Received December 18, 2018, accepted January 4, 2019, date of publication February 14, 2019, date of current version February 27, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2894183

Intrusion Detection Systems for Intra-Vehicle Networks: A Review

OMAR Y. AL-JARRAH[1], CARSTEN MAPLE[1], MEHRDAD DIANATI[1], DAVID OXTOBY[2], AND ALEX MOUZAKITIS[2]

[1]Warwick Manufacturing Group, The University of Warwick, Coventry CV4 7AL, U.K.
[2]Jaguar Land Rover, International Digital Laboratory, The University of Warwick, Coventry CV4 7AL, U.K.

Corresponding author: Omar Y. Al-Jarrah (omar.al-jarrah@warwick.ac.uk)

This work was supported by Jaguar Land Rover and the UK-EPSRC as a part of the jointly funded Towards Autonomy: Smart and Connected Control (TASCC) Programme under Grant EP/N01300X/1, and in part by the Alan Turing Institute through EPSRC under Grant EP/N510129/1.

ABSTRACT A modern vehicle is a complex system of sensors, electronic control units, and actuators connected through different types of intra-vehicle networks to control and monitor the state of the vehicle. In addition, modern vehicles are becoming increasingly connected to the outside world through V2X technologies. However, these provide new attack surfaces that increase the cybersecurity risk to modern vehicles. To this end, there are two distinct and key challenges that need to be addressed to ensure safety and consumer trust. While modern vehicles must be equipped with the best countermeasures against cybersecurity threats, a reliable mechanism shall be also in place to detect the potential intrusions of the system while in operation, which is termed as intrusion detection. This paper provides a structured and comprehensive review of the state of the art of the intra-vehicle intrusion detection systems (IDSs) for passenger vehicles. We first provide an overview of intra-vehicle networks before reviewing contemporary research in intra-vehicle IDSs. The approach employed is to categorize the reviewed works based on their detection technique and to examine the used feature and feature selection methods, evaluation dataset, attack type, performance metrics, and benchmark models. This paper also presents outstanding research challenges and gaps in intra-vehicle IDS research.

INDEX TERMS CAN bus, intra-vehicle network, intrusion detection, intrusion detection systems (IDSs).

I. INTRODUCTION

The rapid adoption of various modern technologies by car manufacturers in the last decade has revolutionized the shape and functions of modern vehicles (*i.e.*, cars). This includes advanced functions such as automation features and interconnectivity with the external world (*e.g.*, Vehicle to Vehicle (V2V) and Vehicle to Infrastructure (V2I) communications) to improve safety and enable cooperation between vehicles.

A typical modern vehicle integrates a set of networked components including sensors, actuators, Electronic Control Units (ECUs), and communication devices [1]. Sensors, for example, can help the vehicle perceives its surrounding environment and uses that perception to automate various aspects of driving functions using several ECUs [2]. Each ECU has usually a specific function (*e.g.*, steering angle control) and ECUs are grouped based on their functions into subnetworks (*e.g.*, powertrain, infotainment, *etc.*). The subnetworks are interconnected through several gateways, structuring the intra-vehicle network of the vehicle. Typically, the ECUs communicate via the Controller Area Network (CAN), a de-facto network protocol for in-vehicle communication.

The low cost, relatively high reliability, and fault tolerance properties of CAN motivate its use as a standard for intra-vehicle networking. However, it has been recognized that CAN is vulnerable to cyberattacks, meaning that many modern vehicles can be exposed to new threats owing to the connectivity to external networks. Two of the main reasons of such vulnerability lie in the fact that the CAN protocol lacks message authentication and the broadcast transmission [2]. An intruder can send messages through the CAN once he/she gain access to it, since the CAN protocol does not provide an authentication mechanism to verify the origin of the messages. Checkoway *et al.* [3] have shown that remote exploitation of a vehicle is possible via a broad range of attack vectors (such as CD players, bluetooth, and cellular), which

TABLE 1. A Comparison of This Survey With Existing Survey Articles.

Work	Intra-vehicle Network IDSs	Categorisation of Existing Work	Features and Feature Selection	Evaluation Data	Attack Type	Performance Metrics	Benchmark Models	Research Gaps
[8]		✓			✓			
[10]					✓			✓
[11]		✓		✓	✓			✓
[12]					✓			✓
[13]		✓			✓			✓
[14]					✓			✓
[15]		✓		✓	✓			✓
This survey	✓	✓	✓	✓	✓	✓	✓	✓

can lead to remote vehicle control [4]. This makes vehicular security a major concern in the automotive industry, as well causing alarm to the public.

Protecting vehicles against cyberattacks is a challenging task since historically vehicles have been designed without comprehensive security requirements in mind, having relied on the assumption that vehicles operate independently with no communication capabilities. In this respect, conventional proactive security countermeasures (such as encryption algorithms, access control and the like) might not be applicable to modern vehicles due to the high connectivity (giving a large attack surface), time-sensitivity, resource constraints, and complexity of modern vehicles [5], [6]. Recently, a considerable attention has been paid to reactive systems, such as Intrusion Detection Systems (IDSs), as complementary solutions to proactive security countermeasures, since they can detect potential cyberattacks in intra-vehicle networks as well as misbehaviors in connected vehicular networks [7].

Although there is an abundance research in the literature discussing vehicles cybersecurity (see [8], [9]), to the authors' best knowledge, a systematic review of the state-of-the-art of intra-vehicle IDS can be a valuable contribution to provide a fresh and critical review of the most recent studies on this particular topic. This we believe will complement the existing broad surveys such as that in [1], [10], and [11], that mainly focus on the security of cyber-physical systems with broad reference to IDS. Two other related articles were published by Studnia *et al*. [8] and Liu *et al*. [12] that mainly survey security threats and protection mechanisms in embedded automotive networks and briefly mentioned IDSs. Also, van der Heijden *et al*. [13] and Sakiz and Sen [14] have studied attacks and detection mechanisms in intelligent transportation systems, concentrating on vehicular ad-hoc networks rather than on intrusion detection in intra-vehicle networks. Zarpelão *et al*. [15] presented a survey about IDSs for Internet of Thing (IoT), identifying leading trends, open issues, and future research possibilities. Zarpelão *et al* classified the IDSs proposed in the literature according to detection method, IDS placement strategy, security threat and validation strategy.

Unlike the aforementioned surveys, this paper provides a thorough, comprehensive, and systematic review of state-of-the-art IDSs for intra-vehicle networks. To help position the contributions of this paper with respect to the existing published work, Table 1 shows a comparison between this survey and relevant survey articles.

In this paper, we review contemporary research of intra-vehicle IDSs and discuss in detail their current state focusing on the used detection technique, features and feature selection method, evaluation data, performance metrics, and benchmark models, and the targeted attack types. We also discuss the current issues and challenges of intra-vehicle IDSs. The contributions of this paper are as follows:

1) Providing an overview of intra-vehicle networks, discussing merits and shortcomings of prevalently used intra-vehicle networks.
2) Reviewing and categorizing contemporary research of intra-vehicle IDSs (42 works) based on their detection technique, and examining each work based on the used features and feature selection method, evaluation data, performance metrics and benchmark models, and targeted attack types.
3) Providing a comprehensive summary of contemporary research in intra-vehicle IDSs (Tables 5–7).
4) Discussing and identifying challenges and current gaps in the landscape of intra-vehicle IDSs research, supported by statistical analysis of existing work.

The remainder of this paper is organized as follows: Section II gives an overview of intra-vehicle networks. Section III reviews recent researches into intra-vehicle IDS and categorizes them into two main categories: flow-based IDSs and payload-based IDSs and then further categorizes them based on the detection technique employed. It then discusses most commonly used/studied features and feature selection methods, datasets, attack types, performance metrics and benchmark models in intra-vehicle IDSs research. Section IV provides a discussion and identifies research gaps and challenges before Section V concludes this paper.

II. AN OVERVIEW OF INTRA-VEHICLE NETWORKS

Today's vehicle is a formidable sensor platform transmitting around 2500 signals internally with an approximately 70 ECUs connected via an intra-vehicle network [4], [16]. The intra-vehicle network facilitates data sharing among sensors, ECUs, and actuators, enabling the operation of the vehicle. There are five widely used intra-vehicle networks in the modern intra-vehicle communication systems: Local

TABLE 2. Selected Comparisons of Intra-vehicle Networks [16].

Network	Relative System Cost	Bandwidth (bits/s)	Max. Protocol Efficiency	Relative Fault Tolerance	MAC Mechanism	Typical Topology	Layers in OSI Model	Security Threat	Typical Apps in Vehicles
LIN	Low	11.2K or 19.6K	51.6%	Low	Polling	Linear Bus	1,2,7	Low	Battery Monitoring, Window Lifter Control, Steering Wheel Button Assembly, Temperature Sensors, Blower Control, Sunroof Module, Alternator Module
CAN	Low to Medium	125K or 500K	59.6%	Low to Medium	CSMA/CA	Mostly Linear Bus	1,2,7	High	Engine Controller, Transmission Unit, Electrical Stability Control, Seat Module, Cluster Control, Upper Body Control, Climate Control, Smart Electrical Centers, Headlamp Assembly, Trailer Module, Standard OBD-II Interface
FlexRay	High	5M or 10M	96.95%	High	TDMA	Linear Bus, Star, or Hybrid	1,2,7	Medium	Steering Angle Sensor, Safety Radar, All-Wheel Drive, Throttle Control, Dynamic Suspension Control, Supplementary Restrain System, Active Safety System, Network Backbones
Ethernet	Medium	100M	97.53%	Medium to High	CSMA/CD	Point-to-Point	1,2,7	High	ECU Flash Interface, Cameras, Lidar, Safety Radar, Entertainment Unit, Wireless or Consumer Electronics Connector, Network Backbones
MOST	Medium to High	25M, 50M,or 150M	96.88%	Medium	Time Division Multiplex/CSMA	Ring	All Layers	Medium to High	Infotainment Head Unit, Central Console Display, Amplifire Control, Rear Seat Entertainment Unit, Audio Module, Navigation System

Interconnection Network (LIN), CAN, FlexRay, Ethernet, and Media Oriented Systems Transport (MOST). Each has its own advantages and limitations. Table 2 provides a comparison of intra-vehicle networks.

LIN provides a low communication speed suitable for applications that do not require stringent time performance such as battery monitoring, window lifter control, *etc.*. Although LIN has low cost and security threat, it has a low fault-tolerant capability. On the other hand, FlexRay, Ethernet, and MOST have a higher bandwidth than LIN. However, they have higher relative system cost and security threat compared to LIN. The high bandwidth and efficiency of FlexRay, Ethernet, and MOST make them suitable for stringent time performance and bandwidth demanding applications. For example, FlexRay is used in steering angle sensor, safety radar, throttle control, *etc.*, whereas Ethernet and MOST are typically used in ECU flash interface and infotainment system, respectively [16]. Among intra-vehicle networks, CAN is the most popular due to its relatively low cost, mature full-scale tool chains (*e.g.*, data dictionary design, production code generation, network simulation, *etc.*), acceptable performance in noise-resistance, and fault tolerance. However CAN is vulnerable to security threats. CAN is commonly used in powertrain and upper body electronics [16].

Typically, a CAN comes with a bus topology that supports a baud rate higher than 125 Kbit/s and is characterized by two 120Ω terminal resistances at the end of the transmission cable. CAN protocol does not require any global synchronization or timing to regulate the communication since nodes connected to the CAN bus synchronize their timing with the sender when receiving messages. Each node connected to the CAN has the right to access the bus when a transmission is ready and the bus is idle. The CAN bus is a broadcast domain in which all nodes connected to the CAN bus receive the transmitted message. The reception filter of each node decides which message to select using the ID of the message. When multiple nodes attempt to transmit messages at the same time, they compete for access to the bus through a non-destructible Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) and Arbitration on Message Priority (AMP) where the message with the lower ID wins the arbitration process [16]; the message with the lower ID has the higher priority. Figure 1 shows the structure of a standard CAN frame.

As can be seen in Figure 1, a CAN frame consists of seven fields as follows:

- Start of Frame: a single dominant bit that informs a start of transmission to all nodes.
- Arbitration Field: it consists of two main parts; the identifier field that represents the ID of the message/frame and is used during the arbitration process; and the Remote Transmission Request (RTR) which is determined according to the kind of the CAN frame.
- Control Field: it has two reserved bits and four Data Length Code (DLC).
- Data Field: holds the actual data transferred to other nodes.
- CRC Field: it guarantees the validity of the message. All other nodes that receive the message verify the message using this code.
- ACK Field: it consists of two bits: ACK part and delimiter part. A node that receives a valid message replaces the ACK part, which is a recessive bit (*i.e.*, logical 1), with a dominant bit (*i.e.*, logical 0).

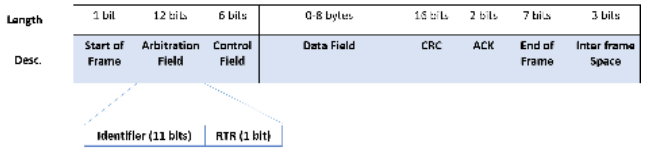

FIGURE 1. Structure of CAN Frame.

- End of Frame: a flag that consists of seven recessive bits and indicates the end of the frame.

III. IDSS FOR INTRA-VEHICLE NETWORKS

Several security countermeasures have been developed to defend systems against cyberattacks. Proactive countermeasures, such as encryption algorithms and access control, aim to prevent attackers from gaining access to the systems. Although proactive countermeasures can protect systems from external cyberattacks, they have limited capability in front of internal attacks. On the other hand, reactive countermeasures (*e.g.*, IDSs) identify cyberattacks once they occur. Conventionally, IDSs are categorized based on their detection technique into: misuse/knowledge and behavior/anomaly - based IDSs [11].

A knowledge-based IDS compares observed events with patterns (*i.e.*, signatures) of known attacks. The IDS reports an intrusion when it finds a match between the observed events and the known attacks' patterns. Generally speaking, knowledge-based IDSs have low false positive rate; as they react only to previously known attacks. However, such approaches have limited capabilities in detecting novel attacks (*e.g.*, zero-day attack). In addition, the signatures database must stay current, which is a challenging task especially with the constant growth in the number of novel attacks. Moreover, storing large signature database and performing pattern matching on it is a demanding process in terms of memory, CPU time, and power [17]. Monther Aldwairi and Jarrah [17] presented a parallel IDS approach to accelerate the pattern matching operation through parallelizing a matching algorithm on a multi-core CPU.

An anomaly-based IDS identifies normal system behavior and considers significant deviations from the normal behavior as intrusions. Such an approach does not need generating a signature for each attack, making it capable of detecting novel attacks. However, such an approach has a relatively high false positive rate. In addition, modeling of normal system's behavior requires attacks-free data, which are not always available in real-world. Generally speaking, anomaly-based IDSs require less memory compared to signature-based IDSs since anomaly-based IDSs do not store attacks' signatures [1].

A. CATEGORIES OF INTRA-VEHICLE IDSS

We categorize intra-vehicle IDSs into: flow-based, payload-based, and hybrid IDSs. A flow-based IDS monitors the internal network of a vehicle, typically the CAN bus, and extracts distinct features (*e.g.*, message frequency and interval) from the messages transmitted on the network. It then uses the extracted features to identify intrusions or abnormal behaviors without inspecting the payload/content of the messages. On the other hand, a payload-based IDS examines the payload of the messages to identify intrusions. A hybrid IDS is a combination of the former two categories. We further categorize the main categories based on their detection technique into sub-categories as in Figure 2. In the following subsections, we review contemporary researches that belong to each sub-category. Despite our effort, this paper might not cover all work reported in the literature of intra-vehicle IDSs.

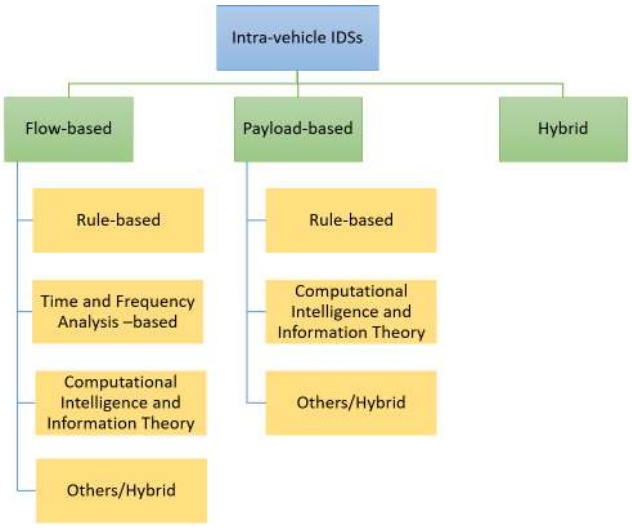

FIGURE 2. Categories of Intra-vehicle IDSs.

1) FLOW-BASED IDSS

(I) Rule-based IDSs: Vuong *et al.* [18] developed a decision tree-based detection model for cyberattacks. The detection model is built using eight on-board physical and cyber features. Experimental evaluation of the proposed model, against four types of attack on a small-scale robotic vehicle, revealed that considering physical features noticeably improves the detection accuracy for two of the four types and reduces the detection latency of all types of attack. However, the proposed system has a high detection latency of about 1sec.

Fu *et al.* [19] proposed an FPGA-based IDS based on novel data model named Link-NFA. The proposed system prototype was applied in vehicular environment and achieved a high real-time performance, a throughput of more than 39 Gbps, which is about 15% higher than state-of-the-art techniques. The total power consumption of the prototype is about 7.5W and the processing latency is about 4 μs, which is about one sixtieth part of the popular software IDSs.

However, the proposed model is specified for FPGA, making it unlikely to be implemented in other hardware platforms such as GPU and CPU. In addition, the number of NFAs in Link-NFA is related to RegEx, which

may grow rapidly in some extreme cases. Moreover, the implemented rule-set is not an Internet of Vehicle (IoV) rule-set, which makes the applicability of the proposed system to vehicular system questionable.

(II) Time and Frequency Analysis -based IDSs: Hoppe *et al*. [20] discussed the main requirements of an automotive application of intrusion detection approaches and how they differ from the requirements of conventional computer networks applications. With reference to a practically demonstrated attack on an automotive network, the authors presented a prototypically anomaly-based IDS. The presented system tracks all CAN messages having a target message type (*e.g.*, 0x395) and evaluates two different characteristics; the current frequency of these messages and the semantical meaning of the previous messages. In addition, the authors discussed the reaction to attacks once detected. The authors proposed to utilise multimedia devices (*e.g.*, visual display, acoustic, haptic) of a vehicle and to use them as computer-human-interfaces to adaptively report security incidents to the driver/occupants of the vehicle while considering the conditions of the surrounding environment (*e.g.*, high noise). The authors argued that different reactive measures should be used to report security incidents depending on driving conditions. For instance, visual display/dashboard might be used to report security incidents when there is a high noise.

Ling and Feng [21] presented an algorithm for intrusion detection in CAN. The concept is based on combining the IDs of messages transmitted on the CAN bus with their interruptible occurrence frequency. For a given input ID, the algorithm counts the number of continuous messages that belong to the given message type (*i.e.*, ID). If the count of messages in the interruptible sequence is greater than a predefined threshold, the algorithm raises an alarm of a possible attack. Although its simplicity, the proposed algorithm has limited capability in detecting attacks that manipulate the content of messages while maintaining their frequency.

Taylor *et al*. [22] proposed and compared the performance of a frequency-based detector with One-Class Support Vector Machine (OCSVM). Experimental results on real data of a Ford Explorer over a range of packet/message insertion and deletion showed that the frequency-based can detect anomalies with an Area Under Curve (AUC) between 0.8720 and 1.0000 over a time window of 1sec. On the other hand, the OCSVM can detect anomalies with an AUC between 0.9874 and 0.9893 over a time window of 0.4sec. However, the proposed approach is suitable only for periodic messages.

Song *et al*. [23] proposed a lightweight IDS based on a statistical analysis of time intervals of CAN messages. The authors concluded that the analysis of the time interval of CAN messages is a meaningful feature to detect packet/message injection attack. The main concept is to analyze traffic anomalies based on message frequency. Under normal operation conditions, messages generated by ECUs have their own regular frequency or interval. When a vehicle under message injection attack, these frequencies or intervals are unexpectedly changed. Typically, the frequency of messages under injection attack increases by 20–100 times higher than the normal case. Experimental results on CAN messages of a real anonymized vehicle showed that the proposed method is very effective and efficient in detecting message injection attack without any false alarms.

Young *et al*. [24] introduced a road map towards a security solution for intra-vehicle networks. The proposed solution can detect anomalies, identify failed states of the network, and adaptively respond in real-time to maintain a fail-operational system. The authors argued that observing message sequences is essential to detect semantic attacks that span multiple state transition. Based on the observation that control messages are high priority, periodic and predictable messages, the proposed IDS partitions incoming messages into control and non-control messages. It then uses an algorithm to examine the control messages exploiting the high predictability of such messages and a kernel-based Machine Learning (ML) algorithm to detect sequential anomalies. However, generally speaking, kernel-based models have a substantial time complexity, deteriorating their efficiency for intrusion detection tasks in vehicular systems. In addition, such an approach is not applicable to aperiodic messages (*e.g.*, even-driven messages), as the high variability of such messages might lead to a high number of false positives.

Cho and Shin [25] built an effective IDS called Clock-based Intrusion Detection System (CIDS), which can detect various types of attack including the masquerade attack. Since the CAN protocol does not provide the identity of the transmitter in the CAN message, the authors fingerprinted ECUs with other ''leaked'' information. The authors exploited message periodicity to extract and estimate transmitters' clock skews, which can be used to fingerprint the transmitter ECUs. The total amount of offset (the accumulated clock offset) is obtained by summing up the absolute values of the average clock offsets. By definition, the slope of the accumulated clock offset would thus represent the clock skew, which is constant. This enables the proposed CIDS to estimate the clock skew from arrival timestamps and thus fingerprint the message transmitter for intrusion detection.

For a given message ID, CIDS derives the accumulated clock offset inherent in the arrival timestamps. Since the clock skew is constant, the accumulated clock offset is linear in time, and hence CIDS describes it as a linear regression model. A linear parameter identification

problem is thus formulated and solved by Least Square algorithm.

The authors used a CAN prototype and a Honda accord for experiments. Approximately, 2.25 million messages of 30 minutes of driving were collected and the CIDS was tested against three types of attacks: suspension (*i.e.*, deletion), fabrication (*i.e.*, injection), and masquerade attacks. Experimental results have shown that the CIDS not only can detect different types of attacks with high accuracy but also can identify the origin of the attack. The proposed system achieved a True Positive Rate (TPR) of 100% and False Positive Rate (FPR) of 0.055%. However, the CIDS can pin-out the origin of the attack only for periodic messages. It would be difficult to fingerprint ECUs that send aperiodic messages. The case when multiple ECUs are compromised and participating in a masquerade attack is not considered. In this case, the CIDS might fail in identifying the origin of the attack. In addition, the authors assumed that the clock skew is inimitable. However, this assumption was refuted by Choi *et al*. [2].

Mabrouka Gmiden and Trabelsi [26] proposed an IDS based on time interval analysis of periodic messages. The proposed system considers messages that do not conform to a learned time periodicity model as attacks or anomalies. However, the authors did not provide any performance evaluation of the proposed system and the proposed model is suitable for detecting anomalies in periodic messages only, limiting it's applicability to anomalies in aperiodic messages.

Lee *et al*. [27] proposed an IDS called Offset Ratio and Time Interval based Intrusion Detection System (OTIDS) based on the analysis of the offset ratio and the time interval between remote frame requests and responses in the CAN. The concept is based on the observation that the offset and time interval of a window, which is defined as the messages between a remote frame request and it's response, of a particular identifier in attack-free state are different compared to when under-attack. Based on this property, intrusions can be detected by monitoring response performance based on the offset ratio and time interval of windows and comparing it to response performance of attack-free state. The OTIDS considers a response to remote frame of an identifier that have an offset ratio or time interval beyond a threshold as an indication of intrusion. The authors argued that the OTIDS is capable of detecting Denial of Service (DoS), Fuzzy, and impersonation attacks in CAN. However, the impact of inserting remote frame on the performance of the vehicle was not studied. This is of interest as inserting remote frames with a high priority might delay transmitting of messages with a lower priority, which might disturb the normal functions of the vehicle. In addition, the detection accuracy and other performance metrics of the proposed system were not presented. Thankfully, the authors made the evaluation dataset available online, facilitating further experiments and research.

Avatefipour [28] proposed a ML-based model that links CAN packets to their sources by learning specific artifacts derived from the physical signal attributes of the received packets. Material and design imperfections in the physical channel and digital device, which are the main contributing factors behind the device-channel specific unique artifacts, were leveraged to link the received electrical signal to the transmitter. A feature vector, which consists of 11 time and frequency domain statistical signal attributes including higher-order moments, spectral flatness measure, minimum, maximum, and irregularity K, was made up of both time and frequency domain physical attributes and then employed to train a neural network-based classifier. Performance of the proposed fingerprinting method was evaluated by using a dataset collected from 16 different channels and four identical ECUs transmitting same message. Experimental results indicated that the proposed model can achieve correct detection rates of 95.2% and 98.3% for channel and ECU classification, respectively.

(III) Computational Intelligence and Information Theory - based IDSs: Müter and Asaj [29] proposed an information theoretical approach based on entropy to detect three types of attack, namely Message Injection (MI), DoS and Plausibility of Interrelated Events. The authors observed that traffic in automotive networks is more restricted than conventional computer networks since every message and its content is specified before transmitting it. This means the entropy (*i.e.*, uncertainty) of the data in the normal network operation is almost fixed and relatively low compared to conventional computer networks. Thus, intrusions that change the entropy of the data might easily be detected by observing the entropy value. For example, MI attack will reduce the entropy value since the number of specific messages will increase. This change in entropy can thus be detected by the IDS and used as an indicator of attacks. The main advantage of the proposed approach is that it requires only records of in-vehicle network traffic as input. However, the proposed approach has limited capability in detecting small-scale attacks which could be part of the normal behavior. The evaluation of the proposed approach has shown that it can successfully detect attacks that deviate from the normal network behavior.

Marchetti *et al*. [30] introduced an entropy-based IDS and evaluated its effectiveness applied to networks included in modern vehicles. The use of entropy as a mean to describe the normal behavior of an information source relies on the following underlying assumptions: (1) the entropy of messages generated by the information source exhibits stable statistical

characteristics; (2) relevant anomalies introduce significant deviations in the statistical characteristics of the entropy. Marchetti *et al*. observed that the entropy values are stable and their distributions are similar to the normal distribution. Since entropy values appear to be rather stable over time and distributed according to a normal distribution, the authors proposed an anomaly detection algorithm based on the assumption that entropy values that are too distant from the average entropy are unlikely, and should be considered as anomalies.

An anomaly is reported if the entropy value is not included in an acceptable parametrized range that defines the sensitivity of the algorithm with respect to deviations from the expected mean. Attacks to in-vehicle networks were simulated by injecting different classes of forged CAN messages in traces captured from a modern licensed vehicle. Experimental results showed that if entropy-based anomaly detection is applied to all CAN messages, it is only possible to detect attacks that comprise a high volume of forged CAN messages. On the other hand, attacks characterized by the injection of few forged CAN messages can be detected only by applying several independent instances of the entropy based anomaly detector (one for each class of CAN messages). The main advantage of this approach is the complete independence with respect to the content of CAN messages, hence it can be applied immediately to the CAN bus of any vehicle without the need of proprietary information that is necessary to interpret the semantic of CAN messages. However, this approach requires several anomaly detectors (one for each ID) to be executed in parallel. Moreover, this approach proves to be ineffective for a small subset of IDs whose entropy exhibits large variations even in normal conditions.

Levi *et al*. [6] proposed a new temporal based detection technique using Hidden Markov Model (HMM) and regression model for vehicle fleet. Important data are collected and then tested against the HMM trained on vehicles' normal behavior. A regression model is built based on temporal features, which is then used to predict an estimated log-likelihood and compare the result with the actual log-likelihood. Experimental results showed that the proposed model has a superior performance, AUC of 0.96, of detecting real-life anomalies. However, the authors did not evaluate the performance of the proposed model on a real dataset. In addition, the authors did not show the detection latency and the training time of the proposed model.

Rieke *et al*. [31] proposed an intrusion detection model based on Petri nets. In the first stage of the proposed model, the discovery stage, a model representing the normal behavior of the vehicle is derived offline where the Alpha algorithm is used to derive a Petri net from traces recorded from the CAN bus. In the second stage, the conformance checking stage, the model is utilized to identify anomalies. Experimental results have shown that the proposed model can achieve a high throughput with a low complexity. However, important performance measures (*e.g.*, detection rate) were not presented in this work. In addition, evaluating the performance of the proposed method against other cyberattacks than message injection would be of great value.

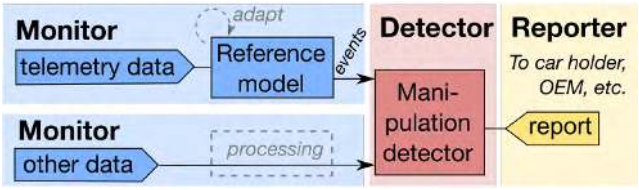

FIGURE 3. Data flow in CAID [32].

Wasicek *et al*. [32] presented a behavior-based Context-Aware Intrusion Detection (CAID) framework that recognizes physical manipulations of the system using cyber means. CAID refines existing IDS approaches by integrating a physical model of the system to establish a context-awareness in the IDS. It integrates three modules: monitors, detectors, and reporters, as in Figure 3. The monitors collect telemetry data by sending queries via the chip/reply protocol and extract features from data points within a time window from a set of ECUs of the vehicle. It then uses the extracted features to build a reference model, which represents the typical behavior of the monitored control system, using bottleneck Artificial Neural Network (ANN).

During operation, CAID checks the data against the reference model by performing a plausibility check using the reference model. If the actual behavior and the reference behavior differ, it generates an event that indicates a potential intrusion. The detector module aggregates the events generated form the monitor modules with other relevant information in the detector module. This implements the second level of context. Once an intrusion is detected by the detector module, reporters perform the actual reporting of the event to a party (*e.g.*, the driver). Experiments using a test vehicle have shown that CAID is able to recognize the chip tuning with a very high probability. However, the data used in the experiments were collected in a well-defined environment on urban and highway roads, which might not the case in real-world where different driving conditions and environments are expected. For instance, the impact of road conditions was not presented in this work.

Choi *et al*. [2] proposed a novel automotive IDS called VoltageIDS, which examines distinct characteristics of the electrical CAN signal corresponding to CAN messages in order to identify the sender of the messages. Conceptually, if two different ECUs send

the same message, there would be an inconsistency between the two signals in the physical layer due to the different cables lengths and wiring resistances, which increase with the wire length.

VoltageIDS is composed of three phases: i) signal measurement and preprocessing, ii) feature extraction, and iii) intrusion detection. In the first phase, the VoltageIDS pre-processes the electrical signal of CAN messages to obtain the dominant state, positive and negative-slope portions of the signals, where the signal state changes from recessive to dominant state and vice-versa, which contain unique properties generated by several passive (*e.g.*, resistance) and active (*e.g.*, capacitor) transmitter components.

In the feature extraction phase, the VoltageIDS extracts all possible features found to be outstanding on node identification for preprocessed signals (*i.e.*, dominant, positive-slope, and negative-slope). It considers 20 time and frequency features for each signal, resulting in 60 features for each CAN message. This is followed by a feature selection method, the Sequential Forward Selection (SFS), that selects most important features out of the 60 features.

The intrusion detection phase can be divided according to the attack type. For masquerade attack detection, the VoltageIDS can detect a masquerade attack by building a supervised multi-class classifier where the number of classes is equal to the number of ECUs in the network. The multi-class classifier (*i.e.*, Support Vector Machine (SVM) or Bagged Decision Tree (BDT)) is trained on an attack-free dataset of labeled data where the CAN ID is the target class. To classify a new CAN message, the VoltageIDS extracts 60 features or the most important features from the message signal and the multi-class classifier is then used to predict the CAN ID of the message. If the prediction is different from the actual CAN ID, VoltageIDS reports an intrusion. To detect buss-off attack, the VoltageIDS adopts a simple thresholding approach based on the assumption that the bus-off attack is an unknown signal that deviates from the normal behavior. The VoltageIDS builds an OCSVM classifier on unlabeled normal signals from legitimate ECUs and considers the signals below the normal behavior threshold as intrusions. The authors extended the VoltageIDS to learn incrementally and makes it robust against temperature changes.

Experimental results on CAN prototype and vehicles, Hyundai Sonata 2010 and Kia Soul 2014, have shown that the VoltageIDS can detect the masquerade attack with a high accuracy. The experimental results also have shown that the VoltageIDS is capable of detecting the bus-off attack with a FPR of 0%. However, signal preprocessing and feature extraction of CAN messages might be time consuming processes, which might increase the time required to detect possible intrusions, hindering the use of VoltageIDS in real-world. Unfortunately, the training and testing time of the proposed method were not provided.

(IV) Others/Hybrid IDSs: Boudguiga *et al.* [33] proposed a simple IDS for CAN where each ECU periodically registers with other ECUs by sending a *Domain_Activation Frame*. The registered ECU monitors the CAN bus for messages (*i.e.*, messages that have identifiers associated with the ECU) that have been sent on its behalf by a malicious entity. Once a forge message is detected, the ECU erases the forged message by sending an error frame. It then notifies the other ECUs about the detected intrusion by sending a *Domain_Violation Frame*. However, the authors did not present experiments to show the effectiveness of the proposed IDS. In addition, the proposed method needs key management capability, which might demotivate its use in real-world.

Marchetti and Stabili [34] proposed an anomaly-based IDS based on the analysis of messages flow sequences on the CAN bus. A normal behavior model is built based on recurring patterns observed within the sequence of message identifiers. The training phase of the proposed model includes analyzing traces of real CAN bus traffic of an unmodified licensed vehicle in normal operation conditions (*i.e.*, attack-free conditions). The output of the training phase is a transition matrix that identifies all legit transitions between the messages identifiers of two consecutive messages. A CAN message that does not conform with the transition matrix is identified as an attack. Experimental evaluations based on real CAN traces demonstrated a high performance represented by a high detection and a low FPR. However, the proposed system has a low detection rate, between 20% and 40%, for the replay attack.

2) PAYLOAD-BASED IDSS

(I) Rule-based IDSs: Bezemskij *et al.* [35] developed a detection mechanism that monitors real-time cyber and physical features from different on-board sources (*e.g.*, sensors, networks, and processing) of a robot vehicle. In the learning phase, the vehicle learns the normal value range of the features (*i.e.*, normal behavior profile). The normal behavior profile is based on signature characteristics of each data source. If an observed value of a feature is out of its normal range, the detection mechanism reports an attack. The tolerance of the proposed mechanism towards false positives and false negatives is controlled by a sensitivity index and individual weights for features are used to fine tune their importance in detecting anomalies, resulting in an improved detection accuracy. The proposed mechanism was evaluated on three types of attack: compass manipulation, MI and rouge node attack. The proposed method achieved an AUC of 1, 1, and 0.875 for the three attacks, respectively. However, the performance

of the proposed mechanism on other attacks was not presented. In addition, the proposed mechanism was evaluated on a robot vehicle with a limited mobility capability, which might not be a good representative of the real-world where vehicles are more capable and various driving conditions exist.

Abbott-McCune and Shay [36] presented an IDS that monitors the CAN bus and finds anomaly by matching the Start Of Frame (SOF) field of messages to the ones programmed in the ECU connected to the CAN bus. If there is a match and the ECU is not transmitting, the ECU identifies a replay attack and would send an alert to the detector alerting that a replay attack has taken place. An invalid message is the message that has an arbitration identifier not associated with the ECUs on the CAN bus segment. Each logical segment would require a device, which could be implemented on the gateway, to monitor all traffic and compare messages' identifiers to the preprogrammed valid identifiers from the manufacturer. Messages with identifiers that are not part of the logical CAN segment are reported as unknown arbitration identifiers.

Markovitz and Wool [37] presented the design and evaluation of a novel domain-aware anomaly detection system for intra-vehicle CAN bus network traffic. The authors developed a classifier that is able to split the CAN messages into fields and identify the field type and its boundaries without any prior knowledge of the message format. The authors observed the presence of three type of fields: Constant, Multi-value and Counter or Sensor fields. The detection system then builds a model for each ECU based on the characteristics (*i.e.*, field type and boundary) of the messages of the ECU obtained from the classifier. Each field has a type, and specific properties according to the type: the constant value for Const, the list of all the observed values for Multi-value, and the minimal and maximal observed values for Counter/Sensor. The model is based on Ternary Content-Addressable Memory (TCAM), which is a special type of high-speed memory usually used by modern switches and routers for fast look-up tables and packet classification.

The TCAM holds a database D of patterns $d_1, ..., d_k$, each consists of three symbols: $0, 1$ and $*$ (''don't care"). When presented with a message m (consisting only of 0 and 1 bits), the TCAM identifies (in parallel) whether the database includes a matching pattern d_i, (*i.e.*, whether for every bit position j, if $d_i[j] \neq *$ then $d_i[j] = m[j]$). For each ECU, the authors built a set of TCAMs that only matches messages that fit the properties of all the ECU's fields. Thus, any message that does not match the TCAMs is flagged as an anomaly. Evaluation results showed that the proposed system on simulated CAN bus traffic can achieve a median FPR of 1% with a median of only 89.5 TCAMs. However, the sensitivity of the system to detect attacks was not presented.

Dario *et al.* [38] proposed a novel intrusion detection algorithm for CAN bus based on computing the Hamming distance between consecutive payloads of different classes of ID. This is motivated by the low computational complexity of the Hamming distance. As such, the proposed algorithm requires small memory, which promotes its use on the ECUs of modern vehicle. During the learning phase, the proposed algorithm builds a normal range of valid hamming distances. It then analyses the sequences of payloads of all messages transmitted via the CAN bus and compares the Hamming distance between consecutive payloads of the same ID with respect to the normal range of valid Hamming distances. The proposed algorithm is evaluated on traffic of a licensed unmodified vehicle. Experiments' results have shown that the proposed algorithm is able to detect the MI attack with percentages close to 100% injection of fuzzing messages in cases of both NoRange and SmallRange classes. However, the proposed algorithm has achieved a low Detection Rate (DR) between 20% and 30% of MidRange attacks (*i.e.*, having hamming range > 6), making it unsuitable for detecting replay attack.

(II) Computational Intelligence and Information Theory based IDSs: Theissler [39] proposed an OCSVM-based model that is capable of dealing with multivariate time-series data to detect errors or faults. Experimental results on a real dataset showed that the proposed model can detect anomalies/faults with Training Time (TrT) between 20843 and 63631 sec, Testing Time (TsT) between 4845 and 24145 sec, False Negative (FN) between 0.0/h and 10.5/h, True Negative (TN) between 9/h and 45/h, True Negative Rate (TNR) between 42.9% and 76.9%, and precision between 32.3% and 100%. Although the proposed model was evaluated on a real dataset that contains errors or faults, the performance of the proposed model on a dataset that contains cyberattacks was not presented. In addition, in term of computational complexity, the SVM is considered demanding, especially when dealing with big datasets, as it has a complexity of $O(N^3)$ [40]. This might hinder the applicability of the proposed in real-world.

Narayanan *et al.* [41] developed a HMM to detect anomalous states of vehicles. The authors have collected data from licensed vehicles and formulated the anomalous detection problem as classification problem. HMM was used to generate a detection model which is then used to detect unsafe or anomalous states from the data flowing on the CAN bus. Data captured from the CAN bus were interpreted as a sequence of observations (*i.e.*, sensor reading) using sliding window approach. For example, Speed is 20 mph, RPM is 3000, State of door is closed, *etc.* are modeled as a vector sequence. The generated model predicts

the posterior probabilities of observations of a given sequence. If any probability of observations in the sequence is below a threshold, based on the generated model, it implies that getting that observation in that sequence is very low and hence an anomalous state is identified.

The generated model was evaluated by generating multiple anomalous scenarios. Two cases were considered, data from a single sensor and data from multiple sensors. Experimental results showed that the proposed technique could be successfully used to identify anomalies and hence unsafe states in a vehicle. However, it is not clear how the proposed technique will work with rare state or aperiodic messages. In addition, it is not clear how the value of the threshold has been specified and what is the effect of changing the value of the threshold on the performance of the proposed model.

Kang and Kang [42] proposed an IDS using Deep Neural Network (DNN). The detection model is trained on high-dimensional features extracted from bit streams of in-vehicle network packets exchanged between ECUs. Once the features are trained and stored in the profiling module, the proposed system examines the packets exchanged in the vehicular network to decide whether the system was being attacked or not. Experimental results have shown that the proposed system could provide a low detection latency between 7 and 8 *m*s for processing 3900 packets, a high detection ratio of 99.9%, and a low FPR of 4.3%. However, the values of the learning parameters, such as alpha and the number of iteration, were not provided. In addition, it was assumed that an attacker targets an instrumental panel to deceive a driver by showing a wrong value of the Tyre-pressure Monitor System (TPMS). However, the attacker can target not only the instrumental panel but also any system in the car such as ECUs and the braking system.

Kang and Kang [43] proposed an efficient IDS for in-vehicular network. Deep belief networks were used to pre-train the parameters of a DNN using probability-based feature vectors extracted from in-vehicular packets. The feature is designed with computational efficiency in-mind, where it is extracted directly from the bitstream of a CAN packet by investigating the probability distribution of the bit-symboles of the data field, which excludes the need for decoding during the extraction process. Experimental results showed that the proposed system could provide a high real-time detection ratio of $\sim$ 98% and FPR between 1% and 2%.

Taylor *et al*. [44] proposed a deep learning approach (recurrent Long-Short Term Memory (LSTM)) for detecting five types of attack namely interleave, drop, discontinuity, unusual and reverse attacks. These attacks can be considered as insertion and dropping/deletion attacks. The authors deemed that the proposed approach does not require knowledge of the communication protocol and have the capability of detecting novel attacks. However, the proposed approach does not consider the inter-dependencies between different message types as it treats each ID's data sequence as independent sequence. It is likely that there are inter-dependencies between IDs as functionality of an ECU depends on the data received from other ECUs. Experimental results showed that the proposed approach can detect certain sequences of messages with a high detection rate of 100%. However, this is not the case for all message sequences.

Theissler [45] proposed an ensemble ML model to detect known and unknown faults in different driving scenarios. The proposed model consists of two-class and one-class classifiers to detect anomalies in univariate and multivariate time-series data. In general, the two-class classifiers yield good results for known fault types whereas the one-class classifiers perform best for previously unseen fault types. The final prediction of the ensemble model is a combination of the individual predictions of the constituent classifiers. The proposed model was evaluated on data from road trials of a Renault Twingo I (model year 2002). The results showed that it is possible to detect different types of faults, namely erroneous injection, erroneous ignition, unavailable engine temperature, and erroneous engine temperature, with a high F2-score between 68.5% and 83.3%.

Ganesan and Shin [46] hypothesized that it is possible to address compromised sensors by exploiting the natural redundancy, which occurs when a physical phenomenon causes symptoms in multiple sensors, found in vehicles. For instance, pressing the accelerator pedal will cause the engine to pump faster and increase the speed of the vehicle. Engine RPM and vehicle speed are multiple sensors which respond in a related fashion to the same cause of the accelerator pedal. The idea is based on identifying the relationship between different sensors under normal operation and detecting anomalous behaviors accurately.

The proposed method uses pairwise correlation between key variables and cluster-analysis to identify distinct behavior of drivers and detect possible attacks. For each time window, the proposed method identifies the cluster describing the context of driving. Then, it performs pairwise cross correlation and compares the computed correlation values with those expected for that cluster. For each pair, the proposed method calculates the deviation from the mean correlation value for that cluster and reports it in terms of number of standard deviations from the mean. It considers and reports deviations that exceed a threshold as attacks. Although the idea of creating driving context and exploiting redundancy is a potential idea to detect abnormality, however how to choose the threshold value in order

to report possible attacks was not presented in this work. In addition, the cause of abnormality could be a number of situations including malicious attacks, ECU faults or extreme driving conditions, however how to differentiate between attacks and faults was not discussed.

Martinelli *et al*. [47] considered the data field of CAN messages as feature vector to discriminate between normal and malicious messages. The authors used four Fuzzy algorithms, namely Fuzzy-RoughKNN, K-nearest Neighbour (KNN), DiscernibilityClassifier and Fuzzy Unordered Rule Induction Algorithm (FURIA), which have been applied to the eight features (*i.e.*, 8 data bytes of the data field of CAN message). Experimental results on real-world data demonstrated that using fuzzy classification algorithms can obtain a FPR between 0 and 0.038, precision between 0.963 and 1, recall between 0.823 and 1, F-Measure between 0.981 and 1, and AUC between 0.986 and 1 in detecting three types of attack, namely DoS, Fuzzy and MI of two types of messages, targeting CAN protocol identification.

Haas *et al*. [48] gave a brief overview of IDSs and discussed some of the major cyber security threats against connected cars. Haas *et al* then explained how IDSs can be implemented using ANN. A similar approach has been adopted in [49] where the authors proposed a Recurrent Neural Network (RNN) -based IDS.

Wang *et al*. [50] proposed a distributed IDS for modification and replay attacks based on Hierarchical Temporal Memory (HTM) learning, which is capable of predicting data flow in real-time based on the status of previous data sequences. The HTM network continuously learns online. When the input data stream changes, the memory of the model is updated and the detection system will also continue to learn new patterns in the CAN network. The proposed method monitors all data sequence and detect anomalies without *a prior* knowledge of underlying bus protocols. Because HTM network has the capability to learn online from streaming data, the algorithm can detect not only the known attacks of CAN bus, but also the unknown attacks.

The overall evaluation of the system proved that the proposed method has a more reliable detection compared with other existing CAN data domain anomaly detection methods, such as HMM and RNN. However, more work is needed to improve the performance of the system in more complex situations. For instance, removing redundant fields in the data can significantly reduce the training time of the model, improving its efficiency. In addition, real datasets, that have more attacks, are needed to evaluate the performance of the proposed system.

Loukas *et al*. [5] proposed a ML-based IDS for robot vehicle. Loukas *et al* have shown experimentally that RNN-based deep learning, which considers the temporal elements of cyberattacks, enhanced by LSTM can improve the intrusion detection accuracy in comparison with standard ML classifiers (*e.g.*, Multi-Layered Perceptron (MLP)). In order to address the high time complexity of deep learning approaches, a cloud-based computational offloading framework was adopted. Loukas *et al.* discussed when offloading is practical from the detection latency perspective. The proposed IDS achieved a detection accuracy of 86.9%.

(III) Others/Hybrid: In a more recent work, Bezemskij *et al*. [51] proposed a method based on Bayesian network that can identify attacks and their sources (*e.g.*, cyber or physical domain). The Bayesian network takes the output of the mechanism proposed in [35] as an input to further identify the origin of the attack. The Hill-Climbing algorithm was adopted to construct a closed Direct Cyclic Graph (DAG) taking into account all entities given to the Bayesian network. Experimental results have shown that the proposed method can detect cyberattacks with high accuracy (AUC value between 0.950 and 0.995), and a lower accuracy for normal observations (AUC value between 0.862 and 0.983) and physical attacks (AUC value between 0.686 and 0.954).

3) OTHERS/HYBRID

Müter *et al*. [52] introduced a set of detection sensors: formality sensor, location sensor, range sensor, frequency sensor, correlation sensor, protocol sensor, plausibility sensor and consistency sensor, which allow the recognition of attacks during the operation of vehicles without causing false positives. This is because the detection sensors are based on unambiguous and reliable information only, which are the network protocol specifications, the defined cooperative networking behavior of the devices, redundant data sources in the vehicle, or a combination of these. Therefore, an incident is reported only when the system is in an abnormal state. Though these sensors can be used to detect attacks without false positive, not all attacks are detectable by these sensors. For example, if the attacker is able to inject messages that are fully compliant to the network's normal behavior and plausible to previous values. In addition, it is difficult to determine if the abnormality is caused by an attack, error, or failure.

Zhang *et al*. [53] identified and discussed the unique challenges of malware detection in autonomous vehicles. The authors presented a cloud-assisted vehicle malware protection framework.

Olga Berlin *et al*. [54] introduced a Security Information and Event Management System (SIEM) called Security Management of Services in Connected Cars (SeMaCoCa). The proposed system uses data from vehicles (*e.g.*, odometer values) as well as additional information from other sources (*e.g.*, data from third parties and service garages) to recognise attacks. The SeMaCoCa uses a combination of rule-based, ML, deep learning, real-time-based, security and big data

algorithms. Although using data from different sources might improve the detection capability of the proposed system, the proposed system wasn't numerically evaluated. As it utilises data from different sources, which are expected to be large in size, scalability of the proposed system should be studied as the system might not scale-up to large-scale heterogeneous data.

Vasistha [55] proposed three techniques to detect anomalies on the CAN bus. Vasistha proposed a cross-correlation technique to validate values of multiple sensors and detect anomalies based on the observation that sensors values are highly correlated in normal behavior. For example, the speed of the front right wheel is highly correlated with the speed of the front left wheel. However, such technique requires knowing the semantics of the protocol to decode the values of the sensors. Vasistha also proposed a timing-based detector to detect changes in the timing behavior of messages using deterministic and statistical approaches. The author concluded that attacks can be detected by comparing delay distribution of messages with normal delay distribution behavior. A deviation from the normal behavior is considered as an attack. This method can detect attacks without the need to know the semantic of the messages.

Moreover, Vasistha proposed a detection technique that observes the order of messages from a single ECU to detect attacks. This is based on the observation that messages from an ECU have a specific order that always seen in. Any deviation from the order can be flagged as an attack. Experiments on Honda Civic, Toyota Camry, and KIA showed that the proposed techniques can detect several types of attacks (such as MI, deletion, and DoS) with a detection latency of 2 sec and FPR between 0% and 3.5%.

Weber *et al*. [56] proposed a hybrid IDS that uses a specification-based system and a ML-based system sequentially. At the first stage, the specification-based system is applied because it does not generate false positives and has high efficiency. The specification-based system examines six specification-based sensors classes classified by Müter *et al*. [52], including formality, location, range, frequency, correlation, and protocol sensor. The authors did not realise the formality and protocol sensors at this stage as the information required to implement such type of protocol checks is mostly available in an OEM-specific and not standardized specifications. In the second stage, ML algorithms, Replicator Neural Network, OCSVM and Lightweight Online Detector for Anomalies (LODA), are trained on features extracted from data for advanced contextual and collective anomalies that cannot be detected by the specification-based system such as an unnatural time series of a communication signal (*i.e.*, plausibility sensor). The decisions of the ML algorithms are fused to decide if there is an intrusion or not. In this work, the authors proposed LODA as a potential classifier for detecting anomalies in CAN.

Zhang *et al*. [57] followed a similar approach to the one proposed by Weber *et al*. [56]. Zhang *et al* proposed a two-stage IDS based on rule-based and deep-learning-based

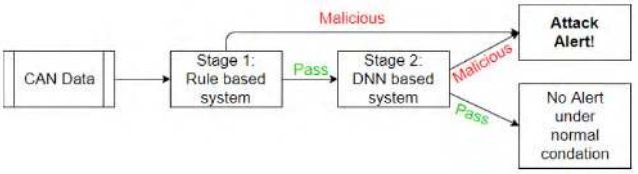

FIGURE 4. The Two-stage IDS proposed in [57].

systems to detect attacks in real time. The first stage is rule-based detection whereas the second stage is a DNN-based system. The rule-based system is used to offset the computational requirements of the deep-learning system. As in Figure 4, the lightweight rule-based system quickly detects attacks that violate the periodicity and regularity of major CAN traffic while the DNN-based system catches missed attacks from the rule-based system. Performance evaluation of the proposed system on real traces from three vehicles, namely Honda Accord, Asia brand vehicle and US brand vehicle, showed that the proposed system can detect five types of attacks (MI (random ID or Zeros ID), spoofing, replay, and deletion or drop attacks) with a DR between 99.91% and 99.97%, FPR between 0.18% and 0.090%, and TsT between 0.53 *m*s and 0.61 *m*s per message.

B. FEATURES AND FEATURE SELECTION

Intrusion detection depends mainly on observing data exchanged among connected entities/nodes (*e.g.*, ECUs). A dataset comprises a set of records/instances, which represent events, objects, or processes. Each instance is characterized by a set of features which describe/measure properties of the instance. For example, a message can be considered as a record and its time-stamp could be considered as a feature. These features are used to build a detection model that can distinguish between normal and intrusive behaviors or messages. In this paper, we distinguish between two types of features: physical and cyber features. Physical features refer to the features that describe the physical state of the system (*e.g.*, speed, engine RPM) whereas cyber features refer to the features that describe the communication and data aspects of the system (*e.g.*, number of messages, data sequences).

As can be seen in Tables 5–7, most of the reviewed works in this paper (*e.g.*, [19], [23], [31], [33], [38], [42], [56]) used cyber features to characterize and detect intrusions. Out of 42 works reviewed in this paper, only two works, [32] and [28], have used physical features to detect attacks. Four works [5], [18], [35], [51] have used a combination of cyber and physical features to detect intrusions, and two works did not provide description of the used features. Figure 5 provides statistics of the reported features used in the reviewed works.

In theory, having more features should result in a more discriminating power [58]. However, practically, adding irrelevant or redundant features to a dataset often negatively impacts the discrimination capability of a learning algorithm that learns from the data. Thus, selecting right features is of critical importance since it could not only reduce the

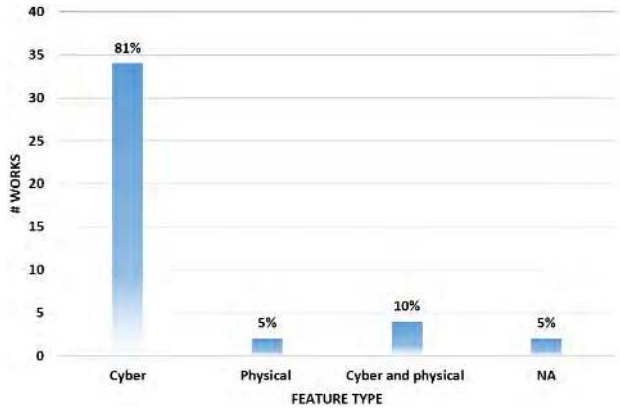

FIGURE 5. Statistics of Feature Sets.

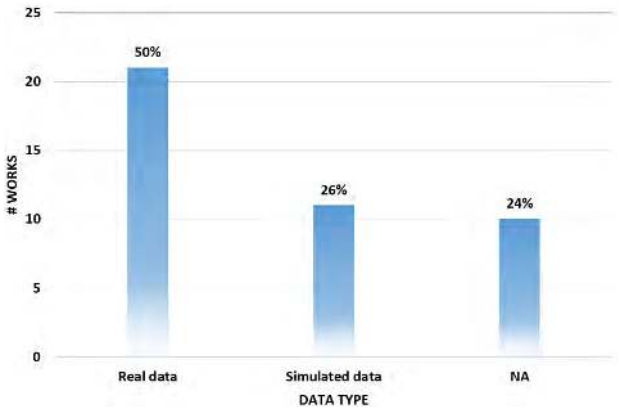

FIGURE 6. Statistics of Data Types.

computational cost but also improve the generalization capabilities of the learning system and give a better understanding of the data.

Feature selection can be done manually, based on understanding the problem and the importance of each feature in discriminating between different classes in a classification problem, such as intrusion detection. However, sometimes it is difficult to have a deep understanding of the data and how it can be used to solve a problem, especially when dealing with high-dimension datasets (*e.g.*, gene microarray [59]). In such cases, automated methods are useful. Feature selection methods can be broadly categorized into two main categories, filter methods and wrapper methods.

Filter methods depend on the characteristics of the data to select features independently of the learning algorithm. Because of their simplicity and success in practical applications, ranking methods (*e.g.*, Pearson correlation and mutual information) are used as the principle criteria for variable selection in filter methods. Each feature is given a score based on a ranking method and a threshold is used to remove the features with scores below the threshold [59]. However, the selected feature set by a filter method might not be optimal as a redundant subset might be obtained. In addition, features that are less informative in their own but more informative when combined with other features could be discarded [59].

Wrapper methods optimize a learning algorithm as a part of the feature subsets evaluation and selection processes [60]. As the number of features grows, finding an optimal feature set becomes infeasible using a wrapper method since it requires evaluating the performance of the learning algorithm in each round. Thus, search algorithms (*e.g.*, sequential feature selection and Genetic algorithm) are used to find a subset of features that maximizes the performance of the learning algorithm and is computationally feasible.

Out of all works reviewed in this paper, three works [2], [28], and [57] have employed feature selection methods, namely Joint Mutual Information and Sequential Forward Selection (SFS). The SFS is a bottom-up search procedure that starts with empty feature set and sequentially add features to the set. The SFS select features based an evaluation function that minimizes the misclassification error rate. A feature is selected if adding this feature to the feature set produces a minimum classification error compared with adding other feature.

C. DATASETS

Thorough evaluation of an IDS is of critical importance as many approaches fail to meet what is expected from them in real-world scenario [61]. This requires an appropriate dataset that represents the real-world scenarios [62]. We categorize the datasets used in the works reviewed in this paper into: real data that are extracted from test vehicles, simulated data refer to the data generated by simulation or prototyping, and data that are not comprehensively described in the source work. As can be seen in Figure 6, 21 works out of 42 works reviewed in this, such as [18], [31], [34], [38], and [50], have used real data, whereas 11 works, such as [5], [6], and [42], have used simulated data, and 10 works did not provide comprehensive description of the used data.

D. ATTACK TYPES

Autonomous vehicles are susceptible to several cyberattacks with different severity levels ranging from eavesdropping to threatening road users' safety, and even paralyzing the whole transportation system [16]. Generally speaking, cyberattacks can be classified into two categories: passive and active attacks. Passive cyberattacks (*e.g.*, eavesdropping) mainly breach the confidentiality requirement of the target system's security and cause privacy leakage (*e.g.*, accessing private data, such as position information, conversation data, and camera records) [63], [64]. Active cyberattacks can obstruct systems' functionality by insertion, deletion, or modification of messages. For instance, it is possible to control primary functions and components (*e.g.*, disabling the braking system and engine) of an autonomous vehicle by compromising its internal network [4]. In what follows, based on our review of existing works, we discuss most common cyberattacks against intra-vehicle networks.

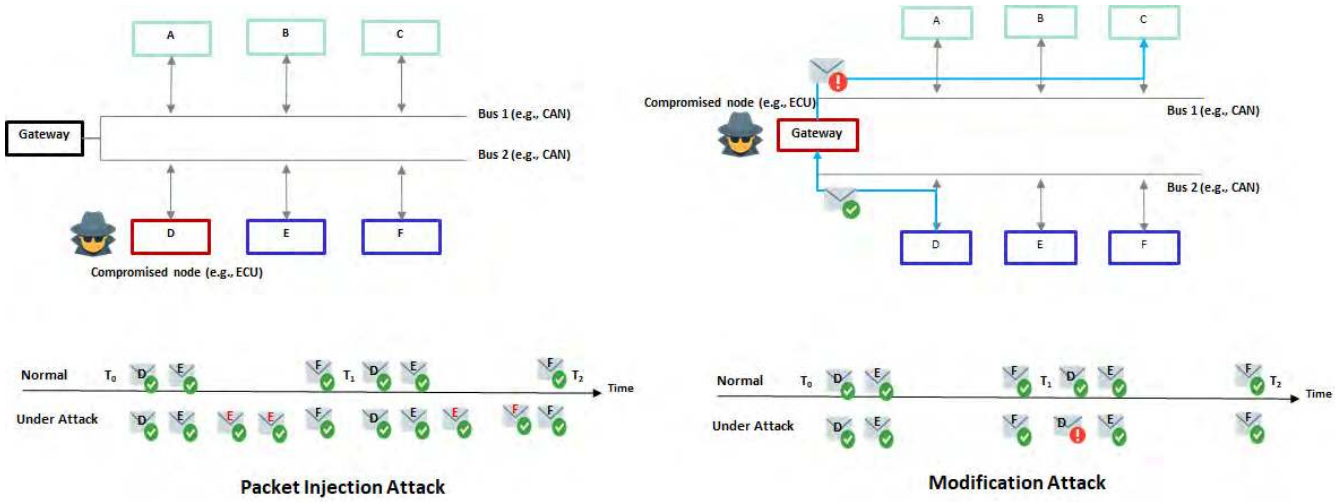

FIGURE 7. Typical Examples of Cyberattacks Against Intra-vehicle Network.

1) DENIAL OF SERVICE (DOS) ATTACK

A DoS attack aims to diminish or disturb the expected functionality of the system. For example, an attacker may send many legitimate requests beyond the ability of a target system to handle them, draining the resources of the system and paralyzing its functions. In vehicular ad-hoc networks for instance, the attacker might shutdown the communication network established by an RSU by sending many request messages, resulting in overloading the RSU and inability of data sharing from and to it. This may prevent sharing of safety related messages such as road status and warning messages, leading to fatal consequences. JellyFish attack [65], intelligent cheater attack [66], flooding attack, and jamming attack [67] are well-known types of DoS attack observed in vehicular ad-hoc networks. In CAN, an attacker might exploit the arbitration mechanism of CAN protocol to hinder the transmission of legitimate messages by sending high priority messages, forcing other ECUs to stop their transmission [8]. It is worth mentioning that an attacker with limited knowledge can launch DoS, thus the likelihood and impact of such attack is considered very high [14].

2) MESSAGE INJECTION (MI) AND REPLAY ATTACK

In MI attack, the attacker sends a valid message over the network. As in Figure 7, the attacker gains an access to ECU (D) and controls it. The attacker then injects fabricated messages to the CAN bus as shown in the timeline. In a replay attack, an attacker stores a valid message at a certain time and uses it at later stages [68]. For example, the attacker can store the speedometer reading and broadcast it again to the network later on.

3) MESSAGE MANIPULATION

This attack impacts the integrity of the data by altering/modifying or deleting messages. For instance, an attacker can modify the content of a message. As in Figure 7, the attacker manages to compromise the Gateway, which connects two CAN buses, and intentionally modifies a message originated from ECU (D) and intended to ECU (C). The attacker might modify the content of the message without affecting its timing. Note that field modification and deletion attacks are subtypes of message manipulation attack. In a deletion attack, the attacker deletes messages in the output buffer of the compromised ECU before transmitting them on the CAN bus.

4) MASQUERADE ATTACK

To mount a masquerade attack, also known as impersonation attack, the attacker needs to compromise two ECUs (A and B). The attacker monitors the CAN bus to learn which messages are sent by A? and at what frequency?. Once the attacker learned the IDs and the frequency of the messages sent by A, the attacker stops the transmissions of A and exploits B to fabricate and inject messages on behalf of A [25].

5) MALWARE ATTACK

Malware may exist in various forms such as viruses, worms, and spyware, which can be injected into the system by exploiting vulnerabilities of the communication interfaces. For example, an attacker can exploit the input vulnerabilities of the media player firmware of a vehicle to add a malware to music files. The malware runs and sends malicious messages into the CAN of the vehicle when the infotainment system plays these music files [69].

As can be seen in Figure 8, 69% of the reviewed works considered MI and replay attacks whereas 26%, 17%, 10%, and 7% of the works focused on detecting message manipulation, DoS, malware, and masquerade attack, respectively. About 29% of the reviewed works focused on other types of attacks, such as reversing of messages order, or did not consider specific types of attack.

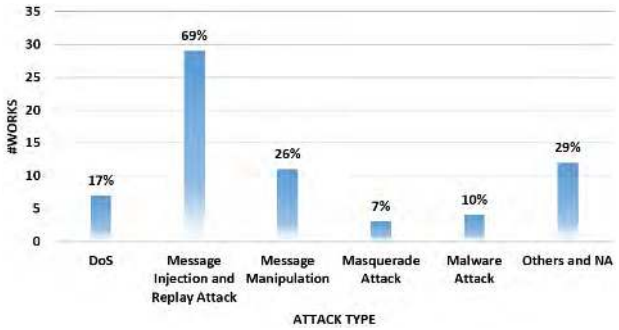

FIGURE 8. Statistics of Attack Types.

TABLE 3. Confusion matrix of a classifier on a two-class problem.

Actual	Predicted Attack	Predicted Normal
Attack	TP	FN
Normal	FP	TN

E. EVALUATION METRICS

Several performance metrics are used to evaluate the performance of proposed methods. In the following, we discuss most commonly used performance metrics to evaluate the performance of IDSs in intra-vehicle networks.

1) Confusion Matrix (CM): A CM is a table that represents the performance of a classifier. Table 3 represents the CM of a binary classifier in a two-class classification problem. True Positive (TP) is the number of correctly classified intrusions, True Negative (TN) is the number of correctly classified normal records, False Negative (FN) is the number of incorrectly classified intrusions as normal traffic, and False Positive (FP) is the number of incorrectly classified normal traffic as intrusions. A good classifier has high TP and TN, and low FP and FN
2) Detection Accuracy (Acc): Acc reveals a classifier's ability to correctly classify normal and intrusive traffic. Acc is given by [62] :

$$Acc = \frac{TP + TN}{TP + TN + FP + FN}. \quad (1)$$

However, Acc, as a single evaluation metric, might be misleading when dealing with imbalanced/skewed data. For example, a classifier that always predicts the class label of a test record as negative achieves an Acc of 95% on a dataset of 95 negative records and 5 positive records. As such, having a high Acc does not necessarily mean that the classifier performs well in detecting all class types.
3) Detection Rate (DR): DR, which also known as recall or True Positive Rate (TPR), is the number of intrusions/attacks detected by the model divided by the total number of attacks in the test set. A DR value of 1 means that the detection model correctly detects all intrusions whereas a DR value of 0 means the detection model fails in detecting all intrusions. DR is given by [62]:

$$DR = \frac{TP}{TP + FN}. \quad (2)$$

4) False Positive Rate (FPR): FPR refers to the percentage of normal traffic classified as intrusion. FPR is given by:

$$FPR = \frac{FP}{TN + FP}. \quad (3)$$

5) False Negative Rate (FNR): FNR refers to the percentage of attack traffic classified as normal. FNR is given by:

$$FNR = \frac{FN}{TP + FN}. \quad (4)$$

6) F-measure: The F-measure of a classifier is the harmonic average of the precision and recall (*i.e.*, DR or TPR) of the classifier. The precision is the number of correct positive results divided by all positive results returned by the classifier. Precision and recall are given by:

$$Precision = \frac{TP}{TP + FP}, \quad (5)$$

$$Recall = 1 - FNR. \quad (6)$$

The value of the F-measure is between 0 and 1, where value of 1 represents a good classifier and value of 0 represents a bad classifier. The F-measure is given by:

$$F - measure = \frac{2 \times TP}{2 \times TP + FP + FN}. \quad (7)$$

Although the F-measure is advocated as a single metric to capture the effectiveness of a classifier, it still ignores the TN which can vary freely without affecting the statistics [70].
7) Receiver Operating Characteristic (ROC) curve and AUC: A ROC curve is a visualization of the performance of a classifier. A ROC curve can be drawn by plotting the FPR vs TPR of a classifier, as in Figure 9. A perfect classifier will score in the top left corner of a ROC curve. A random classifier would be expected to score along the diagonal line of the ROC curve. The AUC quantifies the performance of a classifier where a value of 1 represents a perfect classifier and a value near 0.5 indicates random detection. ROC curve and AUC are suitable metric for model selection and comparison especially when dealing with skewed data.

In addition to the above metrics, some works use other metrics to evaluate the performance of proposed methods. For example, [19] and [31] used throughput to evaluate the performance of the proposed models. TrT and TsT have been used in [42] to evaluate the performance of the proposed method. The TrT is the time required to build the detection model whereas TsT represents the time required to classify new messages. The TsT implies the throughput of the system where a low TsT means a high throughput. Throughput

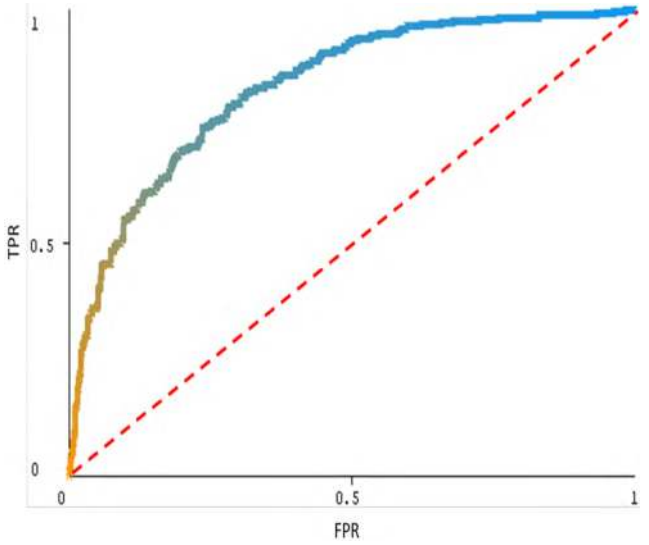

FIGURE 9. Typical ROC graph.

and power consumptions were used in [19] to evaluate the performance of the proposed model. Monther Aldwairi and Jarrah [17] used the execution time, which is the time from reading the pattern until the end of matching process of a knowledge-based IDS, speedup, and memory usage to evaluate the performance of the proposed model.

F. BENCHMARK MODELS

It can be observed from Tables 5–7 that minority of existing works provide a comparison of the proposed methods with the existing ones. However, some ML-based detection models have been used for comparison purposes, including Decision Tree (DT), ANNs and deep learning, SVM and OCSVM, and Random Forest (RF).

- A DT is a graphical representation of possible decisions through a sequence of certain conditions or tests. It consists of nodes, edges, and branches. Nodes are arranged in a tree hierarchy and represent a test on an attribute/feature, based on which the data are partitioned, each branch represents the outcome of the test and each leaf node represents a class label or decision. Each node has a number of edges, which are labeled according to the possible value of the test and connect between nodes. A root node represents the topmost node and has no incoming edges [62]. Decision trees take labeled training data, which might contain numerical or categorical values, as an input and construct decision models that can be used to predict the class type of a new instance or message [71]. The classification process of a test instance starts from the root node to the appropriate leaf node, and the path from the root node to the leaf node represents the classification rule [71].
- ANNs are data processing units inspired by human brain neurons. An ANN consists of multiple layers of simple processing units called nodes or neurons. Each node or neuron is connected with other nodes by a weighted connection that specifies cross-nodes effects. Such networks are able to compute values by feeding data from the input layer till the output layer. ANNs learn through adjusting the weights of the connections between the processing nodes to achieve the desired results [72]. Feed-forward and back-propagation learning are well-known learning methods of ANNs; however, back-propagation performs superbly at classification and prediction tasks in the literature. In addition to its ability to learn and generalize from noisy and incomplete data [73], ANN can be used to map nonlinear statistical relationship of high-dimensional data to two-dimensional, thus, it can extract relationships among complex datasets [74]. The performance of ANN is affected by the number of neurons in use. The higher the number of neurons, the better the performance. However, increasing the number of neurons results in a dramatic increase in the computational cost [75]. Recently, ANNs and deep learning approaches, which incorporate several layers, have proven distinguished performance in several applications including IDS for intra-vehicle networks. Generally speaking, deep learning approaches require large-scale data to learn from.
- Originally, SVM deals with two-class problems in which data are separated into two classes; however, it can be extended to multiple-class applications. SVM builds hyper-plans that separate different classes. SVM tries to set a linear boundary between different classes in such a way that the margin area is maximized. Datapoints/instances that define the margins are called support vectors and are used to define classifiers independently of the input features. This property gives SVM the advantage that it can generalize well [76]. SVM uses kernel functions to transform nonlinear SVM into linear SVM by mapping the original feature or attribute space into higher dimensional space in which data are linearly separable. Hence, SVM can find linear, nonlinear, and complex decision boundaries accurately [77]. However, parameter selection of SVM is not easy and still follows a trial-and-error approach. Moreover, the running time of SVM quadruples when the size of the training data is doubled, which demotivates its use when dealing with large-scale data [77]. Several works proposed OCSVM-based IDSs, such as [2] and [22]. OCSVM learns decision boundaries of one-class training dataset, typically normal or attack-free data, and detects anomalies based on the learned decision boundaries.
- A RF is an ensemble classifier that consists of a collection of base classifiers typically random decision trees. In order to reduce the variance error, base classifiers are required to differ in the error that they commit on testing instances. This is provided by building the base classifiers on non-identical training datasets. Typically, bootstrap resampling is used to generate training data for each tree in the ensemble model, and also a randomly selected subset of input features is used to find the best

TABLE 4. General Summary of Intra-vehicle Intrusion Detection Techniques.

Technique	Advantages	Disadvantages
Rule-based IDSs	Fast, when small rule set is used, and simple	Relatively low generalisation capabilities and prior-knowledge dependent
Time and Frequency Analysis IDSs	Suitable for attacks that affect packets' frequency and timing (*e.g.*, MI) and average complexity	Completely blinds to attacks that affect the data field of packets (*e.g.*, modification attack) and prior-knowledge dependent
CI & IT based IDSs	High generalisation capabilities and prior-knowledge free	High complexity and data dependent

Computational Intelligence & Information Theory (CI & IT)

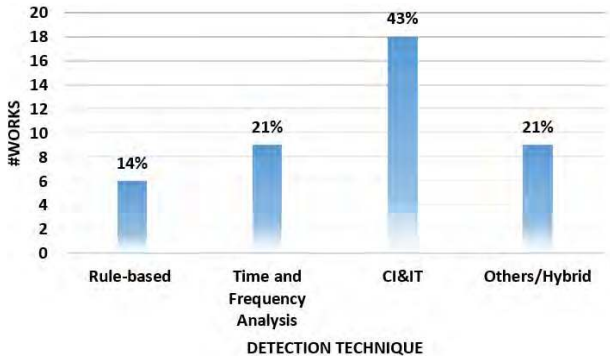

FIGURE 10. Statistics of Detection Technique.

splitter at each node of the DTs [78]. To classify a new input instance, feed the input instance down each of the DTs in the ensemble model and the final decision is the majority vote among all individual decisions in the ensemble [60].

IV. RESEARCH GAPS AND CHALLENGES

In this section, we discuss research challenges and identify research gaps, supported by the trends observed in Tables 5–7.

A. DETECTION TECHNIQUES AND DESIGN

Figure 10 shows that 14%, 21%, and 43% of the reviewed works utilized rule-based, time and frequency analysis, and CI&IT techniques, respectively, to detect intrusions in intra-vehicle networks, and the remaining works adopted hybrid approaches of the former techniques.

Rule-based techniques need prior knowledge such as underlying distribution of data, the known models regarding a system, and empirical assumptions [1]. Since they perform rule matching to detect intrusions, rule-based IDSs are fast and computationally cheap when small rule set is used. However, the matching process might become a burden in terms of memory and computation when a large number of rules is used. In addition, such techniques may fail short of what expected from them when applied to application scenarios that do not fit their prior knowledge. Time and frequency analysis based IDSs also require some prior knowledge that is normally learned by observing data flow under normal vehicle operation. On the other hand, computational intelligence based techniques don't need any prior knowledge as they set no assumptions on the underlying distribution of the data. This type of IDS learns the normal network profile from multivariate training data using a learning procedure. Among other detection techniques, this technique has the strongest capability for generalization. However, for some such systems it is hard to understand what happens inside the system. For example, it is difficult to explain what is happening inside a large ANN and to understand the meaning of the values of the weights. Additionally, most of them are resource intensive in terms of computation and memory [56].

Having considered the shortcomings and merits of existing techniques, it is possible to hypothesize that no single technique can satisfy all the requirements of intra-vehicle IDSs, such as high generality and throughput, and low false positive rate. Accordingly, the intra-vehicle IDSs should be designed while bearing in mind the requirements of modern vehicles. For instance, safety related applications are time stringent applications that require packet delivery in less than 100 *m*s. In this case, an intra-vehicle IDS should have a high throughput in order to fill this requirement. As such, rule-based techniques is a potential candidate. However, their limited generality demotivate their use as novel cyberattacks might emerge with time, for example bus-off attack which wasn't known until recently [2]. The primary question here which detection technique turn out to be the suitable technique for detecting intra-vehicle intrusions. In addition, the placement of the IDS is an important issue that fairly discussed in the literature. For example, placing the IDS at a gateway will allow observing data from different networks, based on which it is possible to find correlation among data pieces, leading to identify intrusions. This will not be possible if the IDS is placed on an ECU within a network with no access to data in other networks. Thus, the design of intra-vehicle networks should consider the selection of suitable detection technique as well as the placement of the IDS.

B. FEATURES AND FEATURE SELECTION

As can be seen in Figure 5, most of reviewed works detect intrusions by examining cyber features only. However, it has been shown that incorporating physical features in the detection process can result in improved performance, see [5], [18], [35], and [51]. Conceptually, a cyberattack might, and indeed is likely to, compromise some data sources of the vehicle while leaving others unaffected. As such, data fusion of multiple sources is required to effectively detect attacks and malfunctioning sensors. However, most of works reviewed in this paper do not analyze multiple data sources, especially

TABLE 5. Summary of Flow-based Approaches.

Work	Technique	Features/Feature Selection	Dataset	Attack Type	Performance Metrics	Benchmark Models
[18]	Rule-based	Cyber&physical/NA	Real data of 52,215 records of a small robot vehicle	DoS, MI and two types of malware	Acc (66.7%–85.24%), FPR (5.43%–29.60%), FNR (5.74%–41.44%), ROC, AUC (0.73–0.97), and detection latency (around 1 sec)	NA
[19]	Rule-based	Cyber/NA	NA	NA	Throughput (39Gbps), power consumption (7.5 w), and latency (4 μs)	Snort software
[20]	Time and Frequency Analysis	Cyber/NA	Simulation	MI	NA	NA
[21]	Time and Frequency Analysis	Cyber/NA	Simulation	NA	NA	NA
[22]	Time and Frequency Analysis	Cyber/NA	Real data of a Ford Explorer 2011	MI and deletion	ROC and AUC (0.8720–1.0000)	OCSVM
[23]	Time and Frequency Analysis	Cyber/NA	Real data of an anomanyised vehicle	MI	Acc (36%–100%)	NA
[24]	Time and Frequency Analysis	Cyber/NA	NA	Replay and MI	NA	NA
[25]	Time and Frequency Analysis	Cyber/NA	Real data of approximately 2.25 million messages	MI, deletion and masquerade	ROC, FAR (0.055%), TPR (100%)	NA
[26]	Time and Frequency Analysis	Cyber/NA	NA	MI and Invalid messages	NA	NA
[27]	Time and Frequency Analysis	Cyber/NA	Real data captured from a Kia Soul	DoS, Fuzzy, and masquerade	NA	NA
[28]	Time and Frequency Analysis	Physical properties (a feature vector consisting of 11 time and frequency domain statistical signal attributes)/Joint Mutual Information Criterion	A dataset collected from 16 different channels and four identical ECUs transmitting the same message	MI	CM, DR (95.2% and 98.3% for channel and ECU classification, respectively)	NA
[29]	CI&IT	Cyber/NA	Real data	MI, DoS (flooding), Plausibility of Interrelated Events	NA	NA
[30]	CI&IT	Cyber/NA	Real data from 2011 Ford Fiesta (48 million CAN messages (about 3.3k messages per second) having 45 distinct IDs)	MI and Fuzzy	NA	NA
[6]	CI&IT	Cyber/NA	Simulated data	Attack scenarios (Out of order scenarios, USB firmware update attack, communication with unknown vendor, OTA malicious updates, and malicious application installation)	AUC (0.81–0.96), F-Measure (0.61–0.93)	NA
[31]	CI&IT	Cyber/NA	A log-file containing 1,014,070 records of real-time traffic of a Renault Zoe 2016	MI	Throughput and complexity (>2000 events/s)	NA
[32]	CI&IT	Physical/NA	Real data of a 2015 passenger vehicle	Chip tuning (manipulation)	NA	NA
[2]	CI&IT	20 features (time and frequency)/SFS	A prototype setup of 12 ECUs where 70 sample per ECU. This gives a total of 840 samples for training and 100 per ECU for testing as well as data from Hyundai Sonata 2010 and Kia Soul 2014	Masquerade and Bus-off	F-score (54.24%–99.61%), precision (92%–99%), recall (92%–99%), FPR (0%)	NA
[33]	Others/Hybrid	Cyber/NA	NA	Masquerade, DoS, and replay	NA	NA
[34]	Others/Hybrid	Cyber/NA	Real data of 120 m CAN messages gathered from an unmodified licensed vehicle	Replay and MI	DR (20%–100%)	NA

Computational Intelligence and Information Theory (CI&IT), Accuracy (Acc), False Positive Rate (FPR), False Alarm Rate (FAR), False Negative Rate (FNR), True Positive Rate (TPR), Detection Rate (DR), Confusion Matrix (CM), Message Injection (MI), Training Time (TrT), Testing Time (TsT), Receiver Operating Characteristic (ROC), Area Under Curve (AUC), One Class Support Vector Machine (OCSVM).

across the cyber and physical features, in detecting attacks. Unlike IDS research for computer networks, there is no specific well-defined feature set used in intra-vehicle IDS research. In addition, as mentioned in the previous section, having more features is likely to improve the detection performance of a learning algorithm. However, it should be recognized that noisy features may negatively impact the performance of the learning algorithm as well as increasing computation time. Therefore, selecting the right features can improve the detection performance and reduce the required computation and resources. Therefore, a potential research direction is defining and selecting discriminating features for intrusion detection applications in intra-vehicle networks.

C. ATTACK TYPES AND REACTIVE MEASURES

Figure 8 shows that most existing works propose algorithms designed to target specific types of attack (*i.e.*, MI). However, novel attacks are likely to emerge over the life-time of the vehicle. For example bus-off attacks were not known until recently [2]. A potential research area is to investigate and study protocols vulnerabilities in order to define potential unknown attacks. Once all possible attacks are defined, an intrusion detection model can be designed to fill

TABLE 6. Summary of Payload-based Approaches.

Work	Technique	Features/Feature Selection	Dataset	Attack Type	Performance Metrics	Benchmark Models
[35]	Rule-based	Cyber and physical/NA	NA	Replay, MI, rogue node, compass manipulation, and broken wheel	ROC and AUC (0.406–1)	NA
[36]	Rule-based	Cyber/NA	Simulation	Replay and Invalid messages (MI)	NA	NA
[37]	Rule-based	Cyber/NA	Data (20 recordings, each with 100,000 messages and approximately 43 seconds long) from2012 Ford Focus for training and synthetic CAN bus traffic simulating 10 different ECUs (500 messages per ECU)	MI	Field Classification Distance (10%–40.8%), Field Classification FPR (0.1%–2.2%) and Model size in TCAMs ($<$1642)	NA
[38]	Rule-based	Cyber/NA	Real data of different traffic traces gathered from a test vehicle (2011 Ford Fiesta)	MI	DR (20%–100%)	NA
[39]	CI&IT	Cyber/NA	Real data from a real vehicle	Faults	TrT (20843s–63631s), TsT (4845 sec–24145 sec), FN (0.0–10.5/hour), TN (9–45/hour), TNR (42.9%–76.9%), Precision (32.3%–100%)	NA
[41]	CI&IT	Cyber/NA	Real data from vehicles from different manufacturers which include âĂIJHonda AccordâĂİ, âĂIJToyota CorollaâĂİ and âĂIJChevrolet CruzeâĂİ	MI	NA	NA
[42]	CI&IT	Cyber/NA	Simulated data of 60,000 records	Field modification and MI	CM, TPR (97.6%–99.8%), TNR (93.7%–99.9%), TrT (4.741 sec–11.977 sec) and TsT (7.957 ms–8.120 ms/3900 packets)	ANN
[43]	CI&IT	Cyber/NA	Simulated data of 200,000 records	MI and manipulation attacks	CM, ROC, DR (99%), FPR (1.6%), Acc (97.8%), TrT (4.15 sec –10.81 sec), and TsT (2.05 ms–3.78 ms)	SVM and ANN
[44]	CI&IT	Cyber/NA	45 million packets from a 2012 Subaru Impreza	Interleave, drop, discontinuity, unusual and reverse	ROC, AUC (0.176471–1), TPR (0%–100%) with FPR of 0%, FPR (0.0010%–0.6341%) with TPR of 100%	NA
[45]	CI&IT	Cyber/NA	Real data from a Renault Twingo I (model year 2002)	Faults injection	TPR (80%–100%), precision (35.4%–100%), F2-score (68.5-83.3%), and diversity (0.093–0.223)	NA
[46]	CI&IT	Cyber/NA	Integrated Vehicle-Based Safety Systems (IVBSS) dataset collected by the University of Michigan Transportation Research Institute. IVBSS contains naturalistic driving behaviour of 108 drivers for 16 cars between April 2009 âĂŞ May 2010 of over 213,000 miles of driving	MI	NA	NA
[47]	CI&IT	Cyber/NA	Real-world data	DoS, Fuzzy, and MI	FPR (0–0.038), Precision (0.963–1), Recall (0.823–1), F-Measure (0.981–1) and AUC (0.986–1)	NA
[48]	CI&IT	Cyber/NA	NA	NA	NA	NA
[49]	CI&IT	Cyber/NA	NA	NA	NA	NA
[50]	CI&IT	Cyber/NA	20 hours of real-time data from high CAN bus of Impreza	Field modification and replay attack	AUC (0.85–1), recall (0.2–0.8), and precision (0.95–1)	RNN and HMM
[5]	CI&IT	Cyber and physical/NA	Simulated data of 3,114 records of DoS, 3,432 records of command injection, and 2,390 records of malware	DoS, MI, and malware	Acc (86.9%) and TsT (1.163–1.704)	Logistic regression, decision tree, SVM, random forest, and deep learning (MLP)
[51]	Others/Hybrid	17 Cyber and physical/NA	Data of a robot vehicle	MI, rogue node, and magnetic interference attacks	ROC and AUC (0.995)	NA

Computational Intelligence and Information Theory (CI&IT), Accuracy (Acc), False Positive Rate (FPR), False Alarm Rate (FAR), False Negative Rate (FNR), True Positive Rate (TPR), Detection Rate (DR), Confusion Matrix (CM), Message Injection (MI), Training Time (TrT), Testing Time (TsT), Receiver Operating Characteristic (ROC), Area Under Curve (AUC), Multi-Layer Perceptron (MLP), Artificial Neural Network (ANN), Hidden Markov Model (HMM), Support Vector Machine (SVM), Recurrent Neural Network (RNN).

the detection and performance requirements of automotive networks.

Once an attack is perceived to have been detected, the system must decide what action to take. Traditionally in computer networks, the IDS raises an alert to the user reporting the incident and asking for user input on what action to take. However, such an action could cause a distraction if the alert is made to driver and could lead to traffic accident. Moreover, the driver is unlikely to have the technical knowledge to advise on the most appropriate action and, in any case, likely

TABLE 7. Summary of Others/Hybrid Approaches.

Work	Technique	Features/Feature Selection	Dataset	Attack Type	Performance Metrics	Benchmark Models
[52]	Detection sensors	NA	NA	NA	NA	NA
[53]	A Cloud-assisted vehicle malware protection framework	Cyber/NA	NA	Malware	NA	NA
[54]	A back-end SIEM	NA	NA	NA	NA	NA
[55]	Cross-correlation-based detector, timing-based detector, and messages order detector	Cyber/NA	Real data of Honda Civic, Toyota Camry, and KIA	MI, deletion, and DoS	Detection latency (2 sec) and FPR (0%–3.5%)	NA
[56]	A Hybrid approach of specification-based and ML-based system	Cyber/NA	Synthetic CAN signal	Limitation of value range, value freeze, alternative signal sequences, peak signal and signal jump	NA	NA
[57]	A Hybrid approach of rule-based and ML-based system	Cyber/forward feature selection	Real data of three vehicles	MI (random ID or Zero ID), spoofing, replay, and deletion attacks	DR (99.91%–99.97%), FPR (0.18%–0.090%), and TsT (0.53 ms–0.61 ms/message)	NA

False Positive Rate (FPR), Detection Rate (DR), Message Injection (MI), Testing Time (TsT).

to require significant time to make a decision, thereby failing to prevent the attack being realized successfully [52]. Studying and designing appropriate alerting methods in the case of vehicular IDSs is a further research direction that requires investigation.

D. BENCHMARKING

1) DATASETS

Evaluation benchmark datasets are essential to evaluate and compare the performance of IDSs. For many years, the publicly available DARPA/Lincoln packet traces [79], [80] and KDD Cup [81] datasets have been used as benchmark datasets to evaluate the performance of IDSs in conventional computer networks. Although they have been criticized due to their lack of recent and modern normal traffic and attack styles, these datasets have served as de-facto for evaluating IDSs in conventional computer networks. Having benchmark datasets enables researchers to replicate experiments and verify results. Although most of the work reviewed in this paper use simulated or real-time data from test vehicles, there is a shortage of public benchmark dataset comprising real data for intra-vehicle IDS research. As such a potential research direction is generating benchmark datasets for IDS in intra-vehicle networks.

2) PERFORMANCE METRICS

Most of works reviewed in this paper use Acc, DR, and FPR to evaluate the performance of a proposed IDS. However, detection latency, which is sometimes referred to as throughput, and training time are critical metrics that researchers seldom report. While it is important to detect attacks with a high accuracy, detection latency is also important. For example, while it is desirable to have an accuracy of 99%, if detecting an attack takes an hour, the attacker may still be able to cause significant harm to the system. It is also desirable that the IDS be efficient and not be a bottleneck of the network. It is critical to have a high throughput so that the IDS does not inhibit data communication. This is of a paramount importance as some applications are time stringent applications (*e.g.*, safety applications which require delays of less than 100 ms). Researchers should, therefore, consider time complexity as a key metric to evaluate the performance of an IDS or even develop a new evaluation metric that takes into account the complexity of the system.

3) BENCHMARK MODELS

It can be observed in Tables 5–7 that there is no standard benchmark detection model to compare the performance of proposed detection models with it. This might result in discrepancies in the results reported in different work as there is no baseline to which to refer to. For instance, in conventional computer networks IDS research, Snort is considered as a standard rule set to be compared to. SVM, DT, and ANN are well-known models that are commonly used to compare the performance of newly proposed ML-based IDSs. However, there is no standard benchmark detection models in intra-vehicle IDSs research.

E. OTHERS

1) CONTEXT-AWARE SYSTEMS

As discussed in Section III, existing works examine the validity of data against some rules (*e.g.*, data range), data sequences, features and content to detect intrusions. Different approaches, such as ML and expert knowledge, have been used to build the detection model and capture the correlation among data pieces. However, these approaches haven't considered the semantics of the data. Attackers who exploit the semantics of messages can remain undetected since the current value of a physical variable depends on many factors, including context and semantics [32]. For example, it is normal to have a value 120 Km/h for the speedometer of the vehicle. However, it is not normal to have the same speed with an rpm value of 1. Developing a context-aware IDS requires processing of data multi-sources in order to build a global

normal-behavior profile of the vehicle. Currently, this area is still under-investigated, with the exception being the work presented in [32].

2) ML ALGORITHMS

In recent years, ML algorithms (*e.g.*, deep learning) have attracted a tremendous attention from academia as well as industry as they have been seen to have distinguished performance in several fields, such as computer vision, autonomous driving, pattern recognition, natural language processing, as well as intrusion detection.

ML algorithms can be broadly categorized into supervised, unsupervised, and semi-supervised learning algorithms. Supervised learning algorithms, such as ANN and SVM, deal with data that have explicit outputs of given inputs. These construct a model based on the training data. Then, this model can be used to label new instances or records that have never previously been seen by the model. When the audit data do not include any sort of outputs, this is known as unsupervised learning and an example of this type of method is a clustering algorithms; unsupervised learning refers to finding structures and patterns in the data. Unlike supervised and unsupervised learning, semi-supervised learning algorithms, such as self-learning and co-training techniques, rely on audit data of labeled and unlabeled instances or records. The labeled instances are used to build a target function that is used to classify the unlabeled instances, in order to generate more labeled data, and then to build a model based on the new audit data.

Several works have used ML algorithms, such as [5], [18], [42], [43], and [50]. Most of these works use supervised ML algorithms to build their detection models. However, labeled data are not always available since it requires domain knowledge and has a high cost to produce. Although unsupervised and semi-supervise ML algorithms have been used to build intrusion detection models in conventional networks, there is a limited use of them in intra-vehicle IDSs. Furthermore, the use of online learning, which allows learning from streaming data, in intra-vehicle remains relatively unexplored.

V. SUMMARY AND CONCLUSIONS

Modern vehicles are vulnerable to a wide range of security threats that can be exploited by attackers to gain access to the vehicles and eventually control them. Although conventional security mechanisms can protect targeted systems from external attacks, they are not usually applicable to intra-vehicle networks. Given this, IDSs represent an essential component of vehicles security suit.

This paper provided an overview of intra-vehicle networks and examined contemporary research in intra-vehicle networks IDSs in an effort to identify research challenges and gaps. The research developments have been examined against six criteria, namely detection technique, features and feature selection methods, data, attack type, performance metrics, and benchmark models. A summary of works examined in this paper was presented in Tables 5–7. The reviewed works have been classified into two main categories: flow-based and payload-based IDSs. Flow-based IDSs are sensitive to timing and frequency change of messages. However, they fall short when confronted with attacks that only change the message content. On the other hand, payload-based IDSs can detect attacks that change the content of messages. However, such systems are blind to attacks that change message timing and frequency. The reviewed works were further sub-classified into seven subcategories based on their detection technique.

Based on observations and analysis from Tables 5–7, six potential research areas have been identified. Firstly, no single solution can provide a full detection capability of intra-vehicle attacks since each detection technique has its own shortcomings and merits, and thus the design process of IDS for intra-vehicle should consider the requirements of the system and have the ability to generalize to unknown attacks. Secondly, most existing works rely solely on cyber features to identify intrusions. However, as vehicles are cyber-physical systems, it is more appropriate to also examine the physical features of the vehicle during the detection process. While including more features is important, it is necessary to select the most discriminative features since this can significantly improve the performance of the detection model and reduce its complexity. Thirdly, there is a dearth of work trying to identify all possible attacks against intra-vehicle networks. In addition, our study has also shown that no work as yet has studied how the system should respond to intrusions. This is of critical importance as inappropriate actions can lead to safety issues. Fourthly, there is a lack of benchmark datasets for intra-vehicle IDS research. As well as the absences of datasets, performance measures, such as detection and training time, are rarely reported in existing work. There is also no benchmark detection model with which to compare the performance of different detection models. This makes the use of proposed model contentious. Collectively this demonstrates an urgent need to generate benchmark dataset, developing a single representative performance metric, and selecting a benchmark model for evaluating the performance of proposed detection models. This work has also shown that most of existing research examines cyber features to detect intrusions and fails to recognize the context of the data. Realization of the semantics of the data is likely to improve the performance of a detection model. Finally, ML-based IDSs seem to be promising candidates for detecting intrusion in intra-vehicle, due to their generalization capability. However, the use of unsupervised, semi-supervised and online learning techniques for intrusion detection in intra-vehicle networks have not adequately investigated and require further study.

REFERENCES

[1] S. Han, M. Xie, H. H. Chen, and Y. Ling, "Intrusion detection in cyber-physical systems: Techniques and challenges," *IEEE Syst. J.*, vol. 8, no. 4, pp. 1052–1062, Dec. 2014.

[2] W. Choi, K. Joo, H. J. Jo, M. C. Park, and D. H. Lee, "VoltageIDS: Low-level communication characteristics for automotive intrusion detection system," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 8, pp. 2114–2129, Aug. 2018.

[3] S. Checkoway *et al.*, "Comprehensive experimental analyses of automotive attack surfaces," in *Proc. USENIX Secur. Symp.*, San Francisco, CA, USA, 2011, pp. 77–92.

[4] K. Koscher *et al.*, "Experimental security analysis of a modern automobile," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2010, pp. 447–462.

[5] 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*, vol. 6, pp. 3491–3508, 2018.

[6] M. Levi, Y. Allouche, and A. Kontorovich. (2017). "Advanced analytics for connected cars cyber security." [Online]. Available: https://arxiv.org/abs/1711.01939

[7] 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.

[8] I. Studnia, V. Nicomette, E. Alata, Y. Deswarte, M. Kaâniche, and Y. Laarouchi, "Survey on security threats and protection mechanisms in embedded automotive networks," in *Proc. 43rd Annu. IEEE/IFIP Conf. Dependable Syst. Netw. Workshop (DSN-W)*, Jun. 2013, pp. 1–12.

[9] P. Mundhenk *et al.*, "Security in automotive networks: Lightweight authentication and authorization," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 22, no. 2, p. 25, Mar. 2017.

[10] A. Humayed, J. Lin, F. Li, and B. Luo, "Cyber-physical systems security—A survey," *IEEE Internet Things J.*, vol. 4, no. 6, pp. 1802–1831, 2017.

[11] R. Mitchell and I.-R. Chen, "A survey of intrusion detection techniques for cyber-physical systems," *ACM Comput. Surv.*, vol. 46, no. 4, p. 55, 2014.

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

[13] R. W. van der Heijden, S. Dietzel, T. Leinmüller, and F. Kargl. (2016). "Survey on misbehavior detection in cooperative intelligent transportation systems." [Online]. Available: https://arxiv.org/abs/1610.06810

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

[15] B. B. Zarpelão, R. S. Miani, C. T. Kawakani, and S. C. de Alvarenga, "A survey of intrusion detection in Internet of Things," *J. Netw. Comput. Appl.*, vol. 84, pp. 25–37, Apr. 2017.

[16] W. Zeng, M. A. S. Khalid, and S. Chowdhury, "In-vehicle networks outlook: Achievements and challenges," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 3, pp. 1552–1571, 3rd Quart., 2016.

[17] M. Aldwairi, A. M. Abu-Dalo, and M. Jarrah, "Pattern matching of signature-based IDS using Myers algorithm under MapReduce framework," *EURASIP J. Inf. Secur.*, vol. 1, no. 9, 2017.

[18] T. P. Vuong, G. Loukas, and D. Gan, "Performance evaluation of cyber-physical intrusion detection on a robotic vehicle," in *Proc. IEEE Int. Conf. Comput. Inf. Technol.; Ubiquitous Comput. Commun.; Dependable, Autonomic Secure Comput.; Pervasive Intell. Comput. (CIT/IUCC/DASC/PICOM)*, Oct. 2015, pp. 2106–2113.

[19] W. Fu, X. Xin, P. Guo, and Z. Zhou, "A practical intrusion detection system for Internet of vehicles," *China Commun.*, vol. 13, no. 10, pp. 263–275, Oct. 2016.

[20] T. Hoppe, S. Kiltz and J. Dittmann, "Applying intrusion detection to automotive IT-early insights and remaining challenges," *J. Inf. Assurance Secur.*, vol. 4, no. 6, pp. 226–235, 2009.

[21] C. Ling and D. Feng, "An algorithm for detection of malicious messages on can buses," in *Proc. Nat. Conf. Inf. Technol. Comput. Sci.* Paris, France: Atlantis Press, 2012.

[22] A. Taylor, N. Japkowicz, and S. Leblanc, "Frequency-based anomaly detection for the automotive CAN bus," in *Proc. WCICSS*, 2015, pp. 45–49.

[23] H. M. Song, H. R. Kim, and H. K. Kim, "Intrusion detection system based on the analysis of time intervals of CAN messages for in-vehicle network," in *Proc. Int. Conf. Inf. Netw. (ICOIN)*, Jan. 2016, pp. 63–68.

[24] C. Young, J. Zambreno, and G. Bloom, "Towards a fail-operational intrusion detection system for in-vehicle networks," in *Proc. 1st Workshop Secur. Dependability Critical Embedded Real-Time Syst.*, 2016, p. 37.

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

[26] M. Gmiden, M. H. Gmiden, and H. Trabelsi, "An intrusion detection method for securing in-vehicle CAN bus," in *Proc. 17th Int. Conf. Sci. Techn. Autom. Control Comput. Eng.*, Dec. 2016, pp. 176–180.

[27] H. Lee, S. H. Jeong, and H. K. Kim, "OTIDS: A novel intrusion detection system for in-vehicle network by using remote frame," in *Proc. IEEE 15th Annu. Conf. Privacy, Secur. Trust (PST)*, 2017, pp. 57–5709.

[28] O. Avatefipour, "Physical-fingerprinting of electronic control unit (ECU) based on machine learning algorithm for in-vehicle network communication protocol 'CAN-BUS,'" M.S. thesis, Dept. Comput. Eng., Univ. Michigan-Dearborn, Dearborn, MI, USA, 2017.

[29] M. Müter and N. Asaj, "Entropy-based anomaly detection for in-vehicle networks," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2011, pp. 1110–1115.

[30] M. Marchetti, D. Stabili, A. Guido, and M. Colajanni, "Evaluation of anomaly detection for in-vehicle networks through information-theoretic algorithms," in *Proc. IEEE 2nd Int. Forum Res. Technol. Soc. Ind. Leveraging Better Tomorrow (RTSI)*, Sep. 2016, pp. 1–6.

[31] R. Rieke, M. Seidemann, E. K. Talla, D. Zelle, and B. Seeger, "Behavior analysis for safety and security in automotive systems," in *Proc. 25th Euromicro Int. Conf. Parallel, Distrib. Netw.-based Process. (PDP)*, Mar. 2017, pp. 381–385.

[32] A. Wasicek, M. D. Pesé, A. Weimerskirch, Y. Burakova, and K. Singh, "Context-aware intrusion detection in automotive control system," in *Proc. 5th ESCAR USA Conf.*, 2017, pp. 21–22.

[33] A. Boudguiga, W. Klaudel, A. Boulanger, and P. Chiron, "A simple intrusion detection method for controller area network," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2016, pp. 1–7.

[34] M. Marchetti and D. Stabili, "Anomaly detection of CAN bus messages through analysis of ID sequences," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2017, pp. 1577–1583.

[35] A. Bezemskij, G. Loukas, R. J. Anthony, and D. Gan, "Behaviour-based anomaly detection of cyber-physical attacks on a robotic vehicle," in *Proc. Int. Conf. Ubiquitous Comput. Commun. Int. Symp. Cyberspace Secur. (IUCC-CSS)*, Dec. 2016, pp. 61–68.

[36] S. Abbott-McCune and L. A. Shay, "Intrusion prevention system of automotive network CAN bus," in *Proc. IEEE Int. Carnahan Conf. Security Technol. (ICCST)*, Oct. 2016, pp. 1–8.

[37] M. Markovitz and A. Wool, "Field classification, modeling and anomaly detection in unknown CAN bus networks," *Veh. Commun.*, vol. 9, pp. 43–52, Jul. 2017.

[38] D. Stabili, M. Marchetti, and M. Colajanni, "Detecting attacks to internal vehicle networks through Hamming distance," in *Proc. IEEE AEIT Int. Annu. Conf.-Infrastruct. Energy ICT (AEIT)*, Sep. 2017, pp. 1–6.

[39] A. Theissler, "Anomaly detection in recordongs from in-vehicle networks," in *Proc. 1st Int. Workshop Big Data Appl. Princ. (BIGDAP)*, Madrid, Spain, Sep. 2014, pp. 23–37.

[40] A. Abdiansah and R. Wardoyo, "Time complexity analysis of support vector machines (SVM) in LibSVM," *Int. J. Comput. Appl.*, vol. 128, no. 3, pp. 28–34, 2015.

[41] S. N. Narayanan, S. Mittal, and A. Joshi. (2015). "Using data analytics to detect anomalous states in vehicles." [Online]. Available: https://arxiv.org/abs/1512.08048

[42] M.-J. Kang and J.-W. Kang, "A novel intrusion detection method using deep neural network for in-vehicle network security," in *Proc. IEEE 83rd Veh. Technol. Conf. (VTC Spring)*, May 2016, pp. 1–5.

[43] M.-J. Kang and J.-W. Kang, "Intrusion detection system using deep neural network for in-vehicle network security," *PLoS ONE*, vol. 11, no. 6, p. e0155781, 2016.

[44] A. Taylor, S. Leblanc, and N. Japkowicz, "Anomaly detection in automobile control network data with long short-term memory networks," in *Proc. IEEE Int. Conf. Data Sci. Adv. Anal. (DSAA)*, Oct. 2016, pp. 130–139.

[45] A. Theissler, "Detecting known and unknown faults in automotive systems using ensemble-based anomaly detection," *Knowl.-Based Syst.*, vol. 123, pp. 163–173, May 2017.

[46] R. J. A. Ganesan and K. Shin, "Exploiting consistency among heterogeneous sensors for vehicle anomaly detection," SAE Tech. Paper 2017-01-1654, 2017.

[47] F. Martinelli, F. Mercaldo, V. Nardone, and A. Santone, "Car hacking identification through fuzzy logic algorithms," in *Proc. IEEE Int. Conf. Fuzzy Syst. (FUZZ-IEEE)*, Jul. 2017, pp. 1–7.

[48] R. E. Haas, D. P. F. Möller, P. Bansal, R. Ghosh, and S. S. Bhat, "Intrusion detection in connected cars," in *Proc. IEEE Int. Conf. Electro Inf. Technol. (EIT)*, May 2017, pp. 516–519.

[49] P. Sharma and D. P. F. Möller, "Protecting ECUs and vehicles internal networks," in *Proc. IEEE Int. Conf. Electro/Inf. Technol. (EIT)*, May 2018, pp. 0465–0470.

[50] C. Wang, Z. Zhao, L. Gong, L. Zhu, Z. Liu, and X. Cheng, "A distributed anomaly detection system for in-vehicle network using HTM," *IEEE Access*, vol. 6, pp. 9091–9098, 2018.
[51] A. Bezemskij, G. Loukas, D. Gan, and R. J. Anthony, "Detecting cyber-physical threats in an autonomous robotic vehicle using Bayesian networks," in *Proc. IEEE Int. Conf. Internet Things (iThings) IEEE Green Comput. Commun. (GreenCom) IEEE Cyber, Phys. Social Comput. (CPSCom) IEEE Smart Data (SmartData)*, Jun. 2017, pp. 98–103.
[52] M. Müter, A. Groll, and F. C. Freiling, "A structured approach to anomaly detection for in-vehicle networks," in *Proc. 6th Int. Conf. Inf. Assurance Secur. (IAS)*, Aug. 2010, pp. 92–98.
[53] T. Zhang, H. Antunes, and S. Aggarwal, "Defending connected vehicles against malware: Challenges and a solution framework," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 10–21, Feb. 2014.
[54] O. Berlin, A. Held, M. Matousek, and F. Kargl, "POSTER: Anomaly-based misbehaviour detection in connected car backends," in *Proc. IEEE Vech. Netw. Conf.*, Dec. 2016, pp. 1–2.
[55] D. K. Vasistha, "Detecting anomalies in controller area network for automobiles," M.S. thesis, Texas A&M Univ., College Station, TX, USA, 2017.
[56] M. Weber, S. Klug, E. Sax, and B. Zimmer, "Embedded hybrid anomaly detection for automotive CAN communication," in *Proc. 9th Eur. Congr. Embedded Real Time Softw. Syst. (ERTS)*, Toulouse, France, Jan. 2018. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01716805
[57] L. Zhang, L. Shi, and N. Kaja, "A two-stage deep learning approach for can intrusion detection," in *Proc. Ground Vehicle Syst. Eng. Technol. Symp. (GVSETS)*, 2018, pp. 1–11.
[58] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical Machine Learning Tools and Techniques*. San Mateo, CA, USA: Morgan Kaufmann, 2016.
[59] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Comput. Elect. Eng.*, vol. 40, no. 1, pp. 16–28, Jan. 2014.
[60] O. Y. Al-Jarrah, O. Alhussein, P. D. Yoo, S. Muhaidat, K. Taha, and K. Kim, "Data randomization and cluster-based partitioning for botnet intrusion detection," *IEEE Trans. Cybern.*, vol. 46, no. 8, pp. 1796–1806, Aug. 2016.
[61] R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2010, pp. 305–316.
[62] O. Y. Al-Jarrah, Y. Al-Hammdi, P. D. Yoo, S. Muhaidat, and M. Al-Qutayri, "Semi-supervised multi-layered clustering model for intrusion detection," *Digit. Commun. Netw.*, vol. 4, no. 4, pp. 277–286, Nov. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2352864817302912
[63] P. Kleberger, T. Olovsson, and E. Jonsson, "Security aspects of the in-vehicle network in the connected car," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2011, pp. 528–533.
[64] H. Schweppe and Y. Roudier, "Security and privacy for in-vehicle networks," in *Proc. IEEE 1st Int. Workshop Veh. Commun., Sens., Comput. (VCSC)*, Jun. 2012, pp. 12–17.
[65] I. Aad, J.-P. Hubaux, and E. W. Knightly, "Denial of service resilience in ad hoc networks," in *Proc. 10th Annu. Int. Conf. Mobile Comput. Netw.*, 2004, pp. 202–215.
[66] A.-S. K. Pathan, *Security of Self-Organizing Networks: MANET, WSN, WMN, VANET*. Boca Raton, FL, USA: CRC Press, 2016.
[67] A. D. Wood and J. A. Stankovic, "Denial of service in sensor networks," *Computer*, vol. 35, no. 10, pp. 54–62, Oct. 2002.
[68] X. Lin, X. Sun, P.-H. Ho, and X. Shen, "GSIS: A secure and privacy-preserving protocol for vehicular communications," *IEEE Trans. Veh. Technol.*, vol. 56, no. 6, pp. 3442–3456, Nov. 2007.
[69] Q. Luo and J. Liu, "Wireless telematics systems in emerging intelligent and connected vehicles: Threats and solutions," *IEEE Wireless Commun.*, vol. 25, no. 6, pp. 113–119, Dec. 2018.
[70] D. M. W. Powers, "Evaluation: From predcision, recall and F-factor to ROC, informedness, markedness & correlation," *J. Mach. Learn. Technol.*, vol. 2, no. 1, pp. 37–63, 2011.
[71] D. K. Bhattacharyya and J. K. Kalita, *Network Anomaly Detection: A Machine Learning Perspective*. London, U.K.: Chapman & Hall, 2013.
[72] I. Ahmad, A. B. Abdullah, and A. S. Alghamdi, "Application of artificial neural network in detection of dos attacks," in *Proc. 2nd Int. Conf. Secur. Inf. Netw.*, 2009, pp. 229–234.
[73] M. Zamani and M. Movahedi. (2013). "Machine learning techniques for intrusion detection." [Online]. Available: https://arxiv.org/abs/1312.2177
[74] O. Depren, M. Topallar, E. Anarim, and M. K. Ciliz, "An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks," *Expert Syst. Appl.*, vol. 29, no. 4, pp. 713–722, 2005.
[75] J. Z. Lei and A. Ghorbani, "Network intrusion detection using an improved competitive learning neural network," in *Proc. 2nd Annu. Conf. Commun. Netw. Services Res.*, May 2004, pp. 190–197.
[76] W.-H. Chen, S.-H. Hsu, and H.-P. Shen, "Application of SVM and ANN for intrusion detection," *Comput. Oper. Res.*, vol. 32, no. 10, pp. 2617–2634, 2005.
[77] S. Dua and X. Du, *Data Mining and Machine Learning in Cybersecurity*. Boca Raton, FL, USA: Auerbach Publications, 2016.
[78] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, 1996.
[79] R. Lippmann *et al.*, "Results of the DARPA 1998 offline intrusion detection evaluation," in *Proc. Recent Adv. Intrusion Detection*, vol. 99, 1999, pp. 829–835.
[80] R. Lippmann, J. W. Haines, D. J. Fried, J. Korba, and K. Das, "The 1999 DARPA off-line intrusion detection evaluation," *Comput. Netw.*, vol. 34, no. 4, pp. 579–595, 2000.
[81] *KDD Cup Dataset*. Accessed: Jul. 25, 2018. [Online]. Available: http://kdd.ics.uci.edu/databases/kddcup99

OMAR Y. AL-JARRAH received the B.S. degree in computer engineering from Yarmouk University, Jordan, in 2005, the M.S. degree in computer engineering from The University of Sydney, Sydney, Australia, in 2008, and the Ph.D. degree in electrical and computer engineering from Khalifa University, United Arab Emirates, in 2016.

He is currently a Postdoctoral Fellow with the Warwick Manufacturing Group, The University of Warwick, U.K. His main research interests include machine learning, intrusion detection, big data analytics, autonomous and connected vehicles, and knowledge discovery in various applications.

CARSTEN MAPLE is currently a Professor of cyber systems engineering with the Warwick Manufacturing Group, The University of Warwick, where he is also the Director of research in cyber security. He has an international research reputation having published over 200 peer-reviewed papers, and his research has attracted millions of pounds in funding and has been widely reported through the media. He is currently a Principal Investigator (PI) with the EPSRC/GCHQ Academic Centre of Excellence in Cyber Security, a Local PI of the U.K. Research Hub for Cyber Security of the Internet of Things, PETRAS, and FAIRSPACE, and the U.K. Research Hub for Future AI and Robotics in Space, and a Co-Investigator of the CARMA project, all funded by EPSRC. His research interests include authentication, privacy, the value of information, and cyber-physical systems. He is a fellow of the Alan Turing Institute.

MEHRDAD DIANATI was a Professor with the University of Surrey. He is currently a Professor of autonomous and connected vehicles with the Warwick Manufacturing Group, The University of Warwick, and a Visiting Professor with the 5G Innovation Centre, University of Surrey. He has been involved in a number of national and international projects as the Project Leader and a Work-Package Leader in recent years.

Prior to his academic endeavor, he has worked in the industry for over nine years as a Senior Software/Hardware Developer and the Director of R&D. He frequently provides voluntary services to the research community in various editorial roles; for example, he has served as an Associate Editor for the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, *IET Communications*, and the *Journal of Wireless Communications and Mobile* (Wiley).

DAVID OXTOBY received the B.Eng. degree in electronic engineering from the University of York, U.K., in 1993. He worked in the field of telecommunication for Nortel Networks, from 1993 to 2002, before making a career change into Automotive, in 2003, first working for Nissan on audio/navigation, telephone, and camera systems. Since 2013, he has been with the Jaguar Land Rover's Electrical Research team on a wide variety of projects, where he is currently responsible for a team delivering new electrical technologies from initial idea to concept ready for production.

ALEX MOUZAKITIS has over 15 years of technological and managerial experience especially in the area of automotive embedded systems. He is currently the Head of the Electrical, Electronics and Software Engineering Research Department, Jaguar Land Rover (JLR). In his current role, he is responsible for leading a multidisciplinary research and technology department dedicated to deliver a portfolio of advanced research projects in the areas of human–machine interface, digital transformation, self-learning vehicle, smart/connected systems, and on-board/off-board data platforms. In his previous position within the JLR, he has served as the Head of the Model-based Product Engineering Department responsible for model-based development and automated testing standards and processes.