IEEE Open Journal of
Vehicular Technology

Received 25 December 2022; accepted 26 December 2022. Date of publication 18 January 2023;
date of current version 9 February 2023. The review of this article was coordinated by Editor Lina Mohjazi.

Digital Object Identifier 10.1109/OJVT.2023.3237802

A Novel Detection Approach of Unknown Cyber-Attacks for Intra-Vehicle Networks Using Recurrence Plots and Neural Networks

OMAR Y. AL-JARRAH [1], KARIM EL HALOUI [2], MEHRDAD DIANATI [2] (Senior Member, IEEE), AND CARSTEN MAPLE [2] (Member, IEEE)

[1]Network Engineering and Security Department, Jordan University of Science and Technology, Irbid, Jordan
[2]Warwick Manufacturing Group (WMG), University of Warwick, Coventry CV4 7AL, U.K.

CORRESPONDING AUTHORS: OMAR Y. AL-JARRAH; KARIM EL HALOUI (e-mail: omar.jarrah2012@gmail.com; karim.el-haloui@warwick.ac.uk)

This work was supported by the U.K.-EPSRC under Grant EP/N01300X/1.

ABSTRACT Proliferation of connected services in modern vehicles could make them vulnerable to a wide range of cyber-attacks through intra-vehicle networks that connect various vehicle systems. Designers usually equip vehicles with predesigned counter-measures, but these may not be effective against novel cyber-attacks. Intrusion Detection Systems (IDSs) serve as an additional layer of defence when conventional measures that are implemented by the designers fail. Several intrusion detection techniques have been proposed in the literature but these techniques have limited capability in detecting novel cyber-attacks. This paper proposes a new Machine Learning (ML)-based IDS for detecting novel cyber-attacks in intra-vehicle networks, specifically in Controller Area Networks (CANs). The proposed IDS generates high-level representations of CAN messages transmitted on the bus exploiting their temporal properties as well as the intra and inter message dependencies through the use of Recurrence Plot (RP), which are then fed into a bespoke Neural Network, designed and trained to detect novel intrusions. Evaluation of the performance of the proposed IDS in comparison with that of the state-of-the-art existing IDS schemes demonstrates the superiority of the proposed IDS.

INDEX TERMS Cybersecurity, intrusion detection, intra-vehicle networks, LSTM.

I. INTRODUCTION

The ubiquitous nature of emerging V2X connectivity systems offers novel services and applications that enable advanced functions and features for modern vehicles, such as Advanced Driver Assistance Systems (ADAS), infotainment, productivity and maintenance services. For a safe, efficient, and comfortable operation of modern vehicles, data is transmitted through intra-vehicle and over inter-vehicle networks depending on system-level requirements [1]. This provides new cyber-attack surfaces for potential intrusions into the vehicle systems, which can put road users' lives at risk if exploited by malicious agents [2], [3].

Modern vehicles are complex cyber-physical systems that embed different components, including Electrical Control Units (ECUs), sensors and actuators. The Controller Area Network (CAN) forms the communication backbone of most vehicles over which these components exchange data. Unfortunately, the CAN protocol has inherent cybersecurity vulnerabilities due to the lack of authentication mechanisms and the broadcasting nature of its communication method. These vulnerabilities can make vehicles subject to a wide range of cyber threats (e.g., fuzzing and Denial of Service (DoS) attacks) [4]. In addition, the external V2X connectivity of vehicles could expose their intra-vehicle networks to remote attacks, allowing hackers to get access to safety-critical sub-systems of connected vehicles (e.g., braking system). Supplementing the functionality of conventional security measures (e.g., encryption algorithms), Intrusion Detection Systems (IDSs) serve as real-time monitoring systems that are becoming an integral part of the security architecture of modern vehicles to detect cyber-attacks which may have evaded conventional counter-measures.

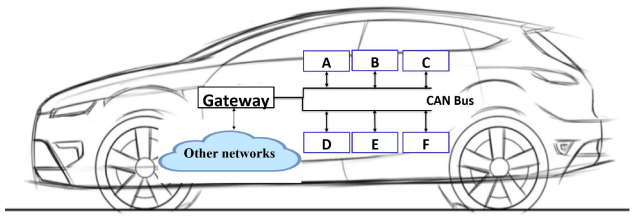


FIGURE 1. Typical CAN network with six nodes (i.e., ECUs).

Intra-vehicle IDSs can be categorised as flow-based, payload-based or hybrid [5]. Whereas a payload-based IDS inspects the content of messages to detect potential intrusions, a flow-based IDS examines the transmission patterns of messages. Generally, flow-based IDSs are suitable for detecting intrusions that affect the frequency and order of messages (e.g., [6], [7], [8]) but their performance is inadequate when the attack affects the content of the messages [5] while payload-based IDSs (e.g., [9], [10], [11]) are effective to such attacks but display weaknesses in detecting attacks that affect the timing and sequence of messages (e.g., message injection attack) [5]. Combining the aforementioned categories, hybrid IDSs aim to combine the strengths of both approaches [5].

Most recently, Machine Learning (ML)-based techniques have been studied in the literature. Unlike rule-based IDSs, where the system designers hard-code the rules to detect intrusions, deep learning models are designed to learn and detect abnormal patterns from training datasets that contain samples of both the normal operation of the system and when under attack. A detailed review of the existing ML-based IDSs as well as their strengths and weaknesses are given in Section II.

This paper presents a novel ML-based hybrid IDS for detecting novel attacks in intra-vehicle networks (i.e., CAN) with a typical deployment such as in Fig. 1. The proposed IDS extracts representative features from the data by looking at the relative local context of a subject message by generating a 2D representation based on the Recurrence Plot (RP) concept. The content of the subject message together with its relative time, with respect to the previous message, are then fed into a Long-Short Term Memory (LSTM) neural network while the generated 2D representation is fed into a Convolutional-LSTM. The main advantage of the proposed technique stems from its ability to combine the learning of the intra-message data dependencies and inter-messages temporo-contextual dependencies, thus, enhancing the learning of the detection model as demonstrated by the improved Key Performance Indicators (KPIs), such as detection accuracy, compared to the state-of-the-art ML models. Our performance evaluation results demonstrate that the proposed IDS outperforms the state-of-the-art ML techniques, in terms of its ability to detect novel cyber-attacks.

The main contributions of this work can be laid out as follows:

- The design of a new ML-based IDS for detecting novel cyber-attacks in intra-vehicle networks, specifically in the CAN of modern vehicles.
- A method to generate inter-messages representative features based on RPs to capture the required temporo-contextual dependencies of messages.
- An evaluation of the performance of the proposed IDS in comparison with state-of-the-art IDSs, showing the superiority of the proposed method.

The rest of the paper is structured as follows: Section II provides a comprehensive review of the state-of-the-art of the related ML-based IDSs. After a short overview of LSTM-based neural networks and RP, Section III details the proposed detection model and positions it within the current landscape of IDSs. Section IV compares and discusses the performance of the proposed technique against the state-of-the-art solutions. To conclude, Section V summarises the findings of this paper and indicates potential future research directions.

II. RELATED WORK

In this section, we discuss related studies, provide an overview of their approaches, as well as their advantages and limitations. Finally, we articulate how the proposed work in this paper fits into the current landscape.

The concept of intrusion detection in intra-vehicle networks was first coined by Hoppe et al. [12]. Since then, a tremendous effort has been carried out to improve IDS KPIs with a notable shift towards ML approaches enabled by the increase of available processing power and promising initial results. Current ML techniques are used at various levels or interfaces to leverage and learn semantic representations of data to detect anomalies. At the data link layer, two natural directions of research were taken based on the payload and flow of data.

Several researchers have developed IDSs for intra-vehicle networks. Levi et al. [13] described a Hidden Markov Model (HMM)-based detection system trained on data collected from vehicles. The trained HMM together with a regression model are used to detect anomalies from the normal expected operation. Their approach monitors different interfaces (communication, CAN and operating system interfaces) across the system, extracts relevant pieces of information based on configurable rules and sends them to a trained model to detect anomalies. A configurable data collector provides a higher level of data abstraction (i.e., events), by modelling the time series data to states, which has an inherent noise-filtering effect and eliminates the need to retrain the model. The objective of the regression model is to calibrate the likelihood threshold for detecting anomalies. Choi et al. [14] went into a different direction and proposed VoltageIDS, an automotive IDS, leveraging the fact that electrical signals used to transmit CAN messages depend on the physical configuration of the network such as cables length, true value of termination resistors, true voltage values of bit zero and one, for example. The operation of VoltageIDS in a CAN is constituted of three phases, namely, the feature extraction, the feature selection, and the intrusion detection phase. In the feature extraction phase, the VoltageIDS extracts 60 features from the electrical signal of normal CAN messages which are then filtered out

by the feature selection phase, selecting only the most significant features. In the intrusion detection phase, the VoltageIDS builds a supervised ML multi-class classifier (e.g., Support Vector Machine) using attack-free CAN data. When deployed, the multi-class classifier predicts the class label (i.e., normal or intrusion) of messages.

Kang and Kang [15] built a Deep Neural Network (DNN)-based IDS trained on high-dimensional features extracted from bit streams of CAN messages. Song et al. [16] adopted Inception Resnet to develop an IDS for CAN. A dataset composed of CAN messages transmitted on a CAN bus of a real vehicle was used to evaluate the proposed system with results outperforming conventional ML methods. Lin et al. [17] developed an IDS based on the Visual Geometry Group (VGG)-DNN. Taylor et al. [18] proposed a LSTM neural network for detecting cyber-attacks in intra-vehicle network, including interleave, drop, discontinuity, unusual and reverse attacks. In the same vein, Loukas et al. [19] presented an LSTM-based IDS but it focused on attacks that are particularly meaningful for robot vehicles.

Martinelli et al. [20] used four Fuzzy algorithms applied to eight features (the eight data bytes of the data field in a CAN message) to detect cyber-attacks. The authors showed that the fuzzy classification algorithms can achieve high performance in detecting three types of attack, namely, Denial of Service (DoS), Fuzzy and message injection.

In [21], Zhu et al. proposed a literal multi-dimensional anomaly detection approach using a distributed LSTM framework. The proposed model uses both time and data dimensions of CAN messages to detect cyber-attacks. The experimental results showed that the proposed model could accomplish a detection accuracy of $\sim 90\%$.

Derhab et al. [2] proposed a Histogram-based Intrusion Detection and Filtering (H-IDFS) framework. The proposed framework, first, groups CAN frames into windows and calculates their histograms which are, then, fed into a multi-class classifier to identify windows containing malicious CAN frames. Thereafter, a one-class SVM filters out malicious CAN frames from each malicious window.

Basavaraj and Tayeb [22] designed a DNN-based IDS and evaluated its performance on two real datasets where they achieved a detection accuracy of 98.67% on known attacks.

He et al. [23] proposed Hybrid Similar Neighbourhood Robust Factorisation Machine Model (HSNRFM) for detecting anomalies in the in-vehicle network. Firstly, the HSNRFM performs a dimensionality reduction of the original data, to enhance its robustness. Then, it combines the information of the target message as well as neighbour messages to form the final input features vector of a factorisation ML model used to derive the final prediction value.

Generally, existing studies for intrusion detection in intra-vehicle networks

- Focus on detecting specific types of cyber-attacks, ignoring novel cyber-attacks [5].
- Neglect the context of messages whereby a message that appears as normal in a given context (i.e., message sequence) may appear as abnormal in another one.
- Do not discriminate between malicious and normal frames within a malicious window, which could cause the system to drop all the frames within the window, causing potentially undesirable effects with an impact on the overall safety of the vehicle.

Diverging from existing works, we propose an IDS for intra-vehicle networks that generates two independent views of the CAN traffic to detect different types of cyber-attacks, augmenting the overall detection capability for both known and novel cyber-attacks. This approach incorporates intra-message features and features derived from the inter-dependencies among CAN messages captured through RPs.

There are several ways in which the proposed IDS differs from the state-of-the-art:

- Use of machine learning: We adopt a ML-based approach to identify potential intrusions, rather than relying on predefined rules or patterns. This allows the system to adapt to changing patterns of normal and anomalous behaviour, and to potentially identify novel attacks that may not have been anticipated by the designers of the system.
- High-level representations of CAN messages: The model generates high-level representations of CAN messages using RP, which captures the complex relationships and temporal dependencies among the messages. These representations provide a more detailed and nuanced view of the data than some other methods, which may allow the system to more accurately identify potential intrusions.
- Analysis of individual messages: Unlike some IDS approaches, the operational granularity of the proposed system is at the message level. It is designed to label each individual message as either normal or anomalous, rather than labeling an entire window of messages as a whole. This allows the system to more accurately identify potential intrusions in individual messages, rather than relying on the presence of a pattern of anomalies within a group of messages.

III. PROPOSED APPROACH

In this section, we give some background on CAN, Recurrent Neural Networks (RNNs), with a focus on LSTM, and RP. We also describe in detail the proposed model.

A. BACKGROUND

CAN plays a preponderant role in the communication architecture of modern vehicles. Messages transmitted over a CAN exhibit temporal relationships. For example, stepping on the accelerator pedal allows more air into the engine. The engine control unit senses the increased airflow and acts accordingly by pumping more fuel into the engine. As a result, the vehicle accelerates, and the rotation-per-minute increases. These

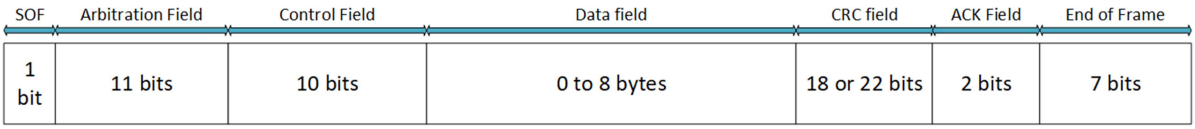


FIGURE 2. Principal fields of the CAN frame.

actions happen in a certain sequence and translate into well-structured time-series traffic under usual driving conditions. However, the temporal relationships observed between messages in intra-vehicle networks can deviate from these typical patterns when the vehicle is under abnormal conditions, such as cyber-attacks or faults.

Conventional intrusion detection techniques in intra-vehicle networks are prominently based on the temporal relationships between messages and their content [21]. From a timing perspective, many messages in the CAN are periodic, meaning that they normally appear at a regular frequency and show a sequential pattern [24]. From a data perspective, the data content transported by CAN frames, with the same CAN ID, also exhibits certain patterns and trends under normal conditions. However, the characteristics of these patterns change when the vehicle is under cyber-attack. On the one hand, a DoS attack, where the attacker injects malicious messages into the network at high rates, affects the frequency of the messages and their sequence. On the other hand, an integrity attack, where the malicious agent tampers the data content, affects the data pattern observed within a CAN frame, despite the fact that it might appear to be valid from a timing perspective.

B. DESCRIPTION OF THE PROPOSED MODEL

Based on these established insights, we propose a ML-based IDS to detect cyber-attacks in CAN by looking at both the content transported by a message and its relative context. For this, two views are generated from the received CAN data. The first view is generated from the intra dependencies of one CAN message and the other from its context. Concatenated, these two views form the input feature vector of a dense neural network to classify each message as normal or as an intrusion.

Messages received by a CAN node are timestamped either in hardware or software. This information is not, per se, part of the CAN protocol but it might be deemed important by upper layer protocols if the sequentiality and order of messages arrival is required. In our model, this piece of information proves to be vital as it gives a temporal context to CAN messages. A typical CAN frame is described in Fig. 2 where the most salient features are the arbitration and data fields. In particular, the arbitration field contains an 11-bit or 29-bit subfield, called message ID, identifying and describing uniquely the data field of each CAN message whose maximum length is 8 bytes.

In our model, we call an ordered set of messages a message window and the last message in this sequence the subject message of the message window. To capture the temporal relationship in a message window, the relative-time stamp (RTS) between a message and its predecessor is calculated, and together with the message ID (ID), Data Length (DL) and data fields (D1,..., D8) form the input feature vector (Input 1) of a RNN to generate the first view; thus, capturing the intra-message dependencies, as in Fig. 3.

RNNs are a type of neural networks with the ability to learn temporal relationships in a data sequence. They can be thought of as a sequence of fully connected neural networks where the state at time t of an RNN, $a^{<t>}$, is updated based on the current input, $x^{<t>}$, and previous state, $a^{<t-1>}$, through weight matrices W_{aa} and W_{ax}. The output $\hat{y}^{<t>}$, at time t, follows a standard calculation and is based on the value of the current state, $a^{<t>}$, and weight matrix W_{ya}. These weight matrices are shared through the sequence as shown in Fig. 4.

RNNs can be used to detect intrusions or cyber-attacks in CAN. For example, it is possible to consider CAN messages in a message window as a data sequence. The likelihood of classifying a subject message, as intrusive or normal, depends on the information collected from previous messages (i.e., prior states) and the subject message (i.e., the current input). Generally speaking, conventional RNNs have limited capabilities in capturing long-term temporal-dependencies in long data sequences due to the vanishing gradient problem, which refers to the exponential decrease in the gradient when updating the weight matrices through back-propagation [19]. An LSTM can address the vanishing gradient problem by introducing gates to control the flow of information shared within an LSTM unit [25]. Traditionally, three gates are in use: the forget gate, the input gate and the output gate, activated by f_t, i_t and o_t respectively as in Fig. 5. Of particular importance is the forget gate as it controls how much information from the previous state is passed to the current state.

The proposed model generates the second view by taking a message window as input to create a 2D texture using the notion of RP. Eckmann et al. [26] proposed RP as a method to visualise recurrent states of dynamical systems [27]. Often, the current state of a dynamical system, represented by a geometrical manifold in the appropriate space, is followed by one future state governed by some transition rule that describes the evolution of the states of the dynamical system over time [28].

This evolutionary concept seems to be typical to vehicles where the communication system exhibits recurring patterns. This implies the presence of an internal mechanism that generates regular and recurrent behaviours/patterns in the data [29]. In this context, we adopt RP to provide high-level explicit representations/features capturing the periodicity of the data, which can hypothetically lead to an improved detection rate.

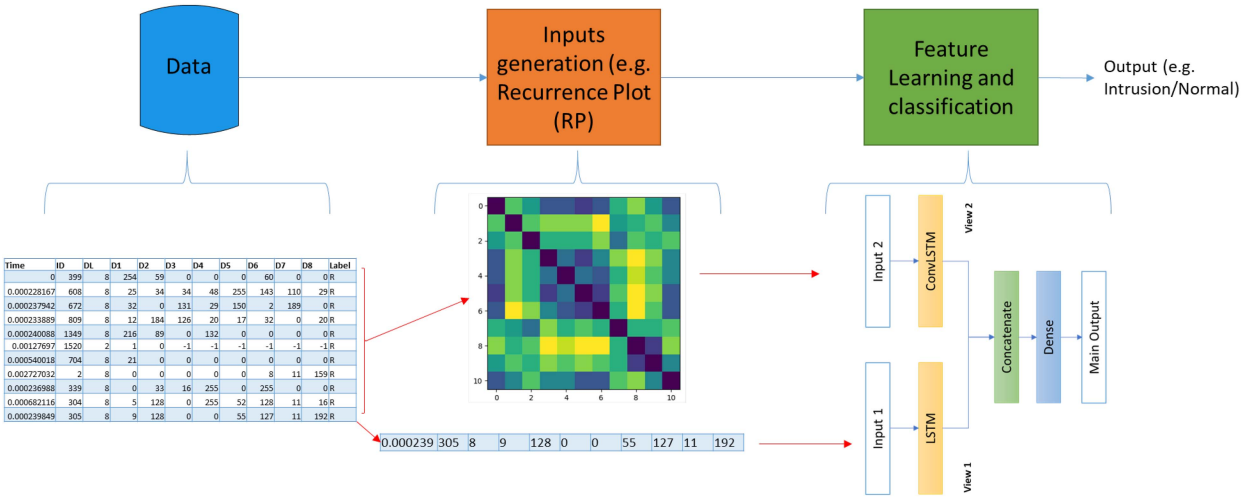


FIGURE 3. System model of the proposed IDS.

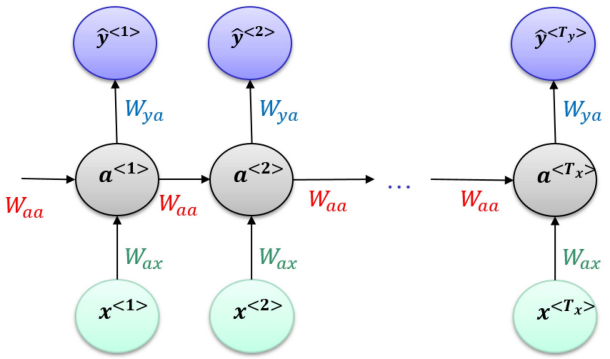


FIGURE 4. Architecture of a conventional RNN with one layer and T_x states. $x^{<t>}$ denotes the input at time t, T_x the number of inputs in a data sequence and W_{aa}, W_{ax}, W_{ya} weight matrices.

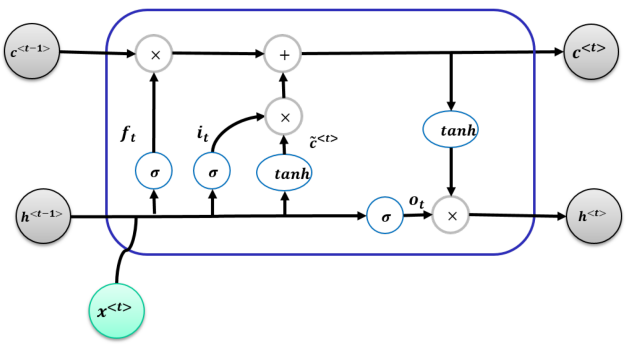


FIGURE 5. An LSTM unit.

The RP is defined as follows [29]:

$$R_{i,j} = H\left(\epsilon - \|\mathbf{x}_i - \mathbf{x}_j\|\right), i, j \in \{1, \ldots, n\}, \tag{1}$$

where $\mathbf{x}_i, \mathbf{x}_j$ are the states of length d (i.e., CAN messages) observed at position/time i and j, respectively. $\|.\|$ denotes a norm between the observations, ϵ a threshold for closeness, n the number of states (i.e., number of messages) and H the Heaviside function defined as [29]:

$$H(z) = \begin{cases} 0, & \text{if } z < 0 \\ 1, & \text{otherwise} \end{cases} \tag{2}$$

The calculation of RP requires setting the value of the closeness threshold ϵ, but determining its value is not intuitive. Heuristics such as setting the threshold to 10% of the largest observed distance or a certain percentage of black points can be used. However, these do not generalise well to multiple RPs and can make it difficult to determine the similarity between two RPs [29]. Following [29], we eliminate the closeness threshold, and also the Heaviside function to keep the granularity provided by the norm function. RP is now a $n \times n$ square matrix whose entries $R_{i,j}$ are given by:

$$R_{i,j} = \|\mathbf{x}_i - \mathbf{x}_j\|, i, j \in \{1, \ldots, n\}. \tag{3}$$

The output represents the distance between different messages in a sequence, and can be viewed as a coloured map. As such, the RP is no longer a tool to analyse recurrence considering neighbourhoods but it quantifies how close each pair of messages in a sequence is. This is known as unthresholded RP, distance plot, or self-similarity matrix [29].

The norm $\|.\|$, and its induced distance, should be carefully chosen. A simple and widely used distance measure is the Euclidean distance. However, the Euclidean distance does not look at the neighbourhood of each entry as the calculation is performed coordinate-wise. This way, the context of each data field is not captured. We need a distance function that looks not only at the neighbouring context but also at the individual fields. Dynamic Time Warping (DTW) is able to measure the similarity between two data sequences while also considering the neighbourhood of each data field, i.e., its context. However, the complexity of the DTW algorithm is quadratic. In order to reduce the time and memory needed to calculate RPs, we used FastDTW [30], which approximates

Algorithm 1: RP Generation Using DTW Distance.

Output: An RP of a sequence of CAN messages
Input: s *// a sequence of CAN messages*
initialization: RP=[] *// an empty list to hold the result/output*;
for *i in s* **do**
  d = [] *//an empty list to hold the similarity between message i and all messages in s*;
  for *j in s* **do**
    sim=DTW(i, j) *// calculate the similarity between messages i and j using the DTW technique*;
    d.append(sim) *// append sim to d*;
  end
  RP.append(d) *// append d to RP*;
end
return RP;

the DTW algorithm with the added advantage to only be of linear complexity.

Algorithm 1 shows the implementation of RP using DTW and Algorithm 2 the DTW calculation adopted in this paper. In Algorithm 2, $|\mathbf{x}_i[k] - \mathbf{x}_j[z]|$ denotes the cost of matching the two entries $\mathbf{x}_i[k]$ and $\mathbf{x}_j[z]$ at indices k and z of CAN messages $\mathbf{x}_i$ and $\mathbf{x}_j$, respectively. Each CAN message is captured in our implementation with 11 entries/fields (e.g., time-stamp, ID, *etc.*).

The generated 2D texture form the input feature vector (Input 2) of a ConvLSTM layer which extracts high-level features of the message window. This captures different data representations taking advantages of the properties of Convolutional Neural Networks (CNNs) [31]. The model concatenates view 1 and 2 together, where both the intra and inter message dependencies are encoded. The output of the "concatenate" layer is then fed to a dense layer to predict whether the subject message is a normal message or an attack, as in Fig. 3.

Algorithm 2: DTW Calculation.

Output: DTW between two messages $\mathbf{x}_{(i)}$ and $\mathbf{x}_{(j)}$
Input: $\mathbf{x}_{(i)}$ and $\mathbf{x}_{(j)}$ *// Two messages of dimension d*
initialization: DTW := array [infinity] *// a $d \times d$ array whose entries are infinity*
DTW[0, 0] := 0
Calculation of the DTW between $\mathbf{x}_{(i)}$ and $\mathbf{x}_{(j)}$:
for *k = 1 to d* **do**
  for *z = 1 to d* **do**
    DTW[k, z] := $|\mathbf{x}_{(i)}[k] - \mathbf{x}_{(j)}[z]|$+ min{DTW(k-1, z), DTW(k, z-1), DTW(k-1, z-1)}
  end
end
return DTW[d, d]

TABLE 1. Statistics of CAN-Intrusion-Dataset

Attack Type	# Normal msg	# Attack msg	# Total msg
DoS	3,078,250	587,521	3,665,771
Gear	3,845,890	597,252	4,443,142
RPM	3,966,805	654,897	4,621,702
Fuzzy	3,347,013	491,847	3,838,860

IV. EXPERIMENTS AND ANALYSIS

In this section, we describe the initial dataset used and the required preprocessing needed to generate the inputs of our model. We also give the complete setup, including the method employed to tune hyper-parameters. We complete the section by a comparative performance evaluation between our model and the state-of-the-art ML IDS solutions.

A. DATASET

To evaluate the performance of our model, we used the "CAN-intrusion-dataset" presented in [32]. The dataset contains four types of attacks:

- DoS attack: high priority CAN messages (e.g., messages with ID '0x000') injected to the CAN bus with a short time cycle (every 0.3 milliseconds).
- RPM/Gear attack: CAN messages with specific message IDs related to RPM/Gear messages injected to the CAN bus with a time cycle of 1 ms.
- Fuzzy attack: CAN messages with spoofed random message IDs and data injected to the CAN bus with a time cycle of 0.5 milliseconds.

Table 1 shows the statistics of the dataset used in this work. We considered the DoS, Gear, and RPM as our training dataset and the Fuzzy dataset as testing dataset. To evaluate and compare the performance of IDSs, it is essential that the testing dataset contains CAN IDs that are not specific to only DoS, Gear or RPM attacks but to more generic attacks. This justifies why the Fuzzy dataset is selected to model novel attacks and serves as a benchmark to measuring performance metrics of the different models used in this study.

B. PREPROCESSING AND GENERATING OF INPUTS

Let $D = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$ be the initial dataset, with $\mathbf{x}_t = \{x_t[1], \ldots, x_t[11]\}$, $y_t \in \{normal, attack\}$, where N denotes the number of messages in the dataset, $\mathbf{x}_t$ the message at position/time t and y_t the class label of $\mathbf{x}_t$. To prepare the data for training and testing, we have performed the following steps:

- Data conversion and padding: Initially, we converted the content of all messages in D to a decimal data representation. The input feature vector, Input 1, has to be of fixed length but the data length of CAN messages varies from 0 to 8 bytes. Hence, to guarantee that all messages have the same length, we padded messages with $DL < 8$ with extra bytes with a value of '-1'. This value has to be fixed and chosen so as to never occur in the original dataset.

(a) *Normal* (b) *Attack*

FIGURE 6. Visualisation of RPs.

- Relative-time derivation: We replaced the time stamp of the messages in D with the relative-time between consecutive messages in the dataset as follows:

$$x_t[1] = \begin{cases} 0, & \text{if } t = 0 \\ x_t[1] - x_{t-1}[1], & \text{otherwise} \end{cases} \quad (4)$$

The preprocessed dataset D serves as the first input (i.e., Input 1) of the proposed detection model. In order to generate the second input (i.e., Input 2), we normalised the data and adopted a sliding window of size 11 moving one message at a time.

The choice of the window size in an IDS is an important design decision that can affect its performance. A large window size can provide a wide context for analysing messages, which may be beneficial for detecting certain types of anomalies or attacks. However, a large window size can also increase the complexity of the IDS, which can lead to longer training times and possibly lower performance if the IDS becomes too complex to effectively learn from the data.

The setting of the window size depends on the specific requirements and constraints of the IDS. Some factors to consider may include the types of attacks or anomalies the IDS is designed to detect, the amount of available training data, and the resources available for training the IDS (e.g., time, computational power).

One approach to finding an optimal window size is to perform experiments with different window sizes and evaluate the IDS's performance using metrics such as accuracy, precision, and recall. This can allow identifying a window size that strikes a good balance between detecting anomalies and maintaining reasonable training and runtime complexity.

In our case, the number of feature vector elements is 11. Therefore, the minimum number of independent parameters required for linear correlation is 11. Additional parameters may be added to include nonlinear relationships. As such, the output of this process is a dataset D' where each data point in D' is an 11×11 array. A new dataset D'' is created, whose elements are RPs generated for each data point in D', which constitutes the second input of the proposed detection model. Fig. 6 gives an example of a normal RP vs. an intrusive RP.

C. EXPERIMENT SETTINGS AND HYPER-PARAMETERS TUNING

The proposed IDS has been implemented using Keras[1] deep learning library with TensorFlow[2] as back-end. The fine tuning of the hyper-parameters of both our proposed models and ML state-of-the-art models was performed using Autonomio Talos.[3] The dataset employed for this purpose is a subset of the concatenation of the DoS, Gear, and RPM datasets. The hyper-parameters of the DT and RF models were tuned using a grid search method.[4] To evaluate the generalisation capability of the proposed model to novel attacks, we considered the Fuzzy dataset as testing dataset. The selected hyper-parameters are given in Table 2.

D. PERFORMANCE EVALUATION

State-of-the-art metrics are used to measure and compare the performance of the proposed model against well-known IDSs: Accuracy (Acc), Detection Rate (DR), False Positive Rate (FPR), $Precision$, $F1$ score and Matthew's Correlation Coefficient (Mcc). The positive class is composed of intrusive messages and the negative class of normal or non-intrusive messages.

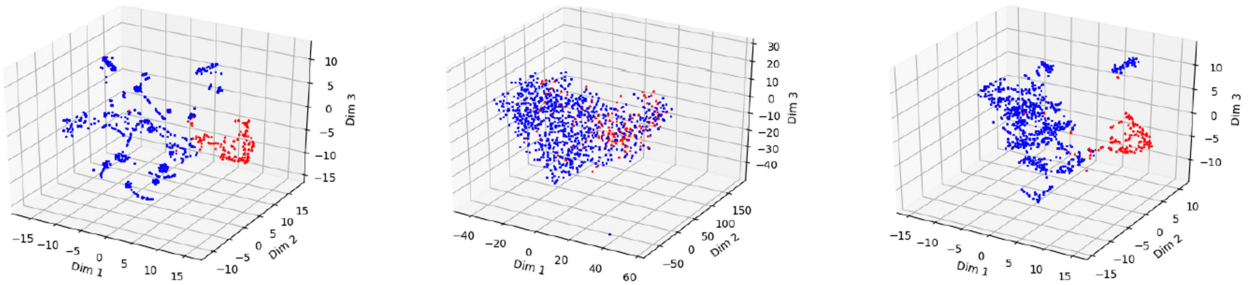
Each metric plays a specific role where some are crucial in the case of imbalanced classes. Acc indicates the ability of a binary classifier to correctly classify messages as being intrusive or normal [5]. DR measures the capability of a system to detect intrusive messages. In our scenario, the DR value can be interpreted as the probability of an actual intrusive message classified as intrusive. For example, a DR value of 0.5 would mean that half of the intrusive messages are detected as intrusive. Similarly, it is also important to class non-intrusive messages correctly and not detect them as intrusive. The FPR indicator helps measuring this unwanted behaviour by looking at the number of actual non-intrusive message classified as intrusive, the closer FPR is to zero the better. It is critical for IDSs if a fail-safe policy is adopted. In such a scenario, a

[1] [Online]. Available: https://keras.io/
[2] [Online]. Available: https://www.tensorflow.org/
[3] [Online]. Available: https://github.com/autonomio/talos
[4] [Online]. Available: sklearn.model_selection.GridSearchCV

TABLE 2. Summary of models performance

Model	Acc%	DR%	Precision%	F1	FPR	Mcc
Model using View 1	93.78642	51.50420	99.99804	0.67813	1.48391×10^{-6}	0.69253
Model using View 2	83.66093	18.81113	28.31386	0.22235	0.06809	0.14178
Model using View 1&2	95.10476	61.79610	99.99473	0.76240	4.82023×10^{-6}	0.76427
DT [34]	88.19493	7.86179	1	0.14578	0	0.26314
RF	87.18911	1.13857	1	0.00023	0	0.00996
Deep Learning [19]	88.39167	9.39740	99.99990	0.16322	9.95916×10^{-9}	0.26272

Hyper-parameters of the proposed models: optimizer=Nadam (lr=0.0008), loss=categorical_crossentropy, filters=32, LSTM_neurons=196, LSTM_activation=relu, Main_output_activation=softmax, batch_size=1024, epoch=1. Hyper-parameters of Deep Learning [19]: Lstm_neurons=512, Dense_neurons=156, Dropout_rate=0.3, activation=relu, output_activation=sigmoid, loss=binary_crossentropy, optimizer=adam(). Hyper-parmeters of the DT: criterion= entropy, max_depth= None, max_features= 9, min_samples_leaf= 5. The best hyper-parameters of RF are :bootstrap= True, max_depth= None, max_features= 4, min_samples_split= 10, n_estimators= 113. The results presented in the table for the proposed models and the deep learning approach presented in [19] is the average of 30 runs.



(a) Outputs of the LSTM layer (*i.e.,* View 1).

(b) Outputs of the ConvLSTM layer (*i.e.,* View 2).

(c) Outputs of the Concatenation layer.

FIGURE 7. Extracted features of different layers.

high FPR would distract the system from its normal operation mode and require the need to investigate the incident by an expert to determine if a genuine intrusion was present or not, hence, increasing costs. *Precision* is the percentage of actual intrusions amongst all predicted intrusions classified by the classifier. It gives us the confidence of an intrusion being an actual intrusion. For imbalanced data sets, two further performance measures are selected, namely, $F1$ and Mcc. $F1$ is the harmonic mean of DR and *Precision*, thus, giving no precedence of one over the other. As to Mcc, it takes into account the ratios of the four classes in the confusion matrix and provides a correlation coefficient between the result of the classifier and the actual data. The range of values that Mcc takes is between -1 and $+1$. A perfect classification results in an Mcc value of 1 whereas a -1 value indicates a total disagreement between the decision of the classifier and the actual data. A zero score indicates that the classifier is not better than a random classifier [33]. Formulas of the performance measures used to evaluate the performance of different models can be found in [5].

We evaluated and compared the performance of our model with ML algorithms proposed in the literature for intrusion detection tasks, namely, DT used in [34] and RF, as well as the state-of-the-art deep learning approach presented in [19]. Delgado et al. [35] have shown that RF is the best performing classifier among 179 tested methods across 121 different classification tasks. Also, Belavagi and Muniyal [36] have shown that RF outperforms other ML algorithms (e.g., SVM and logistic regression) for intrusion detection tasks. We also evaluated the performance of the proposed IDS when the detection model is built using a single input (e.g., Input 1) or both inputs (Input 1&2). Table 2 gives the performance of all models.

Table 2 shows that DT and RF, as well as the deep learning model presented in [19], have a limited capability in detecting novel cyber-attacks denoted by their low *Acc, DR, F1* score, and Mcc value. These models classify most of the instances in the testing data as normal messages, which also indicates a low generalisation ability.

As seen in Table 2, the proposed model (using View 1&2) achieves the highest *Acc* (95.10476%), with a six-fold improvement in the DR (61.79610%), a marginal drop in *Precision* (99.99473%) by 0.00517 compared with the Deep Learning model ([19]) and an increase by more than four-fold of $F1$ (0.76240). A similar trend is observed with Mcc which increases by slightly less than three-fold whilst keeping a very low FPR. It is worth noting that the proposed model using only View 2 has a lower performance than both the proposed model using only View 1 and the concatenated views. The low performance of View 2 indicates that using only the context into which a message appears is not sufficient to distinguish between normal and intrusive messages. The exploration of the content of each message given by View 1 shows already excellent results on its own. However, providing a context in which the message occurs, provided by View 2, enhances significantly the overall performance for each measure.

To visualise the distinctive features that View 1, View 2, and both combined bring to separate intrusive messages from regular traffic, we have employed a data reduction and visualisation technique, namely t-SNE [37], results of which are depicted in Fig. 7, with a reduction of the dimensions to three, where blue points represent regular messages and red points intrusive messages. As can be seen in Fig. 7(a), features provided by View 1 allow a separation of both classes with a separability that can be improved further. In Fig. 7(b), we have a cloud of points where points in both classes are intermingled. It indicates that View 2 alone cannot lead to obtaining a good-performance classifier, however, when concatenating it with View 1 as in Fig. 7(c), we can notice an increase of the separability of points belonging to both classes with a reduction of the scattering compared to Fig. 7(a). These constitute strong visual indicators for achieving better performances as evidenced by the results presented in Table 2.

V. CONCLUSION

This paper proposed an ML-based approach for intrusion detection in intra-vehicle networks. The proposed approach generates two representations/views of the CAN data leveraged by machine learning techniques. The views provide high-level features capturing the time and intra-message dependencies of the CAN messages as well as their context. These views are concatenated and used to predict the class label of each message. The performance of the proposed approach was evaluated and compared with the state-of-the-art detection techniques. The results demonstrated that combining both views lead to better performance compared to a single view. The results also demonstrated that the proposed approach outperforms other state-of-the-art methods in detecting novel intrusions as it achieved the highest accuracy (95.10476%), detection rate (61.79610%), F1-score (0.76240), and Matthew's correlation coefficient (0.76427), with a low false positive rate (4.82023×10^{-6}).

Although the proposed approach outperformed other techniques and achieved promising results, it was not able to perfectly detect all novel cyber-attacks. The possibility of improving the detection capability of the proposed approach could be investigated further. Despite that our method relies on the structure of CAN messages, we believe it could be easily extended for typical message addressing protocols, like CAN FD, and potentially, FlexRay. In addition, it could be worthwhile to investigate how the current work can be extended to inter-vehicle networks.

REFERENCES

[1] C. Jichici, B. Groza, R. Ragobete, P.-S. Murvay, and T. Andreica, "Effective intrusion detection and prevention for the commercial vehicle SAE J1939 CAN bus," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 17425–17439, Oct. 2022.

[2] A. Derhab, M. Belaoued, I. Mohiuddin, F. Kurniawan, and M. K. Khan, "Histogram-based intrusion detection and filtering framework for secure and safe in-vehicle networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 3, pp. 2366–2379, Mar. 2022.

[3] H. Sun, M. Sun, J. Weng, and Z. Liu, "Analysis of id sequences similarity using DTW in intrusion detection for CAN bus," *IEEE Trans. Veh. Technol.*, vol. 71, no. 10, pp. 10426–10441, Oct. 2022.

[4] G. Dupont, J. d. Hartog, S. Etalle, and A. Lekidis, "Network intrusion detection systems for in-vehicle network-technical report," 2019, *arXiv:1905.11587*.

[5] O. Y. Al-Jarrah, C. Maple, M. Dianati, D. Oxtoby, and A. Mouzakitis, "Intrusion detection systems for intra-vehicle networks: A review," *IEEE Access*, vol. 7, pp. 21266–21289, 2019.

[6] A. Taylor, N. Japkowicz, and S. Leblanc, "Frequency-based anomaly detection for the automotive CAN bus," in *Proc. World Congr. Ind. Control Syst. Secur.*, 2015, pp. 45–49.

[7] C. Ling and D. Feng, "An algorithm for detection of malicious messages on CAN buses," in *Proc. Nat. Conf. Inf. Technol. Comput. Sci.*, 2012, pp. 627–630.

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

[9] 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.*, 2016, pp. 61–68.

[10] S. Abbott-McCune and L. A. Shay, "Intrusion prevention system of automotive network can bus," in *Proc. IEEE Int. Carnahan Conf. Secur. Technol.*, 2016, pp. 1–8.

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

[12] T. Hoppe, S. Kiltz, and J. Dittmann, "Security threats to automotive can networks–practical examples and selected short-term countermeasures," in *Proc. Int. Conf. Comput. Saf., Rel., Secur.*, M. D. Harrison and M.-A. Sujan, Eds. Berlin, Germany: Springer, 2008, pp. 235–248.

[13] M. Levi, Y. Allouche, and A. Kontorovich, "Advanced analytics for connected cars cyber security," in *Proc. IEEE 87th Veh. Technol. Conf.* (VTC spring), 2018, pp. 1–7.

[14] 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 Secur.*, vol. 13, no. 8, pp. 2114–2129, Aug. 2018.

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

[16] H. M. Song, J. Woo, and H. K. Kim, "In-vehicle network intrusion detection using deep convolutional neural network," *Veh. Commun.*, vol. 21, 2020, Art. no. 100198.

[17] H.-C. Lin, P. Wang, K.-M. Chao, W.-H. Lin, and J.-H. Chen, "Using deep learning networks to identify cyber attacks on intrusion detection for in-vehicle networks," *Electronics*, vol. 11, no. 14, 2022, Art. no. 2180.

[18] 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. Analytics*, 2016, pp. 130–139.

[19] 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.

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

[21] K. Zhu, Z. Chen, Y. Peng, and L. Zhang, "Mobile edge assisted literal multi-dimensional anomaly detection of in-vehicle network using lstm," *IEEE Trans. Veh. Technol.*, vol. 68, no. 5, pp. 4275–4284, May 2019.

[22] D. Basavaraj and S. Tayeb, "Towards a lightweight intrusion detection framework for in-vehicle networks," *J. Sensor Actuator Netw.*, vol. 11, no. 1, 2022, Art. no. 6.

[23] Y. He, Z. Jia, M. Hu, C. Cui, Y. Cheng, and Y. Yang, "The hybrid similar neighborhood robust factorization machine model for CAN bus intrusion detection in the in-vehicle network," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 16833–16841, Sep. 2022.

[24] A. Taylor, "Anomaly-based detection of malicious activity in in-vehicle networks," Ph.D. dissertation, Ottawa-Carleton Inst. Electr. Comput. Eng., Université d'Ottawa/University of Ottawa, Ottawa, ON, Canada, 2017.
[25] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[26] J. Eckmann, S. O. Kamphorst, and D. Ruelle, "Recurrence plots of dynamical systems," *Europhysics Lett.*, vol. 4, no. 9, pp. 973–977, 1987.
[27] E. Garcia-Ceja, M. Z. Uddin, and J. Torresen, "Classification of recurrence plots' distance matrices with a convolutional neural network for activity recognition," *Procedia Comput. Sci*, vol. 130, pp. 157–163, 2018.
[28] A. Katok and B. Hasselblatt, *Introduction to the Modern Theory of Dynamical Systems*, vol. 54. New York, NY, USA: Cambridge Univ. Press, 1995.
[29] D. F. Silva, V. M. DeSouza, and G. E. Batista, "Time series classification using compression distance of recurrence plots," in *Proc. IEEE 13th Int. Conf. Data Mining*, 2013, pp. 687–696.
[30] S. Salvador and P. Chan, "Toward accurate dynamic time warping in linear time and space," *Intell. Data Anal.*, vol. 11, no. 5, pp. 561–580, 2007.
[31] N. Hatami, Y. Gavet, and J. Debayle, "Classification of time-series images using deep convolutional neural networks," in *Proc. 10th Int. Conf. Mach. Vis.*, 2018, Art. no. 106960Y.
[32] E. Seo, H. M. Song, and H. K. Kim, "GIDS: GAN based intrusion detection system for in-vehicle network," in *Proc. 16th Annu. Conf. Privacy Secur. Trust*, 2018, pp. 1–6.
[33] 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.
[34] T. P. Vuong, G. Loukas, D. Gan, and A. Bezemskij, "Decision tree-based detection of denial of service and command injection attacks on robotic vehicles," in *Proc. IEEE Int. Workshop Inf. Forensics Secur.*, 2015, pp. 1–6.
[35] M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim, "Do we need hundreds of classifiers to solve real world classification problems?," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 3133–3181, 2014.
[36] M. C. Belavagi and B. Muniyal, "Performance evaluation of supervised machine learning algorithms for intrusion detection," *Procedia Comput. Sci.*, vol. 89, pp. 117–123, 2016.
[37] L. V. D. Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.

OMAR Y. AL-JARRAH received the B.Sc. degree in computer engineering from Yarmouk University, Irbid, Jordan, in 2005, the M.Sc. degree in computer engineering from The University of Sydney, Sydney, NSW, Australia, in 2008, and the Ph.D. degree in electrical and computer engineering from Khalifa University, Abu Dhabi, UAE, in 2016. He has more than 12 years of combined academic and industrial experience. He is currently Assistant Professor at Jordan University of Science and Technology, Irbid, Jordan. His main research interests include machine learning, intrusion detection, big data analytics, autonomous and connected vehicles, unmanned aerial vehicles, and knowledge discovery in various applications.

KARIM EL HALOUI received the M.Sc. degree in electronic and electrical engineering from the Institut National des Sciences Appliquées of Lyon, Villeurbanne, France, in 2004, the MMath degree from the University of Oxford, Oxford, U.K., in 2013, and the Ph.D. degree from the University of Warwick, Coventry, U.K., in 2017. He has more than 15 years of combined industrial and academic experience. He is currently Assistant Professor of future mobility at the University of Warwick. His research interests include connected and intelligent vehicles, and more generally, applications of communication technologies and artificial intelligence on future mobility systems.

MEHRDAD DIANATI (Senior Member, IEEE) is currently the Head of the Intelligent Vehicles Research Department and the Technical Research Lead in the area of Networked Intelligent Systems with the Warwick Manufacturing Group, the University of Warwick, Coventry, U.K. His research interests include applying digital technologies (information and communication technologies and artificial intelligence) to develop connected and cooperative autonomous systems with a particular interest in their application in future mobility systems. He has more than 30 years of combined industrial and academic experience, with 20 years in various leadership roles in multi-disciplinary collaborative R&D projects. He works closely with the Automotive and ICT industries as the primary application domains of his research. He is also the Director of Warwick's Centre for Doctoral Training on Future Mobility Technologies, training doctoral researchers in the areas of intelligent and electrified mobility systems in collaboration with the experts in the field of electrification from the Department of Engineering of the University of Warwick. In the past, he was the Editor of the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY and several other international journals, including *IET Communications*. He is currently the Field Chief Editor of *Frontiers in Future Transportation*.

CARSTEN MAPLE (Member, IEEE) is currently the Director of the NCSC-EPSRC Academic Centre of Excellence in Cyber Security Research and Professor of cyber systems engineering at the University of Warwick, Coventry, U.K. He is also a Co-Investigator of the PETRAS National Centre of Excellence for IoT Systems Cybersecurity, where he leads on Transport and Mobility, and is a Fellow of the Alan Turing Institute, London, U.K. He has an international research reputation, and has authored or coauthored more than 350 peer-reviewed papers and being coauthor of the U.K. Security Breach Investigations Report 2010, supported by the Serious Organised Crime Agency and the Police Central e-crime Unit. His research has attracted millions of pounds in funding and has been widely reported through the media.