This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIA.2020.3009906, IEEE Transactions on Industry Applications

An Evolutionary Deep Learning Anomaly Detection Framework for In-Vehicle Networks - CAN Bus

Yubin Lin, Chengbin Chen, Fen Xiao, Omid Avatefipour, Khalid Alsubhi, and Arda Yunianta

Abstract— **Modern vehicles are no longer considered as mere mechanical-based device, instead, they have been hugely replaced by sophisticated electric devices known as Electronic Control Unit (ECU). These ECUs are communicating with each other by publishing/receiving messages that complies with well-known protocol called Control Area Network (CAN). CAN bus is responsible to ensure all critical parts in vehicle e.g. engine, braking, airbag deployment, steering wheel, acceleration, etc. are functioning properly. This indicates that CAN bus is considered as the back-bone network protocol in modern vehicles. Unfortunately, the CAN bus protocol is vulnerable to various cyberattacks due to the lack of security mechanism in the protocol which has introduced several attack surfaces and allows attackers to have legitimate access to the bus and launch malicious activities. This paper proposes a new effective security solution that can detect three different types of message injected attacks namely Denial of Service (DoS), fuzzy, and impersonation attacks in the CAN traffic based on deep learning model. Moreover, the proposed method makes the use of evolutionary optimization algorithm to avoid premature convergence and manual selection of deep learning network architecture. To assess the practicality and effectiveness of the proposed method, CAN traffic is logged using unmodified license vehicle. Furthermore, the proposed method is evaluated using two other different CAN traffic dataset that proves the proposed method can be applied for different car make and models.**

Index Terms—**Controller Area Network (CAN Bus), Anomaly Detection, Deep Learning, Deep Denoising Autoencoder, Optimization Algorithm**

I. INTRODUCTION

WITH the huge advancements in emerging technology, most of the critical and non-critical parts of modern vehicles are controlled by Electronic Control Unit (ECU) [1]. Automotive electric systems are considered as a heterogeneous distributed real-time systems and ECUs are performing their task using sophisticated software which has been flashed into their memory unit. ECUs are communicating with other nodes by publishing/receiving messages which complies with some well-known protocols, namely CAN, CAN Flexible Data-Rate (CAN FD), Local Interconnect Network (LIN), FlexRay, and Media Oriented Systems Transport (MOST) [2]. CAN Bus is the most famous and widely used protocol in the automotive industry and is considered the de-facto standard for vehicular networks. CAN bus not only have been utilized in automotive industry, but also it has range of applications in other industries, such as aerospace, agriculture, medical devices, and even in some home and commercial appliances [2].

Although there are other protocols with more security features available, e.g. Ethernet, they cannot entirely replace the CAN bus for in-vehicle network communication due to the following reasons: 1) The CAN bus is designed to be perfectly applicable for hard real-time environments and guarantees deterministic communication with minimal time latency. 2) In the CAN bus protocol, there is a method of prioritization where lower priority messages do not interfere with higher priority messages. For instance, a message transferring more critical function, such as engine control or airbag control, has more priority than a message for door or climate control. 3) The CAN bus protocol is used in all modern vehicles as the backbone of in-vehicle network communication; replacing this protocol entirely with another protocols requires re-designing the whole vehicle network architecture and a tremendous amount of changes in vehicle software, which runs based on the CAN protocol. Therefore, other protocols will not entirely replace the role and application of the CAN bus but rather augment the CAN bus.

The main challenge in CAN bus protocol is the lack of intrinsic security mechanism in the protocol design which can pave the way for attackers to penetrate into the network and launch malicious activities that can eventually endangers the life of driver and passengers. CAN bus protocol is invented by Robert Bosch GmbH [2] and during the design of CAN protocol, vehicles were not as complicated as nowadays in terms of internal and external communication with the outside environment. Also, since CAN bus is designed to work in environment that requires of real-time reliability with the existence of processing and memory constraints, adding security layers in the CAN protocol e.g. data encryption and message authentication may lead to creating delay in real-time communication and make them often inapplicable. This ease the work for attackers to penetrate into the network and launch malicious activities more easily than when other protocols, like the Transmission Control Protocol/Internet Protocol (TCP/IP), are used. Fortunately, with the advancement of machine learning techniques, this type of attack has been addressed by researchers [3]-[7] in such a way that any anomalous communication traffic activities can be detected and ignored.

Generally speaking, anomaly is an observation or occurrence

Yubin Lin is with State Grid Fujian Power Economic Research Institute, Fuzhou, 350012, CHN. (Email: 18606932711@163.com)
Chengbin Chen is College of Physics and Information Engineering, Fuzhou University, Fuzhou 350108, CHN. (***Corresponding Author***) (Email: M171110009@fzu.edu.cn, fzu-ccb@qq.com)
Fen Xiao is with State Grid Fujian Electric Power Co., Ltd., Fuzhou 350003 CHN. (Email: 12911429@qq.com)
Omid Avatefipour is with Department of Electrical and Computer Engineering, University of Michigan–Dearborn, Dearborn, MI 48128, USA. (Email: oavatefi@umich.edu)
Khalid Alsubhi is with Faculty of Computing and Information Technology King Abdulaziz University, Jeddah, Saudi Arabia. (Email: kalsubhi@kau.edu.sa)
Arda Yunianta is a Faculty of Computing and Information Technology in Rabigh, King Abdulaziz University, Jeddah, Saudi Arabia and Faculty of Engineering, Mulawarman University, Samarinda, Indonesia (Email: ayunianta@kau.edu.sa)

which deviates qualitatively from what is defined as normal, based on expert knowledge [8]. Due to the aforementioned security vulnerabilities existed in the CAN protocol, anomaly detection has become a very important topic in vehicular network security been investigated by researchers, industry, and governments, which has been studied within various research domains. In the scope of CAN protocol, anomaly detection is defined as an intelligent model that monitors communication traffic among ECUs and can be accurately detect any abnormal behavior in the traffic. Recently, machine learning algorithms have been utilized hugely in the area of anomaly detection. Designing intelligent model for intrusion detection systems (IDS) is one of the application of utilizing machine learning algorithms in the cybersecurity area. In particular, anomaly detection in automotive networks has also attracted the attention of researchers in this area, which is discussed more on in the Related Work section.

II. Related Work

To reinforce vehicle security and ensure driver and passengers safety, several security mechanisms and solutions are being actively developed and investigated in both academia and industry, which reflects the fact that this topic has been considered as one of the most critical issues to governments industry, and academia. Recently, numerous research studies have been carried out to expose the vehicular network vulnerabilities and different solutions have been proposed to detect attack on these.

Carsten et al. [9] discussed the vulnerabilities of in-vehicle networks such as lack of integrity in CAN protocol as the CAN messages do not carry the address of sender/receiver. In addition, they discussed lack of message authentication in CAN protocol which resulted in paving the way for attackers to launch spoofing attacks. Authors provided some solutions for attack prevention which focus on data management, monitoring algorithm approach, attestation of node identification, etc.

Hoppe et al. [10] launched four different attacks on different modules in vehicle such as control of electric vehicle window, warning lights, airbag control systems and central gateway. They were able to open the car window automatically when it reaches the speed of 200 (kph) by adding few lines of malicious code in an appropriate ECU. Researchers also could manipulate the airbag module in which the module will be removed from the system and leads to unexpected consequences during the car accident (air bog does not work in emergency cases). Last attack was launched on gateway ECU that interconnect to several internal sub-network or even external ones. This attack can lead to eavesdropping of sensitive internal information by attacker. On the other side, researchers also proposed different countermeasure techniques to detect any abnormal behavior in CAN-Bus which can prevent hackers penetrating to the vehicle internal bus.

Authors in [11] evaluate an anomaly detector for CAN bus protocol by comparing the current and historical packet timing to yield an anomaly signal by measuring inter-packet timing over a sliding window. Authors validated the effectiveness of their proposed method within the confines of injection type, duration range, and frequency of CAN messages.

Marchetti et al. [12] proposed anomaly detection algorithm to identify anomalies in the sequence of CAN messages by using a transition matrix defined based on reiterative CAN ID pattern sequence. Initially, the matrix populated with "False" value (abnormal) and if the combination of CAN ID is correct, it will be replaced by "True" value to populate with the valid values.

Authors in [13] introduced a method for intrusion detection system by leveraging the fact of randomness occurring in the CAN communication. This randomness which can be utilized by the information-theoretic measure, e.g. entropy. The entropy can describe how much information-content has been transferred by a given message in CAN-Bus. The normal value of entropy is learnt by recording the CAN-Bus traffic in normal operation mode. During launching an attack, the level of randomness will be decreased and as a result will alter the entropy value.

Another anomaly detection method for CAN bus which proposed by Taylor et al. [14] is based on long short-term memory neural network to detect anomaly attack. Detector can identify the attack by learning to predict the next data word originating from each sender on the bus and the error is used as detection signal. If the subsequent CAN message exists with higher bit error compared to the normal range, it would be considered as anomalous behavior.

Ansari et al. [15] propose anomaly detection mechanism to address the masquerade and replay attacks in CAN bus using the principle of self-identifier violation. In CAN protocol remote frame is used when receiver needs a certain type of information to run a given task. To this end, receiver broadcast remote frame on the bus in which the identifier of the target ECU is normally included. Assume Y is the CAN ID of a given node. If any frame that is not remote frame received from another node with the same CAN ID Y is considered as masquerade attack. The proposed IDS method was implemented as part of the CAN controller on a test-bed with synthetic vehicle behavior.

Lee et al. [16] introduced anomaly detection method based on the analysis of the offset ratio and time interval between request and response message in CAN. The underlying idea behind their proposed method was benefiting from *remote frame* feature in CAN. If *remote frame* transmitting a given identifier, receiver node should respond to that remote frame immediately. In normal operation, each node has a fixed response offset ratio and time interval. However, during attack, these values can be different. For performance evaluation, authors developed their idea with Raspberry Pi 3 with PiCAN2 shield.

Martinelli et al. [17] modeled the normal CAN traffic behavior by fuzzy technique. They applied four fuzzy classification models to discriminate between the legitimate CAN messages which is generated as a result of human action compared to the malicious messages which are injected as a result of attack. They employed 8 specific bytes in a given CAN message as the feature vector. Their fuzzy NN algorithm's accuracy ranging from 0.963 to 1 for injection attack.

Pajic et al. [18] developed an attack-resilient state estimator, which functions in the presence of sensor noise. They demonstrated this on an automatic cruise-control for a ground vehicle. The downside of this approach is that their system

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIA.2020.3009906, IEEE Transactions on Industry Applications

requires a model of how its components interact. In a vehicular context, this is hard to devise due to many factors outside of our control.

The authors of [19] presented an analysis of CAN broadcasts and subsequent testing of statistical methods to detect timing changes in the CAN traffic that were indicative of some of the predicted attacks. In the simulated attacks, they used two unsupervised methods, namely Autoregressive Integrated Moving Average (ARIMA) and Z-score, as their anomaly detection method. Even though their method requires no prior knowledge about normal activity as reference, the results indicated that not all traffic behavior can be stochastically modeled and their detection performance degraded in lower priority packets.

Recently, machine learning algorithms namely deep learning, support vector machine, fuzzy logic, and artificial neural network [20-22] have been successfully employed to solve various complicated problems in different areas such as cybersecurity [23], medical imaging [24], autonomous vehicles [25], image classification [26], electric load demand forecasting [27], and social media [28].

According to the above discussion provided in the area of CAN bus anomaly detection, this paper proposes a novel model in this field based on using deep learning method (deep denoising autoencoder) and ecogeography-based optimization algorithm. The high accuracy detection and robustness of the proposed model is examined using experimental data gathered from an unmodified licensed vehicle. Moreover, to manifest that the suggested method can be applicable with different vehicles irrespective of the car make and model, we examined the proposed method using two other popular datasets in the area of CAN bus anomaly detection. In this study, three different attacks namely Denial of Service (DoS) attack, fuzzy attack, and impersonation attack have been injected to assess the performance of the introduced method against them. The contributions of this study are summarized as follows:

1. Implementing a novel framework for anomaly detection in CAN bus traffic using deep denoising autoencoder machine learning algorithm and utilizing ecogeography-based optimization algorithm. The optimization algorithm aims to adjust the anomaly detection model parameters optimally for maximizing the efficiency and performance of the model against cyberattacks
2. Assessment of the introduced anomaly detection method on three different real datasets. This greatly helps to manifest the independency of the proposed method from the varied datasets gathered by specific car manufacturers. Moreover, three different message injection attacks namely Denial of Service (DoS), fuzzy attack, and impersonation attack have been applied to evaluate the efficiency and robustness of the method against them.

The rest of this paper is organized as follows: 1) section III provide a preliminary overview for CAN Bus protocol in general. Section IV discusses the introduced deep learning anomaly detection model. Section V provides the simulation results and experimental evaluation. Finally, the paper concludes in Section VI.

III. CAN Bus Protocol – BACKGROUND

Generally, there are two formats of CAN-Bus: standard format which has 11-bit for identifier and extended-format which includes 29-bit identifier frame. Data Frame, Remote Frame, overload frame, and error frame are four major frame types in controlled area network (CAN-Bus). Data Frame is used to carry the data from a transmitter to a receiver, which consists of the following bit fields: start of frame, (one dominant bit), arbitration field which consists of 12 bits, control field which has 6 bit, and data field (in range of 0 to 64 bytes), CRC field (16-bit), ACK field (2-bit), and End of Frame (7-bit) [2]. The complete illustration of data frame is shown in Fig. 1.

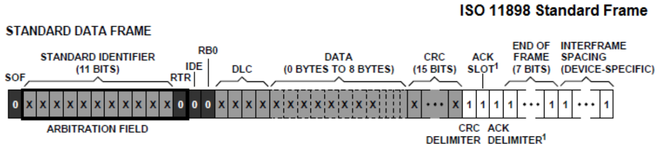

ISO 11898 Standard Frame
STANDARD DATA FRAME
SOF
STANDARD IDENTIFIER (11 BITS)
RTR
IDE
RB0
DLC
DATA (0 BYTES TO 8 BYTES)
CRC (15 BITS)
ACK SLOT1
END OF FRAME (7 BITS)
INTERFRAME SPACING (DEVICE-SPECIFIC)
0 X X X X X X X X X X X 0 0 0 X X X X X X X X X X X X X ••• X 1 1 1 1 ••• 1 1 ••• 1
ARBITRATION FIELD
CRC DELIMITER
ACK DELIMITER1

Fig. 1. CAN Bus Data Frame.

Bit stuffing technique is used in CAN-Bus which indicates that if there are six consecutive identical bits transmitted in the bus, it is considered as an error because bit stuffing law is violated [2]. Bit stuffing can be applied in different frames in CAN-Bus e.g. arbitration field, control field, and CRC field which means a complementary bit will be added to the frame when the transmitter finds that there are five identical bits consecutively. Therefore, six consecutive identical bits during the transmission is considered as bit-stuffing violation and error frame will be transmitted by each node which detects this situation.

CAN-Bus consists of three major layers namely physical layer, transfer layer, and object layer. Physical layer includes the actual bit transfer between different nodes and the electrical properties of transmission and also the medium which is used for communication [29].

IV. Proposed CAN Bus Anomaly Detection Model

This section explains the intrinsic fact how the proposed method can be applicable to the CAN bus traffic and more theoretical aspects of the suggested method will be elaborated as well. In CAN traffic, significant number of messages are transmitting in a periodic fashion followed by a specific message. This periodicity can be leveraged to identify the frequency patterns of each message transferring in the CAN traffic and any deviation from these normal traffic patterns can be triggered as abnormal behavior consequently and the ultimate goal of the proposed method is to detect those deviations with high accuracy. The suggested method contains two main phases as follows: Training phase and testing/validation phase. In the training phase, which is performed off-line, each CAN packet is trained to extract sets of features that can distinguished the underlying statistical properties between normal and abnormal traffic data. For this purpose, deep neural network (here deep denoising autoencoder) has been utilized for statistical patterns training in order to provide effective classification. In the detection phase, the same sub-set of features in the training phase will be

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIA.2020.3009906, IEEE Transactions on Industry Applications

extracted from the incoming CAN traffic and the proposed deep learning model will play role as anomaly detection. Although, deep learning methods have achieved huge success, its efficient application on new problems may be still questionable due to the issues such as premature convergence or the optimal selection of network structure. Hence, we also proposed an evolutionary optimization algorithm for training the deep learning more efficiently. The conceptual illustration of the proposed method is shown in Fig. 2 Each of the aforementioned components will be discussed in the following sections.

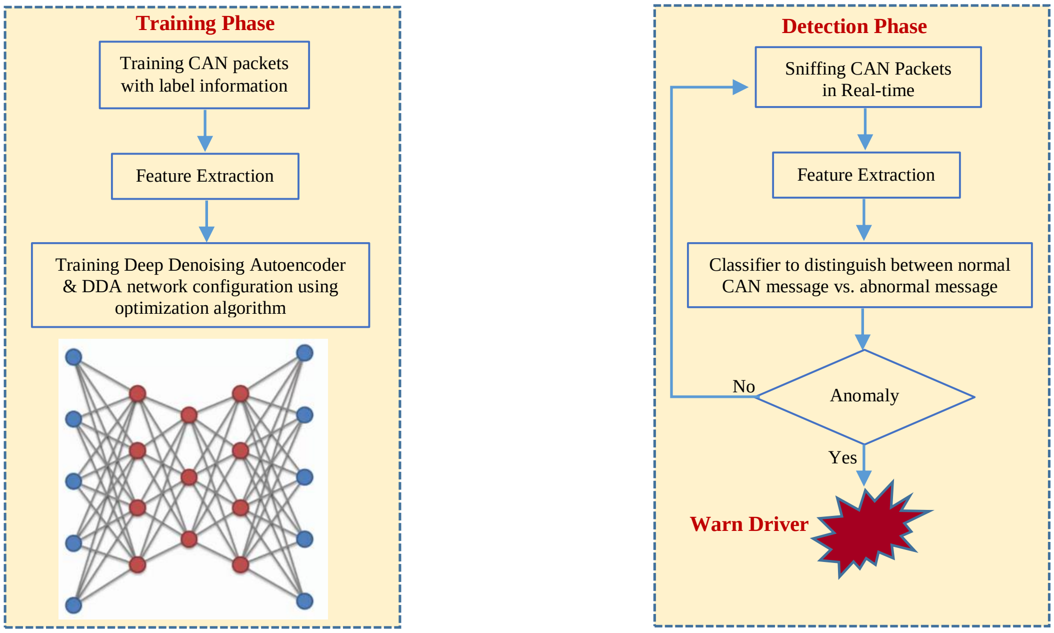

Training Phase
Training CAN packets with label information
Feature Extraction
Training Deep Denoising Autoencoder & DDA network configuration using optimization algorithm
Detection Phase
Sniffing CAN Packets in Real-time
Feature Extraction
Classifier to distinguish between normal CAN message vs. abnormal message
Anomaly
No
Yes
Warn Driver

Fig. 2. Block Diagram of CAN bus anomaly detection platform

A. *Deep Denoising Autoencoder DAE*

Autoencoder is considered as a particular type of Artificial Neural Network (ANN) that consists of one single layer that aims to reconstruct the input $x \in [0,1]^D$ at the output [30]. Input layer encoded to a hidden layer $y \in [0,1]^{D'}$ using following mapping function:

$$f_\theta(x) = s(Wx + b) \tag{1}$$

In above function, $\theta = [W, b]$ and W and b are defined as weight matrix with $D' \times D$ dimension and bias matrix with D' dimension, respectively. Encoded hidden layer then will be decoded to reconstruct to $z \in [0,1]^D$ in input space albeit with updated parameters as $\theta = [W, b]$. So, the output is defined as follows:

$$g_{\theta'}(x) = s(W'x + b') \tag{2}$$

The ultimate goal of training phase in autoencoder is to minimize the average of reconstruction error that formulated as follows:

$$\varsigma(x, z) = \|x - z\|^2 \tag{3}$$

Denoising autoencoder is an extension of conventional autoencoder that aims to reconstruct the original input layer from a manipulated one. As it is depicted in Fig. 3 , deep autoencoder first manipulate the initial input layer x into $\tilde{x}$ using the mapping function and then routed the manipulated input $\tilde{x}$ to the hidden layer form $y = f_\theta(\tilde{x}) = s(W\tilde{x} + b)$ that will eventually reconstruct the output layer as $z = g_{\theta'}(x) = s(W'y + b')$.

The ultimate goal of deep autoencoder is to minimize the average reconstruction error throughout a training set.

$$\arg\min_{\theta,\theta'} = \frac{1}{|D|} \sum_{x \in D} \varsigma(x, g_{\theta'}(f_{\theta'}(\tilde{x}))) \tag{4}$$

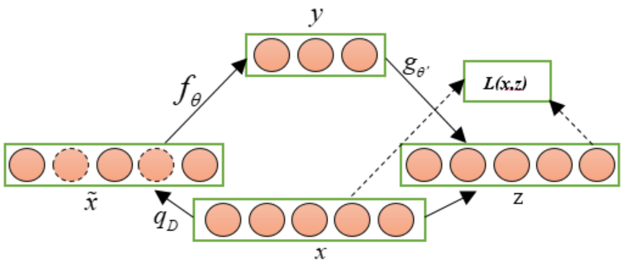
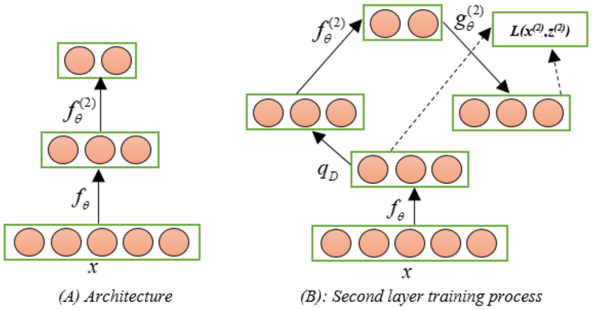


Fig. 3. Denoising Autoencoder.



(A) Architecture

(B): Second layer training process

Fig. 4. Architecture and training of the 2nd layer for deep denoising autoencoder

Deep denoising autoencoder is a special type of DAE which consists of multiple hidden layer in which every layer is responsible to capture the higher-order correlations between the hidden nodes in the layer below [31]. In the deep denoising autoencoder, the input manipulation is only applied for the purpose of initial denoising-training for each layer; Having performed the mapping function in the learning phase, it will be utilized for intact inputs. There is no manipulation employed to generate the representation which will utilize an intact input to train the next layer. An example of a two-layer deep denoising autoencoder is shown in Fig. 4 in which the second layer has been trained in a way that following objective function can be optimized:

$$\arg\min_{\theta^2,\theta'^2} = \frac{1}{|D|}\sum_{x\in D}\varsigma(x^{(2)}, g_{\theta'}^{(2)}(f_{\theta}^{(2)}(\tilde{x}^{(2)}))) \tag{5}$$

In above objective function, $\tilde{x}^{(2)}$ is the manipulated version of original input $x^{(2)}$ to the second layer. After completing the denoising (pre-training) process, which is unsupervised learning, the deep denoising autoencoder is utilized to initialized a deep neural network for the supervised learning on the CAN traffic dataset. The output of deep neural network will be generated using following formula:

$$y = \sum_{i=1}^{D^{(L)}} w_i y_i^{(L)} \tag{6}$$

Where $D^{(L)}$ states as the dimension of the layer.

B. An ecogeography-based learning algorithm

As discussed earlier, even though deep learning method has achieved promising results in different applications, it can face some issues such as premature convergence and optimum network construction in some new problems. In this section, an evolutionary-based optimization algorithm known as ecogeography-based optimization (EBO) [32] is employed for the training of the deep denoising autoencoder. In EBO, each solution *x* has a corresponding immigration rate $\lambda(x)$ and an emigration rate $\mu(x)$ that are formulated as follows:

$$\lambda(x) = I\left(\frac{f_{\max} - f(x) + \varepsilon}{f_{\max} - f_{\min} + \varepsilon}\right) \tag{7}$$

$$\mu(x) = E\left(\frac{f(x) - f_{\min} + \varepsilon}{f_{\max} - f_{\min} + \varepsilon}\right) \tag{8}$$

In which, $f(x)$ defines as the fitness function of *x*, $f_{\min} and f_{\max}$ are defined as the minimum and maximum fitness in the population, respectively. Immigration and emigration rates are denoted as *I* and *E*, respectively. These two rates are normally set to 1 and ε is considered as a constant number to prevent zero-division error condition.

This optimization algorithm is employed for each layer in the deep denoising autoencoder. For *l* layer's optimization process, the output is generated as a variable-size vector as $W = \{w_1, ..., w_d, ..., w_D\}$. In this vector, D indicates the number of neurons in the current layer, $w_d = \{w_{d,1}, ..., w_{d,k}, ...w_{d,D^{l-1}}, b_d\}$ is defined as a sub-vector which populates according to the connections between neuron *d* and other neurons in the layer below and b_d denotes as bias factor for the neuron *d*. At the beginning, the optimization algorithm populates solution in a random fashion and then progress the solution gradually by migrating features from high-quality generated solutions to the low-quality ones. During each generation, every element $w_{d,k}$ contains a probability for $\lambda(W)$ that needs to be adjusted using one of the following operations:

$$w_{d,k} = w_{d,k} + a(w_{d,k}^N - w_{d,k}) \tag{9}$$

$$w_{d,k} = w_{d,k}^F + a(w_{d,k}^N - w_{d,k}) \tag{10}$$

In above operations, *a* is denotes as coefficient ranging from 0 to 1. W^N, W^F is defined as neighbor and non-neighbor of a *W*, respectively. In the context of ecogeography-based learning algorithm, above operations are known as local migration and global migration, respectively. The probability of each migration to be local migration is formulated as follows:

$$\eta = \eta_{\min} + \frac{g}{g_{\max}}(\eta_{\max} - \eta_{\min}) \tag{11}$$

$g_{\max}$ is defined as maximum generation number, $\eta_{\min}, \eta_{\max}$ is stated as the lower and upper limit of variable η, respectively. As a result, the optimization algorithm will have a tendency to carry out more global migration to have better outcome for the exploration in the early phases and also will carry out more local migration in order to achieve improved exploitation in later phases.

V. Results And Discussions

Having discussed the underlying idea behind the proposed method and also explained the main intelligent components in the method, this section presents simulation results according to the CAN traffic logged from a licensed unmodified vehicle. Logged CAN traffic is collection of text files that contains message occurrence timestamp, message ID, and message data field. Each dataset of CAN traffic is divided into three sub-sets as: training set (70% of the dataset), validation set (10% of the dataset, aims to prevent training overfitting), and testing set (20% of the dataset – separated into normal traffic and imitated attacked traffic). It is worth nothing that although CAN bus protocol specification is available to public, message ID, data field content, message frequencies for each vehicle is unique and is not available to public due to the car manufactures ownership. However, the advantage of the introduced method in this paper is its independency from the message IDs and data content. This means that the proposed method does not depend on any specific car manufacturer and can be applied to any CAN traffic traces. For this purpose, we collected CAN traffic from a licensed vehicle using OBD-II connection and VN1630A CAN network interface device. The normal driving scenarios were considered while capturing the CAN traffic in which the car drove away from a normal residential driveway and pulled up on the street. After three right hand turns, the car was backed up into a parking space.

Having captured the CAN traffic from vehicle, some pre-processing processes are performed before import it to the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIA.2020.3009906, IEEE Transactions on Industry Applications

proposed method. These processes include: 1) feature reduction, which unnecessary fields are excluded such as data load and CRC field. 2) feature conversion, which hexadecimal data is converted to the decimal data because deep learning based model cannot process string data. 3) feature normalization, the data is scaled using absolute normalization.

we also assess the performance of the proposed method using two other different datasets that one of them is simulated and generated by CANoe software and the other CAN traffic dataset, which is available online [33], is generated by capturing CAN traffic via the OBD-II port from a real vehicle while message injection attacks were performing, e.g. Denial of Service (DoS) Attack in which messages of '0x000' CAN ID were injected in a short cycle. The dataset contains 2,369,868 attack free states CAN message and 656,579 messages where DoS attacked was launched.

In order to evaluate the performance of the proposed method, three different attacks have been launched as follows: Denial of Service (DoS) attack, fuzzy attack, and impersonation attack.

DoS Attack:

In this attack scenario, attackers try to inject highest priority identifier e.g. `0x000` on the bus to gain the dominant state in the CAN and does not allow other nodes to publish their data on the bus. Since CAN bus protocol use a single bus and all nodes need to publish/subscribe to the same bus, increasing occupancy of the bus by a single mode can create unacceptable time latencies for other nodes which can lead to system failure.

Fuzzy Attack:

In this attack scenario, attackers will inject messages with manipulated data field with an arbitrary data to the spoofed message IDs which lead to unexpected behavior of vehicle e.g. instrument panel links constantly, steering wheel shakes significantly, gear changes abruptly, etc.

Impersonation Attack:

In this attack scenario, attackers can stop message transmission by controlling the target node and can manipulate an impersonating node. For instance, attackers can intentionally manipulate the vehicle components, which relates to the RPM and gear, to a given constant value e.g. 0x000 or 0xFFF. Impersonation attack resulted in making legitimate RPM and gear component in the vehicle experiencing abnormal behavior.

TABLE I
CAN IDENTIFIER AND FREQUENCIES

CAN Identifier	Frequency
6FF	101.010101
308	85.74311927
340	50
2A0	48.7804878
670	99.00990099
3F0	100.1666667
D21	38.7804878
210	51.02040816
238	108.6956522
410	93.45794393
200	61.02040816
A7F	49.01960784
B61	10
212	68.54368932
240	78.01010101
4EB	113.6363636
2C1	110.3595506
312	50
5AE	80.01960784

From capturing CAN traffic in the vehicle while normal driving, it is observed that each message occurrence has its own frequency pattern in the logged file. The proposed anomaly detection method leverages this fact during the training phase. Any deviation from the pre-defined message occurrence frequency can then be detected by this method for DoS attacks. Table I provides some of the message IDs and their corresponding frequency in the captured CAN traffic.

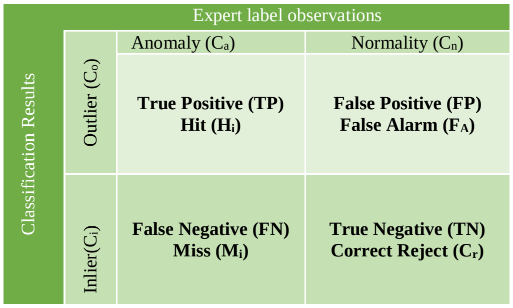
Table II summarized number of unique message ID in each dataset, number of packets in training and testing set, and number of injected packets.

TABLE II
CAN TRAFFIC DATASET DETAILS

Dataset #	Attack Type	Training Packets	Testing Packets	
		Normal Packets	Normal Packets	Injected Packets
Dataset #1	DoS Attack	1,220,341	615,490	598,821
	Fuzzy Attack	1,220,341	615,490	522,730
	Impersonation Attack	1,220,341	615,490	570,000
Dataset #2	DoS Attack	1,658,907	710,961	656,579
	Fuzzy Attack	1,658,907	710,961	591,990
	Impersonation Attack	1,658,907	710,961	659,990
Dataset # 3	DoS Attack	950,000	156,000	95,189
	Fuzzy Attack	950,000	156,000	89,900
	Impersonation Attack	950,000	156,000	90,100

A. Performance Evaluation of proposed Anomaly Detection algorithm

Making comparison between the proposed algorithm predicted results {outlier, inlier} and the real-world observation labels {anomaly, normality} is an essential step toward proving the efficiency and robustness of the anomaly detection method. For this purpose, confusion matrix is mainly utilized for performance evaluation, which represents the four possible results when any actual data point given by an expert compared with its corresponding data point resulted by a given classification algorithm. Four possible outcomes can happen as follows: true positive (*TR*) or hit (*Hi*), false positive *(FP)* or false alarm (*FA*), false negative (*FN*) or miss (*Mi*), and true negative *(TN)* or correct reject (*CR*). If any message in the CAN traffic is labeled as an anomaly or injected message and the anomaly detection algorithm can detect that message as anomalous message, the outcome in the confusion matrix considered as true positive *(TR)* or "hit.*"* In case that proposed anomaly detection model can detect a normal message in the CAN traffic as a normal message as well, the result in the confusion matrix will be true negative *(TN)* or "correct reject." If a given CAN message is a malicious one in reality but the anomaly detection algorithm detects that message as normal, it will be categorized as false negative *(FN)* or "miss". Similarly, if a CAN message in the CAN traffic is a normal packet but the anomaly detection algorithm wrongly detects it as a malicious message, the result in confusion matrix is labeled as false positive *(FP)* or "false alarm". Fig. 5 shows the four possible outcomes of the confusion matrix namely true positive, false positive, false negative, and true negative, graphically.



		Expert label observations	
Classification Results		Anomaly (C_a)	Normality (C_n)
	Outlier (C_o)	**True Positive (TP)** **Hit (H_i)**	**False Positive (FP)** **False Alarm (F_A)**
	Inlier(C_i)	**False Negative (FN)** **Miss (M_i)**	**True Negative (TN)** **Correct Reject (C_r)**

Fig. 5. Confusion matrix for performance measurement

The four confusion matrix outcomes have four associated performance rates, as follows:

$$Hit\ Rate = \frac{|H_i|}{|C_A|} \tag{12}$$

$$False\ Alarm\ Rate = \frac{|F_A|}{|C_N|} \tag{13}$$

$$Miss\ Rate = \frac{|M_i|}{|C_A|} \tag{14}$$

$$Correct\ Reject\ Rate = \frac{|C_R|}{|C_N|} \tag{15}$$

Furthermore, mean square error (MSE) and mean absolute error (MAE) are utilized as error metrics when comparing between actual data and predicted data. MSE can determine the average of squared difference between actual data and predicted errors and MAE can calculate the average absolute difference between actual data and predictions. These two error metrics can be formulated as follows:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{16}$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{17}$$

Where N is the number of data points, y_i represents the observed values and $\hat{y}_i$ represents predicted values. Table III shows the four performance measurement rates in confusion matrix and two error metrics for the suggested method on three different attack scenarios.

TABLE III - CONFUSION MATRIX AND ERROR METRICS FOR THREE DIFFERENT ATTACKS

Dataset #1	Hit Rate	Miss Rate	False Alarm Rate	Correct Reject Rate	MSE	MAE
DoS Attack	98.20%	1.8%	5.23%	94.77%	98.12%	98.30%
Fuzzy Attack	96.30%	3.7%	7.33%	92.67%	91.03%	91.50%
Impersonation Attack	97.10%	2.9%	6.54%	93.45%	97.14%	97.30%
Dataset #2	Hit Rate	Miss Rate	False Alarm Rate	Correct Reject Rate	MSE	MAE
DoS Attack	96.12%	3.88%	5.30%	94.70%	96.6%	96.90%
Fuzzy Attack	89.10%	10.9%	12.89%	87.11%	90.88%	90.5%
Impersonation Attack	94.88%	5.12%	6.70%	93.30%	98.30%	98.60%
Dataset #3	Hit Rate	Miss Rate	False Alarm Rate	Correct Reject Rate	MSE	MAE
DoS Attack	95.80%	4.20%	6.23%	93.77%	99.45%	99.8%
Fuzzy Attack	90.90%	9.10%	10.40%	89.60%	93.12%	93.20%
Impersonation Attack	96.17%	3.83%	8.72%	91.28%	96.6%	96.72%

As it is shown in Table III the performance of proposed anomaly detection is degraded when tested on fuzzy attack. This may stem from the differences between normal and abnormal messages are marginal compared with the actual data. This reflects the fact that fuzzy attack contains complete random and complex patterns that mostly resemble the normal patterns.

In addition, the suggested anomaly detection method is compared with other well-known machine learning algorithms such as ANN (with 2 hidden layer), decision trees, and k-nearest neighbors in terms of precision, recall, F1-score, and error rate. Table IV summarized the detection performance of the suggested method compared to the aforementioned ML algorithm. Precision is defined as the fraction of actual packets among the packets detected as attacks and can be formulated as follows:

$$precision = TP/(TP + FP) \tag{18}$$

High precision rate indicates that the false positive rate of the anomaly detection method is low. Having low false positive rate is essential because at the end, users do not want to experience frequent false alarms for attack detection. Recall is defined as a fraction of the correctly detected attack packets and can be formulated as:

$$recall = TP/(TP + FN) \tag{19}$$

F1-score shows a balance between precision and recall. This score is normally utilized to quantify the performance of classification when a given dataset contains uneven class distribution. F1-score can be formulated as follows:

$$F1 = 2\times(precision \times recall)/(precision + recall) \tag{20}$$

Error rate is defined as the fraction of incorrectly classified packages and can be formulated as follows:

$$ErrorRate = (FN + FP)/(TP + TN + FP + FN) \tag{21}$$

TABLE IV - DETECTION PERFORMANCE OF DIFFERENT ML ALGORITHM

DoS Attack	**Error Rate**	**Precision**	**Recall**	**F1-score**
Deep Denoising Autoencoder	0.03%	1.0	0.9820	0.9909
ANN (with 2 hidden layer)	0.08%	0.9993	0.9444	0.9710
k-Nearest Neighbors (k=5)	0.25%	0.9810	0.8901	0.9332
Decision Trees	1.12%	0.9790	0.8681	0.9202
Fuzzy Attack	**Error Rate**	**Precision**	**Recall**	**F1-score**
Deep Denoising Autoencoder	0.12%	0.9995	0.9630	0.9809
ANN (with 2 hidden layer)	0.71%	0.9832	0.8990	0.9392
k-Nearest Neighbors (k=5)	24.40%	1.0	0.3461	0.5142
Decision Trees	6.41%	0.9550	0.8419	0.8948
Impersonation Attack	**Error Rate**	**Precision**	**Recall**	**F1-score**
Deep Denoising Autoencoder	0.04%	0.9999	0.9710	0.9848
ANN (with 2 hidden layer)	0.13%	0.9899	0.9124	0.9495
k-Nearest Neighbors (k=5)	0.72%	0.9821	0.9009	0.9397
Decision Trees	1.81%	0.9770	0.8899	0.9314

According to the results provided in table IV, it can be observed that the deep denoising autoencoder algorithm outperforms the other ML algorithms. Comparing the conventional machine learning algorithms, ANN has better detection performance in DoS and impersonation attacks. From attack's complexity standpoint, fuzzy attack is the most complicated attack due to the injection of random message compared to DoS attack where specific messages injected. Hence, the performance of anomaly detection algorithms may degrade for this attack compared with the other attacks e.g. DoS and impersonation attack. It is also observed that k-Nearest Neighbors (kNN) algorithm have satisfactory performance for precision in fuzzy attack scenario. However, it has low recall which eventually lead to have low F1-scores. This indicates that the false alarm rate is very low but these classification algorithms are biased toward the normal CAN packets. Also, it is appeared that k-Nearest Neighbors classifier shows a satisfactory outcome for DoS and impersonation attacks but it failed in fuzzy attack detection and classified majority of data as normal. In fact, Fuzzy attack data are created randomly and scattered; since the kNN is a clustering based algorithm, it seems that kNN algorithm could not create proper clusters.

VI. Conclusion

This research study introduced an effective anomaly detection framework that learns sequential patterns existed in the CAN bus traffic with the goal to detect anomaly messages based on traffic behavior change. The proposed framework constructed based on particular type of deep learning known as deep denoising autoencoder. In addition, an evolutionary-based optimization algorithm known as ecogeography-based optimization (EBO) is integrated with the deep denoising autoencoder to avoid premature convergence and promoting the optimum network learning structure. The proposed intrusion detection framework is employed to identify injected malicious messages in CAN traffic. The underlying idea behind the proposed method is to establish a model based on the normal CAN bus traffic, which contains recurring patterns in message IDs that are transmitted in a given normal traffic. Three different categories of message injection attacks, which are likely to be launched in a vehicle environment connected to the external networks, namely DoS, fuzzy, and impersonation attacks are evaluated. The experimental results indicated that the proposed deep denoising autoencoder method outperforms the other machine learning models on three different CAN traffic datasets by achieving highest hit rate and lowest miss rate. It manifested that the proposed method is reliable and robust intrusion detection method to detect malicious injected messages in CAN traffic. From a cyber-resilience point of view, the proposed model can provide a highly secure and accurate model to prevent vehicles from being harmed by attackers. Last but not least, the proposed MBA could show high search ability and convergence characteristics, making it a good algorithm for optimization applications.

I. Acknowledgement

This project was funded by the Deanship of Scientific Research (DSR), King Abdulaziz University, Jeddah, under grant No. (D-700-611-1441). The authors, therefore, gratefully acknowledge DSR technical and financial support.

References

[1] Markovitz, M., & Wool, A. (2017). Field classification, modeling and anomaly detection in unknown CAN bus networks. *Vehicular Communications*, *9*, 43-52.

[2] Specification, C. A. N. (1991). Bosch. *Robert Bosch GmbH, Postfach*, *50*.

[3] Mo, X., Chen, P., Wang, J., & Wang, C. (2019, April). Anomaly Detection of Vehicle CAN Network Based on Message Content. In *International Conference on Security and Privacy in New Computing Environments* (pp. 96-104). Springer, Cham.

[4] M. Dabbaghjamanesh, A. Kavousi-Fard, and J. Zhang. "Stochastic Modeling and Integration of Plug-In Hybrid Electric Vehicles in Reconfigurable Microgrids With Deep Learning-Based Forecasting." IEEE Transactions on Intelligent Transportation Systems (2020).

[5] Marchetti, M., Stabili, D., Guido, A., & Colajanni, M. (2016, September). Evaluation of anomaly detection for in-vehicle networks through information-theoretic algorithms. In *2016 IEEE 2nd International Forum on Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI)* (pp. 1-6). IEEE.

[6] Ni, K. Y., & Payton, D. W. (2019). *U.S. Patent No. 10,484,411*. Washington, DC: U.S. Patent and Trademark Office.

[7] Avatefipour, O., Hafeez, A., Tayyab, M., & Malik, H. (2017, December). Linking received packet to the transmitter through physical-fingerprinting of controller area network. In *2017 IEEE Workshop on Information Forensics and Security (WIFS)* (pp. 1-6). IEEE.

[8] Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM computing surveys (CSUR)*, *41*(3), 1-58.

[9] M. Dabbaghjamanesh, A. Kavousi-Fard, and S. Mehraeen. "Effective scheduling of reconfigurable microgrids with dynamic thermal line rating." IEEE Transactions on Industrial Electronics 66, no. 2 (2018): 1552-1564.

[10] Hoppe, T., Kiltz, S., & Dittmann, J. (2011). Security threats to automotive CAN networks—Practical examples and selected short-term countermeasures. *Reliability Engineering & System Safety*, *96*(1), 11-25.

[11] Taylor, A., Japkowicz, N., & Leblanc, S. (2015, December). Frequency-based anomaly detection for the automotive CAN bus. In *2015 World Congress on Industrial Control Systems Security (WCICSS)* (pp. 45-49). IEEE.

[12] Marchetti, M., & Stabili, D. (2017, June). Anomaly detection of CAN bus messages through analysis of ID sequences. In *2017 IEEE Intelligent Vehicles Symposium (IV)* (pp. 1577-1583). IEEE.

[13] Müter, M., & Asaj, N. (2011, June). Entropy-based anomaly detection for in-vehicle networks. In *2011 IEEE Intelligent Vehicles Symposium (IV)* (pp. 1110-1115). IEEE.

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

[15] Ansari, M. R. (2016). *Low-cost approaches to detect masquerade and replay attacks on automotive Controller Area Network* (Doctoral dissertation, University of New Hampshire).

[16] Lee, H., Jeong, S. H., & Kim, H. K. (2017, August). OTIDS: A novel intrusion detection system for in-vehicle network by using remote frame. In *2017 15th Annual Conference on Privacy, Security and Trust (PST)* (pp. 57-5709). IEEE.

[17] Martinelli, F., Mercaldo, F., Nardone, V., & Santone, A. (2017, July). Car hacking identification through fuzzy logic algorithms. In *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)* (pp. 1-7). IEEE.

[18] M. Dabbaghjamanesh, A. Kavousi-Fard, and Z. Dong. "A novel distributed cloud-fog based framework for energy management of networked microgrids." IEEE Transactions on Power Systems (2020).

[19] Tomlinson, A., Bryans, J., Shaikh, S. A., & Kalutarage, H. K. (2018, June). Detection of automotive CAN cyber-attacks by identifying packet timing anomalies in time windows. In *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)* (pp. 231-238). IEEE.

[20] Samani, Z. R., & Shamsfard, M. (2011). A fuzzy ontology model for qualitative spatial reasoning. In *2011 6th International Conference on Computer Sciences and Convergence Information Technology (ICCIT)* (pp. 1-6). IEEE.

[21] Buczak, A. L., & Guven, E. (2015). A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications surveys & tutorials*, *18*(2), 1153-1176.

[22] Huybrechts, T., Vanommeslaeghe, Y., Blontrock, D., Van Barel, G., & Hellinckx, P. (2017, November). Automatic reverse engineering of CAN bus data using machine learning techniques. In *International Conference on P2P, Parallel, Grid, Cloud and Internet Computing* (pp. 751-761). Springer, Cham.

[23] B. Wang, M. Dabbaghjamanesh, A. Kavousi-Fard, and S. Mehraeen. "Cybersecurity enhancement of power trading within the networked microgrids based on blockchain and directed acyclic graph approach." IEEE Transactions on Industry Applications 55, no. 6 (2019): 7300-7309.

[24] Samani ZR, Alappatt JA, Parker D, Ismail AAO, Verma R. QC-Automator: Deep Learning-Based Automated Quality Control for Diffusion MR Images. *Front Neurosci*. 2020;13:1456. Published 2020 Jan 22. doi:10.3389/fnins.2019.0145

[25] Hodges, C., An, S., Rahmani, H., & Bennamoun, M. (2019). Deep Learning for Driverless Vehicles. In *Handbook of Deep Learning Applications* (pp. 83-99). Springer, Cham.

[26] Ghaffari, Saeed, and M. Ashkaboosi. "Applying Hidden Markov M Recognition Based on C." (2016).

[27] Avatefipour, O., & Nafisian, A. (2018). A novel electric load consumption prediction and feature selection model based on modified clonal selection algorithm. *Journal of Intelligent & Fuzzy Systems*, *34*(4), 2261-2272.

[28] Samani, Z. R., Guntuku, S. C., Moghaddam, M. E., Preoţiuc-Pietro, D., & Ungar, L. H. (2018). Cross-platform and cross-interaction study of user personality based on images on Twitter and Flickr. *PloS one*, *13*(7).

[29] Hafeez, A., Malik, H., Avatefipour, O., Rongali, P. R., & Zehra, S. (2017). *Comparative study of can-bus and flexray protocols for in-vehicle communication* (No. 2017-01-0017). SAE Technical Paper.

[30] M. Dabbaghjamanesh, B. Wang, S. Mehraeen, J. Zhang, and A. Kavousi-Fard. "Networked microgrid security and privacy enhancement by the blockchain-enabled Internet of Things approach." In 2019 IEEE Green Technologies Conference (GreenTech), pp. 1-5. IEEE, 2019.

[31] Grozdić, Đ. T., Jovičić, S. T., & Subotić, M. (2017). Whispered speech recognition using deep denoising autoencoder. *Engineering Applications of Artificial Intelligence*, *59*, 15-22.

[32] Zheng, Y., Lu, X., Zhang, M., & Chen, S. (2019). Ecogeography-Based Optimization: Enhanced by Ecogeographic Barriers and Differentiations. In *Biogeography-Based Optimization: Algorithms and Applications* (pp. 69-87). Springer, Singapore.

[33] Lee, H., Jeong, S. H., & Kim, H. K. (2017, August). OTIDS: A novel intrusion detection system for in-vehicle network by using remote frame. In *2017 15th Annual Conference on Privacy, Security and Trust (PST)* (pp. 57-5709). IEEE.

Yubin Lin was born in Fujian Province, China in 1989. He received his B.S. degree in electrical and electronic engineering from the North China Electric Power University, China, in 2012, and an M.S. degree in electrical engineering from the University of Hong Kong, Hong Kong, China, in 2014. Since 2014, he has been a researcher and project reviewer at State Grid Fujian Power Economic Research Institute. His research interests include machine learning, optimization algorithm, and power system analysis.

Chengbin Chen was born in Fujian Province, FJ, CHN in 1992. He received the Bachelor in Communication Engineering from Fuzhou University, Fuzhou, China, in 2014 and received the Master degree in Electronics and Communication Engineering from Fuzhou University, Fuzhou, China, in 2017.

He is currently pursuing a Ph.D. Degree in the College of Physics and Information Engineering at Fuzhou University, China. His research interests are in sensor fusion, smart grid, UAV Control System, UAV design, fully-autonomous UAV. He is the author of more than 9 articles. And he won more than 20 national science and technology competitions in China.

Fen Xiao was born in Fujian Province, FJ, CHN in 1981. He received a B.S. degree in Communications Engineering and a B.S. degree in Business Administration from the North China Electric Power University, China, in 2003.

From 2003 to 2009, he was a Communications Engineer, State Grid Fujian Electric Power Co., Ltd, Fujian, Chian. Since 2010, he was a senior engineer of power system planning, State Grid Fujian Electric Power Co., Ltd, Fujian, China. His research interests are Power system and transmission line design.

OMID AVATEFIPOUR received the master's degree in computer engineering from the University of Michigan–Dearborn. He is currently with Valeo North America Inc., as a System Engineer with the Advanced Engineering Research and Development Group. He has work experience at Vector CANTech Company, as an Embedded Software Engineer. He has also worked as a Researcher in Information System, Security, and Forensics (ISSF) Laboratory, Department of Electrical and Computer Engineering (ECE), University of Michigan–Dearborn. His research interests include in-vehicle network communication protocol security, autonomous vehicles, embedded systems, machine learning, intelligent control systems, and robotics.

Khalid Alsubhi received his M.Math and Ph.D. degree in computer science from the University of Waterloo, Waterloo, Canada, in 2009 and 2016, respectively. He received his BSc. degree in computer science from King Abdulaziz University (KAU) in 2003. He is currently an Assistant Professor of computer science at KAU. His research interests are focused on network security and management, cloud computing, and security and privacy of healthcare applications.

Arda Yunianta received the bachelor's degree in Teknik Informatika from Universitas Pembangunan Nasional "Veteran" Yogyakarta, Indonesia, in 2007, the M.Eng. degree in Information Technology from Gadjah Mada University, Yogyakarta, Indonesia, in 2010, and the Ph.D. degree in Computer Science from Universiti Teknologi Malaysia, Johor Bahru, Malaysia, in 2015.

He is currently an Assistant Professor with the Department of Information Systems, Faculty of Computing and Information Technology in Rabigh, King Abdulaziz University, Jeddah, Saudi Arabia. He is also a remote faculty member in the Faculty of Engineering, Mulawarman University, Samarinda, Indonesia. He has authored and co-authored articles published in several reputable conferences and journals published by Web of Science, IEEE, SCOPUS, Springer, and many more. His research interests include ontology, data integration, semantic technology, e-learning system, big/intelligence data, deep learning, machine learning, data mining, and data science.

Dr. Arda Yunianta is an Editor and Reviewer in some international journals and IEEE conferences, he is also a committee member in some IEEE conferences.