Detecting Vehicle Anomaly by Sensor Consistency: An Edge Computing Based Mechanism

Zichang Wang*, Fei Guo*, Yan Meng*, Huaxin Li*, Haojin Zhu* and Zhenfu Cao†
*Shanghai Jiao Tong University, 200240 Shanghai, China
{wzc1214, feiguo, yan_meng, lihuaxin0033}@sjtu.edu.cn, zhu-hj@cs.sjtu.edu.cn
†School of Computer Science and Software Engineering, East China Normal University, Shanghai, China
zfcao@sei.ecnu.edu.cn

***Abstract*—Autonomous vehicles are expected to be a disruptive technology that has the potential to revolutionize the human mobility. However, the recent research progress on intra-vehicle network (e.g., the revealing of a series of security vulnerabilities of CAN design) has demonstrated that the security issue still represents one of the major challenges of future self-driving cars. In this study, we propose a novel edge based anomaly detection system, coined VeAnDe, which exploits edge based sensor data fusion to identify the anomaly events. VeAnDe analyzes multiple correlations between different intra-vehicle sensors, and utilizes these correlations to examine whether an anomaly has occurred within the vehicle. More specifically, multiple correlations are organized as ring architecture to reduce the computation overhead. Furthermore, the major components of VeAnDe are embedded in edge computing devices, which enables VeAnDe to be more efficient and privacy-preserving. We evaluate the performance of VeAnDe under different scenarios, and our experimental results demonstrate its feasibility and efficiency.**

I. Introduction

The increasing adoption of sensors and Electronic Control Units (ECUs) in modern vehicles brings intelligence and convenience to our daily life. However, several attacks on the vehicles have been demonstrated, leading to an increasing attention to automotive security in the academia and the industry. Such attacks are mainly launched by accessing the internal Controller Area Network (CAN) bus through wireless channels and injecting/spoofing instruction messages inside a vehicle. By this way, the attackers can take over the control of the vehicle and deviate its system from a safe operational regime. For instance, Charlie *et al.* [1] proposed an attack which could control the multi-media, the power system and the braking system of vehicles without any physical access. Many approaches are proposed to thwart attacks on vehicles. For example, cryptology-based CAN bus protocols are the most intuitive solutions [2]. However, the practicability of this kind of solutions might be limited in practice, because the resource-constrained property of CAN bus cannot meet the high demand of real-time response [3].

Recently, designing practical and efficient anomaly detection solutions for intra-vehicle systems is becoming an important research topic, because of its advantage of identifying the attacks at an early stage and the ease of being compatible with existing vehicle systems. Machine learning based mechanisms [4], [5] have been proposed to achieve the anomaly detection. In the existing solutions, some behavior patterns of a vehicle are extracted to train a model in non-attack scenarios. Then the model is deployed to discover abnormal patterns and protect the vehicle against the various attacks. Considering the fact that, today there are in excess of 100 sensors onboard, which generate massive autonomous vehicle sensor data. Advances in the more powerful sensors (camera, lidar) and in-vehicle networking (e.g., Automotive Ethernet) will produce richer data, which call for a more scalable and time/bandwidth efficient anomaly detection scheme.

In this study, we propose a scalable and efficient vehicular anomaly detection system named VeAnDe. VeAnDe exploits multiple correlations between different intra-vehicle sensors as the criterion to detect anomalies. The basic observation is that some sensors readings are mutually correlated due to the existence of certain physical phenomena of a vehicle and any abrupt changes of correlations will indicate an occurrence of anomalies. Therefore, the correlations of sensors can be organized as a ring architecture that can bring significant advantages in our work. On the one hand, detecting the anomaly using a ring architecture incurs a lower computation overhead. On the other hand, the ring architecture takes multiple in-vehicle sensors into consideration simultaneously, which increases the robustness of the detection mechanism. Furthermore, VeAnDe leverages edge computing paradigm to offload the computing task to the nearest edge node, which is expected to further speed up the data aggregation from the multiple on-board vehicle sensors and achieve the real-time anomaly detection. Lastly, since the data can be processed by the local edge nodes rather than the cloud, it is expected to protect the sensitive data of the users from leaking to the untrusted cloud.

The main contributions of this paper are summarized as follows:

- We present a novel edge based anomaly detection architecture, which is expected to achieve high efficiency, bandwidth resource saving and privacy preservation based on emerging edge computing paradigm.
- We present VeAnDe, a real-time vehicle anomaly detection system, by analyzing the multiple correlations

Haojin Zhu is the corresponding author.

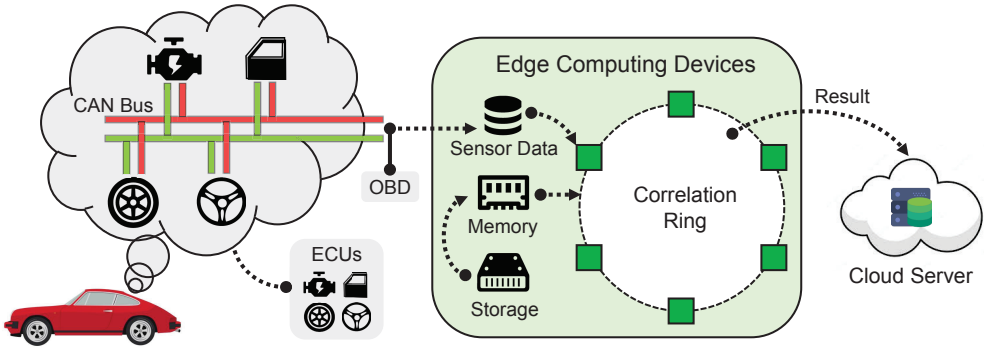


Fig. 1. The system overview.

between different in-vehicle sensors.

- We propose some novel algorithms to calculate the correlations, identify the correlation ring, and detect the vehicle anomaly.
- We implement and evaluate the anomaly detection performance of VeAnDe in different attack scenarios on a real-world vehicular data set. Our experimental results quantitatively show that VeAnDe achieves overall 95.3% detection accuracy with 1% false positive rate and demonstrate the effectiveness and robustness of VeAnDe.

The remainder of this paper is organized as follows. We introduce some preliminaries in Section II, and Section III describes the framework of the anomaly detection system, VeAnDe. In Section IV, we illustrate the evaluation and performance of the VeAnDe, and Section V brings the discussion and future work. Finally, Section VI provides some related work and Section VII concludes the whole paper.

II. Preliminaries

A. Controller Area Network

Controller Area Network (CAN), the *de facto* standard in-vehicle network protocol, prompts the modern automobile an integrated system that achieves real-time interactions with roads, vehicles and people [6]. As the central bus connecting all the ECUs, CAN contains information of each sensor as long as the related ECU is transmitting messages to the CAN. Thus, we can collect information from different ECUs for anomaly detection through CAN bus. With the employment of the Global Positioning System (GPS) sensor, the Inertial Measurement Unit (IMU) sensor and the external system, the robustness of the anomaly detection system is further improved since these systems are independent from the vehicles. And it is much more difficult for attackers to invade both the ECUs and external system simultaneously.

B. Correlation of In-vehicle Sensors.

The correlation between two variables describes how close these two have a relationship with each other. Since vehicles are cyber-physical systems, the correlation between different vehicle sensors reflects how similar they react to the same physical phenomenon, which forms a criterion for anomaly detection. For instance, most of vehicles have multiple speed measurements from different sources, such as rotational speed from sensors on wheels, GPS speed measured by location changes, and the speed that is calculated with the assistance of gearbox principal axis. Therefore, the correlation of in-vehicle sensors can be used to detect in-vehicle anomaly [7]. In this paper, we propose a novel method that leverages the correlations to detect the anomaly in the early stage. For example, if the tire speed is nearly 0 miles/hour while the GPS speed is high (e.g., 60 miles/hour), we would conclude that some anomalies have occurred since these two values violate the natural correlation between the tire speed and the GPS speed.

C. Edge Computing

Edge computing refers to the technology that moves the computations to the edge devices of the network, where the downstream and upstream data are on behalf of the cloud services and Internet of Things (IoT) services respectively [8]. In this study, we adopt edge computing for efficient anomaly detection due to the following merits. Firstly, computing at the edge of the network saves the bandwidth resources, since it saves the efforts of transmitting a huge number of intermediately computational data to the cloud server. Secondly, edge computing achieves shorter response time thanks to a closer distance to data sources and a smaller number of data needed to be transmitted. Thus, it can detect and respond to anomalies more quickly to avoid more severe damage to the vehicles. Thirdly, the edge computing prevents most of the data from being leaked through a potentially untrusted cloud server, because edge computing devices can complete most of the services locally so that the corresponding sensitive data won't be exposed to the Internet.

III. System Design

In this section, we propose an edge computing based mechanism named VeAnDe to detect the vehicle anomaly. The architecture of VeAnDe is shown in Fig. 1, and the VeAnDe is embedded in the edge computing device to gain the benefits of the edge computing. VeAnDe consists of four modules. In *Data Collection Module*, VeAnDe connects to the CAN bus through the On-board Diagnostic Interface, monitoring

```
main():
  if the vehicle stops:
    waiting;
  start two threads: Detection() and Fetch();
  when the threads exit:
    \\the intermediate data are the average and the
    \\standard deviation of the samples
    save the intermediate data to the storage;
Fetch():
  while(true):
    fetch the messages from the CAN bus;
    extract the data for the correlation analysis;
    if the engine stop:
      exit;
Detection():
  while(true):
    if the data of a period are not collected totally:
      waiting;
    if the detection is the first one:
      load the data from the storage;
    for i in the number of the correlation ring:
      calculate the new PCC Pi_new;
      if  Pi_new<Pi_old & |Pi_new-Pi_old|/(1-Pi_old)<ϵ:
        send out an alarm;
      else:
        calculate the new intermediate data;
    every 10 times of the period:
      transmit the statement to the transmit terminal;
    if the engine stop:
      exit;
```

Fig. 2. The algorithm of the VeAnDe.

and buffering the messages. In *Correlation Analysis Module*, VeAnDe selects appropriate sensor pairs based on the pre-collected data to build a correlation ring for the target vehicle. In *Anomaly Detection Module*, VeAnDe calculates the multiple correlations of the variables on each node of the correlation ring using real-time collected data, and determines whether an anomaly occurs in the vehicle. Once an anomaly is detected, *Result Submitting Module* enables the edge computing device to alert the driver and transmit the result to the cloud server. The architecture of VeAnDe is lightweight since there are only two memory blocks: one for buffering the new sensor data and the other for storing *intermediate data*. The whole algorithm of VeAnDe is shown in Fig. 2, and we elaborate the details of each module as follows.

A. Data Collection Module

As shown in Fig. 1, VeAnDe can be deployed at an edge computing device, which collects all the vehicle messages through the On-Board Diagnostic (OBD) Interface. VeAnDe is designed to passively read CAN messages through the OBD Interface and perform analyses and detections locally inside the edge computing device that is independent to the vehicle. Without interfering the normal running of the CAN bus, VeAnDe is expected to be resilient to the intrusion attacks towards the vehicles. Before launching the anomaly detection, VeAnDe pre-collects some sensor data, and sends them to *Correlation Analysis Module* to build the correlation ring. Then, in the driving scenario, the real-time collected data are sent to *Anomaly Detection Module* to detect anomalies.

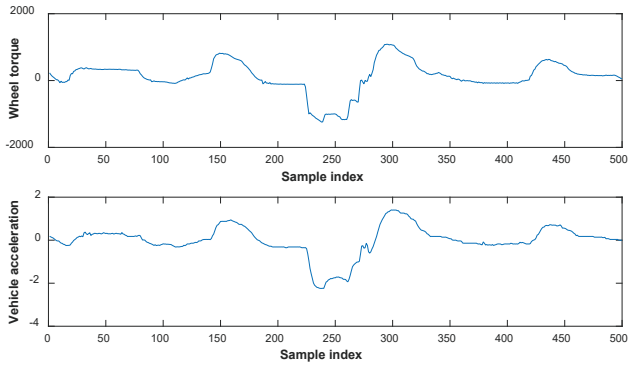


Fig. 3. The top figure shows the torque of wheel and the bottom is the acceleration of the vehicle.

B. Correlation Analysis Module

This module contains two steps: *correlations computing* and *correlation ring building*

1) Correlations Computing: Since VeAnDe detects anomaly based on correlations between different sensors (*e.g.*, speed of wheel VS GPS speed), it is crucial to choose an appropriate criterion to evaluate the correlations. In this paper, the Pearson Correlation Coefficient (PCC) is chosen to calculate the correlations between different sensors, which is formulated as:

$$Corr = \frac{\sum_{i=1}^{n}(X_i - \overline{X})(Y_i - \overline{Y})}{(n-1)\delta_X \delta_Y}, \tag{1}$$

where n is the length of re-sampled sequences X and Y, and δ_X, δ_Y are the sample standard deviations of X and Y, respectively. Generally, the closer the absolute value of $Corr$ is to 1 (or -1), the more positively (or negatively) linear relevant the two variables are assumed to be. For instance, as shown in Fig. 3, the data from the torque of wheel and the vehicle acceleration are relevant with PCC value of 0.9138. And if $Corr = 0$, it is supposed that there is no correlation between these two variables.

However, directly calculating the PCCs of all sensor pairs suffers from the following limitations. Firstly, since not all pairs of sensors have strong correlations, the pairs of sensors that are tightly correlated should be paid more attention than those irrelevant pairs in our anomaly detection mechanism. Secondly, some correlations might not be directly reflected on sensor readings based on our knowledge. For instance, the PCC between the wheel speed and acceleration is small, but the PCC between the differential of wheel speed and acceleration is large (*i.e.*, 0.8713 in our dataset). Lastly, some correlations associate with more than two sensors. For example, the acceleration is correlated with the difference between throttle and brake pedal readings.

Therefore, to capture the correlation between sensors via a more efficient way, VeAnDe firstly identifies the sensor pairs that are related to the same physical phenomenon. In addition, the sensor data in some pairs have been processed empirically such as differentiating the data of wheel speed. Then VeAnDe

calculates the PCCs of these selected sensor pairs' data. Table I illustrates some PCCs of sensor pairs from our dataset. And we can find that the correlations between the variables that measure the same physical quantity is close to 1. In other words, they are highly correlated. However, the correlation involving three or more variables is weakly correlated. Note that, the correlation between the left and the right wheel speed is not always 1, since making a turn requires different motions of two side wheels.

2) Correlation Ring Building: After calculating the correlations between sensor pairs, we can detect the vehicle anomaly based on these correlations. VeAnDe selects multiple correlations and organizes them in a ring architecture, which has the following advantages. First of all, only the correlations related to the sensors in the correlation ring need to be calculated, thus the computation overhead can be reduced. Then, since the computation complexity of the correlation ring is low, it can perform the detection process involving as many correlations as possible within the limited time and resources. Finally, using the ring architecture ensures every node has been examined twice, improving the accuracy of VeAnDe.

The simplest correlation ring contains only three nodes, such as acceleration of IMU in x-axis, angular velocity of IMU in y-axis as well as position of steer in z-axis. To build a more complex correlation ring, an intuitive method is to construct a map structure for all sensor pairs, and find the one with enough correlations. Fig. 4 illustrates one of the correlation rings for our vehicle data, which consists of 10 variables and 9 nodes. In this ring, the difference between the throttle pedal and brake pedal is one node. And the arrow represents that the differential of the variable at the tail is correlated with that at the head.

TABLE I
CORRELATION PAIRS

ID	Variable 1	Variable 2	*Corr*
1	time	GPS time	1.0000
2	speed of left front wheel	speed of left rear wheel	1.0000
3	speed of left front wheel	speed	0.9999
4	speed of left front wheel	speed of right front wheel	0.9998
5	speed	GPS speed	0.9996
6	position of steer	wheel angle	0.9951
7	fuel	integration of GPS speed	-0.9651
8	differential of speed	acceleration	0.9437
9	acceleration of IMU in x-axes	angular velocity of IMU in z-axes	0.9306
10	torque of wheel	acceleration	0.9138
11	acceleration	acceleration of IMU in y-axes	0.9066
12	differential of speed of right front wheel	acceleration	0.8713
13	torque of brake	brake pedal	0.8673
14	wheel angle	angular velocity of IMU in z-axes	0.6558
15	acceleration	throttle pedal - brake pedal	0.6377
16	position of steer	acceleration of IMU in x-axes	0.5331

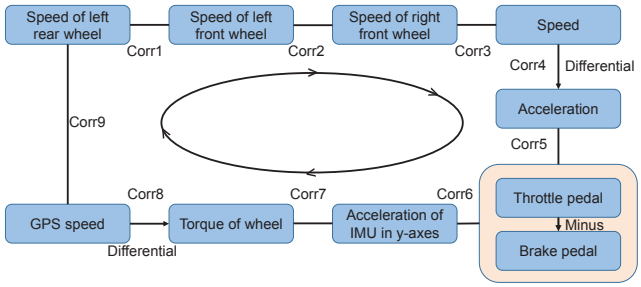


Fig. 4. The correlation ring.

C. Anomaly Detection Module

In this module, VeAnDe calculates n correlations, where n refers to the number of the nodes in the correlation ring, and generates the detection result, which indicates whether there is an anomaly. To eliminate the impact of noises on the PCC and ensure the sensitiveness of the detection, we employ the sliding window method in the detection. In each time window, totally 1000 samples are used for calculating every PCC value, and the sliding window step contains 100 samples, which is represented by n_2. VeAnDe fetches the *intermediate data* from the memory device, which is saved at the end of the last normal driving trip and used in the next detection window. The *intermediate data* includes $Corr_{n_1}$, the average $\overline{X_{n_1}}$ and the standard deviation δ_{n_1} of the former n_1 samples. Then, VeAnDe calculates the PCCs of the sample set in the new detection window and compares it with the former one for every correlation pairs to adaptively determine whether an anomaly occurs. We introduce the method calculating the PCCs in new detection window as follows.

When the *intermediate data* are fetched, we generate the $Corr$ of all the $n_1 + n_2$ samples with the numerical value of the last n_2 samples. After calculating the average $\overline{X}$ of the $n_1 + n_2$ samples, the standard deviation δ is :

$$\delta = [\frac{1}{n_1 + n_2} \sum_{i=1}^{n_1+n_2} (X_i - \overline{X})^2]^{\frac{1}{2}}, \tag{2}$$

Then we have:

$$\begin{aligned}(n_1 + n_2)\delta^2 = \sum_{i=1}^{n_1}[(X_i - \overline{X_{n_1}})^2 + 2(X_i - \overline{X_{n_1}})(\overline{X_{n_1}} - \overline{X}) \\ +(\overline{X_{n_1}} - \overline{X})^2] + \sum_{i=n_1+1}^{(n_1+n_2)} (X_i - \overline{X})^2,\end{aligned} \tag{3}$$

where

$$\sum_{i=1}^{n_1}(X_i - \overline{X_{n_1}})^2 = n_1\delta_{n_1}^2, \tag{4}$$

$$\sum_{i=1}^{n_1}[2(X_i - \overline{X_{n_1}})(\overline{X_{n_1}} - \overline{X})] = 0. \tag{5}$$

Since $(\overline{X_{n_1}} - \overline{X})^2$ and $\sum_{i=n_1+1}^{n_1+n_2}(X_i - \overline{X})^2$ are easy to calculate, the standard deviation of all the samples δ can be

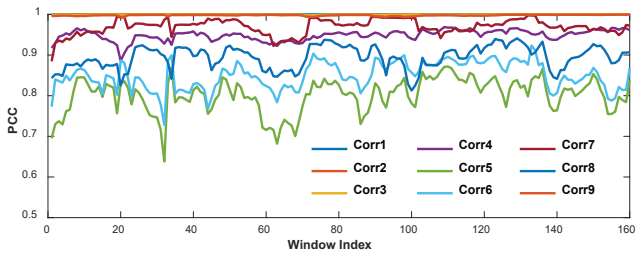


Fig. 5. The PCCs of correlation ring.

calculated. Using the same method, $\sum_{i=1}^{n}(X_i - \overline{X})(Y_i - \overline{Y})$ in Eq. (1) can also be calculated. Thus we obtain the $Corr$ in new detection window based on the current samples and the *intermediate data* of the former samples.

After getting the correlations in the i-th time window, VeAnDe determines if there is an anomaly detected when the following two inequalities are both satisfied:

$$Corr_i \leq Corr_{i-1} < 1, \tag{6}$$

$$\frac{|Corr_i - Corr_{i-1}|}{1 - Corr_{i-1}} \geq \epsilon, \tag{7}$$

where $Corr_i$ and $Corr_{i-1}$ are the correlations in the i-th and $\{i-1\}$-th time window for a certain sensor pair, and ϵ is the threshold to control detection accuracy and false positives.

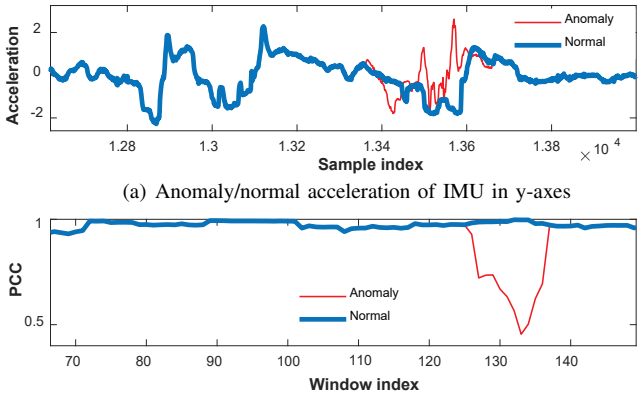
In practice, the detection in this module should be time-efficient enough to guarantee that no data from the CAN bus would be piled up. Otherwise, the memory would be exploded due to the accumulation effect of data transmission in a long term. Our evaluation in Section IV will show that the ring architecture is able to meet the requirement of the time efficiency.

D. Result Submission Module

In this module, VeAnDe gets the result from *Anomaly Detection Module*. If the vehicle is in a normal state, VeAnDe would transmit the result to the cloud server in a fixed period without any other operation. Otherwise, VeAnDe would trigger an alert and the edge computing device would instantly transmit the status to the cloud server. In this module, VeAnDe can save much bandwidth and energy by transmitting only the essential data. In the meantime, VeAnDe also validates the connection of the edge computing device and the cloud server through the periodic acknowledgment (ACK) messages from the cloud server.

IV. Evaluation

In this section, we evaluate the performance of our proposed VeAnDe. We first introduce our dataset and demonstrate the feasibility of exploiting PCC to detect vehicle anomaly. Moreover, we evaluate the effectiveness of the anomaly detection of VeAnDe under different scenarios, and measure the system overhead.



(a) Anomaly/normal acceleration of IMU in y-axes

(b) Anomaly/normal PCC

Fig. 6. The acceleration data and the PCC in normal/anomaly scenario.

A. Dataset

The dataset is from the Open Sourcing 223GB of Driving Data [9], collected in Mountain View, CA by Lincoln MKZ. The ECU messages in this dataset are collected under different weathers and are recorded by the Robot Operate System (ROS) automatically. We first extract the ECU data from the ROS messages, and then filter the non-significant data, such as the pictures taken by cameras on the vehicle.

The processed dataset contains 165 variables and more than 30 million data items. Note that, since the original sample rates of the data are different, we re-sample all the data to 10Hz for analyzing them more efficiently. Since the original dataset is collected in normal driving scenarios, we generate the anomaly data by modifying the normal data to simulate the anomaly scenarios, which will be introduced in details in the following subsections.

B. Analysis of Effectiveness

In this section, we demonstrate the feasibility of VeAnDe on detecting vehicle anomalies. The correlation ring extracted from our dataset is shown in Fig. 4. Fig. 5 illustrates the PCC variations in the correlation ring during the normal driving scenarios. It is observed that all PCCs are higher than 0.6, which reveals the strong correlations among all sensors. Therefore, VeAnDe could regard the vehicle status as normal according to PCCs.

We perform a case study to show the anomaly detection process. In Fig. 6(a), the blue line represents the normal data collected from the IMU in the y-axes while the abnormal data is described in the red line. And the normal PCC between acceleration of IMU in y-axes and wheel torque (*i.e.*, *Corr7* in Fig. 4) is shown as the blue line in Fig. 6(b). We can see the normal PCC in Fig. 6(b) is always higher than 0.9. Then, we simulate abnormal acceleration by replacing the samples around $Sample_{13400}$ with samples collected from another trip, then the corresponding PCC is shown as the red line in the Fig. 6(b). It is observed that when VeAnDe calculates the PCC of *Corr7* for the abnormal data, the PCC drops to 0.45 drastically.

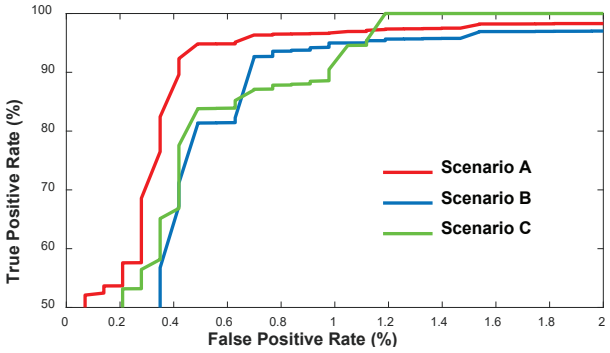


Fig. 7. The ROC curves of the VeAnDe performance. Scenario A, B, C is replacing continuous samples, multiplying samples, randomly replacing samples respectively.

Therefore, VeAnDe can detect the anomaly if a dropping cusp of PCC has appeared.

C. Overall Performance Evaluation

In this subsection, we evaluate the overall performance of VeAnDe under different anomaly scenarios. We simulate the abnormal data by modifying the normal data in the following three attack strategies. The first is replacing the continuous samples with the normal data of a different trip. An attack scenario (Scenario A in Fig. 6) related to this strategy is that a sensor is hijacked by the attack and reports the false message. The second strategy is multiplying some continuous samples by a parameter δ (*e.g.*, 1.2 in our experiment). This strategy simulates the scenario (Scenario B) that the sensor is no longer accurate. The final strategy is choosing samples in normal scenarios randomly and intermittently, and replacing them with samples from other trips. This strategy simulates the scenario (Scenario C) that a sensor is suffering from the wireless message injection attack.

We test 10000 attacks for each scenario and adjust the parameter ϵ to get the ROC curves. In each attack, we randomly choose a target node (sensor) from the correlation ring, and modify 300 message samples of this node using one of the above three strategies. As shown in Fig. 7, VeAnDe can achieve the average 95.3% detection accuracy with 1% false positive rate (FPR). Even in the worst case (Scenario C), VeAnDe can still achieve 94.5% detection accuracy with 1% FPR. It means that VeAnDe is promising to accurately detect anomalies under different scenarios.

Besides, the average time required for performing an anomaly detection is only 3.2ms for the correlation ring with 10 variables, which is acceptable in practice. And more, to save the sensor data and the intermediate data of 10 moving windows, VeAnDe only needs 1MB memory for the correlation ring with 10 variables. the In summary, our experimental results prove the effectiveness and the efficiency of VeAnDe on detecting vehicle anomalies.

V. Related Work

Vehicle attack. With the prevalence of ECUs in modern vehicles, security issues have been studied by recent researches. Rouf *et al.* [10] utilized vulnerabilities of Tire Pressure Monitoring Systems (TPMS) to inject spoofed messages illegally turn on the low tire pressure warning lights on a vehicle. The reason behinds this attack is that the TPMS did not employ any countermeasures for intrusion attacks. The Keen Lab[11] demonstrated attacks to Tesla motors remotely, which can control arbitrary CAN bus and ECUs without any physical access.

Defense Mechanisms for vehicles. To enhance the security of modern vehicles against the above attacks, several defense mechanisms have been proposed. Most of them [12], [13] utilized message authentication protocols to protect the messages broadcast on the CAN bus. However, they would make the real-time vehicle systems suffer from heavy communication delays. Furthermore, Lu *et al.* [14] proposed the method for filtering injected false data in wireless sensor networks. They utilized the random graph metrics of sensor node deployment and the cooperative bit-compressed authentication technique to filtering the injected data which may be applied to intra-vehicle sensor network.

Anomaly detections. As the first line to protect a system, there are many research work for the anomaly detection [15]. Narayanan *et al.* [16] utilized the Hidden Markov Model to complete the anomaly detection task. Ganesan *et al.* [7] used the cluster analysis and the pair-wise correlation to determine the context and detect the vehicle anomaly. Different from them, in this paper, we utilize the correlation ring and the edge computing technology to build a more efficient and robust anomaly detection mechanism.

VI. Conclusion

In this paper, we propose VeAnDe, a novel anomaly detection mechanism to enhance the vehicle security. VeAnDe utilizes multiple correlations between different intra-vehicle sensors as the criterion to detect the vehicle anomaly. To reduce the computation overhead and improve the privacy of vehicle information, the correlations are organized in the ring architecture and the edge computing technology is employed. We perform comprehensive experiments to evaluate the performance of VeAnDe, and the results show that VeAnDe can achieve average 95.3% detection accuracy with 1% false positive rate.

Acknowledgment

This work is supported by Shanghai Committee of Science and Technology, China (No. 18511111502).

References

[1] C. Miller and C. Valasek, "Remote exploitation of an unaltered passenger vehicle," *Black Hat USA*, vol. 2015, 2015.

[2] A. V. Herrewege, D. Singele, and I. Verbauwhede, "Canauth - a simple, backward compatible broadcast authentication protocol for can bus," in *Ecrypt Workshop on Lightweight Cryptography*, 2011, p. 7.

[3] P. S. Murvay and B. Groza, "Source identification using signal characteristics in controller area networks," *IEEE Signal Processing Letters*, vol. 21, no. 4, pp. 395–399, 2014.

[4] A. Bezemskij, G. Loukas, R. J. Anthony, and D. Gan, "Behaviour-based anomaly detection of cyber-physical attacks on a robotic vehicle," in *International Conference on Ubiquitous Computing and Communications and 2016 International Symposium on Cyberspace and Security*, 2017, pp. 61–68.

[5] 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 *ACM Sigsac Conference*, 2017, pp. 2531–2533.

[6] J. Zhong, S. Du, L. Zhou, H. Zhu, F. Cheng, C. Chen, and Q. Xue, "Security modeling and analysis on intra vehicular network," in *2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)*, Sept 2017, pp. 1–5.

[7] A. Ganesan, J. Rao, and K. Shin, "Exploiting consistency among heterogeneous sensors for vehicle anomaly detection," SAE Technical Paper, Tech. Rep., 2017.

[8] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.

[9] Open sourcing 223gb of driving data. [Online]. Available: https://medium.com/udacity/open-sourcing-223gb-of-mountain-view-driving-data-f6b5593fbfa5

[10] I. Rouf, R. Miller, H. Mustafa, T. Taylor, S. Oh, W. Xu, M. Gruteser, W. Trappe, and I. Seskar, "Security and privacy vulnerabilities of in-car wireless networks: A tire pressure monitoring system case study," in *Usenix Security Symposium, Washington, Dc, Usa, August 11-13, 2010, Proceedings*, 2010, pp. 323–338.

[11] (2017) New car hacking research: 2017, remote attack tesla motors again. [Online]. Available: https://keenlab.tencent.com/en/2017/07/27/New-Car-Hacking-Research-2017-Remote-Attack-Tesla-Motors-Again/5

[12] B. Groza, S. Murvay, A. V. Herrewege, and I. Verbauwhede, "Libracan: A lightweight broadcast authentication protocol for controller area networks," *Acm Transactions on Embedded Computing Systems*, vol. 16, no. 3, p. 90, 2017.

[13] O. Hartkopp and R. M. SCHILLING, "Message authenticated can," in *Escar Conference, Berlin, Germany*, 2012.

[14] R. Lu, X. Lin, H. Zhu, X. Liang, and X. Shen, "Becan: A bandwidth-efficient cooperative authentication scheme for filtering injected false data in wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 1, pp. 32–43, Jan 2012.

[15] M. Xie, J. Hu, S. Han, and H. H. Chen, "Scalable hypergrid k-nn-based online anomaly detection in wireless sensor networks," *IEEE Transactions on Parallel & Distributed Systems*, vol. 24, no. 8, pp. 1661–1670, 2013.

[16] S. N. Narayanan, S. Mittal, and A. Joshi, "Obdsecurealert: An anomaly detection system for vehicles," in *IEEE International Conference on Smart Computing*, 2016, pp. 1–6.