the vehicle is not in one of these states. Where choosing an only behaviour-based or only knowledge-based approach is impractical or ineffective, researchers have suggested **hybrid** approaches, which combine the two.

Interestingly, in terms of **audit features**, when designing IDS for vehicles and other cyber-physical systems, one does not need to be limited to *cyber sources* of data, such as those related to network traffic or computation, but can also make use of *physical sources* of data, as monitored by the vehicle's own sensors, such as physical speed or energy consumption. The range of data available influences the **audit technique** utilised, which is usually based on statistical and machine learning techniques (we referred to both as *learning*), as well as by checking whether particular specified (rather than automatically learned) rules are satisfied or broken, especially for behaviour-specification approaches. This also depends on the expected types of **attacks targeting** a particular vehicle. Here, attacks targeting integrity and availability can lead to serious physical damage, and as such the research community

has prioritised them over attacks that target confidentiality.

An important goal of this work is to help researchers and developers of vehicles choose or improve on existing approaches. For this reason, we place particular emphasis on the **evaluation approach** (analytical, simulation or experimental) used for each proposed IDS, as well as the resulting technology readiness level (**TRL**) at the time of publication. We use the TRL system for assessing the maturity of different technologies proposed by Mankins in 1995 [14], as adopted by U.S. government agencies and with minor variations by many other countries across the world, as well as the European Union. For clarity, we have translated the generic TRL definitions into indicative characteristics of IDSs for vehicles in Table I.

A. IDS design architecture

The factors identified in Section II influence directly the choice of architecture to be adopted. Figure 2 summarises the conceptual elements of an IDS architecture for vehicular systems. Note that this is an aggregate view of all elements considered and that. There is no single proposed IDS that includes all and there may not need to be. Also, note that the use of the image of a car rather than any other type of vehicle is for presentation reasons only. Below, we summarise the main examples of architectures that can be derived and have been used in the literature.

1) Onboard self-detection: In the conceptually simplest and often most desirable case, the vehicle can self-detect threats against it based on onboard data collection, aggregation and reasoning (Figure 3) [15]. The advantage of relying only on its own capabilities is that the vehicle does not need network connectivity to recognise that it has been infected by malware or compromised otherwise, and the detection latency does not depend on the performance or reliability of the network. It may also be a more secure approach for intrusion detection, because it does not involve sharing security-sensitive data over an external communication medium. The key disadvantage is that the complexity of the reasoning approach used is limited by the onboard data collection and processing capabilities.

2) Collaborative detection: In some application areas, such as platoons of driverless trucks [16] or UAV swarms used in urban sensing [17], a vehicle may operate as part of a network, where it can share the task of detection with other nodes (Figure 4) [18] or carry out the detection for one of its neighbours, assuming a "monitor node" role [19]. For example, it may ask other nodes to report whether it seems to be veering off a route or it may participate in voting on whether another vehicle seems to be misbehaving. Research here benefits considerably from prior work in other areas of distributed computing, such as security in wireless sensor networks. The key advantage of collaborative detection is that it can help detect threats that are invisible to a particular vehicle and usually without considerable processing load or the need to monitor many sources of data on each vehicle. The key disadvantages are that other nodes cannot always be trusted and some types of cyber threats may leave no trace that can be observed from outside the targeted vehicle.

3) Offloaded detection: If being able to self-detect threats is not a requirement and collaboration with neighbouring nodes is not an option, then it may be efficient to offload the detection process onto a remote service (Figure 5), as in [13]. A key benefit is that access to a more powerful system (e.g., a cloud) means access to more powerful algorithms for intrusion detection, for example based on deep learning, which may otherwise be prohibitive for a resource-constrained vehicle. More powerful algorithms lead to lower false positive rates of detection (thus, less potential disruption because of incorrect detection) and lower false negative rates. Offloading can also have benefits in terms of energy consumption and even detection latency, but the latter depends on the reliability of the network supporting it. If the network is reliable and fast enough, then sending all data to a cloud and waiting to receive back the detection result may be faster than doing all this with the limited resources available onboard the vehicle. Also, the longer the task of processing data onboard, the greater the energy consumption for the vehicle.

On the other hand, offloading a cyber security task can be challenging by itself. It requires access to a remote infrastructure, which may be expensive to own or rent. If the network is too slow, detection latency can increase beyond what is acceptable. More importantly, the purpose of offloading a security task, such as intrusion detection, is defeated if this process cannot be carried out securely enough, especially as the vehicle needs to communicate security-sensitive data about its operation over a wireless medium or to a third party cloud provider.

IV. Security threats

In summarising attacks against in-vehicle networks of automobiles, Liu *et al.* [20] have emphasised that frame sniffing can be the foundation of most if not all other attacks (e.g., frame falsifying, frame injection, replay and denial of service attack, etc.). This is also how Koscher *et al.* started their analysis in [21], which led to the identification of the valid CAN frames in the automobile they used as testbed. Then, they used fuzz testing (fuzzing), which is the process of creating CAN frames with all possible combinations of bits in the command fields, and observing the physical impact on the automobile. However, in intrusion detection, the assumption is that this type of *confidentiality* breach for the purpose of reverse-engineering has already happened (e.g., the attacker already knows what frame does what), and as a result, most IDS papers focus almost exclusively on attacks targeting the *integrity* or *availability* of a vehicle's computing systems that control its actuation while in operation. Table II presents a summary of the security threats that have been considered in the literature specifically for evaluation of IDS systems for vehicles. Note that this is not a complete a list of all attacks that are possible or that have been demonstrated against vehicles or vehicular networks. For a more general list, albeit not specific to intrusion detection evaluation, the interested reader can refer to [22].

TABLE I
INDICATIVE LEVELS OF TECHNOLOGICAL MATURITY OF IDS FOR VEHICLES

TRL	Description
1	Basic IDS model in the form of mathematical formulations or algorithms
2	IDS model evaluated via analytical predictions or early simulation results
3	The IDS model has been evaluated in accurate simulation of vehicle states and attack mechanisms, possibly using data from real vehicles
4	A prototype IDS has been implemented, partially integrated in a real vehicle and evaluated against a small number of laboratory attacks on the vehicle
5	The prototype IDS has been integrated in a real vehicle and evaluated in high-fidelity experimental laboratory conditions
6	The prototype IDS has been integrated in a real vehicle and thoroughly evaluated in relevant environment conditions (air, land, sea)
7	The prototype IDS has been demonstrated in a real vehicle and in realistic operation/missions against a wide range of attacks
8	The system development of the IDS has been completed
9	Bug fixing has completed and the IDS is ready for deployment/production

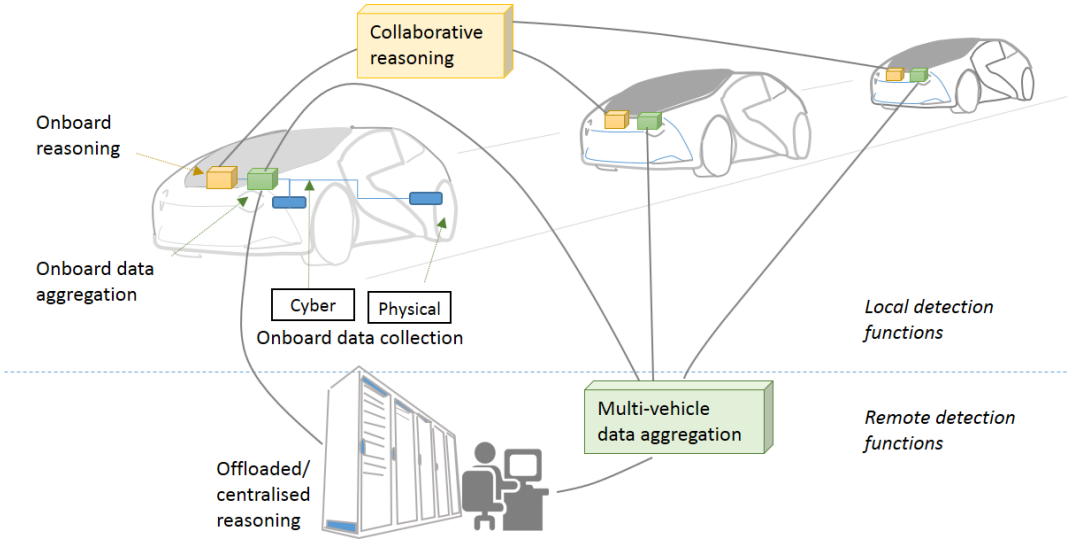

Fig. 2. Aggregate view of the IDS architectural components that can be considered

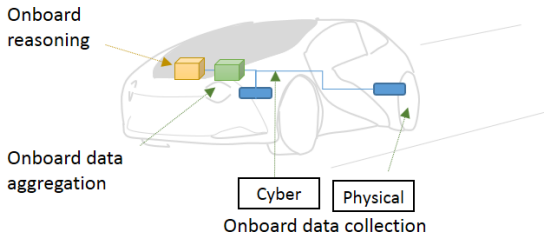

Fig. 3. Conceptual architecture of onboard self-detection

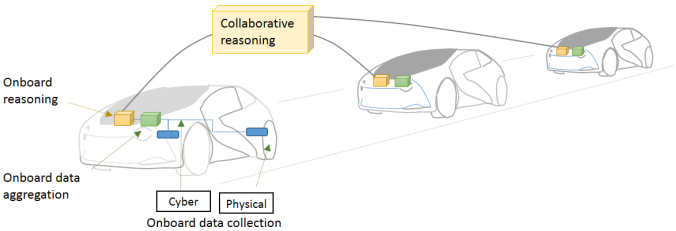

Fig. 4. Conceptual architecture of collaborative detection

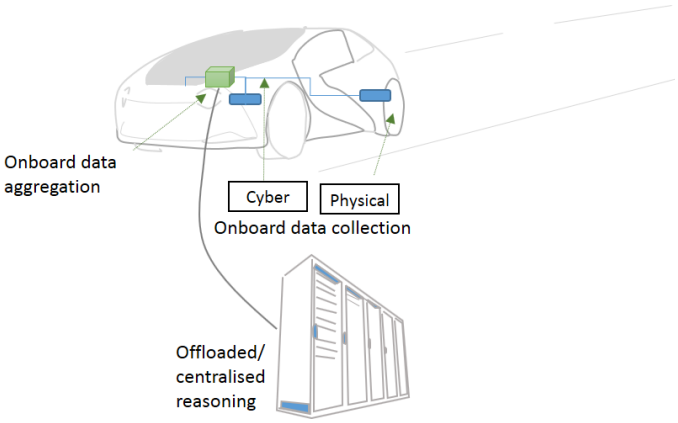

Fig. 5. Conceptual architecture of offloaded detection

V. INTRUSION DETECTION APPROACHES BY TYPE OF VEHICLE

In this section, we survey and classify the individual IDSs reported in the literature, grouped in subsections based on the type of vehicle they have been designed for, starting in

TABLE II
INDICATIVE SECURITY THREATS USED FOR EVALUATION OF IDS FOR VEHICLES

Attack	Description	References
Wormhole	Force a node to transmit its data through a rogue tunnel by pretending to be the shortest and authentic route	[23], [24], [25]
Blackhole	Compromise a node to drop all packets travelling through it without informing their sources	[26], [27], [25], [28], [23]
Greyhole	Compromise a node to selectively drop some packets travelling through it without informing their sources	[29],[28],[27]
Rushing attack	Flood a network with malicious messages so that they are delivered before a legitimate message is received and acknowledged	[29]
Sybil Attack	Generate multiple pseudo-identities in a vehicular network that relies on a reputation system for assessing reliability of information	[24], [25], [27]
Denial of Service (incl. message flooding)	Disrupt communication typically by overwhelming the network with large volumes of meaningless or false data, such as fake alert messages about road accidents and congestion	[30], [31], [32], [26], [33], [27], [34], [35], [36], [37]
Bus-off attack	Exploit the error-handling scheme of in-vehicle networks, by deceiving an uncompromised ECU into thinking it is defective, and eventually forcing itself or even the whole network to shut down	[38]
Message Distortion	Generate distorted reliability message in a vehicular network and activate distribution of this message to a neighbouring vehicle	[39]
Timing attack	An integrity attack that alters message timeslots	[40]
Replay attack	A valid data transmission, such as a command or a sensor reading, is recorded and maliciously repeated at a later point.	[15], [41], [42], [43], [44]
Command Injection	Request execution of existing command with malicious intent, typically to affect actuation	[30], [15], [31], [32]
Impersonation (or masquerade or spoofing) attack	An adversary assumes successfully the identity of one of the legitimate nodes in the vehicular network	[41], [45], [44], [46], [36], [47], [48], [49]
Packet Duplication	Transmit unnecessary network messages to exhaust bandwidth or trigger unnecessary processing	[24], [23], [25]
Selective Forwarding	Retransmit data selectively in a vehicular network	[24], [23], [25]
GPS Jamming	Jam legitimate GPS signals; possibly followed by GPS spoofing	[28]
GPS Spoofing	Transmit false GPS signals to disrupt or hijack navigation of a GPS-dependent vehicle, such as a UAV	[28]
Fuzzing (Fuzz testing)	Send random messages to the in-vehicle network to trigger critical instructions in a brute force manner)	[21]
False Data Injection	Transmit false data to trigger malicious events or affect situational/environmental awareness	[28]
False Information Dissemination	Transmit false data, e.g. a reputation score, to affect a collaborative process in a network	[28], [50]
Location Spoofing	Share false location coordinates within a vehicular network	[39]
Malware	Infect vehicle with malicious software/firmware by compromising supply chain or hijacking an update	[30], [51], [52], [19], [53]
Resource exhaustion attack	Exhaust a vehicle's battery/fuel, network, processing or other resource by repeating requests, infecting with malware, etc.	[24], [23], [25]
Ranging Manipulation	Share incorrect time tags within a vehicular network to disrupt a vehicle's ranging capabilities	[39]
Sensory channel attack	Manipulate the physical environment so as to deceive a vehicle's critical sensors, such as lidar or cameras used by driverless vehicles	[4], [54], [46], [55]
Adversarial machine learning attack on driverless vehicle	Maliciously craft input data to sensors specifically aiming to affect its machine learning policies	[56]
Hardware Tampering	Tamper with hardware or gain physical access to modify/damage components or infect with malware	[51]
Hardware Failure	Physical damage or natural degradation of a vehicle's components	[51]
Fraudulent ADS-B Messages	Transmit false ADS-B messages to affect aircraft safety	[57], [58]
AIS spoofing	Transmit false AIS signals to impede vessel tracking	[59], [60], [61]
Isolation attack	Isolate a node from a network by dropping all messages going to or coming from it	[41]

each subsection with individual vehicles here and extending to networks of vehicles. Below the description of each IDS, we include a bullet-point summary based on the taxonomy of Section III. To the extent that this is possible based on only published information, we provide an estimate of the TRL of each IDS to give the reader an idea of the maturity of the approach. For each type, we have produced a table summarising the characteristics of the IDS approaches proposed and evaluated for it.

A. Aircraft

Aircraft depend on a variety of types of communication both for navigation and safety. For example, a very useful technology for air traffic management and safety is Automatic Dependent Surveillance-Broadcast (ADS-B). Using ADS-B, aircraft determine their position and broadcast it together with other situational data, so that they are received by air traffic control ground stations as well as other aircraft, in this way supporting self-separation. ADS-B messages are unencrypted and unauthenticated, which the Federal Aviation Administration considers to be necessary due to operational requirements [62]. As a result, a malicious user can generate fraudulent ADS-B messages to severely compromise the safety of the aircraft, especially in less dense airspaces where it may be the only means of air traffic surveillance.

Lauf *et al.* [57] were the first to consider ADS-B data in intrusion detection. Their system, HybrIDS, has been designed generally for distributed detection in ad hoc networks, and uses integer labeling with associated probabilities to define a probability density function of the data request interactions between nodes. It has been adapted to take into account ADS-B data. To be appropriate for resource-constrained devices, it does not take into account the actual content, but only the type of request (e.g., "Request Altitude" or "Request Velocity"). It first detects single intruders by analysing peaks in the probability density function from statistics generated from requests made by other nodes, assuming that a local maximum in a normalised distribution indicates misbehaviour. In a second phase, it uses *cross-correlation* between different nodes' behaviours to detect potentially cooperating intruders. However, applying this approach assumes a modification of ADS-B to include an extended list of data transmitted as well as, in addition to broadcast, a provision for direct data requests between aircraft, which at the moment is not the case.

- Scope (S): Aircraft using ADS-B
- Deployment / design architecture (D/DA): Onboard (local or distributed)/ Onboard self-detection
- Audit type / technique (AT/T): Behaviour / (Learning) Deviation from local maxima and cross-correlation
- Audit features (AF): (Cyber) Type of ADS-B request
- Attacks addressed (AA): (Integrity) Fraudulent ADS-B messages
- Evaluation approach (EA): (Simulation) Matlab simulation of mission data and (Experimental) evaluation of embedded system implementation
- TRL: 2

Also for ADS-B, Strohmeier *et al.* [58] have used statistics related to the received signal strength (RSS) as the only audit features, with the assumption that the RSS of spoofed ADS-B signals coming from an attacker on the ground would differ to signals coming from aircraft. The authors have used *standard statistical hypothesis testing*, where the detection system judges the probability that a collected RSS sample comes from a legitimate aircraft. Pearson Correlation Coefficient can be used to test the veracity of the distance claimed via the ADS-B message against the RSS. Autocorrelation Coefficient can then help identify repeated RSS patterns, and hence, attackers that are stationary or do not adapt their sending strength. Also, legitimate ADS-B users use two different antennas transmitting alternatingly. So, a legitimate aircraft's RSS can be divided into two time series of rather different values. A less sophisticated ADS-B spoofer would use only one antenna and would be unlikely to mimic this behaviour. Anomaly detection based on RSS measurements was shown to perform well with a variety of standard classifiers, including Parzen, K-Means, Minimax, Minimum Spanning Tree and K-Nearest Neighbors, with Parzen's false negatives dropping below 2% when the messages per flight exceed 100. To evade it, an attacker on the ground would need to put extraordinary effort to mimic accurately the statistical behaviour of legitimate RSS signals.

- S: Aircraft using ADS-B
- D/DA: Onboard (local) / Onboard self-detection
- AT/T: Behaviour / (Learning) Five standard classifiers applied on Pearson correlation coefficient and autocorrelation coefficient of the RSS
- AF: (Physical) ADS-B RSS
- AA: (Integrity) Fraudulent ADS-B messages
- EA: (Simulation) Attacks simulated in Matlab based on (Experimental) crowdsourced ADS-B data (OpenSky Network)
- TRL: 3

Differing considerably to manned aircraft, UAVs pose considerable challenges to national aviation authorities. In response to recommendations for information security controls introduced by the Federal Aviation Administration in the United States, Schumann *et al.* [53] have set reliability, responsiveness and unobtrusiveness as the key goals of R2U2, their on-board security monitoring framework. R2U2 aims to detect attacks in real-time by monitoring traffic on the flight computer and communication buses, including inputs from the GPS, the ground control station, sensor readings, actuator outputs, and flight software status. In terms of attacks, it looks for ill-formatted and illegal commands, dangerous commands that should not be run in-flight (e.g., "Reset Flight Software"), nonsensical or repeated navigation commands, and transients in GPS signals. It also monitors system behaviour, including oscillations of the aircraft around any of its axes, deviation from the flight path, sudden changes or consistent drifts of sensor readings, as well as memory leaks, real-time failures and other unusual software behaviour. The observations for each of these features are fed into a Bayesian network engine which determines the likelihood of different attack scenarios based on prior experiments. To minimise the overhead, R2U2 has been implemented on a re-configurable field-programmable gate array. Performance evaluation on a NASA DragonEye UAV

has produced promising results in detecting GPS spoofing, denial of service and malicious command injection.

- S: Semi or fully-autonomous UAV
- D/DA: Onboard (local) / Onboard self-detection
- AT/T: Behaviour / (Learning) Bayesian Network
- AF: (Cyber) Ill-formatted, nonsensical, repeated or dangerous commands, memory leaks and other unusual software behaviour, (Physical) oscillations, deviation from flight path, sudden changes in sensor readings
- AA: (Integrity) Command injection, GPS spoofing, (Availability) denial of service, GPS jamming
- EA: Experimental
- TRL: 5

In the same space, Birnbaum *et al.* [51] have focused on addressing hardware failure, malicious hardware and attacks against the flight control computer of a UAV. Their approach monitors features that allow both mechanical degradation and cyber attacks affecting flight control by identifying and tracking flight dynamics. The technique followed uses the *recursive least squares* statistical method to estimate actual UAV airframe and control parameter values, so as to then compare against corresponding nominal values specified beforehand. The feasibility of the approach was demonstrated in a hardware-in-the-loop fashion using the ArduPlane open source flight simulation platform flashed on an Arduino micro controller board for the plane autopilot system, and the Flight Gear open source simulator for the generation of the flight data.

- S: Semi or fully-autonomous UAV
- D/DA: Onboard (local) or external / Onboard self-detection or offloaded detection
- AT/T: Behaviour specification / Rule-based
- AF: (Physical) Roll, pitch, yaw, aileron, rudder, elevator, throttle
- AA: (Integrity) Tampered hardware, hardware failures, suspicious flight control behaviour
- EA: Simulation enhanced in a hardware-in-the-loop fashion
- TRL: 3

The same authors [54] have also argued that instead of looking at flight data in isolation it is preferable to learn to identify the events that correspond to them. For example, aggregating from several data points can help identify the elementary event "sharp left turn", and then detecting "incline" and "turn" can merge into the more complex "spiralling upward" event, and so forth, up to the definition of the UAV's mission. Then, detecting misbehaviour is a matter of checking to what extent the events identified in real-time deviate from the flight plan specified beforehand, in terms of both UAV states and timings. The authors' simulation results exhibited no false positives in the conditions evaluated, but the false negative rate increased considerably as the wind increased (from 3.3% without wind up to 14.4% for 10 m/s wind). Also, there was no provision for telling whether the UAV's misbehaviour were because of a cyber attack, unreliable sensor reading or other hardware failure. The simulation was based on the JSBSim flight simulator, an ArduPlane autopilot and software in the loop model, the MissionPlanner ground control station, the FlightGear visualisation system, and the authors' own Flight Analysis Engine.

- S: Semi or fully-autonomous UAV
- D/DA: Onboard (local) or external / Onboard self-detection or offloaded detection
- AT/T: (Hybrid) Knowledge-based identification of current state and behaviour specification based checking against specified flight plan / (Learning) Identification of current state, and (Rule-based) deviation from specified flight plan
- AF: (Physical) Roll, pitch, yaw, aileron, rudder, elevator, throttle
- AA: Unspecified
- EA: Simulation
- TRL: 3

Aircraft exhibiting full autonomy, such as UAVs, rely almost entirely on the correctness of the GPS signal and their sensing capabilities. Along these lines, Muniraj *et al.* [63] have proposed a framework for self-detecting GPS spoofing attacks onboard a UAV, using three anomaly detectors, based on the sequential probability ratio test, the cumulative sum, and binary hypothesis testing. To minimise the effect of uncertainties on detection accuracy, any attack indicators identified are fed to a Bayesian network. The initial learning for the anomaly detectors was developed based on a simulation dataset but can be re-tuned based on data from flight tests to improve their accuracy. The key assumption is that the sensors of the UAV that require no external input are not vulnerable to malicious interference and can be trusted, in contrast to GPS which cannot be trusted because it depends on an external signal. The IDS uses attack signatures, which correspond to abnormal behaviour in the time evolution of measurements on the trusted sensors, as well as anomaly detection using residuals based on GPS data and the output of a state estimator (an Extended Kalman Filter). The effectiveness of the approach has been assessed on a small fixed-wing UAV subjected to two types of GPS spoofing (with constant bias and with linearly increasing bias on the latitude measurements) in the presence of a variety of exogenous disturbances. The IDS was evaluated based on the data collected during the flights, but was not at the time implemented to run itself on the actual UAV.

- S: Fully-autonomous UAV
- D/DA: Onboard (local) / Onboard self-detection
- AT/T: Hybrid / (Learning) Sequential probability ratio test, cumulative sum and binary hypothesis testing, and a Bayesian network
- AF: (Physical) Body-axis velocities, angular rates, attitude angles, position and altitude
- AA: (Integrity) GPS spoofing
- EA: (Simulation) Attacks simulated using mathematical model, but based on actual flight data previously gathered on a fixed-wing UAV
- TRL: 4

One of the attractive benefits of using UAVs in a wide range of applications is that they can operate in teams, communicating with each other and sharing airspace according

to predefined rules, for example to maximise coverage or to avoid collisions. In the case of collision avoidance, there can be interaction rules, such as "turn left". The problem here is that a member of the team may misbehave, in the sense that it will not abide by such interaction rules, and the rest of the UAVs will need to detect this misbehaviour in time. Martini *et al.* [18] have worked on this challenge assuming the constraint that there can be no centralised mechanism for misbehaviour detection. Instead, the UAVs need to collaborate with each other, in this case using a a Boolean consensus protocol, where each UAV relies not only on its own sensors, but also on data shared by its neighbours to predict the allowed trajectories that an other UAV should follow if it abides by the agreed interaction rules. If it does not, then the other UAVs should detect it by noticing that its actual trajectory is among the ones predicted for it, and thus labelling it as uncooperative. The researchers have evaluated this method on a network of four UAVs (two real and two simulated), where one was uncooperative. Providing an implementation on an actual UAV network is very useful, even if two UAVs were virtual, but the evaluation has not taken into account different environmental conditions, percentages of uncooperative UAVs or realistic network conditions, which may not allow one-hop communication between any pair of UAVs and at any time.

- S: Autonomous UAV Network
- D/DA: Onboard (distributed) / Collaborative
- AT/T: Behaviour / Rule-based
- AF: (Physical) Onboard sensor data
- AA: (Integrity) Malicious tampering of flight control
- EA: Experimental
- TRL: 4

An interesting alternative approach is to use behaviour specification, such as in the work of Mitchell and Chen [52], which uses a behaviour rule state machine. Some of the attack states utilised were "weapons being armed while not in the target location", "thrust being over a threshold while in loitering mode", "gear being deployed while not near the airbase", "destination not belonging to a whitelist", etc. Using a modest range of values for each audit feature (e.g., only thrust being low, medium or high), the state machine produced consisted of 165 safe and 4443 unsafe states, with probabilities assigned for getting from one state to another. Then, each state was binary graded as completely safe or completely unsafe, and the measure of compliance to each behaviour rule was defined as the proportion of time being in safe states. The technique for deciding whether there is an attack or not was based on *maximum likelihood*. In their simulation, the false positive rate was 7.39% and the false negative rate varied from below 0.001 up to 44.3% depending on the sophistication of the attacker, as represented by a random attack probability parameter. This work was extended in [19], which emphasised on the flexibility of the approach on aiming for low false positives if targeted by low-impact attacks or low false negatives if targeted by more sophisticated attackers. Although theoretically very interesting, this approach has not been evaluated experimentally on an actual UAV, which is important because it has the significant disadvantage that it needs an extremely large number of states to accurately capture the behavioural specifications with greater granularity and for different environmental conditions. It also assumes that there is a "monitor node" in the vicinity, which observes the UAV at hand and runs its IDS externally.

- S: Semi or fully-autonomous UAV network
- D/DA: External / Collaborative
- AT/T: Behaviour specification / Rule-based
- AF: (Physical) 18 features, including altitude, rudder, destination, bank, pitch, yaw etc.
- AA: Attacks affecting confidentiality (e.g., mission data exfiltration) and integrity (e.g., unauthorised actuation and wasting energy to decrease endurance).
- EA: Simulation
- TRL: 3

Sedjelmaci *et al.* [28] have focused on civilian applications where UAVs explore an isolated zone to collect and transmit critical information to a ground station for analysis and decision processing. They have proposed a hierarchical intrusion detection scheme, which relies on two IDS mechanisms, one running at the UAV node level, and one running at the ground station level. The scheme combines knowledge (rule-based for each attack, running on each UAV) with behaviour-based detection (running at the ground station and based on support vector machines), with the aim to categorise each monitored UAV as normal, suspect, abnormal, or malicious. Monitoring can be in promiscuous mode, where a UAV acting as detection agent can hear all traffic within radio range and can observe UAVs traversing, and additionally in mutual monitoring mode, where each UAV monitors its neighbours. The authors have shown that their hierarchical scheme can outperform a fully distributed one, where ground stations are not involved, and does not incur considerable communication overhead. In their evaluation, which was based on NS-3 simulation, the false positive rate was consistently below 4%. The particular work has been extended in [25], where the focus was on the optimal next steps following detection, ejecting any node that is anticipated to commence an attack. Misbehaviour can be permanent if the node is always considered malicious, or transitory, where the node is considered malicious if the rate of switching into malicious mode is higher than the rate of switching to a normal mode. Whether a UAV node is ejected depends on the expected accuracy of the detection and the networking overhead that will be incurred, as addressed using game theory to optimally activate monitoring (not all UAVs perform monitoring) and optimally eject attackers (not all detected attackers are ejected) before they damage the network, subject to resource constraints of other network nodes.

- S: Autonomous UAV network
- D/DA: External and onboard (distributed) / Collaborative
- AT/T: (Hybrid) Knowledge-based at UAV level and behaviour-based at ground station level / Rule-based combined with learning (support vector machines)
- AF: (Physical) GPS Signal strength, consistency between neighbours' sensor value reports, (Cyber) number of packets sent, number of packets dropped, jitter, packet

TABLE III
COMPARATIVE ANALYSIS OF INTRUSION DETECTION SYSTEMS FOR AIRCRAFT

			Deployment		Architecture			Type		Features		Techn.		Attacks on			Evaluation			
Ref.	Year	Scope	Onboard	External	Self-detection	Collaborative	Offloaded	Knowledge	Behaviour	Cyber	Physical	Learning	Rule-based	Confid/ility	Integrity	Availability	Analytical	Simulation	Experimental	TRL
[57]	2010	Aircraft with ADS-B	✓	✗	✓	✗	✗	✗	✓	✓	✗	✓	✗	✗	✓	✗	✗	✗	✓	2
[58]	2015	Aircraft with ADS-B	✓	✗	✓	✗	✗	✗	✓	✗	✓	✓	✗	✗	✓	✗	✗	✓	✗	3
[53]	2015	UAV	✓	✗	✓	✗	✗	✗	✓	✓	✓	✓	✗	✗	✓	✓	✗	✗	✓	5
[51]	2014	UAV	✓	✓	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗	✗	✓	✗	3
[54]	2015	UAV	✓	✓	✓	✗	✓	✓	✓	✗	✓	✓	✓	✗	?	?	✗	✓	✗	3
[63]	2017	UAV	✓	✗	✓	✗	✗	✓	✓	✗	✓	✓	✗	✗	✓	✗	✗	✓	✓	4
[18]	2015	UAV network	✓	✗	✗	✓	✗	✗	✓	✗	✓	✗	✓	✗	✓	✗	✗	✓	✓	4
[52], [19]	2014	UAV network	✗	✓	✗	✓	✗	✗	✓	✗	✓	✗	✓	✓	✓	✗	✗	✓	✗	3
[28]	2017	UAV network	✓	✓	✗	✓	✗	✓	✓	✓	✓	✓	✓	✗	✓	✓	✗	✓	✗	3

round trip time, and each UAV's history as detection agent
- AA: (Integrity) False information dissemination, GPS spoofing, (Availability) jamming, and black/greyhole attacks
- EA: Simulation
- TRL: 3

Table III summarises the characteristics of the different IDSs proposed for aircraft. It is worth observing that all follow behaviour-based or hybrid approaches, because even though determining what is a normal state for an aircraft is very challenging, researchers have found it even more impractical to rely solely on signature patterns of known attacks.

B. Land vehicles

Research on IDS for land vehicles has focused on robotic land vehicles, and automobiles, including driverless vehicles and vehicular networks. Note that up to now, all current research on intrusion detection for driverless vehicles ([29], [26], [27], [64], [65], [66]) has been addressed from the perspective of vehicular networks, whether as platoon networks or as networks of individual driverless vehicles, and as such is included in the corresponding subsection.

1) Robotic land vehicles: Such vehicles (Table IV) are particularly attractive for research, because they have a large variety of applications, from surveillance, to emergency response and defence-oriented missions, as well as because they are often less expensive to purchase or develop and easier to conduct experiments with in the constrained physical spaces typically afforded to researchers. An example is the work by Vuong *et al.* [30], [31], [32], who have used a small 4-wheel drive robotic vehicle controlled via an on-board Intel Atom computer running Linux, and have subjected it to denial of service, false data injection and malware attacks. The vehicle's onboard detection method is based on decision trees with a training phase that involves learning the signatures of a range of attacks based on their impact on a set of both cyber and physical features. Given the nature of the attacks, it is not surprising that the cyber features have proven to be the most relevant, especially the network-related ones, but the authors have also shown that introducing physical features too, such as battery consumption and physical vibration of the chassis, noticeably increases the detection accuracy and reduces the detection latency. An example physical manifestation of a cyber attack that was observed in the particular case was a minute physical vibration caused by the vehicle continuously losing network connection to its remote controller and having to enter fail-safe mode for extremely short periods of time. Having used a real robotic vehicle in the evaluation is significant, but the particular were run with the vehicle on stands for reproducibility and to minimise environmental effects. Also, accuracy varied considerably between different attacks. Indicatively, the false positive rate was only 5.4% for malware, but reached as high as 29.6% for command injection, and similarly the false negative rate for command injection was only 5.7%, but reached as high as 41.4% for denial of service.

- S: Remote-controlled robotic vehicle
- D/DA: Onboard (local) or external / Onboard self-detection or offloaded detection
- AT/T: Knowledge / (Learning) Rules generated by decision trees
- AF: (Cyber) CPU consumption, network traffic, disk usage, (Physical) encoder value for each wheel, vibration and power consumption
- AA: (Integrity) Command injection and malware, (Availability) denial of service
- EA: Experimental
- TRL: 4

Along the same lines, Bezemskij *et al.* [55], [68] have also shown that it is highly beneficial to use both cyber features and physical features, and have additionally placed emphasis on

TABLE IV
COMPARATIVE ANALYSIS OF INTRUSION DETECTION SYSTEMS FOR ROBOTIC LAND VEHICLES

			Deploym.		Architecture			Type		Features		Techn.		Attacks on			Evaluation			
Ref.	Year	Scope	Onboard	External	Self-detection	Collaborative	Offloaded	Knowledge	Behaviour	Cyber	Physical	Learning	Rule-based	Confid/ility	Integrity	Availability	Analytical	Simulation	Experimental	TRL
[30]-[32]	2015	R/C robotic vehicle	✓	✓	✓	✗	✓	✓	✗	✓	✓	✓	✗	✗	✓	✓	✗	✗	✓	4
[55], [15]	2017	Semi-auto robotic veh.	✓	✗	✓	✗	✗	✓	✓	✓	✓	✓	✗	✗	✓	✓	✗	✗	✓	5
[46]	2015	Auton. vehicle platoon	✓	✗	✓	✗	✗	✗	✓	✗	✓	✓	✗	✗	✓	✗	✗	✗	✓	3
[13], [67]	2018	R/C robotic vehicle	✗	✓	✗	✗	✓	✓	✓	✓	✓	✓	✗	✗	✓	✗	✗	✗	✓	4

the processing and memory efficiency requirements for implementing IDS on a resource-constrained vehicle. For this, they have proposed onboard detection mechanisms to first monitor data related to four cyber (communication and computation) and 13 physical (actuation and sensing) indicators of the robot in real-time and then using lightweight heuristic techniques decide whether a vehicle is in an attack state or not. As an extension, in [15], they have presented a method based on Bayesian Networks to additionally determine the domain (cyber or physical) from which the attack originated from. Using a purpose-built 4-wheel drive semi-autonomous robotic vehicle following the military-oriented Generic Vehicle Architecture [69], they have shown the feasibility of the approach for most attacks that the vehicle has been subjected to. Represented in the form of Receiver Operating Characteristics graphs, their experimental results yielded area under the curve of 0.995 for attacks coming from the cyber domain and 0.953 or attacks coming from the physical domain.

- S: Semi-autonomous robotic vehicle
- D/DA: Onboard (local) / Onboard self-detection
- AT/T: (Hybrid) Behaviour-based detection of attack, followed by knowledge-based identification of domain of origin / (Learning) Bayesian network
- AF: (Cyber) packet arrival time, action indicator, sequence number, packet rate, (Physical) Battery voltage, pitch, roll, temperature, compass bearing, distances, motors
- AA: (Integrity) False data injection, replay attack, rogue node, physical compass manipulation, (Availability) Denial of service
- EA: Experimental
- TRL: 5

Autonomous vehicles are almost entirely dependent on the robustness of their sensing processes. This makes them particularly attractive targets to sensory channel attacks and network-based false data injection attacks that affect the integrity or availability of a vehicle's sensor data, for instance to disrupt its collision avoidance subsystem. One approach that is commonly used to detect attacks on sensors is to treat them as standard sensor failure events and utilise statistical anomaly detection methods. For example, if it can be assumed that the rate of change of a sensor's data cannot exceed a particular value, then the recursive least-square filter can be used to discard data that do. Gwak *et al.* [46] have demonstrated this approach on small robotic vehicles operating as a platoon, and using a simple obstacle avoidance system that is limited to only ultrasonic sensors and does not have the luxury of cross-checking between different types of sensing. The simple approach followed is that if a sensor's data is deemed to be unreliable, the particular sensor is excluded from the collision avoidance processes. However, in terms of the origin of a sensor's failure, there is no provision to distinguish between malicious threats of cyber origin and natural sensor failures, making this work rather impractical in this context.

- S: Fully autonomous robotic vehicle (as part of platoon)
- D/DA: Onboard (local) / Onboard self-detection
- AT/T: Behaviour / Learning
- AF: (Physical) Sensor values
- AA: (Integrity) Sensor spoofing
- EA: Experimental
- TRL: 3

Contrary to previous approaches that prioritise lightweight approaches, Loukas *et al.* [13], [67] have shown that very accurate, but also computationally heavy machine learning algorithms, such as deep learning, can be used if the detection task is offloaded to a more powerful infrastructure, such as a remote server or cloud. The authors argue that computation offloading can be extremely useful for demanding, real-time and continuous tasks required by resource-constrained and time-critical cyber-physical systems. To demonstrate the effectiveness of offloading, they have conducted experimental evaluations, which reduced both the detection latency and the energy consumption for a particular robotic vehicle. Of course, it has the drawback that it depends on the availability of an offloading infrastructure, which is impractical in many application areas of robotic vehicles. The authors have presented a mathematical model, which predicts the benefits in terms of energy consumption and detection latency based on the complexity of the deep learning processing required, the processing capabilities of the vehicle and the offloading infrastructure, and the performance of the network. The differences in configuration and condition of the latter were emulated using wide

area network emulation software. Obviously, the more reliable the network and the more demanding the processing, the more useful the offloading becomes. Comparison between different deep learning and standard machine learning classification approaches (decision trees, support vector machines, logistic regression, random forest) showed experimentally that recurrent neural networks enhanced with long short term memory can greatly improve detection accuracy. The authors have reported average accuracy of 86.9% across three different attack types, up from a best of 79.9% achieved by the second best, which was support vector machines, for the same attacks.

- S: Remote-controlled robotic vehicle
- D/DA: External / Offloaded detection
- AT/T: Knowledge / (Learning) Deep learning in recurrent neural network architecture
- AF: (Cyber) CPU consumption, network traffic, disk usage, (Physical) encoder value for each wheel, vibration and power consumption
- AA: (Integrity) Command injection
- EA: (Experimental) evaluation with real attacks on real vehicle but emulated network conditions
- TRL: 4

2) Automobiles: The vast majority of large-scale automotive security research projects have focused on cryptographic approaches for ensuring authenticity, integrity, confidentiality and privacy [70]. In recent years, researchers have also started looking into intrusion detection for automobiles' in-vehicle networks (Table V), mostly in relation to the Controller Area Network (CAN), which is the most prevalent of the related protocols. Here, the challenge is that CAN is a broadcast protocol which does not require unique identifiers for the various electronic control units (ECUs). This impedes network-based intrusion detection and facilitates attacks that exploit anonymity, such as denial of service and node masquerading.

One approach is to use behaviour specification with particular detection rules checked on each ECU. For instance, in the first relevant IDS in the literature proposed in 2008, Larson *et al.* [42] have defined detection rules based on the specifications of both the network protocol (individual, dependent and inter-object fields of a message) and the behaviour of each ECU (message transmission, message reception, and rates of message transmission and reception). Their rather insightful observation was that gateway devices are more critical for the security of the in-vehicle network than other ECUs, because they require more complex intrusion detection rulesets, and if compromised, they would allow a more diverse range of attacks to be performed. Indeed, it was a gateway device (the multimedia interface) that was exploited a year later in [21] in the first publication detailing high-impact cyber-physical attacks on a conventional automobile. Larson *et al.* also observed that in most cases, a single ECU is not able to detect an attack, and that cooperation between multiple ECUs is needed. However, the particular IDS proposed was presented at conceptual level and was not evaluated in simulation or experiments with real vehicles.

- S: Automobile using CAN bus
- D/DA: Onboard (local) / Onboard self-detection
- AT/T: Behaviour specification / Rule-based.
- AF: (Cyber) Message structure and content and ECU object directory communication parameters
- AA: Confidentiality, integrity and availability breaches at individual ECU or gateway level
- EA: Analytical
- TRL: 2

Also in 2008, Hoppe *et al.* [71] demonstrated proof of concept cyber-physical attacks through exploitation of CAN bus. These included preventing the actuation of the warning lights, disabling the airbag control module, and malicious code automatically issuing an "open driver window" command every time a "close drive window" command is transmitted. The authors observed particular patterns on the network corresponding to each attack, which they proposed to use in an IDS. Some examples include increased message frequency, misuse of message IDs, and communication characteristics at the physical layer, such as the degree of signal attenuation, the shape of clock edges and propagation delays. In [72], they progressed with evaluation of their IDS concept on a single attack (suppressing the warning lights) and using only two audit features (the current frequency and content of the last eight messages to the targeted ECU). Although perhaps too simple and too limited in scope, this was the first actual implementation of IDS for a vehicle.

- S: Automobile using CAN bus
- D/DA: Onboard (local) / Onboard self-detection
- AT/T: Behaviour specification / Rule-based.
- AF: (Cyber) Message frequency, content of last eight messages
- AA: (Integrity) Command injection
- EA: Experimental
- TRL: 4

Miller and Valasek [73] have also focused on message frequency as an audit feature and have produced a prototype IDS device, which can be attached to an automobile's onboard diagnostics port to detect attacks based on only this feature. The rationale is that ECUs communicate with each other continuously and at a relatively predictable rate. So, any maliciously injected message will increase the rate of messages received. So, the particular IDS learns normal message rates and determines that there is an anomaly if the message rate measured is considerably higher, that is 20 - 100 times higher in their experiments. The fact that there is a working prototype of IDS based on a single feature is indicative of how straightforward detection can be for some types of attacks, and hence there is little excuse for the complete absence of intrusion detection in production automobiles. Of course, for attacks that are more sophisticated or involve no significant change in message frequency, there is a need for equally sophisticated intrusion detection.

- S: Automobile using CAN bus
- D/DA: Onboard (local) / Onboard self-detection
- AT/T: Behaviour / Learning.
- AF: (Cyber) CAN Message frequency
- AA: (Integrity) Command injection
- EA: Experimental

TABLE V
COMPARATIVE ANALYSIS OF INTRUSION DETECTION SYSTEMS FOR AUTOMOBILES

			Deploym.		Architecture			Type		Features		Techn.		Attacks on			Evaluation			
Ref.	Year	Scope	Onboard	External	Self-detection	Collaborative	Offloaded	Knowledge	Behaviour	Cyber	Physical	Learning	Rule-based	Confid/lity	Integrity	Availability	Analytical	Simulation	Experimental	TRL
[42]	2008	Automobile CANbus	✓	✗	✓	✗	✗	✗	✓	✓	✗	✗	✓	✓	✓	✓	✓	✗	✗	2
[71], [72]	2009	Automobile CANbus	✓	✗	✓	✗	✗	✗	✓	✓	✗	✗	✓	✗	✓	✗	✗	✗	✓	4
[73]	2014	Automobile CANbus	✓	✗	✓	✗	✗	✗	✓	✓	✗	✓	✗	✗	✓	✗	✗	✗	✓	4
[74]	2017	Automobile CANbus	✓	✗	✓	✗	✗	✗	✓	✓	✗	✓	✗	✗	✓	✗	✗	✓	✗	3
[37]	2016	Automobile CANbus	✓	✗	✓	✗	✗	✓	✓	✓	✗	✓	✗	✗	✓	✓	✗	✓	✗	3
[44]	2017	Automobile CANbus	✓	✗	✓	✗	✗	✗	✓	✗	✓	✓	✗	✗	✓	✗	✗	✗	✓	3
[75]	2017	Automobile CANbus	✓	✗	✓	✗	✗	✗	✓	✓	✗	✗	✓	✗	✓	✓	✗	✗	✓	4
[36]	2011	Automobile CANbus	✓	✗	✓	✗	✗	✗	✓	✓	✓	✓	✗	✗	✓	✓	✗	✗	✓	4
[43]	2017	Automobile CANbus	✓	✗	✓	✗	✗	✗	✓	✓	✗	✓	✗	✗	✓	✗	✗	✓	✗	3
[41]	2016	Automobile CANbus	✓	✗	✓	✗	✗	✗	✓	✓	✗	✗	✓	✗	✓	✓	✓	✗	✗	2
[48]	2016	Automobile CANbus	✓	✗	✓	✗	✗	✗	✓	✗	✓	✓	✗	✗	✓	✗	✗	✓	✗	3
[45], [49]	2016	Automobile CANbus	✓	✗	✓	✗	✗	✗	✓	✗	✓	✓	✗	✗	✓	✓	✗	✗	✓	5
[76]	2016	Automobile CANbus	✓	✗	✓	✗	✗	✗	✓	✓	✗	✓	✗	✗	✓	✓	✗	✓	✗	3
[47]	2017	Automobile CANbus	✓	✗	✓	✗	✗	✓	✗	✓	✗	✓	✗	✗	✓	✓	✗	✓	✗	3
[77]	2016	Automobile CANbus	✓	✗	✓	✗	✓	✗	✓	✗	✓	✓	✗	✗	✓	✗	?	?	?	?
[78]	2017	Automobile CANbus	✓	✗	✓	✗	✗	✗	✓	✓	✗	✓	✓	✗	✗	✗	✗	✓	✗	2
[79]	2016	Automobile CANbus	✓	✗	✓	✗	✗	✓	✗	✓	✗	✓	✗	✗	✓	✗	✗	✓	✗	3
[80]	2015	Plugin electric Veh.	✗	✓	✗	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗	✗	✓	✗	3

- TRL: 4

Moore et al. [74] have also focused on the regularity of messages on CAN bus and specifically observed that with the vehicle engine being on, the majority of process IDs' signals are regularly occurring, i.e. repeatedly, at a fixed rate and with little noise. So, the authors have built a model for each process ID's signal stream as a Markov process. If the inter-signal arrival time is too short or too long in comparison to a learned value (plus/minus a predefined 15% of the absolute error from expectation), then this is flagged as an anomaly and an alert is raised when three consecutive anomalies are detected.

- S: Automobile using CAN bus
- D/DA: Onboard (local) / Onboard self-detection
- AT/T: Behaviour / (Learning) in the form of Markov process models
- AF: (Cyber) CAN inter-signal arrival times
- AA: (Integrity) Regular-frequency signal injection
- EA: Simulation
- TRL: 3

Along the same lines, Song *et al.* [37] have also based their IDS for CAN in-vehicle networks on the message frequency, but in a hybrid fashion, detecting both deviation from normal behaviour and known signatures of attacks. The rationale is that if the time interval of a new message is shorter than what is deemed to be normal, then this is evidence of message injection, and if it is considerably shorter, then this is evidence of denial of service. By way of evaluation, they have tried three types of message injection attacks (injecting messages of single CAN ID, injecting random or pre-ordered messages of multiple CAN IDs, and injecting massive rates of messages in the form of denial of service). The dataset used was normal speed driving of a production automobile for 40 minutes. The attacks involved injecting messages 30 times for 5-10 s each. Afterwards, 100 one-min samples were chosen randomly and were separated into normal and attack, depending on whether they contained attack messages. The IDS then determines that there is a message injection attack if the message frequency is above double what has been learned to be normal, and that there is a denial of service if it is above five times the normal. Importantly, the particular method achieved 0% false positive and 0% false negative rates for the particular vehicle and the particular configuration of detection rules.

- S: Automobile using CAN bus
- D/DA: Onboard (local) / Onboard self-detection
- AT/T: Hybrid / Learning.
- AF: (Cyber) CAN message frequency
- AA: (Integrity) message injection, (Availability) denial of service
- EA: (Simulation) Learning of what is normal based on real vehicle, but attacks were offline

- TRL: 3

In CAN, receiver nodes may require certain kinds of information to run a given task, and for this reason, they need to broadcast a *remote frame* on the bus, which typically has the identifier of its target ECU. Ansari *et al.* [44] have proposed an approach which uses the principle of self-identifier violation. It assumes that frames with a high value in the Remote Transmission Request (RTR) flag are remote frames. If γ is the CAN ID of a node, any frame that is not a remote frame received from another CAN node with CAN ID γ is assumed to be a masquerade or replay attack. This detection decision is then broadcast on the CAN Bus. The simplicity of the approach has an obvious advantage in speed, with the authors' experiments showing detection latency as low as 40μs, which is important considering automobiles' very strict real-time requirements. Their IDS module was implemented as part of a CAN controller on a prototype CAN system produced in the laboratory. The modification proposed is compliant with the CAN protocol.

- S: Conventional automobiles
- D/DA: Onboard (local) / Onboard self-detection
- AT/T: Behaviour specification / Rule-based
- AF: (Cyber) CAN ID
- AA: (Integrity) Masquerade and replay attacks
- EA: (Experimental) on a CAN system prototype with synthetic vehicle behaviour
- TRL: 3

Lee et al. [75] have developed OTIDS, which is an IDS based on the observation that in normal cases of remote frame, every ECU has a fixed response offset ratio and time interval between request and response, and that these values vary when under attack. The detection decision is then taken based on whether the average time intervals are out of range, as specified by predefined thresholds, or the Pearson correlation coefficient between offsets and time intervals is under a threshold. For evaluation, they have developed a prototype based on Raspberry Pi 3 with PiCAN2 shield and a KIA Soul. Importantly, the authors have released the datasets[1] they developed as part of this work for others to use in their research.

- S: Automobile using CAN bus
- D/DA: Onboard (local) / Onboard self-detection
- AT/T: Behaviour / Rule-based.
- AF: (Cyber) Message response offset ratio and frequency
- AA: (Integrity) Command injection and false data injection (impersonation and fuzzy attacks), (Availability) denial of service
- EA: (Experimental) Attacks using Arduino with CAN shield, and detection based on Raspberry Pi3 implementation partially integrated on a KIA Soul.
- TRL: 4

Instead of the message frequency in an in-vehicle network, Müter and Asaj [36] turned their attention to their randomness. The logic is that, unlike network traffic in computer networks, in-vehicle network traffic exhibits less and somewhat predictable randomness. Timings, message lengths and types of packets are highly predictable. So, their assumption was that a significant change in entropy is a sign of potential malicious activity. Based on this, their proposed IDS collects data at the level of individual bits, fixed size groups of bits, signals and protocols, and uses a variety of metrics from information theory, including conditional self-information (how much information has been transferred with a message), entropy (the expected value of self-information), and relative entropy for measuring the distance between two datasets. Another important dimension is the status of the vehicle, as the number of messages expected is much lower when the vehicle is not moving than when it is, so what is a normal value for entropy needs to be learned for every possible vehicle status. Evaluation on a real vehicle showed that for attacks that involve flooding or repeating messages, entropy can indeed be very helpful. This is expected, especially for simple flooding attacks where the attacker does not inject randomness in the type, rate or source of traffic used. What is also expected, and was shown in the authors' experiments is that false data injection cannot be detected unless the data injected were highly unrealistic (e.g., injecting a 70 mph speed value immediately after a 30 mph value).

- S: Automobile using CAN bus
- D/DA: Onboard (local) / Onboard self-detection
- AT/T: Behaviour / (Learning) entropy in non-attack conditions and comparing against learned value
- AF: (Cyber) Network traffic entropy, (Physical) Sensor value entropy
- AA: (Integrity) Increased message frequency, command injection, (Availability) Message flooding
- EA: Experimental
- TRL: 4

Marchetti and Stabili [43] have placed their focus on the CAN message ID sequences for detecting malicious message injections. In a training phase, the IDs of all frames captured on a vehicle's CAN bus are stored in the form a transition matrix, which contains all legitimate transitions between the message IDs of two consecutive CAN messages. From then on, the matrix can be considered as a whitelist, and sequence analysis can be based on comparing against it. Evaluation in simulation has shown that this approach's detection percentage can reach 95% for attacks that involve two or more simple message injections per second, but drops below 40% for replay attacks. Also, analysis of the memory and computational requirements of the approach has shown that integration in a real vehicle's ECUs should be practical, but this has not yet been confirmed with a real implementation. The authors have suggested that further improvements in efficiency can be achieved by decentralising the mechanism, to run detection on one ECU per network branch, rather than on a gateway.

- S: Automobile using CAN bus
- D/DA: Onboard (local) / Onboard self-detection
- AT/T: Behaviour / Learning
- AF: (Cyber) CAN bus messages
- AA: (Integrity) Replay, command injection and false data injection
- EA: (Simulation) of attack conditions, but normal be-

[1] http://ocslab.hksecurity.net/Dataset/CAN-intrusion-dataset

haviour collected from real production automobile
- TRL: 3

Boudguiga *et al.* [41] have developed an IDS model for detecting the types of attacks where an attacker impersonates a legitimate ECU by forging or replaying legitimate CAN frames. The model suggests a CAN extension, where every legitimate ECU registers itself periodically with other ECUs, and from then on checks with each ECU register whether any data frames have been sent containing its own identifier. The authors have assumed that each ECU has an embedded hardware security module dedicated for cryptographic computation and key storage, which is the case for newer automotive microcontrollers, and allows authentication of each ECU to other ECUs. The decision to determine that there is an attack is based on whether the number of violations detected exceed a threshold. Such a provision would indeed help protect against a range of impersonation, denial of service, but not against isolation attacks preventing traffic to reach the targeted ECU, because the proposed IDS relies on the targeted ECU checking the bus itself. Evaluation of the approach was based on security analysis.

- S: Automobile using CAN bus
- D/DA: Onboard (local) / Onboard self-detection
- AT/T: Behaviour specification / Rule-based
- AF: (Cyber) CAN frame identifiers
- AA: (Integrity) Impersonation, replay, and (Availability) denial of service
- EA: Analytical
- TRL: 2

Narayanan *et al.* [48] have analysed message streams from different ECUs as sequences of events, which they have formulated into a time series machine learning problem and used Hidden Markov Models to generate a model of normal behaviour. That is because the physical movement of a vehicle can be considered as a sequence of states that are dependent on the previous state. Data collection was based on vehicles from different automotive manufacturers, and included speed, load, engine coolant temperature and other physical sensor values. The features used to train the model were their gradients rather than their absolute values. The authors collected CAN message data from real vehicles and used Hidden Markov Models (HMMs) to generate a model for the prediction of anomalous states in vehicles. Upon detecting unsafe and anomalous states while monitoring CAN messages, the proposed technique aims to issue alerts while the vehicle is in operation. Matlab simulations have shown very high accuracy in detecting false sensor values or unsafe states.

- S: Automobile using CAN bus
- D/DA: Onboard (local) / Onboard self-detection
- AT/T: Behaviour / (Learning) Hidden Markov Models
- AF: (Physical) Speed, load, engine coolant temperature, engine RPM, intake air temperature, absolute throttle position and O2 voltage
- AA: (Integrity) False data injection
- EA: (Simulation) Matlab, using the normal behaviour data from three automobiles
- TRL: 3

Cho and Shin [45] have proposed a behaviour-based clock-based IDS (CIDS), which takes into account the intervals of periodic in-vehicle messages for fingerprinting ECUs. These are used for constructing a baseline of ECUs' clock behaviours with the Recursive Least Squares algorithm. In practice, an ECU's clock skew is its fingerprint. CIDS then uses Cumulative Sum to detect small persistent changes, which are assumed to be signs of intrusion. This allows quick identification of in-vehicle network intrusions with a low false positive rate of 0.055%, as measured experimentally on a 2013 Honda Accord and on data from another two vehicles. The authors have argued that it is not enough to detect that there is an attack on the CAN bus, and that a detection system needs to also identify from which exact ECU the attack originates from, so as to facilitate response to the detected attack or facilitate forensics. The fingerprinting of ECUs provided by CIDS can help in this direction too, but the same authors have produced a more specialised solution for attacker identification in [49]. They have shown that it is possible to pinpoint the attacker ECUs by monitoring their voltage profiles, which can be sufficiently unique. To evaluate the practicality of this approach, they have produced *Viden*, a prototype implementation, which first determines whether the measured voltage signals come from the genuine transmitter ECU, then constructs the voltage profiles for each transmitter ECU to be used as their fingerprints, and uses these to identify the compromised ECU, when an attack is detected. *Viden* takes into account both the momentary behaviour of the voltage outputs and its trend. Experimental evaluation on two real vehicles has yielded a false identification rate of only 0.2%. However, Viden can only work well if the compromised ECU transmits at least one message.

- S: Automobile using CAN bus
- D/DA: Onboard (local) / Onboard self-detection
- AT/T: Behaviour. (Learning) a parameter reflecting the number of standard deviations intended to detect
- AF: (Physical) Timing and voltage measurements
- AA: (Integrity) ECU masquerading, (Availability) rushing attack and isolation/suspension of targeted ECU
- EA: Thorough experimental evaluation on a CAN bus prototype and real vehicles
- TRL: 5

Taylor et al. [76] have proposed a method that uses a Long Short-Term Memory (LSTM) neural network to predict the next bits expected by a sender on the CAN bus. Any next bits to appear that are deemed to be highly "surprising" (forming sequences of bits that have never been seen before or are seen very rarely) are assumed to be anomalies due to malicious attacks. The introduction of a LSTM block can help "remember" values over arbitrary time intervals, which makes it very useful for predicting in the presence of time lags of unknown size and duration. Evaluation was based on real-world data for normal behaviour from a 2012 Subaru Impreza and synthetic data for attacks that were created according to the related literature, including adding messages, erasing messages, replaying messages and modifying the contents of messages.

- S: Automobile using CAN bus
- D/DA: Onboard (local) / Onboard self-detection
- AT/T: Behaviour / (Learning) Neural network enhanced with LSTM
- AF: (Cyber) CAN data sequences
- AA: (Integrity) False data injection, replay (Availability) ECU suppression
- EA: (Simulation) based on data from a 2012 Subaru Impreza and synthetic attack data
- TRL: 3

Martinelli et al. [47] have argued that normal CAN messages that are triggered by human action can be modelled well by fuzzy techniques. So, they have formulated the problem as a fuzzy classification problem and have applied four fuzzy classification algorithms to distinguish between legitimate CAN messages generated as a result of action taken by the human driver and injected ones generated by an attacker ECU. As features, they have used a specific eight bytes from the CAN frames, and the evaluation was carried out offline based on the KIA Soul dataset offered by Lee et al. [75]. The performance was generally high across most types of data injection attacks tried, with, indicatively, their fuzzy NN algorithm's precision ranging from 0.963 to 1 for injection attacks.

- S: Automobile using CAN bus
- D/DA: Onboard (local) / Onboard self-detection
- AT/T: Knowledge / (Learning) Four fuzzy classification algorithms
- AF: (Cyber) Eight bytes selected from the CAN frame
- AA: (Integrity) false data injection, (Availability) denial of service
- EA: Simulation
- TRL: 3

Malinowski *et al.* [77] have filed a patent on a monitoring and analysis system for detecting both malicious activity and harmful hardware/software modifications to a vehicle. The proposed IDS engine looks for inconsistencies when receiving emergency conditions from the vehicle's sensors, by comparing the processed output of one of the sensors to the unprocessed observed value, so as to detect malware attacks that may not have the ability to affect the unprocessed value (e.g., an input to the sensor). The patent specifies that artificial intelligence can be used to determine that an emergency state has been declared maliciously and is incorrect, but does not detail how, and due to the nature of the publication, no evaluation results have been disclosed, and there is no indication as to how malware will be differentiated from other types of misbehaviour or natural faults. As a result, it is not possible to evaluate the maturity of the approach. Interestingly, the design suggests that detection can run onboard or offloaded to a cloud computing system.

- S: Automobile using CAN bus
- D/DA: Onboard (local) / Onboard self-detection or Offloaded detection
- AT/T: Behaviour / Learning
- AF: (Physical) Sensors measurements, such as oxygen, throttle position and tyre pressure
- AA: (Integrity) Malware affecting integrity of systems or sensor values
- EA: Unknown
- TRL: Unknown

In [78], Markovitz and Wool have described an IDS, which first identifies the boundaries and field types of the 64-bit CAN messages of each ECU, and based on this builds a model for these messages, based on Ternary Content-Addressable Memory (TCAM). TCAM is a special type of high-speed memory usually used for fast look-up tables and packet classification in switches and routers. The rationale is that the positional bit fields of CAN messages make them easy to represent as TCAMs. For each ECU, a TCAM database of normal traffic patterns is constructed and used to detect messages that do not match the TCAM-based model. The authors have evaluated their system using an ECU traffic simulator that they have developed. In their experiments, it was able to detect irregular changes in CAN bus messages with a false positive rate that did not exceed 2.5%, but it has not been evaluated against specific attacks.

- S: Automobile using CAN bus
- D/DA: Onboard (local) / Onboard self-detection
- AT/T: Behaviour / Learning
- AF: Cyber and Physical features, as represented in the different CAN fields
- AA: No attack model was implemented
- EA: (Simulation) using synthetic CAN data
- TRL: 3

In contrast to almost all other IDSs designed for CAN, which opt for very lightweight behaviour-based approaches, Kang *et al.* [79] have proposed the use of a Deep Neural Network in a knowledge-based fashion. Their neural network is trained on high-dimensional CAN frame data to figure out the underlying statistical properties of normal and attack CAN frames and extract the corresponding features. After the very lengthy training has been completed offline, the IDS monitors the frames transmitted in the network to decide whether it is under attack or not. Though very promising from the perspective of detection accuracy, deep neural networks are computationally heavy, and such an IDS is challenging to integrate in a real vehicle, especially if it is meant to operate continuously. In the particular case, the authors have used a deep neural network with a small number of layers, so as to keep complexity low and still have acceptable detection accuracy. For 5 hidden layers, the false positive and false negative rates were measured around 2%.

- S: Conventional automobiles
- D/DA: Onboard (local) / Onboard self-detection
- AT/T: Knowledge / (Learning) Deep neural networks
- AF: (Cyber) CAN bus frames extracted as binary bitstream
- AA: (Integrity) Message injection
- EA: (Simulation) Evaluation runs on a PC and using the OCTANE CANbus sniffer and injector [81]
- TRL: 3

A very different problem has been tackled by Abedi *et al.* [80], who have focused on the security of charging of plug-in

electric vehicles and specifically false data injection attacks in relation to energy measurement reporting in the smart grid. For this, they have used two approaches. The first is model-based, making use of the chi-square distribution test to detect whether there is an attack and the largest normal residual test to identify what data it has affected. The variable with the largest measurement residual is assumed to be the suspicious one. The second approach is signal-based, using discrete wavelet transform for timeline analysis, where the detail coefficient values are compared with predefined thresholds to detect anomalies.

- S: Plug-in electric vehicle
- D/DA: External / Offloaded detection
- AT/T: Behaviour specification / Rule-based
- AF: (Physical) Smart meter data, such as line active/reactive power flows
- AA: (Integrity) False data injection
- EA: Simulation
- TRL: 3

3) Automobile vehicular networks: Increased automation in vehicles is followed by increased use of vehicular networks, especially for automobiles, which raises the question of what happens when one of the vehicles is compromised or launches cyber attacks on neighbouring vehicles. Here, we review representative approaches for intrusion detection in vehicular ad hoc networks (VANETs). We have prioritised IDSs that have been evaluated against specific technical breaches of cyber security. For completeness, we also include a small number of representative examples of misbehaviour detection systems (MDSs), which consider cyber security breaches, but do not distinguish against other unidentified misbehaviour attributed to normal failures, physical attacks or selfish drivers wilfully disseminating false information [50]. For more complete surveys of general MDSs in VANETs, we refer the reader to [82] and [11]. Here, our emphasis is on the audit techniques and features used for the detection of the attacks rather than the reputation, trust-oriented or cluster-head selection algorithms.

While there is little doubt that the future for driverless vehicles is promising, what it will exactly look like is still uncertain, and as a result, researchers need to make assumptions as to how the interactions between them will be affected by security breaches. Alheeti et al. [29] have focused on driverless and semi-driverless vehicles communicating warning messages and cooperative awareness messages between each other in a vehicular ad hoc network (VANET). Here, the challenge is to detect greyhole and rushing attacks which aim to disrupt the communication between vehicles and with roadside units. The proposed approach's training and testing was based on machine learning (support vector machines and feedforward neural networks), but the authors have also employed fuzzification for the pre-processing stage, so as to increase the detection rate and reduce false positives. Evaluation based on NS-2 simulations yielded false positive rate of 1.21% and false negative rate of 0.23%. This work was extended in [26], which uses linear discriminant analysis and quadratic discriminant analysis. The evaluations included different types of mobility models (urban, highway and rural), with rushing attacks and greyhole attacks in [29], and denial of service and blackhole attacks in [26] in networks of 30-40 vehicles on two-lane roads.

- S: Autonomous and semi-autonomous VANET
- D/DA: Onboard (local) / Onboard self-detection
- AT/T: Behaviour / (Learning) Support Vector machine and feedforward neural network in [29], and Linear and Quadratic Discriminant Analysis in [26]
- AF: (Cyber) up to 21 features, incl. payload size, packet ID, source, destination, hop counts, etc.
- AA: (Availability) Greyhole, rushing, blackhole and denial of service
- EA: (Simulation) NS-2
- TRL: 3

In [27], Alheeti et al. have extended their IDS techniques for external communication attacks to also include measurable properties extracted from sensors, such as the magnetometers used by driverless and semi-driverless vehicles. For this, they have used the Integrated Circuit Metrics (ICMetric) technology, which is capable of uniquely identifying a system's behaviour. Specifically, they have added the bias reading of magnetometer sensors to the cyber features used in their previous work, and have applied a simple machine learning approach based on k-nearest neighbours to detect anomalous conditions. For evaluation, they have used measurements from a real sensor system and NS-2 simulation of the rest of their setup. Their results have shown considerable improvement in the detection accuracy when using ICMetric. In [64], they have additionally evaluated the use of gyroscope sensors with similarly positive results.

- S: Autonomous and semi-autonomous VANET
- D/DA: Onboard (local) / Onboard self-detection
- AT/T: Behaviour / (Learning) k-nearest neighbours in [27], and additionally feedforward neural networks in [64]
- AF: (Physical) Sensor bias readings and (Cyber) another 16 features, including payload size, packet ID, source, destination, hop counts, etc.
- AA: Availability
- EA: (Simulation) NS-2 enhanced with bias measurements from real sensors
- TRL: 3

Security research in VANETs is often geared towards MDSs, where it is not necessary that a particular vehicle has been compromised by a cyber attack, but it may also be the driver/operator that selfishly disseminates false information, for example to gain access to a particular lane. For several MDSs, there is no distinction as to the cause of the misbehaviour. Indicatively, Raya et al. [83] have used entropy to represent the anomalous and normal behaviours of nodes, and k-means clustering to identify outliers, which are assumed to be the attackers that should be evicted. Another important assumption for the approach to work is that there is an honest majority. Eviction of a suspected node is based on distance enlargement and deviation from the majority.

- S: Automobile VANETs
- D/DA: Onboard (distributed) / Collaborative

TABLE VI
COMPARATIVE ANALYSIS OF INTRUSION DETECTION SYSTEMS IN VANETS

			Deploym.		Architecture			Type		Features		Techn.		Attacks on			Evaluation			
Ref.	Year	Scope	Onboard	External	Self-detection	Collaborative	Offloaded	Knowledge	Behaviour	Cyber	Physical	Learning	Rule-based	Confid/ity	Integrity	Availability	Analytical	Simulation	Experimental	TRL
[29], [26]	2017	Autonomous VANET	✓	✗	✓	✗	✗	✗	✓	✓	✗	✓	✗	✗	✗	✓	✗	✓	✗	3
[27], [64]	2017	Autonomous VANET	✓	✗	✓	✗	✗	✗	✓	✓	✓	✓	✗	✗	✗	✓	✗	✓	✗	3
[83]	2007	VANET	✓	✗	✓	✗	✗	✗	✓	✓	✓	✓	✗	✗	✗	✓	✗	✓	✗	3
[84]	2010	VANET	✓	✓	✗	✓	✗	✗	✓	✗	✓	✓	✓	✗	✓	✗	✗	✓	✗	3
[85]	2016	VANET	✓	✗	✗	✓	✗	✗	✓	✗	✓	✗	✓	✗	✓	✗	✗	✓	✗	3
[86]	2013	VANET	✓	✗	✗	✓	✗	✗	✓	✓	✗	✗	✓	✗	✗	✓	✗	✓	✗	3
[87]	2014	VANET	✓	✗	✗	✓	✗	✗	✓	✓	✗	✗	✓	✗	✓	✗	✗	✓	✗	3
[23], [24]	2015	VANET	✓	✓	✗	✓	✗	✓	✓	✓	✓	✓	✓	✗	✓	✓	✗	✓	✗	3
[88]	2016	VANET	✓	✗	✗	✓	✗	✓	✓	✓	✗	✗	✓	✗	✗	✓	✗	✓	✗	3
[89], [90]	2014	VANET	✗	✓	✗	✗	✓	✓	✗	✓	✗	✓	✗	✗	✗	✓	✗	✓	✗	3
[91], [92]	2014	VANET	✗	✓	✓	✓	✗	✗	✓	✗	✓	✗	✓	✗	✓	✗	✗	✗	✓	7
[35]	2017	VANET	✓	✗	✗	✓	✗	✗	✓	✗	✓	✓	✓	✗	✓	✓	✗	✓	✗	3
[93]	2012	VANET	✗	✓	✗	✗	✓	✓	✗	✗	✓	✓	✓	✗	✓	✗	✗	✓	✗	3
[94]	2011	VANET	✓	✗	✗	✓	✗	✓	✗	✗	✓	✓	✗	✗	✓	✗	✗	✓	✗	3
[39]	2017	VANET	✓	✗	✗	✓	✗	✗	✓	✗	✓	✓	✗	✗	✓	✗	✗	✓	✗	3
[95]	2016	VANET	✓	✗	✗	✓	✗	✗	✓	✓	✗	✗	✓	✗	✗	✓	✗	✓	✗	3
[96]	2015	VANET	✓	✗	✓	✗	✗	✗	✓	✓	✗	✗	✓	✗	✗	✓	✗	✓	✗	3
[66]	2014	Platoon VANET	✓	✗	✗	✓	✗	✓	✗	✓	✗	✗	✓	✗	✗	✓	✗	✓	✗	3

- AT/T: Behaviour. Rule-based and (Learning) k-means clustering
- AF: (Physical) Coordinates and timestamps
- AA: (Integrity) False information dissemination
- EA: Simulation using NS-2
- TRL: 3

Also, geared towards rogue nodes disseminating false information, is the MDS proposed by Ruj *et al.* [50], all nodes are monitored for their actions after sending out an alert message. Lack of consistency of recent messages and new alerts with regard to the reported and estimated vehicle positions is an indication of misbehaviour, which is penalised with fines by the Certification Authority.

- S: Automobile VANETs
- D/DA: Onboard (local) / Onboard self-detection
- AT/T: Behaviour / Rule-based
- AF: (Physical) node location, event location, and (Cyber) alert type and time
- AA: (Integrity) False information dissemination
- EA: Simulation
- TRL: 3

Kim et al. [84] have combined several of key MDS concepts integrated into a single model, which filters messages based on a confidence level for a received message, as derived by six data sources, both on-board and external. In a linear approach, the confidence level for each message is calculated by each vehicle in isolation, while in a cumulative approach, greater confidence is placed on a message reporting an event that has been previously reported by other vehicles. Simulation involving the injection of false Electronic Emergency Brake Light (EEBL) messages has shown that the cumulative approach can achieve a noticeably higher true positive rate of detection. However, it requires keeping track of all alerts about events sent by others.

- S: Automobile VANETs
- D/DA: Onboard (distributed), external / Collaborative
- AT/T: Behaviour / Rule-based.
- AF: (Cyber) Authentication, other vehicles' messages, reputation, (Physical) location, on-board sensors, RSU-based sensors
- AA: (Integrity) False information dissemination
- EA: Simulation
- TRL: 3

A common weakness of collaborative MDSs is that they require a provision for estimating, storing and sharing securely the levels of trust or reputation for each vehicle. So, a primary motivation for the work of Zaidi et al. [85] was to detect misbehaviour without relying on trust or reputation. For this, they have proposed a detection mechanism running on each vehicle that uses data collected from other nodes in the

vicinity to model the traffic around it. Hypothesis testing is then employed to decide whether the received data is correct and should be accepted or not. The rationale is that if all data shared is correct, then each vehicle's estimate of the traffic flow (vehicles per hour) should not differ considerably from the other vehicles in the vicinity as determined by the fact that they are within communication range. This is expressed as a rule that the flow values need to lie within two standard deviations of the mean. Then, a statistical t-test can be used to detect false values reported by misbehaving vehicles. Evaluation of this technique in the OMNeT++ simulator based on simulations under varying vehicular and network traffic conditions showed that it can achieve high accuracy with relatively low overhead, as long as the ratio of misbehaving vehicles is not extremely high. Indicatively, for a ratio of attackers below 25%, the true positive rate was above 98% and the false positive rate did not exceed 2%.

- S: Automobile VANETs
- D/DA: Onboard (distributed) / Collaborative
- AT/T: Behaviour / Rule-based.
- AF: (Physical) speed, flow, density, location
- AA: (Integrity) False information dissemination
- EA: (Simulation) OMNeT++
- TRL: 3

Daeinabi et al. [86] have focused on detecting attacks that affect packet forwarding in VANETs, including blackhole, duplication of packets and isolation from honest vehicles. The approach is based on the existence of verifier nodes, which are "trustier' vehicles in the vicinity that can detect their neighbours' abnormal behaviour. Every time a vehicle is deemed to be behaving abnormally, its distrust value is decreased, and when this exceeds a threshold, it is reported to the corresponding Certification Authority. Detection of abnormal behaviour is based on monitoring the number of packets received, missed or duplicated by a neighbouring vehicle, as observed by a verifier.

- S: Automobile VANETs
- D/DA: Onboard (distributed) / Collaborative
- AT/T: Behaviour / Rule-based
- AF: (Cyber) Packets received, missed or duplicated
- AA: (Availability) Blackhole, packet duplication, isolation
- EA: Simulation
- TRL: 3

Kumar et al. [87] have developed T-CLAIDS, which is a collaborative IDS using a Learning Automata type of machine learning. Each vehicle is assumed to be equipped with a learning automaton, which is code able to take decisions by learning the optimal action through repeated interaction with the environment [97]. Tuning of the detection is based on Collaborative Trust Index, a parameter computed for each vehicle according to the success or failure of each operation. Any value below a threshold is considered to be an indication of malicious behaviour. The threshold depends on the reliability requirements of the application at hand. This approach has performed well in terms of both detection accuracy and scalability.

- S: Automobile VANETs
- D/DA: Onboard (distributed) / Collaborative
- AT/T: Behaviour / Learning
- AF: (Physical) Density, mobility, direction
- AA: Integrity
- EA: Simulation
- TRL: 3

Sedjelmaci et al. [23], [24] have proposed AECFV, an intrusion detection framework which takes into account node mobility and frequent changes in network topology. At its core, there is a clustering algorithm, where cluster-heads are selected based on each vehicle's trust level and a boundary distance. Trust levels are evaluated based on majority voting and a reputation protocol and are broadcast periodically within the network. Similarly to the same research team's IDS for UAVs [28], it makes use of two detection systems; (a) a local one running at each cluster member and monitoring its neighbouring vehicles and the cluster-head, and (b) a global one running at cluster-head level, evaluating the trustworthiness of its cluster members. Then, a global decision system running at road side unit (RSU) level, computes the level of trust for each vehicle and classifies them based on this. Together, these systems constitute a network IDS as they take a decision based on monitoring of behaviours of nodes within their radio range. The two IDSs use rules and support vector machines to model normal behaviour. The authors have implemented AECFV in the NS-3 network simulator and have reported its performance in terms of accuracy and detection latency to be superior to T-CLAIDS [87] for selective forwarding, blackhole, packet duplication, resource exhaustion, wormhole and Sybil attacks. For a ratio of attackers between 10% and 30%, the false positive rate ranged between 1.5% and 3.5% respectively.

- S: Automobile VANET
- D/DA: Onboard (distributed) and externally at RSUs / Collaborative
- AT/T: Hybrid / Learning and Rule-based
- AF: (Cyber) packet drop ratio, packet sent ratio, message duplication ratio, (Physical) signal strength intensity
- AA: (Integrity) Sybil, packet duplication, wormhole, (Availability) selective forwarding, blackhole, resource exhaustion
- EA: (Simulation) NS-3
- TRL: 3

Kerrache et al. [88] have also developed a trust-based mechanism that is geared specifically towards denial of service attacks by preventing the forwarding of malicious data and by rapidly revoking nodes deemed to be dishonest. For this, they have proposed adding an "opinion" field in the packet header, which corresponds to a message's trustworthiness as evaluated by its last forwarder. The detection is based on both a knowledge-based element looking for signatures of known attacks, and a behaviour-based element with a predefined threshold for the maximum number of messages considered normal for a specific type of traffic and conditions. If a node receives more packets from a particular neighbour than normal, then the latter's honesty score is decreased. There is also a

similar approach for the overall quality of packets received from a particular neighbour. The two parameters are periodically combined into a single weight used to decide whether a denial of service attack exists in the network. Evaluations run on the NS-2 simulator have shown this approach to outperform [87] and [24] in the presence of high ratios of dishonest nodes, with a true positive rate of 88% for a dishonest vehicle ratio of 50%.

- S: Automobile VANETs
- D/DA: Onboard (distributed) / Collaborative
- AT/T: (Knowledge) and (Behaviour) / Rule-based.
- AF: (Cyber) Counters for neighbours' messages sent
- AA: (Availability) Denial of service
- EA: (Simulation) NS-2
- TRL: 3

Verma et al. [89] have proposed detecting denial of service attacks by checking whether the number of TCP SYN packets that have not been acknowledged within a certain amount of time exceeds a predefined threshold, as well as the IP addresses of the sources in [90]. To record the behaviour of each message and achieve very low rate of false positives, they have based their classification on a Bloom filter, which can check very rapidly whether a message belongs to a set or not.Deployment is assumed to be on an edge router, such as on a RSU. The training of the system is carried out on a lightweight neural network with back-propagation. NS2 simulation results yielded a false positive rate of between 4% and 25% as the ratio of malicious vehicles increased from 5% to 30% respectively.

- S: Automobile VANETs
- D/DA: External / Offloaded
- AT/T: Knowledge / Learning.
- AF: (Cyber) Rate of unacknowledged SYN packets, IPs
- AA: (Availability) Denial of service
- EA: (Simulation) NS-2
- TRL: 3

Bissmeyer et al. [91], [92] have produced a MDS, which combines local short-term detection of misbehaviour of neighbouring nodes with centralised long-term identification. Their detection uses plausibility checks in individual modules, which are integrated in a fusion phase, and employs Kalman filters, Particle filters, and a method that detects overlaps of vehicles appearing to virtually drive through each other. Significantly, while the vast majority of IDSs and MDSs for VANETs have been evaluated only in simulation, the authors have instead opted to evaluate this work in field experiments with real vehicles in a dedicated test area which allowed low-speed and high-speed tests without endangering public road traffic. For their evaluation, they have also developed VANET-specific malware, which when deployed on an attack vehicle, it analyses the network and automatically selects a victim vehicle, in front of which it creates a fake ("ghost") vehicle. After some lead time, the attack vehicle broadcasts spoofed EEBL notification messages on behalf of the ghost vehicle, so as to display a false driver warning on the victim vehicle. Experiments were run over 15 weeks with approximately 17 billion messages checked, in a rare example of published IDS research for vehicles progressing to high TRL.

- S: Automobile VANETs
- D/DA: Onboard (distributed) / Self-detection, collaborative
- AT/T: Behaviour / Rule-based
- AF: (Physical) Location
- AA: (Integrity) False information dissemination
- EA: (Experimental) Three test vehicles equipped with VANET communication
- TRL: 7

Recent work by Subba et al. [35] has combined several promising ideas for VANET IDSs into a single multi-layered framework, which they have shown to be effective against a variety of different attacks. In all cases, detection is based on comparison of audit features against thresholds. These include packet delivery rates (PDR) and Received Signal Strength Information (RSSI) for selective forwarding (grayhole) and blackhole attacks; duplicate packet rate and packet forwarding rate for denial of service; RSSI and PDR for wormhole attack; and the z-score of RSSI for Sybil attack. Evaluation based on NS-3 simulation has shown that this framework can achieve greater accuracy and lower overhead in terms of IDS-specific network traffic generated than [86], [24] and [87]. The reduction of IDS traffic overhead is the result of adopting a game theoretic approach in modeling the interaction between the IDS and the malicious vehicle as a two-player non-cooperative game and using the Nash Equilibrium to inform the choice of monitoring strategy.

- S: Automobile VANETs
- D/DA: Onboard (distributed) / Collaborative
- AT/T: Behaviour / Rule-based
- AF: (Cyber) Packet delivery/forwarding rates, duplicate packet rate, (Physical) RSSI
- AA: (Integrity) Wormhole, Sybil, (Availability) Gray/blackhole, Denial of service
- EA: (Simulation) NS-3
- TRL: 3

The majority of IDSs designed specifically for detecting Sybil attacks in VANETs look for similarities in motion trajectories, as inferred from messages shared and timestamps, so as to detect "Sybil" communities. An example of such work is Footprint, proposed by Chang et al. [93], which uses a network of trusted RSUs to track a vehicle's trajectory through active demand of an authorised message from the RSU as proof of its appearance time. Footprint has placed particular emphasis on preserving the vehicles' location privacy by comparing their trajectories anonymously. In terms of accuracy, using the best check window size and the best trajectory length limit, this approach could achieve minimum false positive rate of 3% and minimum false negative rate of 1%.

- S: Automobile VANETs
- D/DA: External (at RSUs) / Offloaded
- AT/T: Knowledge / Rule-based
- AF: (Cyber) Authorisation timestamps, (Physical) locations
- AA: (Integrity) Sybil

- EA: Simulation
- TRL: 3

The IDS for Sybil attacks proposed by Grover et al. in [94] does not need an established infrastructure of RSUs. Instead, its aim is to identify vehicles with fake identifies by looking for similarities in neighbourhood information of nodes at incremental interval of time. The assumption is that a node that is simultaneously observed by neighbouring nodes for significant duration of time is very likely to be a Sybil node. Through simulation with traffic traces based on Swiss road maps, the authors have produced a methodology for choosing a time threshold for varying number of attackers and number of Sybil identities per attacker. The lower the threshold, the higher the true positive rate, but also the more common the false positives, especially in the case of high normal vehicle density. Indicatively, the false positive rate for a 15% ratio of Sybil attackers and 500 m transmission range was 3%.

- S: Automobile VANETs
- D/DA: Onboard (distributed) / Collaborative
- AT/T: Knowledge / Rule-based
- AF: (Cyber) Neighbourhood information
- AA: (Integrity) Sybil
- EA: Simulation
- TRL: 3

The authors of [39] have focused on cooperative vehicular positioning networks and potential malicious attacks targeting these networks. Examples of such attacks may be incorrect time-tags that make ranging with neighbouring vehicles erroneous, or distorted location coordinates and distorted reliability messages sent to neighbouring vehicles. In the proposed detection technique, every vehicle evaluates the reliability of its neighbours in a weighted manner and based on the Mahalanobis distance [98] of the difference observation between the measured range and the Euclidian distance, using the vehiclesâĂŹ shared location coordinates. It then distributes these reliability estimates within the network. The aim is to distinguish inaccuracies caused by non-line-of-sight delays and genuine attacks. The authors have used Monte Carlo simulations with vehicle-to-vehicle radio parameters that were determined empirically to demonstrate that the proposed approach can protect cooperative positioning against malicious manipulation.

- S: Automobile VANETs
- D/DA: Onboard (distributed) / Collaborative
- AT/T: Behaviour / Learning
- AF: (Physical) measured LOS range and Euclidean distance
- AA: Location spoofing, ranging manipulation, message distortion
- Attacks targeting: Integrity
- EA: Simulation
- TRL: 3

Bouali et al. [95] have taken a very different approach to detection, which they have called proactive detection, by extending its scope to include prediction of a vehicle's potential misbehaviour and classification into white, gray and black. As area of application, they have used the threat of authenticated vehicles launching denial of service in the network of a cooperative intelligent transportation system. Their technique uses a clustered architecture, where each clusterhead is responsible for the classification of vehicles that are members of its cluster. The classification is effectively a trust scoring mechanism based on past experience with each member and recommendations from specific other vehicles, which are trusted to act as monitoring agents. From an algorithmic perspective, it uses basic Kalman filtering to integrate the different data sources. To predict misbehaviour in the form of denial of service, the primary audit feature is the packet delivery ratio for each vehicle, as monitored by its neighbours. Evaluation using NS-3 in a Manhattan grid has shown that the inclusion of prediction can reduce the impact of the attack by evicting some of the malicious vehicles before they take part , with the true positive rate ranging from 79% for 400 vehicles to 97% for 100 vehicles.

- S: Automobile VANETs
- D/DA: Onboard (distributed) / Collaborative
- AT/T: Behaviour / Rule-based
- AF: (Cyber) Packet delivery ratio
- AA: (Availability) Denial of service
- EA: (Simulation) NS-3
- TRL: 3

VANET availability can also be affected by a wireless communication jamming attack. There, the challenge is to determine whether a disruption is the result of a genuine attack or a poor radio link. Mokdad et al. [96] have proposed differentiating between the two by monitoring the ratio of packets received that pass the Cyclic Redundancy Check. The jamming attack is modeled based on Markov Chains and the aim of the authors' analysis is to set the appropriate packet delivery ratio threshold for considering a detected disruption as a jamming attack.

- S: Automobile VANETs
- D/DA: Onboard (local) / Self-detection
- AT/T: Behaviour / Rule-based
- AF: (Cyber) Packet delivery ratio
- AA: (Availability) Jamming
- EA: Simulation
- TRL: 3

In the special case of a platoon where only the leading vehicle has a driver, the safe cooperation between the vehicles depends on periodic broadcast messages containing vehicle positions and velocities. In this context, a jamming attack can have severe consequences by delaying transmission of these messages beyond the delay requirements of automotive control systems. This is particularly the case for platoons because of both their highly autonomous nature and the fact that a jammer placed on one of the vehicles can always be within range of the whole platoon. Lyamin et al. [66] have proposed mounting a sniffer device on the leading vehicle listening to the channel and recording the identifiers of the vehicles for which these messages have been successfully received. In addition, the platoon's vehicles are divided into groups in such a way that messages from different groups never collide with each other. Then, if there is at least one group where exactly one such

message has not been received, this is assumed to be due to a jamming attack. Otherwise, at least two nodes should have been involved in a legitimate collision within the same group.

- S: Automobile VANETs
- D/DA: Onboard (local, on leading vehicle) / Collaborative
- AT/T: Knowledge / Rule-based
- AF: (Cyber) Messages lost
- AA: (Availability) Jamming
- EA: Simulation
- TRL: 3

C. Watercraft

The cyber security of the maritime sector has only recently started featuring in governmental and research agendas. The sector is still largely at the point of trying to identify the realistic cyber threat scenarios [99], unique challenges [100] and generalist defences that are applicable. Ntouskas and Gritzalis have found that the main vulnerabilities in the maritime sector are lack of security awareness programs, weak protection of physical access to ship information systems, lack of backup systems and existence of several assets that can be "single points of failure" [101].

Civil maritime transportation depends heavily on the reliability of GPS signals. Having demonstrated the vulnerability of a real surface vessel to GPS spoofing using the White Rose of Drachs yacht as their testbed, Bhatti and Humphreys [3] have proposed a GPS spoofing detection framework that is appropriate for integrating in the Electronic Chart Display and Information System (ECDIS) software available to ships. The framework uses a maritime dynamics model to take into account wind, ocean currents and other environmental disturbances.

- S: Semi-autonomous ship (equipped with course autopilot) using GPS
- D/DA: Onboard (local) / Onboard self-detection
- AT/T: Behaviour / Learning
- AF: (Physical) ECDIS sensor measurements
- AA: (Integrity) GPS spoofing
- EA: Experimental (for the demonstration of the attack) and Monte-Carlo simulations (for the evaluation of the detection)
- TRL: 3

Automatic Identification System (AIS) is a system that supplements marine radar for the purpose of vessel tracking and collision avoidance. It broadcasts signals containing the ship's unique identification, position, course, and speed, which are received by other ships and land based systems to help them track it on their display systems. However, these signals are unauthenticated and unencrypted, which makes them easy to spoof [59], as a result of a cyber security breach or on purpose by the crew (e.g., a fishing boat entering a no-fishing zone). Katsilieris et al. [102] have formulated detecting AIS spoofing as a hypothesis testing problem, using as input any data coming from available radars, as well as information from the tracking system. For detection based on a single AIS report, they have used a Clairvoyant likelihood ratio test for one radar and K radars, and extended as a generalised likelihood ratio test to account for the unknown spoofing distance. This was then further extended to account for sequences of AIS reports. However, the approach developed is not naturally designed for ships changing directions or speeds. Evaluation in the form of Receiver Operating Characteristics (ROC) curves was based on varying the spoofing distance, the number of radars and the sample data available. The data used were from real AIS reports for the normal behaviour, with synthesised reports for the attack behaviour. Indicatively, the true positive rate for spoofing distance of 200 m exceeded 80%.

- S: Ship using AIS
- D/DA: External / Unknown
- AT/T: Behaviour / Learning
- AF: (Physical) Locations as reported by radar
- AA: (Integrity) AIS spoofing
- EA: Simulation
- TRL: 3

Iphar et al. [60], [61] have proposed a methodology for detecting AIS spoofing, which takes into account positions, trajectories and monitoring the link between conceptual specification of a situation and its implementation, as well as comparison of AIS data with historical and predicted data. The detection is based on whether each single data field value is consistent with the possible field values provided by the technical specifications, and whether there is any discrepancy between the fields, between messages of the same type, as well as between the fields values of different kinds of messages. Initial results have shown that it was able to detect the simple case of a vessel changing its identity during travel, but evaluation is still at an early stage.

- S: Ship using AIS
- D/DA: External / Unknown
- AT/T: Behaviour / Rule-based
- AF: Cyber and physical, as included in AIS signal content
- AA: (Integrity) AIS spoofing
- EA: Simulation of AIS spoofing and offline detection
- TRL: 2

Mazzarella et al. [103] have additionally tackled the problem of AIS unavailability where AIS is jammed or maliciously turned on or off. Their approach aims to detect when a reduction in received AIS reports is natural or an indication of malicious manipulation. For this, it monitors the RSSI available at the AIS base stations, taking into account the natural electromagnetic propagation phenomena involved in AIS transponders to base station communication, in the form of path loss propagation (due to curvature of the Earth, multipath effects and ducting from the varying refractive index of the air), as well as the behaviour of each vessel and of the base station. The detection decision depends on whether deviation from a single vessel's normality model and a base station's normality model (based on One-Class Support Vector Machines and geospatial distribution of historical RSSI data) corresponds to a level of risk above a set threshold. In their evaluation, they have used data collected from several base stations, which indicated comparable RSSI dynamics between them, albeit with some degree of asymmetry. The

TABLE VII
COMPARATIVE ANALYSIS OF INTRUSION DETECTION SYSTEMS FOR WATERCRAFT

			Deploym.		Architecture			Type		Features		Techn.		Attacks on			Evaluation			
Ref.	Year	Scope	Onboard	External	Self-detection	Collaborative	Offloaded	Knowledge	Behaviour	Cyber	Physical	Learning	Rule-based	Confid/lity	Integrity	Availability	Analytical	Simulation	Experimental	TRL
[3]	2014	Ship using GPS	✓	✗	✓	✗	✗	✗	✓	✗	✓	✓	✗	✗	✓	✗	✗	✓	✗	3
[102]	2017	Ship using AIS	✗	✓	?	?	?	✗	✓	✗	✓	✓	✗	✗	✓	✗	✗	✓	✗	3
[60], [61]	2017	Ship using AIS	✗	✓	?	?	?	✗	✓	✓	✓	✗	✓	✗	✓	✗	✗	✓	✗	2
[103]	2017	Ship using AIS	✗	✓	?	?	?	✗	✓	✗	✓	✓	✗	✗	✗	✓	✗	✓	✗	3

one-class support vector machine algorithm was shown to achieve particularly high detection accuracy.

- S: Ship using AIS
- D/DA: External / Unknown
- AT/T: Behaviour / Learning
- AF: (Physical) RSSI
- AA: (Availability) AIS on/off manipulation
- EA: Simulation
- TRL: 3

VI. LESSONS LEARNED AND OPEN ISSUES

Research for vehicular IDSs is centered on a rather narrow range of applications, primarily geared towards the in-vehicle network of automobiles and VANETs. This is not surprising. The body of work on IDSs for aircraft, watercraft and robotic land vehicles is growing steadily, but it is the security of automobiles that has already captured the interest of the general public. At the same time, cyber security in VANETs is a key requirement for their successful adoption. It is also not surprising that researchers prefer collaborative detection for the already naturally collaborative VANETs or that in-vehicle automobile IDSs are designed for self-detection, as any other design architecture might introduce additional security requirements and detection latencies. What is perhaps surprising is that despite the cyber-physical nature of most attacks affecting vehicles, the percentage of IDSs using both cyber and physical audit features is a remarkably low 26% (only 17 out of 66). Below, we detail what we consider four key lessons learned and corresponding areas where further research can be highly beneficial.

A. Shortage of research testbeds

The majority of proposed work published in the literature is of relatively low maturity, usually at TRL 3. This is primarily because the evaluation for most research is limited to simulation and often in generalist network simulation packages, such as NS-2 or NS-3. For more of the proposed IDS solutions to reach higher TRL, towards 6 ("technology demonstrated in relevant environment") or 7 ("system prototype demonstration in operational environment"), there is a need for testbeds to become available to more research teams. If purchasing a commercially available vehicle is impractical (e.g., due to costs) or unsuitable (e.g., because it is not possible to meaningfully modify its software or components), the only other option is to build a testbed. However, most researchers specialising in IDS design come from computer networks, computer security and mathematical/algorithmic backgrounds, while developing a vehicle testbed requires electronics, embedded system and mechanical knowledge that can vary significantly in different domains and types of vehicles. This mismatch in skills can be a barrier for research teams. Out of the IDSs surveyed here, only 11 were based on actual testbeds. These include NASA's DragonEye UAV [53], the White Rose of Drachs yacht [3], a remote-controlled [30] and a semi-autonomous robotic vehicle [15] at the University of Greenwich, a platoon of small robotic vehicles at Daegu Gyeongbuk Institute of Science and Technology [46], a 2013 Honda Accord and a 2015 Chevrolet Trax at the University of Michigan [49], a production automobile at Daimler AG [36], three production automobiles (from Toyota, Honda and Chevrolet) at the Univerity of Maryland [48], a production automobile at Korea University [37], and three test automobiles at TU Darmstadt [92]. Beyond acquiring a vehicle or group of vehicles for research, there is the even greater challenge of gaining access to appropriate space for experiments, especially if these involve automobiles at speed, UAVs in flight or watercraft in proximity to land or other vessels. Here, what would help is high-fidelity simulation software to minimise this barrier to entry for researchers, with some attempts already made in [81], as well as industrial collaboration with vehicle manufacturers for real-world prototype implementations, and vehicle testing facilities for collection not only of real normal behaviour data but also of real attack behaviour data.

B. Strengthening knowledge-based approaches with a vehicular IDS hub

The preference shown by researchers for behaviour-based over knowledge-based approaches is natural. Even though they tend to exhibit high false positive rates, they are more practical than knowledge-based approaches due to the absence of a large enough dictionary of attack signatures. This problem can potentially be addressed by extending the offloaded detection

scenario into a vehicular IDS hub scenario, whereby the centralised reasoning benefits from data collected from other vehicles too (Figure 6). A supervised machine learning based detection system running onboard a vehicle may have never been trained on a specific attack and may miss it when the vehicle encounters it. This will not be the case if detection is offloaded to a centralised IDS hub, which has seen this attack previously on another vehicle of the same type, and as a result has access to a more complete or more up to date dictionary of relevant known threats. The IDS hub approach has not been implemented in the context of vehicles yet, but similar concepts have been adopted by collaborative threat intelligence platforms, such as Hewlett Packard's Threat Central [104], and have been briefly discussed in [77].

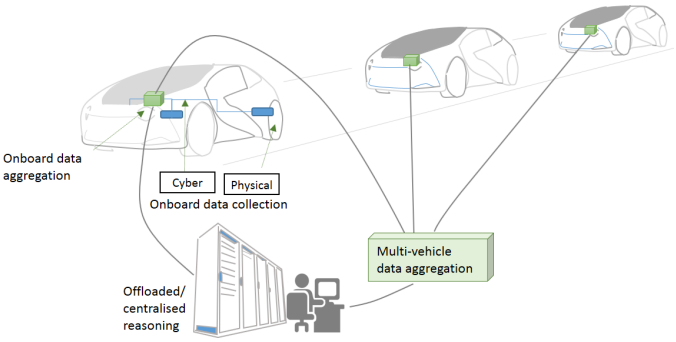

Fig. 6. Conceptual architecture of a vehicular IDS hub

C. Addressing confidentiality

The vast majority of IDSs for vehicles have largely omitted confidentiality-related security threats. There are two reasons for this. The first is that cyber threats against vehicles are still in their infancy and naturally the more important ones in the short-term are the ones affecting the integrity or availability of a vehicle's subsystems and hence its safety. However, it is a confidentiality breach that allows reverse-engineering a protocol or understanding what subsystem affects what when it is disabled. Also, the more connected a vehicle is, the greater the threat to the passengers' or drivers' privacy, from their contact address book to their presence in the vehicle, their location and even their biometrics [105]. The second reason is that current IDSs depend heavily on looking for physical manifestations of a security breach, e.g., in a UAV's deviation from the specified flight plan or a robotic vehicle's excessive energy consumption. However, confidentiality breaches do not have physical manifestations, and as such cannot be detected with approaches that rely heavily on physical behaviour monitoring. For example, out of the eight IDSs proposed for UAVs, only one has been evaluated on a data exfiltration attack [19], even though most of the first real-world attacks against UAVs have targeted specifically the confidentiality of their video streaming [106]. In this direction, we anticipate that IDS techniques will need to emphasise less on physical features and waiting for an attack's physical manifestation, and more on looking for (cyber) traces of the earlier stages of the attack (e.g., of probing or of attempting to install malware).

D. Integrating IDSs for vehicular networks with IDSs for single vehicles

The vast majority of IDSs (and MDSs) for vehicular networks, such as VANETs, focus exclusively on the availability and integrity of data shared on the network, with particular emphasis on variations of false data injection. However, there are several onboard IDSs designed for detecting a much larger variety of attacks on individual vehicles. So, it would seem natural to integrate them in VANET IDSs. For example, if a VANET IDS has access to the onboard CAN bus IDS detection results collected from individual vehicles, then it might be able to determine when anomalous behaviour is the result of a cyber security breach or of a driver purposefully sharing false information, or modify the trust estimate of a vehicle before its neighbours or the VANET is affected. Tables V and VI should be a good place to start for researchers interested in exploring such integration.

VII. Conclusion

As is commonly the case for any relatively young research area, the landscape of IDSs for vehicles is fragmented into isolated families of research ideas employed on a single type of vehicle, and usually evaluated on generalist network simulators. By proposing a single IDS taxonomy for all types of vehicles and identifying areas of future research, we have aimed to help researchers from a diverse range of backgrounds identify where they can contribute in the overall architecture of a vehicle's IDS, adopt ideas tried previously on different types of vehicles, as well as extend existing solutions with both cyber and physical audit features, more diverse design architectures, and evaluation in more realistic conditions and against a greater range of realistic attacks.

Acknowledgment

This research has been funded and supported by the Defence Science and Technology Laboratory as part of DSTL-1000086863 project "Safeguarding military autonomous vehicles against cyber attacks", as well as by the European Commision as part of the Horizon 2020 project ROBORDER under grant agreement No. 740593.

References

[1] G. Loukas, *Cyber-Physical Attacks: A Growing Invisible Threat*. Butterworth-Heinemann (Elsevier), 2015.
[2] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, T. Kohno *et al.*, "Comprehensive experimental analyses of automotive attack surfaces." in *USENIX Security Symposium*. San Francisco, 2011.
[3] J. Bhatti and T. E. Humphreys, "Covert control of surface vessels via counterfeit civil gps signals," *University of Texas*, 2014.
[4] J. Petit, B. Stottelaar, M. Feiri, and F. Kargl, "Remote attacks on automated vehicles sensors: Experiments on camera and lidar," *Black Hat Europe*, vol. 11, p. 2015, 2015.
[5] C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel, and M. Rajarajan, "A survey of intrusion detection techniques in cloud," *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 42–57, 2013.
[6] R. Mitchell and R. Chen, "A survey of intrusion detection in wireless network applications," *Computer Communications*, vol. 42, pp. 1–23, 2014.
[7] I. Butun, S. D. Morgera, and R. Sankar, "A survey of intrusion detection systems in wireless sensor networks," *IEEE communications surveys & tutorials*, vol. 16, no. 1, pp. 266–282, 2014.

[8] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2016.

[9] D. Kwon, H. Kim, J. Kim, S. C. Suh, I. Kim, and K. J. Kim, "A survey of deep learning-based network anomaly detection," *Cluster Computing*, pp. 1–13, 2017.

[10] R. Mitchell and I.-R. Chen, "A survey of intrusion detection techniques for cyber-physical systems," *ACM Computing Surveys (CSUR)*, vol. 46, no. 4, p. 55, 2014.

[11] F. Sakiz and S. Sen, "A survey of attacks and detection mechanisms on intelligent transportation systems: VANETs and IoV," *Ad Hoc Networks*, vol. 61, pp. 33–50, 2017.

[12] V. L. Thing and J. Wu, "Autonomous vehicle security: A taxonomy of attacks and defences," in *2016 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. IEEE, 2016, pp. 164–170.

[13] G. Loukas, Y. Yoon, G. Sakellari, T. Vuong, and R. Heartfield, "Computation offloading of a vehicleâĂŹs continuous intrusion detection workload for energy efficiency and performance," *Simulation Modelling Practice and Theory*, vol. 73, pp. 83–94, 2017.

[14] J. C. Mankins, "Technology readiness levels," *White Paper, April*, vol. 6, 1995.

[15] A. Bezemskij, G. Loukas, D. Gan, and R. Anthony, "Detecting cyber-physical threats in an autonomous robotic vehicle using bayesian networks," in *2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. IEEE, 2017.

[16] A. Broggi, P. Cerri, M. Felisa, M. C. Laghi, L. Mazzei, and P. P. Porta, "The vislab intercontinental autonomous challenge: an extensive test for a platoon of intelligent vehicles," *International Journal of Vehicle Autonomous Systems*, vol. 10, no. 3, pp. 147–164, 2012.

[17] D. Wu, D. I. Arkhipov, M. Kim, C. L. Talcott, A. C. Regan, J. A. McCann, and N. Venkatasubramanian, "Addsen: adaptive data processing and dissemination for drone swarms in urban sensing," *IEEE transactions on computers*, vol. 66, no. 2, pp. 183–198, 2017.

[18] S. Martini, D. Di Baccio, F. Romero, A. JimÃl'nez, L. Pallottino, G. Dini, and A. Ollero, "Distributed motion misbehavior detection in teams of heterogeneous aerial robots," *Robotics and Autonomous Systems*, vol. 74, pp. 30–39, 2015.

[19] R. Mitchell and I. Chen, "Adaptive intrusion detection of malicious unmanned air vehicles using behavior rule specifications," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 44, no. 5, pp. 593–604, 2014.

[20] J. Liu, S. Zhang, W. Sun, and Y. Shi, "In-vehicle network attacks and countermeasures: Challenges and future directions," *IEEE Network*, vol. 31, no. 5, pp. 50–58, 2017.

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

[22] G. De La Torre, P. Rad, and K.-K. R. Choo, "Driverless vehicle security: Challenges and future research opportunities," *Future Generation Computer Systems*, 2018.

[23] H. Sedjelmaci and S. M. Senouci, "A new intrusion detection framework for vehicular networks," in *2014 IEEE International Conference on Communications (ICC)*. IEEE, 2014, pp. 538–543.

[24] ——, "An accurate and efficient collaborative intrusion detection framework to secure vehicular networks," *Computers & Electrical Engineering*, vol. 43, pp. 33–47, 2015.

[25] H. Sedjelmaci, S. M. Senouci, and N. Ansari, "Intrusion detection and ejection framework against lethal attacks in uav-aided networks: A bayesian game-theoretic methodology," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 5, pp. 1143–1153, 2017.

[26] K. M. A. Alheeti, A. Gruebler, and K. McDonald-Maier, "Using discriminant analysis to detect intrusions in external communication of self-driving vehicles," *Digital Communications and Networks*, 2017.

[27] K. M. A. Alheeti and K. McDonald-Maier, "An intelligent intrusion detection scheme for self-driving vehicles based on magnetometer sensors," in *Students on Applied Engineering (ICSAE), International Conference for*. IEEE, 2016, pp. 75–78.

[28] H. Sedjelmaci, S. M. Senouci, and N. Ansari, "A hierarchical detection and response system to enhance security against lethal cyber-attacks in uav networks," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2017.

[29] K. M. A. Alheeti, A. Gruebler, and K. McDonald-Maier, "Intelligent intrusion detection of grey hole and rushing attacks in self-driving vehicular networks," *Computers*, vol. 5, no. 3, p. 16, 2016.

[30] T. Vuong, G. Loukas, and D. Gan, "Performance evaluation of cyber-physical intrusion detection on a robotic vehicle," in *Proceedings of 13th International Conference on Pervasive Intelligence and Computing (PICOM)*. IEEE, 2015.

[31] T. Vuong, A. Filippoupolitis, G. Loukas, and D. Gan, "Physical indicators of cyber attacks against a rescue robot," in *IEEE International Conference on Pervasive Computing and Communications*. IEEE, 2014, pp. 338–343.

[32] T. Vuong, G. Loukas, D. Gan, and A. Bezemskij, "Decision tree-based detection of denial of service and command injection attacks on robotic vehicles," in *Proceedings of 7th International Workshop on Information Forensics and Security (WIFS)*. IEEE, 2015.

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

[34] K. M. A. Alheeti and K. McDonald-Maier, "Hybrid intrusion detection in connected self-driving vehicles," in *22nd International Conference on Automation and Computing (ICAC)*. IEEE, 2016, pp. 456–461.

[35] B. Subba, S. Biswas, and S. Karmakar, "A game theory based multi layered intrusion detection framework for VANET," *Future Generation Computer Systems*, 2017.

[36] M. Muter and N. Asaj, "Entropy-based anomaly detection for in-vehicle networks," in *Intelligent Vehicles Symposium*. IEEE, 2011, pp. 1110–1115.

[37] H. Song, H. Kim, and H. Kim, "Intrusion detection system based on the analysis of time intervals of can messages for in-vehicle network," in *International Conference on Information Networking (ICOIN)*. IEEE, 2016, pp. 63–68.

[38] K.-T. Cho and K. G. Shin, "Error handling of in-vehicle networks makes them vulnerable," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2016, pp. 1044–1055.

[39] S.-H. Kong and S.-Y. Jun, "Cooperative positioning technique with decentralized malicious vehicle detection," *IEEE Transactions on Intelligent Transportation Systems*, 2017.

[40] M.-C. Chuang and J.-F. Lee, "Team: Trust-extended authentication mechanism for vehicular ad hoc networks," *IEEE systems journal*, vol. 8, no. 3, pp. 749–758, 2014.

[41] A. Boudguiga, W. Klaudel, A. Boulanger, and P. Chiron, "A simple intrusion detection method for controller area network," in *2016 IEEE International Conference on Communications (ICC)*. IEEE, 2016, pp. 1–7.

[42] U. Larson, D. Nilsson, and E. Jonsson, "An approach to specification-based attack detection for in-vehicle networks," in *Intelligent Vehicles Symposium*. IEEE, 2008, pp. 220–225.

[43] M. Marchetti and D. Stabili, "Anomaly detection of can bus messages through analysis of id sequences," in *Intelligent Vehicles Symposium (IV), 2017 IEEE*. IEEE, 2017, pp. 1577–1583.

[44] M. R. Ansari, W. T. Miller, C. She, and Q. Yu, "A low-cost masquerade and replay attack detection method for can in automobiles," in *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2017, pp. 1–4.

[45] K.-T. Cho and K. G. Shin, "Fingerprinting electronic control units for vehicle intrusion detection." in *USENIX Security Symposium*, 2016, pp. 911–927.

[46] C. Gwak, M. Jo, S. Kwon, H. Park, and S. Son, "Anomaly detection based on recursive least-square filter for robust intelligent transportation systems," in *Proceedings of the 2015 Korea Institute of Communication Sciences Summer Conferences*. KICS, 2015, pp. 438–440.

[47] F. Martinelli, F. Mercaldo, V. Nardone, and A. Santone, "Car hacking identification through fuzzy logic algorithms," in *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE, 2017, pp. 1–7.

[48] S. N. Narayanan, S. Mittal, and A. Joshi, "OBD SecureAlert: An anomaly detection system for vehicles," in *IEEE Workshop on Smart Service Systems (SmartSys 2016)*. IEEE, 2016.

[49] K.-T. Cho and K. G. Shin, "Viden: Attacker identification on in-vehicle networks," in *24th ACM Conference on Computer and Communications Security (CCSâĂŹ17)*. ACM, 2016, pp. 164–170.

[50] S. Ruj, M. A. Cavenaghi, Z. Huang, A. Nayak, and I. Stojmenovic, "On data-centric misbehavior detection in VANETs," in *IEEE Vehicular technology conference (VTC Fall)*. IEEE, 2011, pp. 1–5.

[51] Z. Birnbaum, A. Dolgikh, V. Skormin, E. O'Brien, and D. Muller, "Unmanned aerial vehicle security using recursive parameter estimation," in *International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2014, pp. 692–702.

[52] R. Mitchell and I. Chen, "Specification based intrusion detection for unmanned aircraft systems," in *Proceedings of the first ACM MobiHoc workshop on Airborne Networks and Communications*. ACM, 2012, pp. 31–36.

[53] J. Schumann, P. Moosbrugger, and K. Rozier, "R2u2: Monitoring and diagnosis of security threats for unmanned aerial systems," in *Proceedings of 15th International Conference on Runtime Verification*. Springer, 2015.

[54] Z. Birnbaum, A. Dolgikh, V. Skormin, E. O'Brien, D. Muller, and C. Stracquodaine, "Unmanned aerial vehicle security using behavioral profiling," in *Unmanned Aircraft Systems (ICUAS), 2015 International Conference on*. IEEE, 2015, pp. 1310–1319.

[55] A. Bezemskij, G. Loukas, R. J. Anthony, and D. Gan, "Behaviour-based anomaly detection of cyber-physical attacks on a robotic vehicle," in *International Conference on Ubiquitous Computing and Communications and 2016 International Symposium on Cyberspace and Security (IUCC-CSS)*. IEEE, 2016, pp. 61–68.

[56] G. Clark, M. Doran, and W. Glisson, "A malicious attack on the machine learning policy of a robotic system," in *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)*. IEEE, 2018, pp. 516–521.

[57] A. Lauf, R. Peters, and W. Robinson, "A distributed intrusion detection system for resource-constrained devices in ad-hoc networks," *Ad Hoc Networks*, vol. 8, no. 3, pp. 253–266, 2010.

[58] M. Strohmeier, V. Lenders, and I. Martinovic, "Intrusion detection for airborne communication using phy-layer information," in *12th Conference on Detection of Intrusions and Malware & Vulnerability Assessment (DIMVA)*. Springer, 2015.

[59] M. Balduzzi, A. Pasta, and K. Wilhoit, "A security evaluation of ais automated identification system," in *Proceedings of the 30th annual computer security applications conference*. ACM, 2014, pp. 436–445.

[60] C. Iphar, A. Napoli, and C. Ray, "On the interest of data mining for an integrity assessment of ais messages," in *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*. IEEE, 2016, pp. 368–373.

[61] ——, "Integrity assessment of a worldwide maritime tracking system for a geospatial analysis at sea," in *20th AGILE International Conference on Geographic Information Science (AGILE 2017)*, 2017, pp. 4–pages.

[62] D. McCallie, J. Butts, and R. Mills, "Security analysis of the ads-b implementation in the next generation air transportation system," *International Journal of Critical Infrastructure Protection*, vol. 4, no. 2, pp. 78–87, 2011.

[63] D. Muniraj and M. Farhood, "A framework for detection of sensor attacks on small unmanned aircraft systems," in *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2017, pp. 1189–1198.

[64] K. M. A. Alheeti, R. Al-Zaidi, J. Woods, and K. McDonald-Maier, "An intrusion detection scheme for driverless vehicles based gyroscope sensor profiling," in *2017 IEEE International Conference on Consumer Electronics (ICCE)*. IEEE, 2017, pp. 448–449.

[65] K. M. A. Alheeti, M. S. Al-Ani, and K. McDonald-Maier, "A hierarchical detection method in external communication for self-driving vehicles based on tdma," *PloS one*, vol. 13, no. 1, p. e0188760, 2018.

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

[67] G. Loukas, T. Vuong, R. Heartfield, G. Sakellari, Y. Yoon, and D. Gan, "Cloud-based cyber-physical intrusion detection for vehicles using deep learning," *IEEE Access*, 2018.

[68] A. Bezemskij, R. J. Anthony, G. Loukas, and D. Gan, "Threat evaluation based on automatic sensor signal characterisation and anomaly detection," in *The Twelfth International Conference on Autonomic and Autonomous Systems (ICAS 2016)*. IARIA, 2016.

[69] F. Bergamaschi, D. Conway-Jones, and N. Peach, "Generic vehicle architecture for the integration and sharing of in-vehicle and extra-vehicle sensors," in *Proc. of SPIE Vol*, vol. 7694, 2010, pp. 76 940B–1.

[70] O. Henniger, A. Ruddle, H. SeudiÃl', B. Weyl, M. Wolf, and T. Wollinger, "Securing vehicular on-board IT systems: The EVITA project," in *VDI/VW Automotive Security Conference*, 2009.

[71] T. Hoppe, S. Kiltz, and J. Dittmann, "Security threats to automotive can networks - practical examples and selected short-term countermeasures," in *SAFECOMP*, 2008.

[72] ——, "Applying intrusion detection to automotive IT - early insights and remaining challenges," *Journal of Information Assurance and Security*, vol. 4, no. 6, pp. 226–235, 2009.

[73] C. Miller and C. Valasek, "A survey of remote automotive attack surfaces," in *BlackHat USA*, 2014.

[74] M. R. Moore, R. A. Bridges, F. L. Combs, M. S. Starr, and S. J. Prowell, "Modeling inter-signal arrival times for accurate detection of can bus signal injection attacks: a data-driven approach to in-vehicle intrusion detection," in *Proceedings of the 12th Annual Conference on Cyber and Information Security Research*. ACM, 2017, p. 11.

[75] H. Lee, S. H. Jeong, and H. K. Kim, "Otids: A novel intrusion detection system for in-vehicle network by using remote frame," in *15th International Conference on Privacy, Security and Trust (PST)*, 2017.

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

[77] R. E. Malinowski and T. Xaypanya, "System and method for detecting malicious activity and harmful hardware/software modifications to a vehicle," Dec. 20 2016, uS Patent 9,525,700.

[78] M. Markovitz and A. Wool, "Field classification, modeling and anomaly detection in unknown can bus networks," *Vehicular Communications*, vol. 9, pp. 43–52, 2017.

[79] M.-J. Kang and J.-W. Kang, "A novel intrusion detection method using deep neural network for in-vehicle network security," in *83rd IEEE Vehicular Technology Conference (VTC Spring)*. IEEE, 2016, pp. 1–5.

[80] S. Abedi, A. Arvani, and R. Jamalzadeh, "Cyber security of plug-in electric vehicles in smart grids: Application of intrusion detection methods," in *Plug-In Electric Vehicles in Smart Grids*. Springer, Singapore, 2015, pp. 129–147.

[81] C. E. Everett and D. McCoy, "Octane (open car testbed and network experiments): Bringing cyber-physical security research to researchers and students." in *6th workshop on cyber security experimentation and test, USENIX*, 2013.

[82] U. Khan, S. Agrawal, and S. Silakari, "A detailed survey on misbehavior node detection techniques in vehicular ad hoc networks," in *Information systems design and intelligent applications*. Springer, 2015, pp. 11–19.

[83] M. Raya, P. Papadimitratos, I. Aad, D. Jungels, and J.-P. Hubaux, "Eviction of misbehaving and faulty nodes in vehicular networks," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 8, 2007.

[84] T. H.-J. Kim, A. Studer, R. Dubey, X. Zhang, A. Perrig, F. Bai, B. Bellur, and A. Iyer, "Vanet alert endorsement using multi-source filters," in *Proceedings of the seventh ACM international workshop on VehiculAr InterNETworking*. ACM, 2010, pp. 51–60.

[85] K. Zaidi, M. B. Milojevic, V. Rakocevic, A. Nallanathan, and M. Rajarajan, "Host-based intrusion detection for VANETs: a statistical approach to rogue node detection," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 8, pp. 6703–6714, 2016.

[86] A. Daeinabi and A. G. Rahbar, "Detection of malicious vehicles (dmv) through monitoring in vehicular ad-hoc networks," *Multimedia tools and applications*, vol. 66, no. 2, pp. 325–338, 2013.

[87] N. Kumar and N. Chilamkurti, "Collaborative trust aware intelligent intrusion detection in VANETs," *Computers & Electrical Engineering*, vol. 40, no. 6, pp. 1981–1996, 2014.

[88] C. A. Kerrache, N. Lagraa, C. T. Calafate, and A. Lakas, "Tfdd: A trust-based framework for reliable data delivery and dos defense in VANETs," *Vehicular Communications*, 2016.

[89] K. Verma, H. Hasbullah, and A. Kumar, "Prevention of dos attacks in VANET," *Wireless personal communications*, vol. 73, no. 1, pp. 95–126, 2013.

[90] K. Verma and H. Hasbullah, "Bloom-filter based ip-chock detection scheme for denial of service attacks in VANET," *Security and Communication Networks*, vol. 8, no. 5, pp. 864–878, 2015.

[91] N. Bissmeyer, K. H. Schroder, J. Petit, S. Mauthofer, and K. M. Bayarou, "Short paper: Experimental analysis of misbehavior detection and prevention in VANETs," in *Vehicular Networking Conference (VNC), 2013 IEEE*. IEEE, 2013, pp. 198–201.

[92] N. Bissmeyer, "Misbehavior detection and attacker identification in vehicular ad-hoc networks," Ph.D. dissertation, Technische Universität, 2014.

[93] S. Chang, Y. Qi, H. Zhu, J. Zhao, and X. Shen, "Footprint: detecting sybil attacks in urban vehicular networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 6, pp. 1103–1114, 2012.

[94] J. Grover, M. S. Gaur, V. Laxmi, and N. K. Prajapati, "A sybil attack detection approach using neighboring vehicles in VANET," in *Proceedings of the 4th international conference on Security of information and networks*. ACM, 2011, pp. 151–158.

[95] T. Bouali, S.-M. Senouci, and H. Sedjelmaci, "A distributed detection and prevention scheme from malicious nodes in vehicular networks," *International Journal of Communication Systems*, vol. 29, no. 10, pp. 1683–1704, 2016.

[96] L. Mokdad, J. Ben-Othman, and A. T. Nguyen, "Djavan: Detecting jamming attacks in vehicle ad hoc networks," *Performance Evaluation*, vol. 87, pp. 47–59, 2015.

[97] K. S. Narendra and M. A. Thathachar, "Learning automata-a survey," *IEEE Transactions on systems, man, and cybernetics*, no. 4, pp. 323–334, 1974.

[98] P. C. Mahalanobis, "On the generalised distance in statistics," *Proceedings of the National Institute of Sciences of India*, pp. 49–55, 1936.

[99] K. D. Jones, K. Tam, and M. Papadaki, "Threats and impacts in maritime cyber security," *Engineering & Technology Reference*, 2016.

[100] D. Egan, N. Drumhiller, A. Rose, and M. Tambe, "Maritime cyber security university research: Phase 1," US Coast Guard New London United States, Tech. Rep., 2016.

[101] T. Ntouskas and D. Gritzalis, "Innovative security management services for maritime environment," in *NATO maritime interdiction operational training Centre 1st Cyber Security Conference*, 2016.

[102] F. Katsilieris, P. Braca, and S. Coraluppi, "Detection of malicious ais position spoofing by exploiting radar information," in *2013 16th international conference on Information fusion (FUSION)*. IEEE, 2013, pp. 1196–1203.

[103] F. Mazzarella, M. Vespe, A. Alessandrini, D. Tarchi, G. Aulicino, and A. Vollero, "A novel anomaly detection approach to identify intentional ais on-off switching," *Expert Systems with Applications*, vol. 78, pp. 110–123, 2017.

[104] H. Packard, "HP Threat Central," April, 2015.

[105] M. Swan, "Connected car: quantified self becomes quantified car," *Journal of Sensor and Actuator Networks*, vol. 4, no. 1, pp. 2–29, 2015.

[106] S. Gorman, Y. J. Dreazen, and A. Cole, "Insurgents hack us drones," *Wall Street Journal*, vol. 17, 2009.