IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 21 July 2022, accepted 6 August 2022, date of publication 19 August 2022, date of current version 29 August 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3200375

RESEARCH ARTICLE

An Intelligent, Two-Stage, In-Vehicle Diagnostic-Based Secured Framework

TASNEEM A. AWAAD[1], M. WATHEQ EL-KHARASHI[1,2], MOHAMED TAHER[1], AND KHALID AMMAR[3], (Member, IEEE)

[1]Department of Computer and Systems Engineering, Faculty of Engineering, Ain Shams University, Cairo 11517, Egypt
[2]Department of Electrical and Computer Engineering, University of Victoria, Victoria, BC V8W 3P6, Canada
[3]Department of Electrical Engineering, Ajman University, Ajman, United Arab Emirates

Corresponding author: M. Watheq El-Kharashi (watheq@engr.uvic.ca)

ABSTRACT Recent research interests have been directed to study the security of vehicles due to the advancement of their technologies. Due to the rapid growth and accelerated development of electronic control units (ECUs), they are countered to be exploited by external attacks. As a result, recent research efforts have been focused on investigating alternative countermeasures that might be implemented by introducing different intrusion detection systems (IDSs). The problem with some of IDSs is the location of their deployment because of the ECU limitations and constraints. Other introduced IDSs require severe changes in the in-vehicle network, which is not preferred by vehicle manufacturers. In this research, we introduce a novel design of a framework to check the state of the vehicle and capture possible attacks by detecting any malicious data in the diagnostic parameters of the vehicle. The framework is divided into two phases: the specific-based detection phase and the anomaly-based detection phase. The proposed system employs the extreme gradient boosting (XGBoost) algorithm to detect anomalies in diagnostic data and it is optimized by a non-dominated sorting genetic algorithm II (NSGA-II). The model is verified against two datasets collected from real vehicles. To generate anomalies in datasets, an attack generation algorithm is introduced. The model is trained on a dataset that contains different attack types and verified blindly against various attacks that have not been seen before. The framework's experimental results show that it can detect abnormalities with accuracy 97.00% for the Seat Leon 2018 dataset and 97.49% for the KIA SOUL dataset.

INDEX TERMS Anomaly detection, cyber-physical security threats, diagnostics, genetic algorithm, intrusion detection, machine learning, NSGA-II, vehicular security, XGBoost.

I. INTRODUCTION

Over the last decade, automobile manufacturers have transformed the shape and function of modern vehicles by rapidly adopting different current technologies. Advanced elements such as automation and connectivity with the outside environment (e.g., Vehicle to Infrastructure (V2I) and Vehicle to Vehicle (V2V) communications) are included to increase safety and enable vehicle collaboration. Modern vehicles now comprise a network of ECUs, accompanied by actuators and sensors, that perform one or more functions including critical ones. ECUs can serve a variety of functions, ranging from simple tasks like opening a door to more complex procedures like regulating the car's braking system. ECUs are managed by sophisticated software components that read data from and send data to other ECUs through one of the in-vehicle communication protocols.

Securing vehicles from cyberattacks is a difficult mission, as vehicles have traditionally been designed without considering full security requirements, relying on the supposition that vehicles function independently with no communication capabilities. ECUs can be exposed to threats through physical access to the on-board diagnostic (OBD)-II port or short and long-range wireless connectivity such as Bluetooth, cellular radio, and telematics control unit (TCU). The firmware updates over the air (FOTA) can reveal ECUs software and hardware faults and vulnerabilities that can be exploited to be one of the reasons for external unwilling control [1]. Thus,

The associate editor coordinating the review of this manuscript and approving it for publication was Wai-Keung Fung.

the safety and security of cyber-physical systems, such as vehicles, became an essential topic to be discussed.

Inter-vehicle communication is a promising paradigm that, by sharing messages, can assist in reducing traffic congestion and crashes. The security of the inter-communication of vehicles and intra-communication received a lot of interest for a discussion in the latest research [2], [3], [4]. The findings of studies into remote attacks on vehicles have astounded automotive companies, and this has resulted in recalling of 1.4 million vehicles [5], [6]. Tesla automobiles were shown by researchers that can be hacked and they demonstrated that the crucial vehicle functions can be manipulated remotely [5]. Electric vehicles, such as Tesla, can be attacked to broadcast false state-of-charge (SoC) data to charging services in order to get greater priority for charging [7]. Current studies are not concerned only with defending electric vehicles but also protecting the charging stations using deep learning techniques [8].

Various methods were mentioned by Zhang *et al.* to attack any vehicle [9]. One method is to use the OBD-II port, which allows users to capture the ECU traffic. Most ECU-controlling attacks are carried out via diagnostic updates, which allow users to download or update software. Wolf *et al.* studied the possible hazards of ransomware attacks on automotive systems [10]. The vehicle may be rendered unusable as a result of the ransomware encrypting one of its critical ECUs until the user pays to release it.

Usually, the focus of recent research in the automotive field tends to detect malicious attacks on Controller Area Network (CAN) buses due to the deficiency of its security features. The problem with such proposed IDSs that they need to be reliable, light, and rapid in the processing due to the characteristics of the CAN bus and the limitations of the provided ECUs. Malicious CAN messages can be detected by capturing the frequency of abnormal messages or manipulated data of CAN messages. The disadvantage of such IDSs that they can ignore some of the diagnostic parameters that are not frequently sent on the CAN bus or the diagnostic data that are related to the ECU that has the IDS. Thus, our research will focus on detecting attacks on the diagnostic data of the vehicle regardless of the source ECU that is in charge of them.

Various protocols, such as the OBD protocol [11] and the universal diagnostic services (UDS) protocol [12] are used to communicate data between ECUs and diagnostic systems. Several PIDs were captured using the OBD and UDS protocols in our test case.

The goal of this study is to detect malicious behaviors in vehicle diagnostics with acceptable accuracy. Thus, we expanded our previous proposed IDS [13] to introduce a framework of two stages: the first stage is responsible for detecting the unreasonable values of diagnostics based on the specification rules for each PID, while the second stage is in charge of detecting malicious semantic values of PIDs using machine learning. Since the current stream of research is interested in the utilization of machine learning and deep learning approaches to detect complex attacks, the XGBoost algorithm is employed in the second stage of our framework to detect suspicious data as it is a widespread and efficient open-source framework for this aim [14]. Selecting proper parameters for XGBoost is a hard problem. Consequently in this research, we use a modified version of the genetic algorithm known as NSGA-II to optimize the hyperparameters of the machine learning model [15].

The following are the key contributions to the literature made by this article:

1) Introducing a novel two-step framework for detecting malicious attacks in diagnostic parameters obtained from real vehicles without overloading the bus based on specification and anomaly detection techniques. The specification detection stage is composed of rules related to the characteristics of each PID and the anomaly detection stage uses the XGBoost in capturing malicious attacks.
2) Optimizing the hyperparameters of XGBoost using NSGA-II to detect malicious diagnostic in-vehicle parameters with optimal results.
3) Building datasets containing benign and malicious data to train and verify our framework. The malicious attacks have been generated through introduced attack models.
4) Comparing the detection performance of our framework against other machine learning and statistical algorithms.
5) Verifying our framework against unknown attacks generated from different attack models that are not used in the training process of the XGBoost model.

The structure of this paper is as follows. A brief background is provided about diagnostic protocols, XGBoost model, and genetic algorithm (GA) used in this research in Section II. The related work will be shown in Section III. The suggested architecture is shown in Section IV, which also depicts the model's flow. It also explains the machine learning technique that is applied to recognize anomalies and how the model parameters have been optimized using the NSGA-II algorithm. An illustration of the datasets used in this study, as well as a clarification of the attack generation models for training and testing, will be found in Section V. Evaluation results are discussed in Section VI and the limitations and advantages of our proposed work are discussed in Section VII. The conclusion of this study and future work are included in Section VIII.

II. OVERVIEW

This section provides an overview on some of the diagnostic protocols, such as OBD-II and UDS. It also provides a brief background about XGBoost and GA techniques.

A. DIAGNOSTIC PROTOCOLS

1) OBD-II

OBD-II is a vehicle's diagnostic protocol that is used to read emission data from ECUs for the purpose of diagnosis and monitoring [11]. OBD-II PIDs are defined by the

Identifier	Number of Bytes	Mode	PID	A	B	C	D	Unused data byte

FIGURE 1. General structure of an OBD-II frame.

Service Request	Service ID	Subfunction	Data Parameters
Positive Response	Service ID	Data Parameters	
Negative Response	Error ID	Service ID	Response Code

FIGURE 2. General structure of a UDS message.

SAE J1979 standard so that each PID has a code to request certain information about one of the vehicle's parameters (e.g., vehicle speed) [16]. The OBD-II PIDs do not cover all vehicle parameters, however, the vehicle manufacturer can customize PIDs using another diagnostic protocol (e.g., UDS). The OBD request is sent when a diagnostic tool is connected to the OBD-II connector and the corresponding ECU responds. Figure 1 shows the general structure of an OBD-II frame, where the identifier field shows whether it is a request or a response message, and the number of bytes field shows the needed number of bytes for each PID. The OBD-II has ten modes, where some of them show the data in real-time (e.g., RPM) and others clear and retrieve the stored trouble codes. The third field of the OBD-II frame shows the mode in which PIDs are defined. The fourth field of the frame is the corresponding PID. Some PIDs have minimum and maximum values and the formula to convert the corresponding PID value to decimal. The A, B, C, and D fields are the sent hexadecimal data bytes that need to be converted to logical values.

2) UDS

UDS is a diagnostic protocol that can be used for different communication buses, such as CAN and Local Interconnect Network (LIN) [12]. This protocol allows the diagnostic devices to interact with ECUs to diagnose and analyze the faults and provide the possibility to reprogram ECUs. The diagnostic communication through UDS is held by sending UDS request to the ECU which replies whether by a positive or negative response with the UDS message structure shown in Figure 2. The first field of the message is Service Identifier (SID) which distinguishes between the response and request messages. The subfunction field is added to some UDS messages as it is an optional field. The data parameter field identifies further information and configuration for the requested parameter.

B. XGBoost

XGBoost is a gradient boosting algorithm implementation that is scalable, portable, and distributed [14]. From ensemble weak learners models, the gradient boosting technique creates predictive models. Decision trees are a common model used in gradient boosting. The trees are built in a sequential order while boosting, with each succeeding tree attempting to reduce the faults of the previous tree. Each tree learns from its predecessors' residual errors and updates them. As a result, the tree that grows next in the series will learn from a modified version of the residuals.

To anticipate the outputs, the tree boosting model employs K additive functions.

$$\hat{y}_i = \varnothing(X_i) = \sum_{k=1}^{K} f_k(X_i), f_k \in \mathcal{F}, \tag{1}$$

where $\mathcal{F}$ is the space of tree regression and f_k is one of the tree structures that its leaves have w_j as a weight score at each $j-th$ leaf. The sum of the score of leaves is given by w. An objective function (2) must be minimized to train the model.

$$obj^{(t)} = \sum_{i} l(\hat{y}_i, y_i) + \sum_{k} \Omega(f_k), \tag{2}$$

$$\Omega(f_k) = \gamma T + \frac{1}{2}\lambda ||w||^2, \tag{3}$$

where l is the differentiable loss function that assesses the discrepancy between the anticipated $\hat{y}_i$ value and expected value y_i, while $\Omega(f_k)$ (3) is a regularization function that represents the tree structure complexity and is used to avoid model overfitting, the number of tree leaves is denoted by T, γ is the score when an external leaf is inserted, λ is the score when the tree cannot be subdivided any further. Because the model is based on the ensemble concept, it will be trained in an additive manner to predict $\hat{y}_t$ at the t-th iteration by minimizing f_t by adding it to (2). Taylor series is used to express some loss functions since they are difficult to simplify in good forms. The objective function can be described as indicated in (4) after the tree model has been re-formulated with respect to the derivation and Taylor series representation.

$$obj^{(t)} = \sum_{j=1}^{T} [\sum_{i \in I_j} g_i w_j + \frac{1}{2}(\sum_{i \in I_j} h_i + \lambda) w_j^2] + \gamma T, \tag{4}$$

where $g_i = \partial_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)})$ and $h_i = \partial^2_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)})$. Each j-th leaf's weight score can be determined as illustrated in (5). Equation (6) will be used to determine how good the supplied tree structure is. Because the number of produced trees is too huge to identify the best one, a greedy approach is used to prune ineffective tree branches one level at a time, utilizing (7) to evaluate split possibilities,

$$w_j = \frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda} \tag{5}$$

$$obj^{(t)}(q) = -\frac{1}{2} \sum_{j=1}^{T} \frac{(\sum_{i \in I_j} g_i)^2}{\sum_{i \in I_j} h_i + \lambda} + \gamma T \tag{6}$$

$$obj_{split} = \frac{1}{2}[\frac{(\sum_{i \in I_L} g_i)^2}{\sum_{i \in I_L} h_i + \lambda} + \frac{(\sum_{i \in I_R} g_i)^2}{\sum_{i \in I_R} h_i + \lambda} - \frac{(\sum_{i \in I} g_i)^2}{\sum_{i \in I} h_i + \lambda}] - \gamma \tag{7}$$

After splitting, I_L and I_R are instances of left and right tree nodes, respectively, where the $I = I_L \cup I_R$. The branch will be trimmed if the gain of (7) is negative.

C. GENETIC ALGORITHM

It is a stochastic optimization technique that is devoid of derivatives and depends on natural selection and biological evolution. GA outperforms other optimization methods in a number of ways [17]. It can be used to solve issues in both continuous and discrete optimization [17]. It is a population genetic-inspired computational model [17]. It has primarily been exploited as a function optimizer, and it has been shown to be a useful global optimization tool, particularly for multi-model and non-continuous functions [17]. GA creates a population of individuals, which is a collection of elements. Each individual represents a potential solution X_i ($i = 1, 2, 3, \ldots, p$), where p is the number of elements in the population, to the problem that needs to be optimized. The individual solution can be represented in a number of parameters which is known as a gene. The combination of genes for a string value is known as chromosomes.

The algorithm starts with the initial population to be evaluated by the fitness function. The selected individuals in the population are paired to apply the typical crossover operator. The matching locations on the two mating chromosomes are cut once, and the parts after the cuts are exchanged. The point of intersection can be randomly selected. The new individuals are subjected to mutation after crossing. A random value from a provided set of values corresponding to each parameter is used to change the value of a variable that is chosen with a particular probability. A tiny percentage of the fittest solutions is also copied into the next generation, which is known as elitism. Elitism ensures that the GA's solution quality does not deteriorate from one generation to another. The algorithm is repeated until a stopping criterion is reached.

III. RELATED WORK

A significant number of academics have focused on malicious attack detection in automotive network communication, reflecting the fact that this is one of the most important challenges for governments, businesses, and research. In this section, previously proposed in-vehicle communication IDSs, such as CAN bus and diagnostic IDSs are discussed.

A. IN-VEHICLE COMMUNICATION IDSs

According to Al-Jarrah *et al.*, IDSs are conventionally classified based on detection techniques into two types; knowledge/misuse and anomaly-based IDSs [18].

1) KNOWLEDGE-BASED IDS

It matches monitored events to known attack patterns (i.e., signatures). When knowledge-based IDS detects a match between the recorded events and known attack patterns, an intrusion is notified. Aldwairi textitet al. [19] proposed a parallel IDS approach for parallelizing a pattern matching algorithm on a multi-core CPU to speed up pattern matching.

2) ANOMALY-BASED IDS

It recognizes typical system behavior and classifies significant departures from it as intrusions. Lo *et al.* proposed a deep learning IDS that is composed of a convolutional neural network (CNN) and long short-term memory (LSTM) to capture the spatial and temporal dependencies in CAN data [20]. Their IDS performs preprocessing on CAN traffic to reduce inconsistency and incomplete data. The processed data are fed to CNN to extract the feature map and then LSTM is applied to extract the temporal dependencies and the extracted features are finally fed to a fully connected neural network (NN) to classify the output. Basavaraj and Tayeb proposed IDS where the data are preprocessed and encoded to be fed to a deep neural network (DNN) to detect anomalies in CAN data [21].

Al-Jarrah *et al.* categorized the intra-vehicle IDSs further into hybrid, payload-based, and flow-based IDSs [18].

3) FLOW-BASED IDS

The internal network of a vehicle, often the CAN bus, is monitored by a flow-based IDS, which extracts distinct characteristics. Vuong *et al.* proposed a detection model based on a decision tree using eight on-board cyber and physical features that were held on small-scale robotic vehicle [22]. Taylor *et al.* proposed a method for computing the mean of inter-packet timing using historical timing that was pre-calculated in normal packet flow to capture anomalies attacks in CAN messages [23]. The mean of inter-packet timing is used in a one-class support vector machine (OCSVM) to indicate whether an abnormal arriving frequency of CAN messages exists [23]. The proposed frequency-based detector, on the other hand, is unable to identify data manipulation attacks (masquerade) in CAN communications. Song *et al.* developed a lightweight technique for detecting unexpected behavior in CAN message frequency [24]. They claimed that each CAN ID is sent at a specific time interval, and if one of the CAN IDs' messages is sent faster than the corresponding time interval, an attack is detected. A modified OCSVM was introduced by Avatefipour *et al.* to capture cyber-attacks on CAN bus [25].

4) PAYLOAD-BASED IDS

A payload-based IDS looks at the payload of transactions to detect intrusions. Stabili *et al.* introduced a Hamming distance-based intrusion detection technique to capture abnormalities across a series of payloads of distinct ID classes [26]. Ganesan *et al.* developed a method for obtaining pairwise correlation between sensors, clustering those pairwise points for various driver behaviors, and comparing the cluster correlation points to the computed correlation values of the given sensor data [27]. Li *et al.* built a model that uses regression to forecast sensor values in relation to other connected sensors, compare the anticipated value to the received value, and raise a harmful alarm if the difference value is larger than a specific threshold [28]. Deep learning was used

by Kang to recognize attacks in CAN messages [29]. High-dimensional features derived from in-vehicle network packet bit streams that are exchanged between ECUs are used to train the detection model.

5) HYBRID IDS

A hybrid IDS is one that combines a payload-based IDS and a flow-based IDS categories. Müter *et al.* developed a set of detection sensors, including a consistency sensor, a plausibility sensor, a protocol sensor, a correlation sensor, a frequency sensor, a range sensor, a location sensor, and a formality sensor that allow for the identification of intrusions while driving without causing false positives [3]. The IDS presented by Zhang *et al.* has two stages: the first is a rule-based model that uses the time interval, message sequence, valid ID, and frequency of messages to detect anomalous behaviors [30]. The second stage of this approach employs a deep learning model to reveal anomaly attacks that may have passed the first stage.

B. VEHICLE DIAGNOSTIC IDSs

Rumez *et al.* proposed diagnostic IDS based on natural language processing (NLP) approach using the n-gram technique [31]. Their idea is based on building diagnostic logs by a gateway and sending the log to a server to analyze and perform anomaly detection. The IDS captures diagnostic sequences in a given window size to be analyzed. If the captured behavior is observed before in the training dataset, then the sequence is normal. If an anomaly in the sequence is detected, byte-based analysis is performed, which is built based on the n-gram algorithm. The byte-based analysis considers the CAN message as sentences and the bytes of payload as words that construct the contextual meaning of the diagnostic message. The work was built on the assumption that the attacker has access to the in-vehicle network. The attack scenarios can be summarized as follows.

- The CAN message can be manipulated and changed within the sequence.
- Additional CAN messages can be inserted within the sequence.
- The CAN messages within the sequence can be exchanged.

The dataset was collected from BMW i3 using a diagnostic device. However, the dataset is small, whereas the proposed IDS is not proven to scale with large diagnostic data and different attack models.

IV. PROPOSED FRAMEWORK

The framework consists of two main detection levels as shown in Figure 3; the first level is the specification-based detection (specification-based system) and the second level is the anomaly attack detection using a machine learning model (anomaly-based system) that is tuned by an optimization technique. Our suggested approach addresses the problem of the location of IDS. The IDSs are usually installed in a single ECU or a gateway ECU. The downside of deploying such an IDS is the ECU's power limits, as the IDS should not impact the ECU's CPU load in general, the complexity of modifying the internal design of each manufacturer's vehicle network, and the cost of additional powerful ECUs. The proposed solution is to introduce an appropriate architecture that allows one powerful ECU to maintain the IDS while reducing the cost of employing many powerful ECUs. The trend nowadays is to consolidate multiple functions distributed over many ECUs into common zone ECUs to reduce the complexity of wiring and architectures in the in-vehicle networks, where these zone ECUs will be connected to each other, and they will carry all the messages of their smaller children ECUs [32], [33]. Hence, these zone ECUs should be more powerful compared to the ordinary ECUs used in the previous architectures.

The end node ECU is attached to the OBD connector, as shown in Figure 3. The diagnostic analysis will be performed on each PID in addition to the basic functionality of the end node ECU, which is to carry out the OBD request. The model should issue a diagnostic request via the connecting bus, and the recipient ECUs either respond directly if they have the needed PID, or they route the request to other ECUs that are not directly connected to the OBD. The targeted ECU sends the PID's value to the framework after receiving the request.

Another ECU can complement our introduced framework to avoid a single point of failure in our system, however, this decision is up to the manufacturer to compromise between the cost of another powerful ECU and the necessity of the vehicle safety and security.

In our proposed approach, sending emission diagnostic messages during the vehicle operation to read the PIDs will be treated as normal messages, otherwise, it will be taken into consideration as an attack.

A. SPECIFICATION-BASED SYSTEM

The input to our framework will be processed by the first stage which is the specification-based system that detects whether this PID is benign based on some defined specification rules for the corresponding PID. For example, if the received value of PID is out of its minimum and maximum ranges, then the framework reports that there is an attack. If the processing PID passes successfully, the PID is sent to the second stage which is the anomaly detection based on the XGBoost technique that is tuned by NSGA-II.

B. ANOMALY-BASED SYSTEM

The second stage of the framework employs XGBoost technique [14], which is divided into two phases; the first phase is responsible for training. In this phase, the model is optimized and validated. The second phase is in charge of testing. XGBoost works well with heterogeneous features as in our case. The training shape of the datasets is in two dimensions (2D) format where the first dimension represents the values of each feature (PID) over time and the second dimension represents different PIDs. XGBoost has several parameters that affect its performance. Consequently, we used NSGA-II [15]

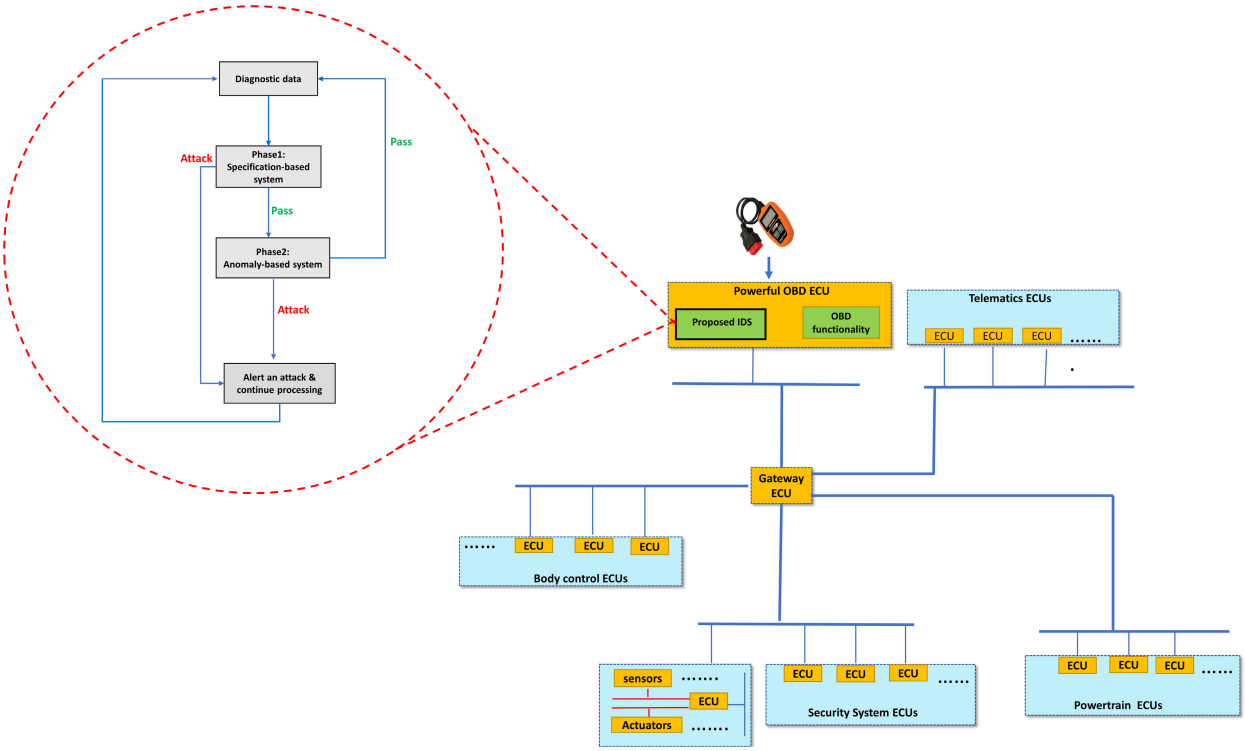


FIGURE 3. The proposed framework for in-vehicle network communication architecture.

to optimize XGBoost parameters in the training phase of the second stage of our framework.

In our case, the problem population is a set of different XGBoost models with different model values. Each parameter of the model is considered in our case as a gene. The chromosome representation in our problem will be a vector of consecutive parameters of the model, which can be identified as

$$C = [\eta, \gamma, \theta, \ldots], \tag{8}$$

where θ is the maximum depth of a tree, γ is the value of minimum loss reduction, and η is the learning rate. The rest of the parameters, such as instance minimum sum weight needed for each child, the method to use to sample the training instances, and the subsample ratio of the training instances can be listed in a sequence to follow the aforementioned parameters that are identified in (8).

Figure 4 provides an overview of the training and testing phases in the second stage of the framework. The population is divided frontwise based on non-dominated sorting. The algorithm selects N XGBoost models by forming N chromosomes of different model parameters (genes). The chosen candidates are trained and evaluated by a fitness function. The fitness function in our problem assesses the loss of the XGBoost in (2), minimizes the false acceptance rate, and increases the detection rate. Individuals are ranked based on calculated crowding distance. The crossover and mutation are held on given XGBoost models by exchanging and mutating some of the parameters' values. The newly built XGBoost models are evaluated to choose the new offspring based on the given criteria. Sometimes, some XGBoost models provide the same loss, false acceptance rate, and detection rate results, so the criteria for selecting the fittest offspring have been updated so that if there are XGBoost models that give the same results, the algorithm chooses the models that have the least value of maximum depth of trees parameter to reduce the model complexity as much as possible. For each population, the process is repeated until the GA converges. Convergence refers to the algorithm's ability to produce children that are not significantly different from earlier generations or reach the criteria. After completing the training process, the final XGBoost model is produced to be evaluated by test data in the testing phase.

V. DATASETS AND ATTACK MODEL FOR TRAINING AND TESTING

This section illustrates the datasets and explains the attack models used in the creation of the training and testing datasets. To verify the framework, two datasets were obtained from real vehicles. They were acquired from genuine automobiles, such as the Seat Leon 2018 vehicle [34] and the KIA SOUL vehicle [35]. The KIA SOUL dataset is available on the Hacking and Countermeasure Research Lab (HCRL) website. It contains 51 vehicle signals collected through OBD-II [35]. The total records of the dataset are 94,401 captured every one second for ten drivers. The Seat Leon 2018 dataset is available on the Karlsruhe Institute of Technology (KIT) website. It contains ten signals collected

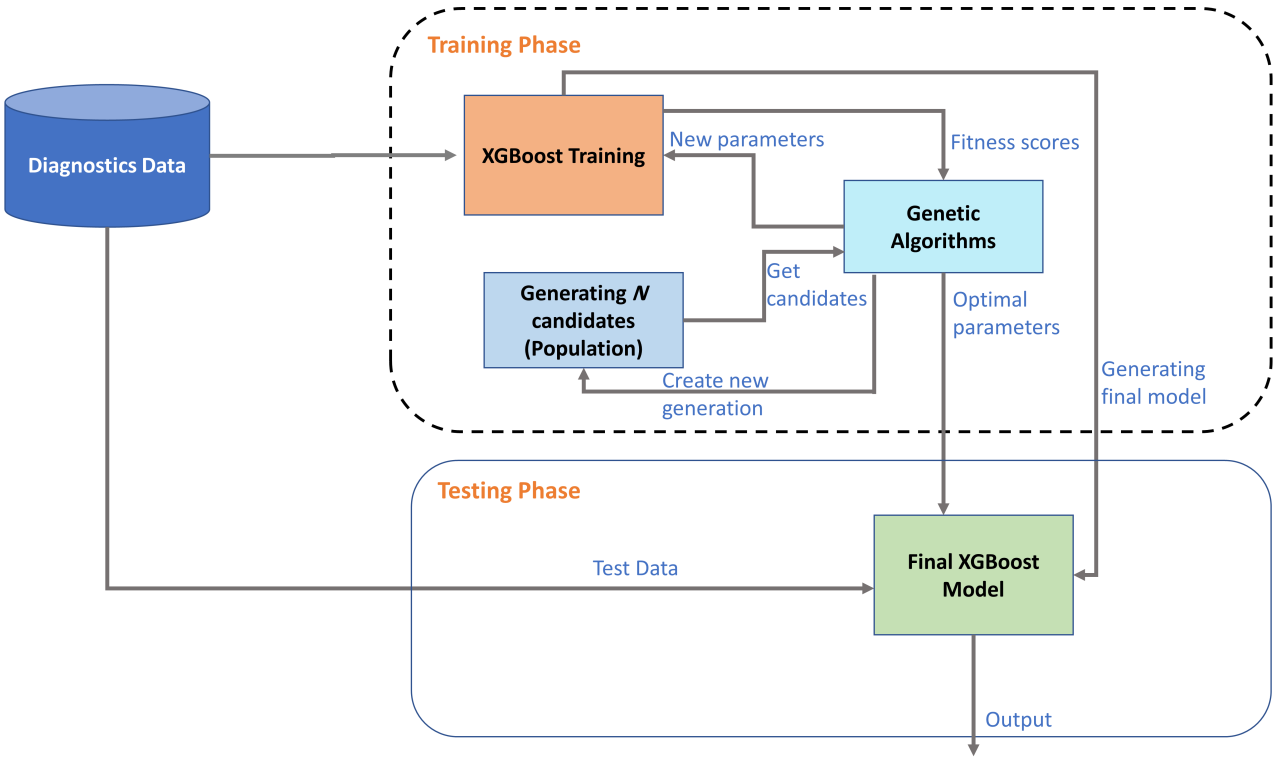


FIGURE 4. Training and testing phases of the anomaly-based detection system in the second stage of the proposed framework.

through OBD-II collected every 100 *ms* [34]. The total records of data used in this work are 54,784.

The autocorrelation function (ACF) is used to check the periodicity in both datasets, as shown in Figure 5, where the cyclic behavior of the signal is repeated over time in both datasets. ACF shows the correlation between the current value of the signal with its previous values over time. The periodicity behavior in given datasets assists machine learning to detect the deviation when abnormal behavior occurs.

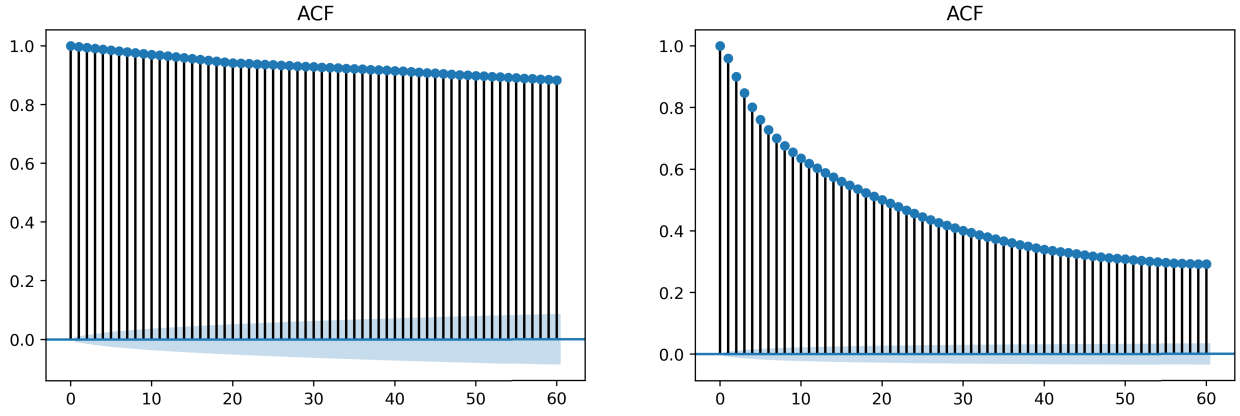
Most of the proposed IDSs mentioned in Section III stated attack model scenarios for CAN messages, such as injection fuzzy and spoofing messages and performing replay, masquerade, and denial of service attacks without sharing the datasets they worked on. Kang *et al.*, Lee *et al.* published benign and malicious datasets for CAN messages [36], [37]. However, we would not work on them as their representation differs from the format that is needed for our framework in the application layer of the OSI model. Our framework works on tabular data where each column contains a particular feature (PID) represented in logical values, while other datasets, such as "Car Hacking: Attack Dataset" [36] that has a different format where each row represents a time stamp, CAN message ID, number of data bytes, and the physical values of different PIDs and other fields. Hence, there is no published dataset containing malicious data for diagnostic parameters. We introduce different attack models depending on randomization that manipulates the benign dataset to create malicious training and testing data.

The attack models are built on the following assumptions:

- Diagnostic tests with values that are anomalous or malicious coming from suspicious sources configure some diagnostic parameters.
- Random values are written by an infected ECU into other ECUs.
- Semantic values are written by an infected ECU into other ECUs for certain PIDs but at an inappropriate time.

A. TRAINING ATTACK MODEL

Algorithm 1 depicts which attack model is used to exploit the datasets for training the XGBoost model. The algorithm iterates over the dataset through a step-wise period n, where N is the number of steps, to capture particular rows at a certain time t to be manipulated by different attacks based on the generated probability p. The value of n should not be too large to avoid generating a biased dataset providing the machine learning model, the space to learn the benign and the malicious behaviors. If a biased dataset is generated, augmentation algorithms, such as the adaptive synthetic sampling approach (ADASYN) can be utilized for an imbalanced dataset. The attacks depend heavily on randomness where Attack A and Attack B are performed on randomly chosen PID at a certain t, while Attack D is performed on two random PIDs at the same particular t and Attack C is performed on all PIDs.



(a) ACF of engine speed graph of the Seat Leon 2018 dataset.

(b) ACF of engine speed graph of the KIA SOUL dataset.

FIGURE 5. ACF graphs of engine speed of the Seat Leon 2018 and KIA SOUL datasets. The blue shaded region indicates the confidence interval. If the signal at any time exists within the shaded region, then this signal becomes less correlated to itself in past times.

Algorithm 1 Mechanism for Generating Attacks

```
p := generate_random(0.1, 0.9)
for t ← n to N * n do
    if p > 0.75 then
        Dataset[t] ← Generate_Attack_D()
    else if p > 0.50 then
        Dataset[t] ← Generate_Attack_C()
    else if p > 0.25 then
        Dataset[t] ← Generate_Attack_B()
    else
        Dataset[t] ← Generate_Attack_A()
    end if
end for
```

Attack A can be formulated as follows.

$$f(t_l, x_k) = \begin{cases} f(t_l, x_k)\alpha, & \text{if } p > 0.5, 0.1 \leq \alpha \leq 0.9 \quad (9) \\ -f(t_l, x_k)\alpha, & \text{otherwise}, 0.1 \leq \alpha \leq 0.9 \quad (10) \end{cases}$$

The aim of Attack A is to manipulate the current value of random PID at a chosen t by reducing it with a positive or negative value (α) determined by random probability p. Since there are negative values for some of the PIDs, the negative part is introduced to fuzz the values.

The second attack can be stated as follows.

$$f(t_l, x_k) = \begin{cases} f(t_{l-1}, x_k)\beta, & \text{if } p > 0.5, 0.1 \leq \beta \leq 0.9 \quad (11) \\ f(t_{l-1}, x_k)\beta, & \text{otherwise}, 1.5 \leq \beta \leq 4 \quad (12) \end{cases}$$

Attack B targets the manipulation of the current value of PID, which is chosen randomly, by replacing it with the previous value of this PID and multiplying it with a random value to scale up or scale down the current value of PID.

The following is an example of Attack C. The attack changes the value of all PIDs at t to zeros.

$$f(t_l, x_k) = 0, \quad \forall_k \quad (13)$$

Attack D is as follows.

$$f(t_l, x_k) = f(t_l, x_k)\alpha, \quad 0.1 \leq \alpha \leq 0.9 \quad (14)$$
$$f(t_l, x_m) = f(t_l, x_m)\beta, \quad 1.5 \leq \beta \leq 4 \quad (15)$$

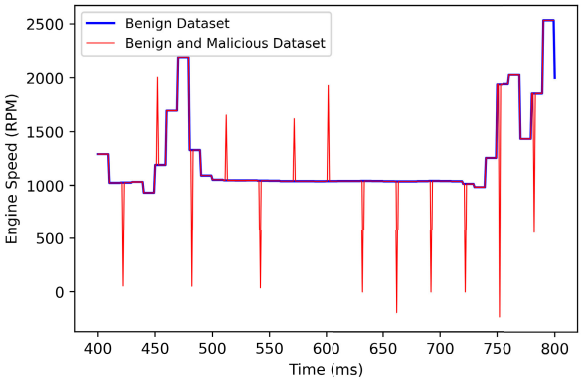
This attack targets change in two random PIDs at the same t to insert more fuzziness in the data. The randomization attacks are more effective and powerful than increasing or reducing the current benign value by a specific amount, as they also cover different types of attacks like replay, chip tuning, and masquerade.

Figure 6 demonstrates the manipulation of engine speed in the two datasets. The graphs show the occurrence of point anomalies, where some of the benign values deviate at a particular time.

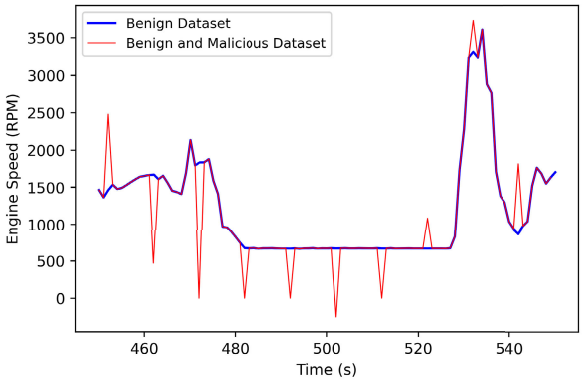
B. TESTING ATTACK MODEL

For testing the IDS with unknown attacks, the framework is verified against different datasets where each dataset contains benign data and malicious data that is created by one of four attacks mentioned in Table 1. To generate testing datasets, the benign dataset is iterated by step-wise period n to obtain certain rows at a particular time t to manipulate random PIDs with a certain attack. Since the testing attack generation depends on randomness, the distribution of each new dataset with one of the attacks is hard to be predicted. This process is done to test the robustness and stability of the model.

The proposed testing attacks cover many scenarios that target manipulation of PID values in a different way from attacks used in training. Increasing the Engine Coolant Temperature PID while the vehicle is in a normal state could lead to poor performance for the engine and may lead to operating the electric radiator fans in case it is not needed. While reducing RPM may give a false alarm for the driver to increase the driving speed which leads to jeopardizing the driver's life.



(a) A snapshot of the benign engine speed data against the manipulated engine speed data of the Seat Leon 2018 dataset.



(b) A snapshot of the benign engine speed data against the manipulated engine speed data of the KIA SOUL dataset.

FIGURE 6. An example of generated malicious engine speed PID in both datasets used in training XGBoost model. The benign data points in (a) and (b) are manipulated at a particular time by applying Algorithm 1 to deviate from the normal behavior.

TABLE 1. Attack models for testing.

Attack no.	Formula
Attack E	$f(t_l,z_k)=\begin{cases} f(t_l,z_k)+f(t_l,z_k)\beta, & \text{if } p>0.5, 0.1\le\beta\le 0.5 \quad (16)\\ f(t_l,z_k)-f(t_l,z_k)\beta, & \text{otherwise, } 0.1\le\beta\le 0.5 \quad (17)\end{cases}$
Attack F	$f(t_l,z_k)=\begin{cases} f(t_l,z_k)+f(t_{l-1},z_k)\beta, & \text{if } p>0.5, 0.1\le\beta\le 0.5 \quad (18)\\ f(t_l,z_k)-f(t_{l-1},z_k)\beta, & \text{otherwise, } 0.1\le\beta\le 0.5 \quad (19)\end{cases}$
Attack G	$f(t_l,z_k)=\begin{cases} f(t_l,z_k)+f(t_l,z_k)\beta, & \text{if } p>0.5, 1.5\le\beta\le 4 \quad (20)\\ f(t_l,z_k)-f(t_l,z_k)\beta, & \text{otherwise, } 1.5\le\beta\le 4 \quad (21)\end{cases}$
Attack H	$f(t_l,z_k)=\begin{cases} f(t_l,z_k)+f(t_{l-1},z_k)\beta, & \text{if } p>0.5, 1.5\le\beta\le 4 \quad (22)\\ f(t_l,z_k)-f(t_{l-1},z_k)\beta, & \text{otherwise, } 1.5\le\beta\le 4 \quad (23)\end{cases}$

VI. EXPERIMENTAL RESULTS

Our framework's output is depicted in this section. On an Intel Core i7-8550U processor running at 1.80 *GHz*, the training and testing are simulated. Currently, we are not concerned about the framework's processing time as the results focus on the proof of concept for the proposed idea. The introduced IDSs in section III for in-vehicle communication buses are not in our scope as we focus on diagnostic IDSs. The introduced IDS for detecting anomalies in diagnostic messages in work [31] did not publish the used dataset and worked on UDS messages over CAN bus which have different formats than needed by our framework. Therefore, a benchmark is built to show the comparison between the performance of our framework against techniques, such as decision trees, random forest, naive Bayes, SVM with the sigmoid kernel, SVM with the radial basis function kernel (RBF), SVM with the linear kernel, OCSVM, isolation forest, and neural network that are used in some of mentioned IDSs in section III.

First, our framework is compared to some well-known machine learning models as well as statistical models mentioned in [21], [22], [23], and [27]. The dataset, which contains benign data and malicious data generated from Attack A, Attack B, Attack C, and Attack D is divided into 70% training data and 30% for testing and validation. The testing part of the dataset is used for the evaluation of our IDS against other models. NSGA-II is used to tune and choose the fittest parameters of the mentioned models for consistency. The size of the population used in our case is 20 and the number of generations is 15. The rates of crossover and mutation used are 0.9.

TABLE 2. Configuration of hyperparameters of XGBoost in our framework after using NSGA-II.

Parameter	Value
Learning rate (η)	0.1
Minimum loss reduction (γ)	0
Tree maximum depth (θ)	15
Instance minimum sum child weight	3
Training instances subsample ratio	0.9
Sampling method	Uniform

Table 2 shows the optimal value of each XGBoost parameter used for anomaly detection of PIDs after using NSGA-II. The learning rate is used to avoid overfitting by tensing the feature weights at each iteration. The maximum depth of the tree is not preferable to be high to avoid the complexity and the overfitting of the model. Minimum loss reduction is a parameter that indicates the minimum reduction value of loss to perform a split on the leaf node. The split will be done only if the resulting reduction is positive. The sampling approach used in our model is uniform, which provides an equal probability for each training set leading to choose the subsample ratio to be greater than 0.5 for getting good results. The partitioning process of the leaf node will stop if the instance sum weight of the leaf node is less than the minimum sum child weight.

Detection accuracy, precision (PR), and recall (R) are chosen as metrics to compare between our framework and other machine learning and statistical models. Detection accuracy can be determined by the ratio of true positively detected attacks and true negatively detected attacks over the total number of samples.

$$Accuracy = \frac{T_P + T_N}{T_P + T_N + F_P + F_N}, \tag{24}$$

where T_P denotes malicious attacks that have been positively identified and T_N represents the true classified benign class.

TABLE 3. The accuracy of the proposed framework versus the accuracy of some common machine learning and statistical models.

Model	Accuracy	
	Seat Leon 2018 (%)	KIA SOUL (%)
Decision tree	95.98	95.29
Random forest	96.34	97.23
Naive Bayes	82.42	60.35
SVM (Sigmoid kernel)	60.00	56.53
SVM (RBF kernel)	50.40	55.65
SVM (Linear kernel)	59.10	60.08
OCSVM	81.00	70.50
Isolation forest	78.96	69.00
Neural network (3 layers)	89.20	89.48
Our framework	97.00	97.49

TABLE 4. The precision and recall of the proposed framework versus the precision and recall of some common machine learning and statistical models.

Model	Seat Leon 2018		KIA SOUL	
	PR (%)	R (%)	PR (%)	R (%)
Decision tree	95.99	95.93	92.23	92.23
Random forest	96.28	96.17	93.31	92.72
Naive Bayes	72.66	63.89	76.05	60.37
SVM (Sigmoid kernel)	77.33	60.75	42.64	49.98
SVM (RBF kernel)	77.37	60.30	74.16	53.36
SVM (Linear kernel)	44.66	43.89	46.60	49.41
OCSVM	79.91	70.53	62.89	52.32
Isolation forest	70.26	61.20	63.21	52.10
Neural network (3 layers)	83.85	78.72	83.87	77.69
Our framework	97.09	96.92	96.65	96.44

The number of benign samples recognized as malicious is represented by F_P, whereas the number of malicious samples detected as benign is represented by F_N.

A precision factor is the number of correct attack detections to the overall number of identifications.

$$PR = \frac{T_P}{T_P + F_P} \tag{25}$$

A recall factor is the number of correct attack detections to the total number of generated attacks.

$$R = \frac{T_P}{T_P + F_N} \tag{26}$$

Table 3 and Table 4 show the superiority of our framework against others regarding the accuracy, precision, and recall. Because the attacks are generated randomly, the resulting value is sometimes close to normal values but at an inappropriate time, causing models like isolation forest and OCSVM to miss such an attack. The linear and statistical classifiers fail to detect the rise in unpredictability and nonlinearity of employed datasets. Random forest and decision tree models are the only models that can achieve results that are close to our framework, however, they are less stable and more sensitive to any change in the data. The results of the framework show its stability against datasets of two different vehicles.

The performance of the XGBoost for different model thresholds is shown in Figure 7 for the two datasets, where the area under the curve (AUC) of the model is 0.97 for the Seat Leon 2018 dataset and 0.98 for the KIA SOUL dataset

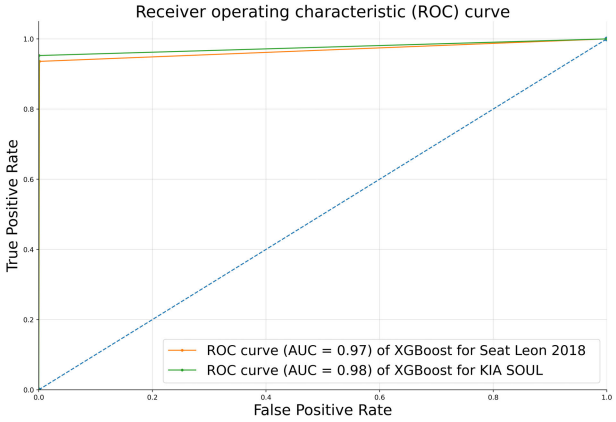


FIGURE 7. Roc curves of our anomaly detection model for the Seat Leon 2018 and KIA SOUL datasets.

clarifying the ability of the model in differentiating between the two classes (i.e., benign and malicious).

Second, we verified the model against four datasets, where each dataset per vehicle contains benign data manipulated by one of the attacks used in training to verify the model accuracy, false acceptance rate, detection rate, precision, and recall for each attack as clarified in Table 5 and Table 6.

The detection rate (DR) is a metric for determining the proportion of truly identified malicious diagnostic readings to the overall number of malicious attacks detected.

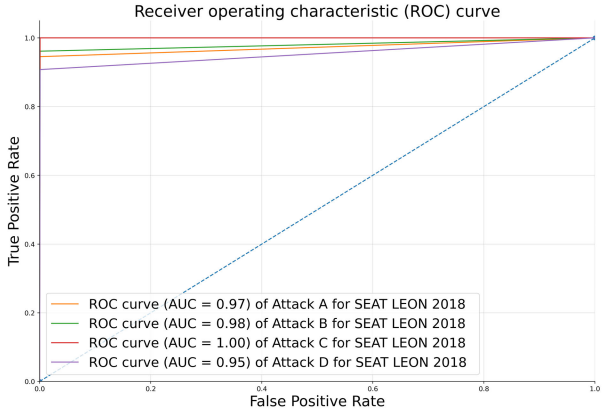
$$DR = \frac{T_P}{T_P + F_P} \tag{27}$$

The false acceptance rate (FA) is the proportion of benign diagnostic values detected as malicious.

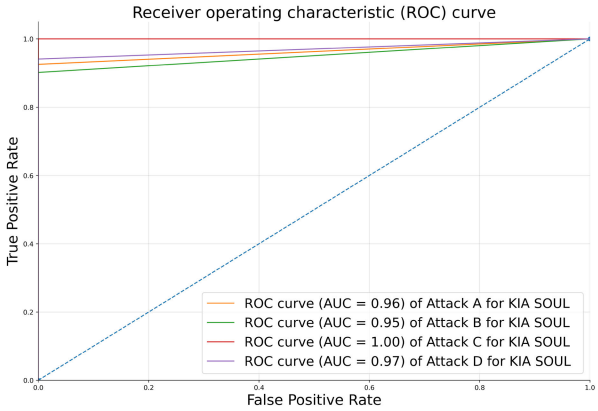
$$FA = \frac{F_P}{T_N + F_P} \tag{28}$$

The framework returns a low false acceptance rate and a high detection rate, indicating that the ratio of benign data misclassification is minimal, reducing user distraction and demonstrating that the framework is effective. Figure 8 shows the good performance of the anomaly detection model used in the second stage of our proposed framework against Attack A, Attack B, Attack C, and Attack D for the two different vehicles.

Third, the framework is evaluated against Attack E, Attack F, Attack G, and Attack H which are considered unknown attacks that are not used in the training process of the XGBoost model. Although the IDS has not seen the attacks before, it is capable to detect them with high accuracy, precision, and recall with a good detection rate, and false acceptance rate for each attack in the two datasets, as shown in Figure 9. Due to the gradient boosting nature of the XGBoost and the capability to build a more expressive model by learning and updating the previous weak models' residual errors, the classifier is efficient to detect the abnormalities in diagnostic parameters even if the classifier has not been trained on the attack models used in testing. The results proved that our



(a) ROC curves of the XGBoost model over the Seat Leon 2018 dataset.



(b) ROC curves of the XGBoost model over the KIA SOUL dataset.

FIGURE 8. ROC curves of the XGBoost model for (a) the Seat Leon 2018 and (b) the KIA SOUL datasets. The model is tested against Attack A, Attack B, Attack C, and Attack D over the Seat Leon 2018 and the KIA SOUL dataset. The model is trained on a dataset that contains benign data and manipulated data generated from Algorithm 1. The model is verified against four datasets, where each dataset contains benign data and manipulated data by one of the aforementioned attacks.

TABLE 5. The accuracy, detection rate, false acceptance rate, precision, and recall of the proposed framework against Attack A, Attack B, Attack C, and Attack D for Seat Leon 2018.

	Attack A	Attack B	Attack C	Attack D
Accuracy	0.9726	0.9804	0.9995	0.9536
DR	0.9460	0.9621	1.0000	0.9084
FA	0.0009	0.0009	0.0010	0.0010
PR	0.9736	0.9809	0.9995	0.9588
R	0.9726	0.9804	0.9995	0.9536

TABLE 6. The accuracy, detection rate, false acceptance rate, precision, and recall of the proposed framework against Attack A, Attack B, Attack C, and Attack D for KIA SOUL.

	Attack A	Attack B	Attack C	Attack D
Accuracy	0.9600	0.9502	0.9998	0.9704
DR	0.9100	0.9006	1.0000	0.9421
FA	0.0003	0.0002	0.0002	0.0001
PR	0.9700	0.9547	0.9998	0.9715
R	0.9600	0.9502	0.9998	0.9704

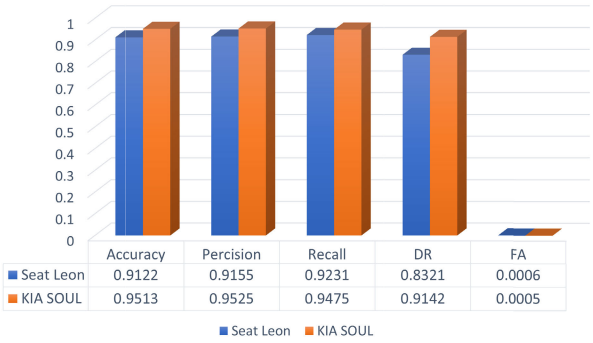
framework is capable of being used in detecting anomalous attacks without causing a distraction for the user.

According to the busload analysis, each PID requires two messages: a request message and a response message, both of which take 400 *us* because one CAN message takes 200 *us* at a common CAN baud rate of 500 *kbps*. If the system is expected to process 200 PIDs, the model will need to send and receive 200 CAN messages, which will take 80 *ms*. If the system checks all of the PIDs once every second, the busload in this instance will be less than 10%. By concentrating on the most critical PIDs and increasing their reading rate while decreasing the rate of reading the less important ones, the busload can be minimized.

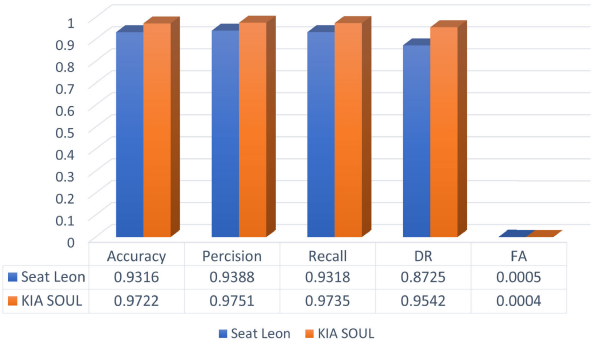
VII. DISCUSSION

Our framework has several advantages that can be summarized as follows. It is a generic framework due to its ability to deal with any diagnostic protocol because of its location in the application layer of the OSI model. It is located in a centralized arrangement to reduce the need for additional ECUs in distributed systems. Querying different PIDs minimizes the busload on the in-vehicle network buses by dividing the messages on different communication buses; therefore reducing the heavy load on a particular bus or ECU. The framework can detect the attack of any ECU by detecting any abnormal change in the PIDs that are not necessarily processed by the aforementioned ECU but are correlated to the signals of that ECU. It can check the state of the vehicle periodically in the diagnostic testing and updates (e.g., firmware updates) and in moving mode. It provides the manufacturer the flexibility to define which important PIDs can be checked due to the generality of the framework. The existence of the first phase accelerates the detection process when the specification-based system can capture malicious diagnostic data and increases the accuracy of the detection. The proposed framework provides a solution to detect attacks in any ECU without changing the internal design of the in-vehicle network structure. The framework is verified on datasets of two different vehicle models, where one of them contains the behavior of ten drivers which shows that the framework is stable and robust.

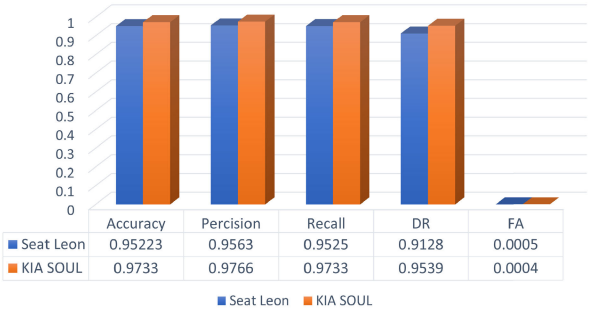
However, there are some challenges that need to be addressed further in future work. The characteristics of some signals could be changed by increasing the age of the vehicle. The problem of vehicle aging could be solved by performing training in the maintenance time to avoid the increase of false alarms. Vehicle maintenance and future software updates may change the behavior of vehicle modules which may affect the framework accuracy. However, the re-training of our framework could solve this issue, but this will expose our framework to attacks. The framework deals with any abnormal behavior as a malicious one without checking the diagnostic trouble code (DTC) for possible issues or faults in sensors [38].



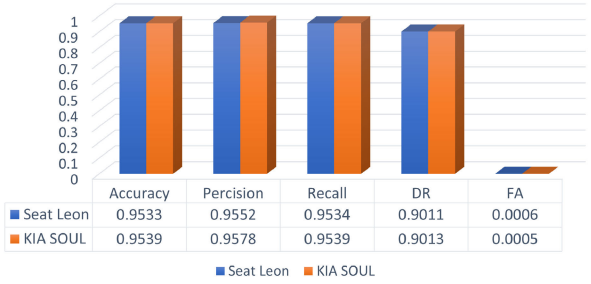
(a) Attack E



(b) Attack F



(c) Attack G



(d) Attack H

FIGURE 9. Accuracy, recall, and precision results of (a) Attack E, (b) Attack F, (c) Attack G, and (d) Attack H for the Seat Leon 2018 and KIA SOUL datasets.

VIII. CONCLUSION

Our research introduced a framework to detect anomalous cyber intrusions in automotive diagnostics. To detect malicious diagnostic parameters without raising the busload, a novel architecture of a vehicle communication network was introduced. Our IDS works with any diagnostic protocol's data. The framework contains two phases; the first phase is a specific-based detection system and the second one is the anomaly-based detection system. The XGBoost machine learning technique is employed in the second stage of the framework. The parameters of the XGBoost were tuned using the NSGA-II optimization technique in the training phase. The model was verified using two datasets gathered from several real-world vehicles. To modify the diagnostic data, different attack models are proposed. Because of the difficulty of attacks, the most prominent machine learning models mentioned in this research failed to detect them, while our proposed framework achieved high detection accuracy with 97.00% for the Seat Leon 2018 dataset and 97.49% for the KIA SOUL dataset. In addition, the proposed framework can detect unknown attacks (Attack E, Attack F, Attack G, and Attack H) with a high detection accuracy of 91.22%, 93.16%, 95.22%, and 95.33%, respectively, for Seat Leon 2018 and 95.13%, 97.22%, 97.33%, and 95.39%, respectively, for KIA SOUL. In future work, we will enhance the framework to deal with period anomalies that affect the behavior of the signal for a period of time. Hardware acceleration for such a framework will be developed to include more complex functionality in our IDS.

REFERENCES

[1] D. K. Nilsson, U. E. Larson, and P. H. Phung, "Vehicle ECU classification based on safety-security characteristics," in *Proc. IET Road Transp. Inf. Control Conf. ITS United Kingdom Members' Conf. (RTIC)*, 2008, pp. 1–7.

[2] U. E. Larson, D. K. Nilsson, and E. Jonsson, "An approach to specification-based attack detection for in-vehicle networks," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2008, pp. 220–225.

[3] M. Muter, A. Groll, and F. C. Freiling, "A structured approach to anomaly detection for in-vehicle networks," in *Proc. 6th Int. Conf. Inf. Assurance Secur.*, Aug. 2010, pp. 92–98.

[4] S. Woo, H. J. Jo, and D. H. Lee, "A practical wireless attack on the connected car and security protocol for in-vehicle CAN," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 993–1006, Apr. 2015.

[5] J. Takahashi, "An overview of cyber security for connected vehicles," *IEICE Trans. Inf. Syst.*, vol. E101.D, no. 11, pp. 2561–2575, 2018.

[6] C. Miller and C. Valasek, "Remote exploitation of an unaltered passenger vehicle," in *Proc. DEFCON Hacking Conf.*, 2015, pp. 1–91.

[7] A. A. Shafee, M. M. Fouda, M. M. E. A. Mahmoud, A. J. Aljohani, W. Alasmary, and F. Amsaad, "Detection of lying electrical vehicles in charging coordination using deep learning," *IEEE Access*, vol. 8, pp. 179400–179414, 2020.

[8] A. Shafee, M. Nabil, M. Mahmoud, W. Alasmary, and F. Amsaad, "Detection of denial of charge (DoC) attacks in smart grid using convolutional neural networks," in *Proc. Int. Symp. Netw., Comput. Commun. (ISNCC)*, Oct. 2021, pp. 1–7.

[9] T. Zhang, H. Antunes, and S. Aggarwal, "Defending connected vehicles against malware: Challenges and a solution framework," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 10–21, Feb. 2014.

[10] M. Wolf, R. Lambert, A. Schmidt, and T. Enderle, "Wanna Drive? Feasible attack paths and effective protection against ransomware in modern vehicles," in *Proc. ESCRYPT*, Nov. 2017.

[11] *Road Vehicles Communication Between Vehicle and External Equipment for Emissions-Related Diagnostics—Part 3: Diagnostic Connector and Related Electrical Circuits: Specification and Use*, document ISO 15031-3:2016-04, 2016.

[12] *Road Vehicles Unified Diagnostic Services (UDS)—Part 1: Specification and Requirements*, document ISO 14229-1:2013-03, 2013.

[13] T. A. Awaad, M. W. El-Kharashi, and M. Taher, "Lightweight diagnostic-based secure framework for electronic control units in vehicles," in *Proc. Int. Symp. Netw., Comput. Commun. (ISNCC)*, Oct. 2021, pp. 1–5.

[14] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 785–794, doi: 10.1145/2939672.2939785.

[15] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[16] *Road Vehicles Communication Between Vehicle and External Equipment for Emissions-Related Diagnostics—Part 5: Emissions-Related Diagnostic Services*, document ISO 15031-5:2015, 2015.

[17] M. J. Er and F. Liu, *Parameter Tuning of MLP Neural Network Using Genetic Algorithms*. Berlin, Germany: Springer, 2009, pp. 121–130.

[18] 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.

[19] M. Aldwairi, A. M. Abu-Dalo, and M. Jarrah, "Pattern matching of signature-based IDS using myers algorithm under MapReduce framework," *EURASIP J. Inf. Secur.*, vol. 2017, no. 1, pp. 1–11, Dec. 2017.

[20] W. Lo, H. Alqahtani, K. Thakur, A. Almadhor, S. Chander, and G. Kumar, "A hybrid deep learning based intrusion detection system using spatial-temporal representation of in-vehicle network traffic," *Veh. Commun.*, vol. 35, Jun. 2022, Art. no. 100471. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214209622000183

[21] D. Basavaraj and S. Tayeb, "Towards a lightweight intrusion detection framework for in-vehicle networks," *J. Sensor Actuator Netw.*, vol. 11, no. 1, p. 6, Jan. 2022. [Online]. Available: https://www.mdpi.com/2224-2708/11/1/6

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

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

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

[25] O. Avatefipour, A. S. Al-Sumaiti, A. M. El-Sherbeeny, E. M. Awwad, M. A. Elmeligy, M. A. Mohamed, and H. Malik, "An intelligent secured framework for cyberattack detection in electric vehicles' CAN bus using machine learning," *IEEE Access*, vol. 7, pp. 127580–127592, 2019.

[26] D. Stabili, M. Marchetti, and M. Colajanni, "Detecting attacks to internal vehicle networks through Hamming distance," in *Proc. AEIT Int. Annu. Conf.*, Sep. 2017, pp. 1–6.

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

[28] H. Li, L. Zhao, M. Juliato, S. Ahmed, M. R. Sastry, and L. L. Yang, "POSTER: Intrusion detection system for in-vehicle networks using sensor correlation and integration," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2017, pp. 2531–2533.

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

[30] L. Zhang, L. Shi, N. Kaja, and D. Ma, "A two-stage deep learning approach for can intrusion detection," in *Proc. Ground Vehicle Syst. Eng. Technol. Symp.*, Aug. 2018, pp. 1–11.

[31] M. Rumez, J. Lin, T. FuchB, R. Kriesten, and E. Sax, "Anomaly detection for automotive diagnostic applications based on N-Grams," in *Proc. IEEE 44th Annu. Comput., Softw., Appl. Conf. (COMPSAC)*, Jul. 2020, pp. 1423–1429.

[32] S. Brunner, J. Roder, M. Kucera, and T. Waas, "Automotive E/E-architecture enhancements by usage of Ethernet TSN," in *Proc. 13th Workshop Intell. Solutions Embedded Syst. (WISES)*, Jun. 2017, pp. 9–13.

[33] GuardKnox. *Zonal Architecture: The Foundation for Next Generation Vehicles*. Accessed: Jun. 30, 2022. [Online]. Available: http://learn.guardknox.com/zonal-architecturethe-foundation-for-next-generation-vehicles

[34] M. Weber, *Automotive OBD-II Dataset*. Karlsruhe, Germany: Karlsruhe Institute of Technology, 2019.

[35] B. I. Kwak, J. Woo, and H. Kim. (2016). *Driving Dataset*. Accessed: Jun. 30, 2022. [Online]. Available: https://ocslab.hksecurity.net/Datasets/driving-dataset

[36] H. Kang, B. I. Kwak, Y. H. Lee, H. Lee, H. Lee, and H. K. Kim, "Car hacking: Attack & defense challenge 2020 dataset," IEEE Dataport, Feb. 3, 2021, doi: 10.21227/qvr7-n418.

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

[38] *Road Vehicles Communication Between Vehicle and External Equipment for Emissions-Related Diagnostics—Part 6: Diagnostic Trouble Code Definitions*, document ISO 15031-6:2015, 2015.

TASNEEM A. AWAAD received the B.Sc. degree (Hons.) in computer engineering and software systems from Ain Shams University, Cairo, Egypt, in 2018, where she is currently pursuing the M.Sc. degree with the Faculty of Engineering. She worked at Siemens EDA, Cairo, as a Hardware Engineer with the Emulation Division for two years. She is currently a Teaching Assistant with the Faculty of Engineering, Ain Shams University. Her current research interests include digital VLSI, embedded systems, security, and machine learning.

M. WATHEQ EL-KHARASHI received the B.Sc. (Hons.) and M.Sc. degrees in computer engineering from Ain Shams University, Cairo, Egypt, in 1992 and 1996, respectively, and the Ph.D. degree in computer engineering from the University of Victoria, Victoria, Canada, in 2002. He is currently a Professor of computer organization with the Department of Computer and Systems Engineering, Ain Shams University, and also an Adjunct Professor with the Department of Electrical and Computer Engineering, University of Victoria. He has published 180 papers in refereed international journals and conferences and authored two books and seven book chapters. His current research interests include advanced system architectures, especially networks-on-chip, systems-on-chip and secure hardware, hardware architectures for networking (network processing units), and security; advanced microprocessor design, simulation, performance evaluation, and testability; and computer architecture and computer networks education.

MOHAMED TAHER received the B.Sc. (Hons.) and M.Sc. degrees in computer engineering from Ain Shams University, Cairo, Egypt, in 1996 and 2001, respectively, and the Ph.D. degree in electrical and computer engineering from The George Washington University, Washington, DC, USA, in 2006. He is currently an Associate Professor with the Department of Computer and Systems Engineering, Ain Shams University. His research interests include high-performance computing, reconfigurable computing, embedded systems, and computer architecture.

KHALID AMMAR (Member, IEEE) received the M.E. degree from Concordia University, Montreal, Canada, in 1986, and the Ph.D. degree in electrical engineering from the University of Sherbrooke, Canada, in 2001. He previously worked with Nortel Networks, Ottawa, Canada, as a Senior ASIC Applications Engineer. He is currently an Assistant Professor at the Department of Electrical and Computer Engineering, Ajman University, United Arab Emirates. His research interests include VLSI and embedded systems.