Vehicular Communications 14 (2018) 52–63

Contents lists available at ScienceDirect

Vehicular Communications

www.elsevier.com/locate/vehcom

Anomaly intrusion detection method for vehicular networks based on survival analysis

Mee Lan Han, Byung Il Kwak, Huy Kang Kim *

Graduate School of Information Security, Korea University, Seoul, Republic of Korea

ARTICLE INFO

Article history:
Received 9 May 2018
Received in revised form 14 September 2018
Accepted 18 September 2018
Available online 26 September 2018

Keywords:
In-vehicle network
Anomaly detection
Intrusion detection
Survival analysis

ABSTRACT

In recent years, alongside with the convergence of In-vehicle network (IVN) and wireless communication technology, vehicle communication technology has been steadily progressing. Furthermore, communication with various external networks—such as cloud, vehicle-to-vehicle (V2V), and vehicle-to-infrastructure (V2I) communication networks—further reinforces the connectivity between the inside and outside of a vehicle. On the contrary, this means that the functions of existing vehicles using computer-assisted mechanical mechanisms can be manipulated and controlled by a malicious packet attack. Therefore, diversified and advanced architectures of vehicle systems can significantly increase the accessibility of the system to hackers and the possibility of an attack. This paper proposes an intrusion detection method for vehicular networks based on the survival analysis model. Our main aims were to identify malicious CAN messages and accurately detect the normality and abnormality of a vehicle network without semantic knowledge of the CAN ID function. To this end, normal and abnormal driving data were extracted from three different types of vehicles and we evaluated the performance of our proposed method by measuring the accuracy and the time complexity of anomaly detection by considering three attack scenarios and the periodic characteristics of CAN IDs. Based on the results, we concluded that a CAN ID with a long cycle affects the detection accuracy and the number of CAN IDs affects the detection speed. The difference in the detection accuracy between applying all CAN IDs and CAN IDs with a short cycle is not considerable with some differences observed in the detection accuracy depending on the chunk size and the specific attack type. High detection accuracy and low computational cost will be the essential factors for real-time processing of IVN security. Taken together, the results of the present study contribute to the current understanding of how to correctly manage vehicle communications for vehicle security and driver safety.

1. Introduction

Modern vehicles are rapidly changing from purely mechanical devices to software-controlled devices. Vehicles are being integrated with information technology, such as wireless communications and GPS, as well as with telematics services or intelligent vehicle information systems. In-vehicle technologies and services are evolving into a large platform. Therefore, the number of in-vehicle electronic devices is rapidly increasing, and in-vehicle systems are becoming more complex to support connections among the many devices. To efficiently receive communication services, in-vehicle devices must be seamlessly connected to an external network system; however, this increases the risk of exposure of such systems to potential internal/external threats.

Vehicle attacks can be analyzed from two viewpoints: the attack surface and the attack vector. First, in the attack surface aspect, there is a direct attack surface through the On-Board Diagnostic II (OBD-II) port and an indirect attack surface through the firmware of the media player manufacturer. In fact, in July 2012, in the UK, thieves stole a parked BMW, bypassing the vehicle's anti-theft system, via a laptop computer and a smartphone connected the OBD-II port [1]. Likewise, Checkoway, Stephen, et al., reported an indirect attack using malicious MP3 files to control a vehicle [2]. With respect to the attack vector, potential attacks are a short-range attack that occurs through the vulnerability of the Bluetooth protocol that supports connections among the in-vehicle audio video navigation (AVN) systems, and a remote wireless attack that occurs through the communication channel between a telematics module and a smartphone application. In 2012, a group of researchers from the Korea University Graduate School of Information Security demonstrated several approaches to vehicle hacking. Specifically, they manipulated the dashboard and the alarm sound through a malware-infected vehicle inspection application [3]. In a

* Corresponding author.
E-mail addresses: blosst@korea.ac.kr (M.L. Han), kwacka12@korea.ac.kr (B.I. Kwak), cenda@korea.ac.kr (H.K. Kim).

https://doi.org/10.1016/j.vehcom.2018.09.004
2214-2096/

2015 study, Charlie Miller and Chris Valasek succeeded in controlling similar functions by injecting an anomalous message on the CAN network by hacking Bluetooth, telematics, and the mp3 parser of the radio in a vehicle [4,5]. Likewise, a team from the Keen Security Lab in China demonstrated how to connect a vehicle to an intentionally designated malicious Wi-Fi hotspot, inducing remote control of the vehicle's brakes, locking system, and navigation by accessing the control system in the vehicle [6].

Previously, attacks through the OBD-II port were possible only in the case of an OBD-II connector directly connected to a laptop computer, so it was difficult to attack continuously against different types of vehicles. However, new dongle applications of the OBD-II connection type have appeared on the market at affordable prices, which can be applied to a number of vehicles, regardless of manufacturer, model, or release year [7]. Remote surveillance and control of a vehicle using a dongle application may cause safety threats and/or driver confusion. As vehicles are basic means of transportation, even a small risk factor in the vehicle can not only lead to vehicle breakdown, but can also significantly compromise the safety of the driver, passengers, and pedestrians. Therefore, vehicle security is a crucial issue that must be considered when it comes to safety and public order.

The intrusion detection method proposed in the present paper is based on an anomaly detection algorithm used to detect the occurrence of a suspicious pattern in the usual pattern information. Unlike a misuse detection method, the proposed method can detect an unknown attack and does not require regular update of signatures concerning attack patterns. Second, data collection and analysis are targeted not outside the vehicle (i.e., the driving data recorded in the cloud or server system), but inside the vehicle (i.e., Controller Area Network (CAN) data). In addition, we propose the mechanically reliable and relatively simple host-based intrusion detection system (IDS) technology. The low-performance of Electronic Control Unit (ECU) devices and an unstable power supply make it difficult to apply existing network security solutions within vehicles. Third, the proposed method is not intended to perform safety diagnosis or check the condition of the vehicle after an incident. Rather, it enables real-time analysis using the data extracted from the vehicle while in operation. Although our method does not allow changing from an anomalous status to normal status, we consider countermeasures to store the log data and to alert the driver, passengers, and other parties involved to the anomalous condition in the vehicle.

The present paper makes the following contributions:

- We extend the survival analysis model which estimates the survival function to detect anomalies to the CAN of a vehicle.
- We define and evaluate three attack scenarios that can directly attack all vehicles via the CAN.
- In the experiments, we propose and evaluate a novel general-purpose intrusion detection method. This means that our method is not subordinate to the CAN ID, which can vary in characteristics depending on the manufacturer, model, and release year of the vehicle.

The remainder of this paper is structured as follows. Section 2 provides a summary of the literature related to our work. Section 3 explains the system setup and the attack scenarios. The detailed implementation of our scheme for the intrusion detection approach is presented in Section 4. Section 5 reports the experimental results and discusses the outcomes. Finally, Section 6 concludes the paper and suggests directions of further research.

2. Background and related work

Vehicle network In the vehicle, various control modules (e.g., electric control unit, transmission control unit, airbag control unit, and anti-lock braking system) are installed for performance, safety, and convenience. Each control module strengthens the overall vehicle function by sharing performance values with other modules depending on their needs. Therefore, a vehicle system requires an internal communication system to organically control dozens of control modules. A CAN communication system shares the signals of the control modules through a network. In other words, this system sends and receives data through a parallel connection of ECUs. The CAN bus connected with ECUs communicates as a multi-master type without a central master that can control all nodes [8].

Fig. 1 depicts the extended CAN message structure and bus arbitration. The CAN transmits data in the form of packets called message frames. The format of a CAN message frame can be categorized into the following three main parts. First, the ID in the arbitration field is used to identify the message and assign its priority. A Remote Transmission Request (RTR) is used to distinguish between a remote frame, and a data frame. A logic 0 RTR represents a data frame, and logic 1 RTR indicates a remote frame. In the event of a collision between CAN messages on the receiving node, each ID is compared by bits. Only the messages having the lowest ID value (i.e., the highest priority) are transmitted continuously and transmission of the remaining messages is immediately stopped [9]. The Data Length Code (DLC) in the control field indicates the number of bytes, and DLC values vary depending on the vehicle type. Finally, the data field contains the data to be transferred from one node to another and consists of 0 to 8 bytes [10,11]. The CAN ID varies in number and type depending on the vehicle model. Even if the CAN ID used in different vehicles is numerically identical, the defined and assigned characteristics of CAN IDs may vary. As the interpretation of the CAN ID is highly restricted if the manufacturer's manual concerning CAN messages is not available, it is necessary to secure an algorithm that is not dependent on the CAN ID (i.e., a general-purpose intrusion detection method applicable to all vehicles).

IDS for vehicle security To reinforce vehicle security and to ensure the safety of drivers and passengers, a variety of security systems and solutions are being currently developed and investigated in both industry and academia. In recent years, numerous vulnerabilities within vehicles have been exposed, and extensive research has addressed algorithms to detect attacks on these.

Taylor et al. applied a flow-based method to the CAN message data. Their flow-based method evaluates several parameters using frequency and the average of CAN message occurrence. The authors validated the effectiveness within the confines of injection type, duration, range, and frequency of CAN messages [12].

Müter et al. and Marchetti et al. suggested several information-theoretic measures, such as conditional self-information, entropy and relative entropy, for anomaly detection. While the entropy of normal status is relatively stable, the entropy of an anomalous status produces considerable deviations in statistical characteristics. This detection method does not require semantic knowledge of the CAN ID function and has the advantage of being instantly applicable to the CAN buses of all vehicles with a low rate of false-positives [13,14].

Research has also been conducted on the time interval of the CAN ID [15]. The available findings suggest that while the time interval in which the CAN ID occurs is stable and regular in the normal status, the time interval of the CAN ID under injection attacks, such as a flooding attack or a fuzzy attack is shorter than in the normal status.

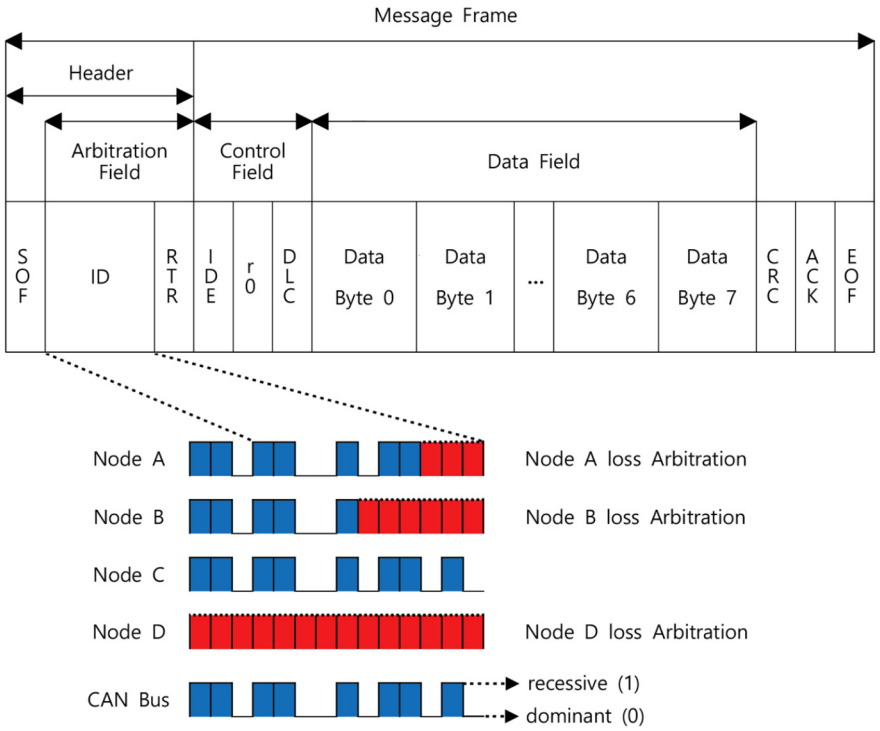

Fig. 1. Extended CAN message structure and CAN bus arbitration.

Marchetti et al. detected anomalous status by using a transition matrix defined as patterns of the reiterative CAN ID sequence. This research was divided into two steps: the training phase (i.e., occurrence of only CAN IDs without attack data) and the detection phase (i.e., the traffic to analyze and identify attack data using the transition matrix). The analysis begins with a "False" value (abnormal) and, if the combination of CAN ID sequences is correct, it is replaced with a "True" value to generate a valid matrix [16].

Another detection method identifies the boundaries of normality and abnormality by learning the CAN ID sequence without prior knowledge of a CAN message. In this respect, Markovitz et al. proposed a detection model based on ternary content addressable memory that can efficiently operate on both hardware and software [17]. To detect anomalies of the CAN network, Taylor et al. proposed an anomaly detector based on long short-term memory using a recurrent neural network trained to predict the ID value of the subsequent CAN message sent from each sender on the bus, and the error is used as the detection signal. If the subsequent CAN message occurs with a higher bit error than the normal range, it may be treated as an anomaly [18].

Ganesan et al. and Li et al. described an intrusion detection method based on correlations and redundancy among multiple sensors for an IVN attack. For instance, the act of pressing the accelerator pedal and the act of changing gears cause the throttle valve open, the engine pump rotates faster, and the RPM value and vehicle speed increase. When an attack or breakdown occurs in a vehicle, the balance of positive or negative correlations among the vehicle sensors may be unstable, and their expected values may deviate from their valid range. This makes it possible to detect attacks or system faults that cause anomalous correlations between pairs of sensors [19,20].

Several companies are attempting to develop a variety of security systems and solutions for IVNs. For instance, ESCRYPT is studying countermeasures against potential vulnerabilities of IVNs; the IDPS system serviced by ESCRYPT monitors CAN traffic and detects anomalous status [21]. Similarly, the security system serviced by Symantec monitors CAN traffic and detects anomalous status in real time using machine learning techniques. After learning the normal status of all vehicle models, it enhances their security and safety by sending a notification if an electronic attack or a simple malfunction occurs in the vehicle [22].

Survival analysis model Survival analysis can also be referred to as time-to-event analysis. This type of analysis embraces statistical methods that focus on the time until an event occurs. Survival analysis is widely used in the medical field to estimate the effects of treatment methods and to determine prognostic factors for patient survival. Alongside with being widely used in engineering and natural sciences, survival analysis is also used in social sciences in reference to the duration of an event (e.g., poverty and non-poverty; marriage and divorce). For a survival analysis, the survival time and current status information are required. The survival time refers to the period from a start point in time to a finish point in time for measuring the current status, and the current status is used to confirm whether the patient has survived or not at that time point. Survival analysis is characterized by the inclusion of uncertain data regarding the occurrence of an event. These uncertain data are called the censored data. Although the survival time of the censored data is unclear, it does include partial information in that the event (i.e., participant(s) death) did not occur before censoring. The advantage of this approach is that the data can be analyzed by making the most of these features in the survival analysis [23,24]. The survival function, which is one of the fundamental concepts of survival analysis, can be explained as follows:

- Definition of the survival function at time (t): the probability that a patient will survive until time (t).
  $S(t) = P$ (an individual survives longer than time (t))
- If there are no censored data, the survival function at time (t) can be estimated as an empirical distribution.
  $S(t)$ = number of survivors beyond time (t)/total number of patients

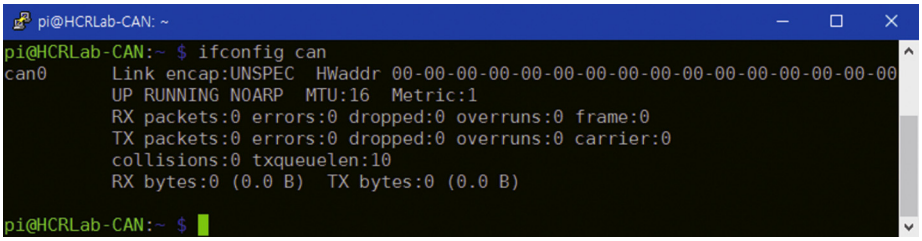

Fig. 2. Can0 interface activated for attack in Raspberry Pi3.

3. Vehicle attack

3.1. Attack model

IVN attacks can be categorized into several types, such as masquerade attack, fabrication attack, suspension attack, fuzzy attack, and replay attack.

The masquerade attack is not a direct vehicle attack, but is a way to conceal an attack. The purpose of this attack is to manipulate an ECU. This attack enables one of the two compromised ECUs to stop the transmission of a CAN message which flows periodically, and then causes another compromised ECU to send a CAN message on its behalf. This attack adopts signal characteristics similar in time interval to CAN messages transmitted from legitimate and safe ECUs [5,9].

A fabrication attack causes a compromised ECU to fabricate and inject CAN messages to a receiver ECU by overriding the CAN messages transmitted from legitimate and safe ECUs [9,25].

As a type of Denial-of-Service (DoS) attack, the suspension attack can prevent an intended CAN message's delivery. That is, it is impossible to send CAN messages from a compromised ECU or receive CAN messages from legitimate and safe ECUs [9]. The DoS attack creates high priority CAN IDs that fully occupy the CAN bus, so it causes a problem—namely, that CAN messages of other legitimate and safe ECUs cannot be transmitted. In addition, there are other ways to directly attack a certain ECU. For instance, Cho et al. succeeded in suspending the sending and receiving of CAN messages through the bus-off status signal for a certain ECU [26]. The bus-off is a unit with respect to the fault confinement of each ECU node. An ECU node goes into the bus-off state when the transmit error counter is greater than 255. In this mode, the ECU node cannot send or receive CAN messages or transmit error frames of any kind. Because the ECU node begins to transmit again if the fault condition is removed, the error states (i.e. error-active, error-passive, bus-off) have no effect on the CAN bus [27]. However, in the bus-off attack, an attacker who remote access to the CAN network performs simultaneous transmission of bits in fields. The bus-off node will be switched off in a target ECU due to the simultaneous transmission. As a result, an attacker can intentionally suspend the target ECU by using the bus-off attack [28].

The fuzzy attack is an attack that randomly injects compromised ID, DLC, and Data fields into the vehicle network. Even if an attacker does not have any specific information regarding CAN messages, s/he can easily attack a vehicle with the fuzzy attack. A vehicle attack may succeed as soon as the compromised IDs used in the fuzzy attack become consistent with the legitimate and safe CAN messages. However, if the compromised IDs are not included in the list of original CAN IDs, vehicle control by an attacker may be limited [29].

The replay attack causes a problem by injecting a set of CAN messages extracted and logged in a certain order into the vehicle network. The advantage here is that it does not require the reverse-engineering process for interpreting the meaning and function of the IDs and Data fields. However, it is difficult to accurately inject the log data into the vehicle network in accordance with a legitimate time and ID sequence [30,31].

3.2. Setup for collecting data

The present study is based on CAN messages obtained through an in-vehicle OBD-II port [2]. To efficiently design and evaluate our proposed algorithm, real vehicle operating data were collected. The collected data were divided into two types of driving data. One dataset is the normal driving data that occurs without an attack. The other dataset is abnormal driving data that occurs when an attack is performed. The data were collected using Raspberry Pi3[1] and the PiCAN2[2] devices connected by a serial peripheral interface, and the DB9 port of PiCAN2 was connected to the vehicle's OBD-II port. We ran a program to perform attacks using the python-can library (Python, version 3.4.2). In accordance with the communication speed of the CAN bus in the vehicle, the speed of the can0 interface of the Raspberry Pi3 was set to 500 Kbps. Fig. 2 shows the can0 interface activated for an attack in the Raspberry Pi3. To validate the general-purpose intrusion detection algorithm proposed in our paper, we extracted driving data from three different types of vehicles: the HYUNDAI YF Sonata 2010, a representative midsize vehicle of South Korea, the KIA compact SUV Soul 2015, and the CHEVROLET mini-compact vehicle Spark 2015 produced by GM Korea company, a subsidiary of General Motors.

3.3. Attack scenarios

In the present study, we focused on the following three attack scenarios that can immediately and severely impair in-vehicle functions or deepen the intensity of an attack and the degree of damage: Flooding, Fuzzy, and Malfunction. To substantiate the three attack scenarios, two different datasets were produced. One of the datasets contained normal driving data without an attack and was used for measuring the ground-truth value explained in Section 4. The other dataset included the abnormal driving data that occurred when an attack was performed. In particular, we generated attack data in which attack packets were injected for five seconds every 20 seconds for the three attack scenarios. Fig. 3 shows the three typical attack scenarios against an IVN.

Flooding attack As mentioned in Section 2, the CAN is a multi-master network, and the collision of CAN messages sent from multiple ECU nodes is arbitrated depending on their priority. The CAN message sent from an ECU node does not contain the address of the sender and receiver. Instead, when CAN messages transmitted from several different sender ECU nodes are simultaneously transmitted to a receiver ECU node, the values of the CAN IDs are compared to determine the priority of the CAN message to be accepted first. The lower the value of a CAN ID is, the higher its priority

[1] A series of small single-board computers developed in the United Kingdom by the Raspberry Pi Foundation.

[2] CAN-bus board for Raspberry Pi.

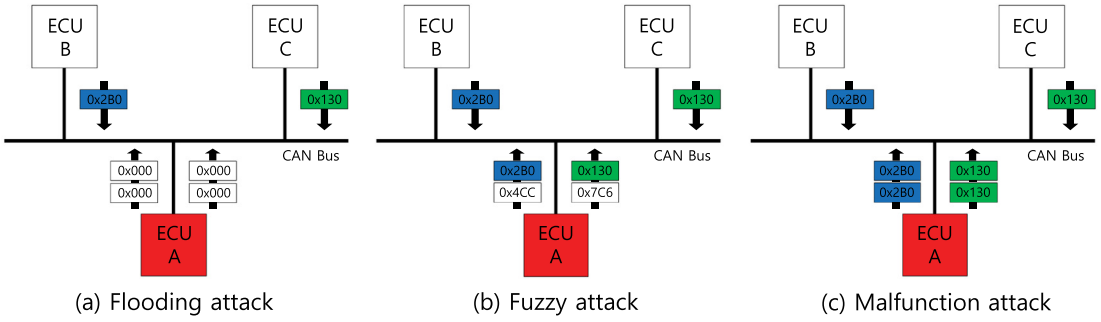

Fig. 3. Three attack scenarios on IVN. ECU A is manipulated and controlled by a malicious packet attack.

(e.g., the priority of 0×130 is higher than 0×545). Low-priority CAN messages are automatically retransmitted on the following CAN bus cycle (the broadcast principle) [27]. The flooding attack allows an ECU node to occupy many of the resources allocated to the CAN bus by maintaining a dominant status on the CAN bus. This attack can limit the communications among ECU nodes and disrupt normal driving. We conducted the flooding attack by injecting a large number of messages with the CAN ID set to 0×000 into the vehicle networks.

Fuzzy attack Fuzzing is a software-testing technique used to find vulnerabilities by entering unexpected values or random data into computer programs [32,33]. In the second attack scenario, the attacker performs indiscriminate attacks by iterative injection of random CAN packets. For the fuzzy attack, we generated random numbers with "randint" function, which is a generation module for random integer numbers within a specified range. Messages were sent to the vehicle once every 0.0003 seconds. This process was conducted for both the ID field and the Data field. The randomly generated CAN ID ranged from 0×000 to 0×7FF and included both CAN IDs originally extracted from the vehicle and CAN IDs which were not. We determined that the reaction of the fuzzy attack using random numbers restricted to the extractable CAN IDs from the vehicle network is more immediate than the reaction using all random numbers. When the attack was executed on the KIA Soul, a short beeping sound repeatedly occurred, and the heater turned on. A navigation system error in which the rear camera turned on regardless of the drive gear status occurred as well.

Malfunction attack The malfunction attack targets a selected CAN ID from among the extractable CAN IDs of a certain vehicle. As CAN IDs for the malfunction attack, we chose 0×316, 0×153 and 0×18E from the HYUNDAI YF Sonata, KIA Soul, and CHEVROLET Spark vehicles, respectively. For a malfunction attack, the manipulation of the data field has to be simultaneously accompanied by the injection attack of randomly selected CAN IDs. When the values in the data field consisting of 8 bytes were manipulated using 00 or a random value, the vehicles reacted abnormally. In the data field for CAN ID 0×153, when data bytes from the first to third were changed to 00, a short beeping sound repeatedly occurred in the KIA Soul vehicle. In addition, for CAN ID 0×43F, when the third data byte was randomly changed, and the remaining bytes were set to 00, the headlights light blinked, and the engine/emissions control lamp icon on the instrument panel was illuminated.

4. Proposed design

Fig. 4 shows the proposed anomaly intrusion detection system based on the survival analysis model. The system contains two main parts: the chunk-based threshold measurement and the detection algorithm. The survival analysis model is a statistical method used to discover what factors affect the survival rate and survival duration of a measurement object. The proposed system focuses on the survival rate of an individual CAN ID in a defined chunk unit. This algorithm aims to accurately detect the three typical attack scenarios described in Section 3.3.

4.1. Threshold measurement

As described in Section 2, based on the time until the event occurs, survival analysis is a statistical method that analyzes the survival time of the subject and whether or not the subject has survived given a specific time point. Performing survival analysis requires information concerning the measurement object, its survival time, and its current status. The current status is needed to check whether or not there is an object to be measured at a specific time or within a section of the survival measurement. To apply the survival analysis method to driving data, we selected only the ID field among the ID, DLC, and Data fields in the CAN message frame. The CAN messages extracted in accordance with time were divided into chunk units, and the chunk size was increased from 50 to 300 in increments of 10. A cycle signified an interval until the same CAN ID occurred again after its first occurrence. A CAN ID of short cycle occurred approximately once every 18 CAN messages, while a CAN ID of long cycle occurred approximately once every 200 CAN messages. Each CAN ID that occurred during normal driving had a unique periodic pattern. CAN ID 0×5A2 of the HYUNDAI YF Sonata vehicle had a unique long cycle pattern—among the CAN messages extracted in accordance with time (approximately, once per second), it appeared on average once every 1950 times. If the defined chunk size was set to 100, CAN ID 0×5A2 appeared for the first time at the 19th chunk, and Fig. 5 shows the unique periodic pattern of CAN IDs extracted from the HYUNDAI YF Sonata vehicle. As a general rule, the higher the priority of the CAN ID is, the more frequent its occurrence.

The survival rate $SR(C_i)$ applied in the present study paper is calculated using Eq. (1).

$$SR(C_i^{CAN\_ID}) = \frac{F(C_i^{CAN\_ID})}{S(C_i^{CAN\_ID})} \tag{1}$$

- $C_i = \{C_1, C_2, \ \ldots, \ C_i \parallel 1 \le i \le n\}$: A defined chunk unit divided by the number of CAN messages logged in accordance with time.
- i: chunk ordering.
- $S(C_i^{CAN\_ID})$: All CAN messages in a defined chunk unit.
- $F(C_i^{CAN\_ID})$: Frequency of an individual CAN message in a defined chunk unit.
- $SR(C_i^{CAN\_ID})$: Survival rate of an individual CAN message in a defined chunk unit.

The CAN ID varies in attributes depending on the manufacturer, model, and release year of the vehicle. Even if the CAN ID

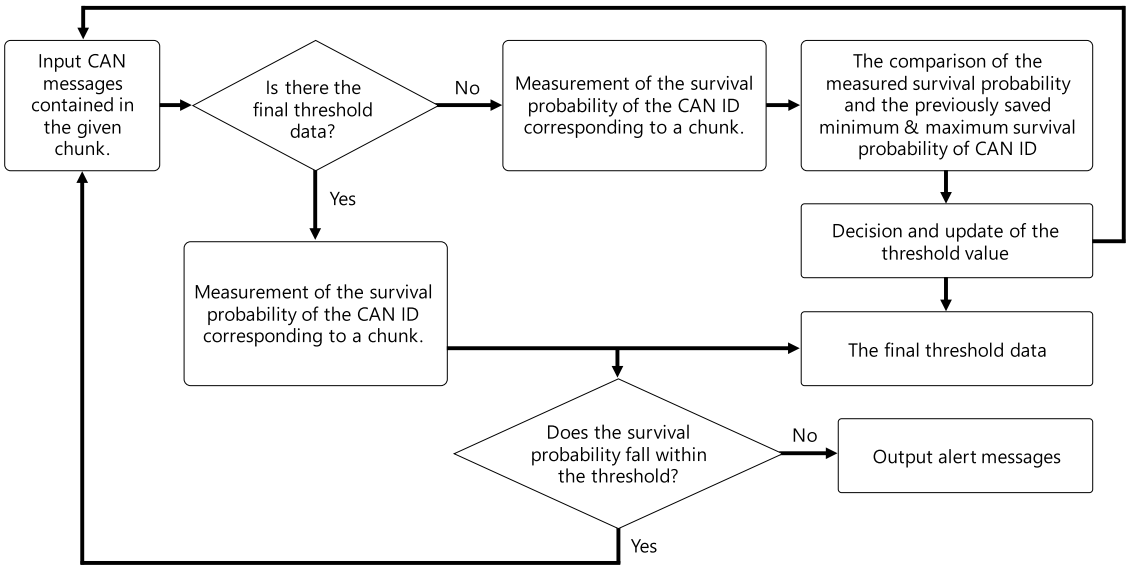

Fig. 4. Overview of the anomaly-intrusion detection scheme based on the survival analysis model.

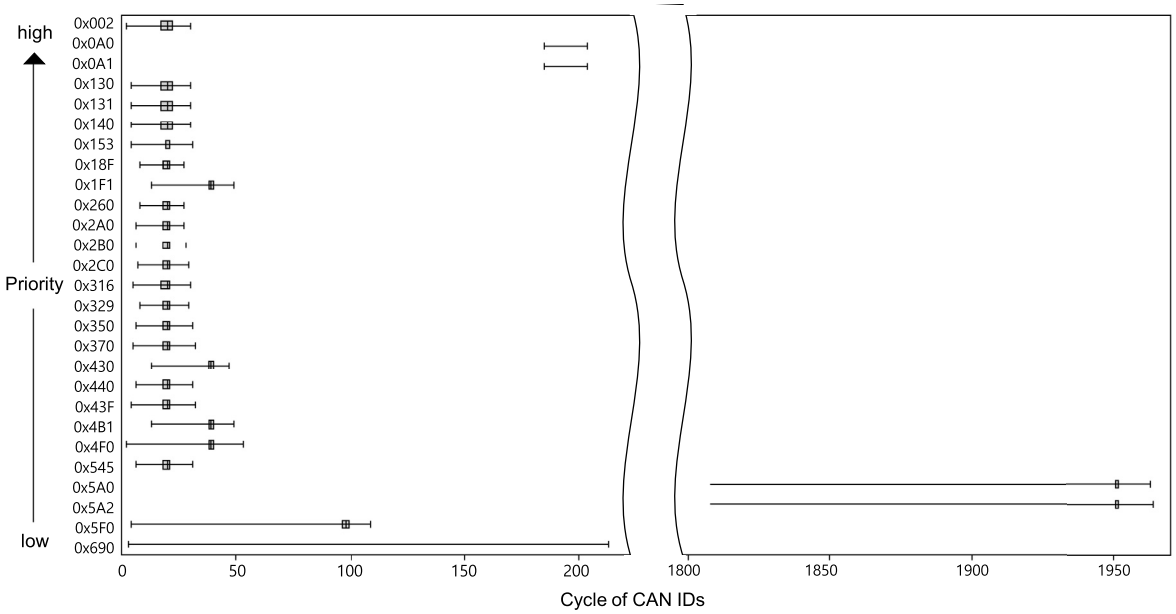

Fig. 5. Unique periodic pattern of CAN IDs extracted from HYUNDAI YF Sonata vehicle.

used among different vehicles is numerically identical, the defined and assigned characteristics of CAN IDs may vary. Therefore, as the prior-setting process for the ground-truth value, first of all, the survival rate operation module is activated once in the beginning. This provides the basis for a unique periodic pattern of the CAN IDs defined depending on the vehicle model of the manufacturer. Based on the unique periodic pattern of the CAN IDs, the survival rate operation module measures the survival rate of a CAN ID in the defined chunk units. When a vehicle is first registered into this system, the ground-truth value is produced using just the survival rate operation module without operation of the anomaly detection module. After completing the creation of the ground-truth value, the survival rate operation module is operated only through function calls of the anomaly detection module.

The survival rate operation module is developed from the following three steps: chunk framing, threshold setting, and min–max value setting. In the chunk framing step, a CAN message is bound depending on the defined chunk unit and is then returned to the previous step of the function call. The survival rate of a CAN ID in a chunk is measured in the threshold setting step. The survival rate of a certain CAN ID is calculated as the rate of how many times the same CAN ID is present in a chunk. Note that all CAN IDs of the vehicle could be present in a defined chunk but a CAN ID with a long cycle may not appear in the defined chunk. The survival rate of the CAN ID measured in the previous chunk is then compared with the survival rate of the CAN ID in the following chunk, and their minimum and maximum values are updated (e.g., the min–max value setting step). In the case of CAN IDs that frequently appear in every chunk, the minimum and maximum values of the survival rate are updated continuously.

Algorithm 1 shows the survival rate operation module measurement steps for the survival rate of the CAN ID in a chunk. This module is operated only through the function calls of the anomaly detection module.

Algorithm 1: Survival rate operation module.

Input:
L1 (list of the CAN IDs in one chunk depending on the defined chunk size)
L2 (list of the unique CAN IDs extracted from vehicles)
Output: min–max value of threshold on the survival probability of CAN IDs

```
L3 ← removing the duplicated CAN IDs in the L1
L3_count, L2_count ← counting the CAN IDs in L3 and L2
declaration of MinMax[3][L3_count] array
for i ∈ {0, ..., L3_count} do
    count ← counting the L3[i][0] among the L1
    prob ← count/the numbers in L1
    for row ∈ {0, ..., L2_count} do
        for col ∈ {0, ..., 3} do
            MinMax[i][0] ← L3[i]
            MinMax[i][1] ← prob     /* Setting the ground-truth
            and threshold value */
            MinMax[i][2] ← 0.0
        end
    end
end
```

4.2. Detection algorithm

The survival analysis model-based detection algorithm is used to determine whether the CAN message is normal by comparing the ground-truth value with the survival rate of the CAN message ID. When an injection attack such as flooding, fuzzy, or malfunction occurs, the survival rate of CAN IDs exhibits a different pattern from the ground-truth value. As shown in Fig. 6, the survival rate of CAN IDs during the flooding and fuzzy attacks is lower than the minimum value of the ground-truth. The injected CAN ID for the flooding and fuzzy attacks was not a CAN ID defined by the manufacturers' guidelines. The flooding attack was performed with a higher priority of CAN IDs, while the fuzzy attack was performed with randomly generated CAN IDs. A large number of attack packets were injected into a chunk during the attacks, resulting in lower survival rates. On the other hand, the survival rate of CAN IDs during the malfunction attack was significantly higher than the maximum value of the ground-truth. Because the malfunction attack was performed through the CAN IDs prescribed by the manufacturer, the number of injected CAN IDs was added to the number of existing CAN IDs, significantly increasing the total number of CAN IDs.

Algorithm 2 indicates how the anomaly detection module operates by comparing the ground-truth value and the survival rate of CAN IDs returned from the threshold measurement step. The goal of the module is to discern and detect accurately whether or not a CAN message is normal for three defined attacks. We also focused on the applicability of the proposed algorithm to various types of vehicles, and this process is valid for the attack scenarios mentioned in Section 3.3. If the injection attack involving large quantities of CAN IDs does not occur continuously, it may be difficult to detect an abnormality correctly; because there is no or insignificant variation in the survival rate of CAN IDs.

5. Performance evaluation

To evaluate the performance of our algorithm, we measured the detection accuracy and the detection speed considering three aspects (i.e., the type of vehicle, the attack scenario, and the periodic characteristics of CAN IDs). The detection accuracy was evaluated by the accuracy metric and the F-measure metric, and the detection speed was evaluated by time complexity and run-time.

5.1. Detection accuracy

First, we used the accuracy metric (Eq. (2)) to evaluate the feasibility of the proposed algorithm. The accuracy is defined as the rate of correctly detected data among all detected data. That is, the correctly detected data indicates that values of true-positives and true-negatives are high, (e.g., when the rate of true-positives and true-negatives among all detected data is high, the accuracy also increases indicating that the error rate is low). Table 1 shows the confusion matrix for the vehicle attack scenarios.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{2}$$

As shown in Fig. 5, a CAN ID was assigned an ID number depending on the priority of the CAN message, and it can be seen that the cycle pattern of the CAN ID also differed depending on its priority. Higher-priority CAN messages were transmitted faster than lower-priority ones, while lower-priority CAN messages were delayed. Therefore, higher-priority CAN IDs appeared more frequently. Based on the periodic characteristics of CAN IDs, we measured the detection accuracy in each chunk by dividing the evaluation object into an individual CAN ID, All CAN IDs, and CAN IDs with a short cycle.

The first evaluation object was an individual CAN ID. Fig. 7 shows the detection accuracy based on the survival rate of an individual CAN ID extracted from the HYUNDAI YF Sonata vehicle. We found that the detection accuracy of certain CAN IDs with a short

Algorithm 2: Anomaly detection module.

Input: G (ground-truth value)
Output: alert message

```
calling the survival rate operation module
M ← returning survival rate of CAN IDs
for i ∈ {0, ..., M} do
    for j ∈ {0, ..., G} do
        count ← counting the M[i][0] among the G[j][0]  /* The CAN ID
        in the two-dimensional array */
        if count ≧ 1 then
            if G[j][0] ≠ M[i][0] then
                continue
            else
                if G[j][1] ≤ M[i][1] then      /* Comparison of the
                ground-truth value with the survival rate
                of CAN ID */
                    if G[j][2] ≥ M[i][1] then
                        flag[M[i][0]] ← 0
                    else
                        output the alert message
                        flag[M[i][0]] ← 1
                    end
                else if G[j][2] ≥ M[i][1] then
                    if G[j][1] ≤ M[i][1] then
                        flag[M[i][0]] ← 0
                    else
                        output the alert message
                        flag[M[i][0]] ← 2
                    end
                end
            end
        end
    end
end
```

Table 1
Confusion matrix for the vehicle attack scenarios.

	Total population	Predicted class	
		Intrusion (class=positive)	Normal (class=negative)
Actual class	Intrusion (class=positive)	TP	FN
	Normal (class=negative)	FP	TN

True-positive (TP): an attack packet classified as "attack".
False-positive (FP): an attack packet classified as "normal".
False-negative (FN): a normal packet classified as "attack".
True-negative (TN): a normal packet classified as "normal".

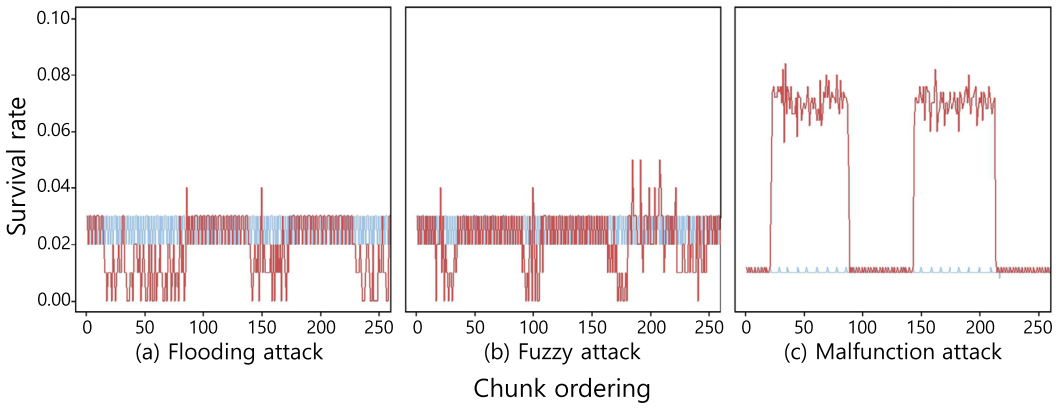

Fig. 6. The survival rates of normal and abnormal data in each chunk: (a) Flooding attack, (b) Fuzzy attack, (c) Malfunction attack.

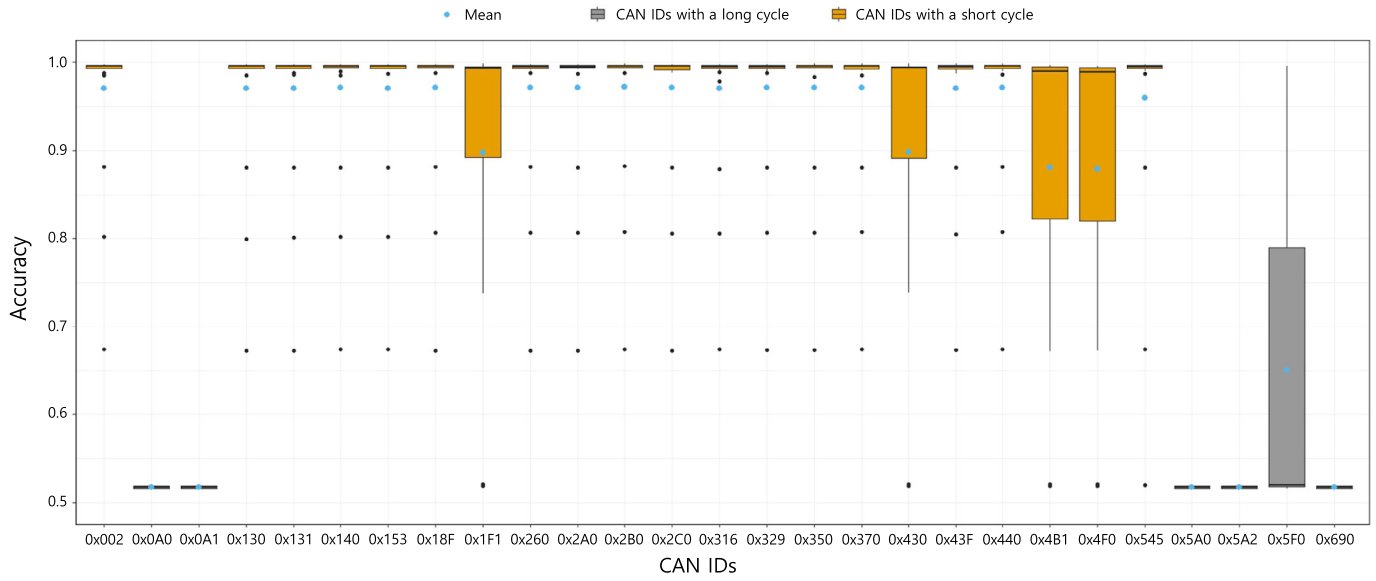

Fig. 7. The detection accuracy by the survival rate of an individual CAN ID extracted from the HYUNDAI YF Sonata vehicle.

cycle ranged from 70% to 100%. At the same time, for CAN IDs with a long cycle (i.e., 0×0A0, 0×0A1, 0×5A0, 0×5A2, 0×5F0, and 0×690), the detection accuracy remained unchanged at approximately 70%. Because a CAN ID with a long cycle may not appear in some chunks, it was difficult to measure the accuracy based on their survival rate. In other words, if accuracy was measured by the survival rate of CAN IDs with a long cycle, the rates of false-positives and false-negatives increased. As a result, accuracy was low.

Next, we measured the accuracy by the survival rate of all CAN IDs for each chunk. The total number of CAN IDs varied depending on the type of vehicle. We were able to extract 27 CAN IDs from the HYUNDAI YF Sonata, 45 CAN IDs from the KIA Soul, and 83 CAN IDs from the CHEVROLET Spark. Among all CAN IDs, the CAN IDs having a cycle less than 50 were 21 CAN IDs from the HYUNDAI YF Sonata, 24 CAN IDs from the KIA Soul, and 17 CAN IDs from the CHEVROLET Spark. Fig. 8 shows the detection accuracy based on survival rate of all CAN IDs and the CAN IDs with a short cycle.

We then evaluated the difference in the detection accuracy depending on the type of vehicle and three attack scenarios. Row (a) shows the results of the detection accuracy measured with all CAN IDs extracted from a vehicle, and row (b) shows the results of the detection accuracy measured with the CAN IDs having a cycle less than 50. The results of the accuracy measured with all CAN IDs and the CAN IDs having a short cycle were similar. However, the accuracy measured with all CAN IDs extracted from the KIA Soul decreased with an increase of chunk size. In the case of the flooding attack, the results confirmed that the accuracy measured with all CAN IDs and CAN IDs having a short cycle was over 97% regardless of the chunk size. In the CHEVROLET Spark, however, detection accuracy was low when the chunk size was less than 90. The next attack scenario was the fuzzy attack. The detection accuracy by the survival rate of all CAN IDs and the CAN IDs with a short cycle was uneven at chunk sizes below 100. However, above 100, the detection accuracy, with the exception of KIA Soul vehicle, was high. The detection accuracy by the survival rate of all CAN IDs decreased with an increase of the chunk size only in the KIA Soul. Finally, for the malfunction attack, the detection accuracy was above 97% for all three types of vehicle, chunk sizes, and the periodic characteristics of CAN IDs. These results demonstrate that our method can detect anomalies in the CAN network with high accuracy for all vehicles used for evaluation. Table 2 shows the detection rate (Recall) of the proposed method, depending on the types of vehicles, the cycle of CAN IDs and the attack types. The proposed method reached high Recall averages over 92% in everything, except the Fuzzy attack of the CHEVROLET Spark.

A second metric for certifying the feasibility of the proposed algorithm is the F-measure. The F-measure is a metric that com-

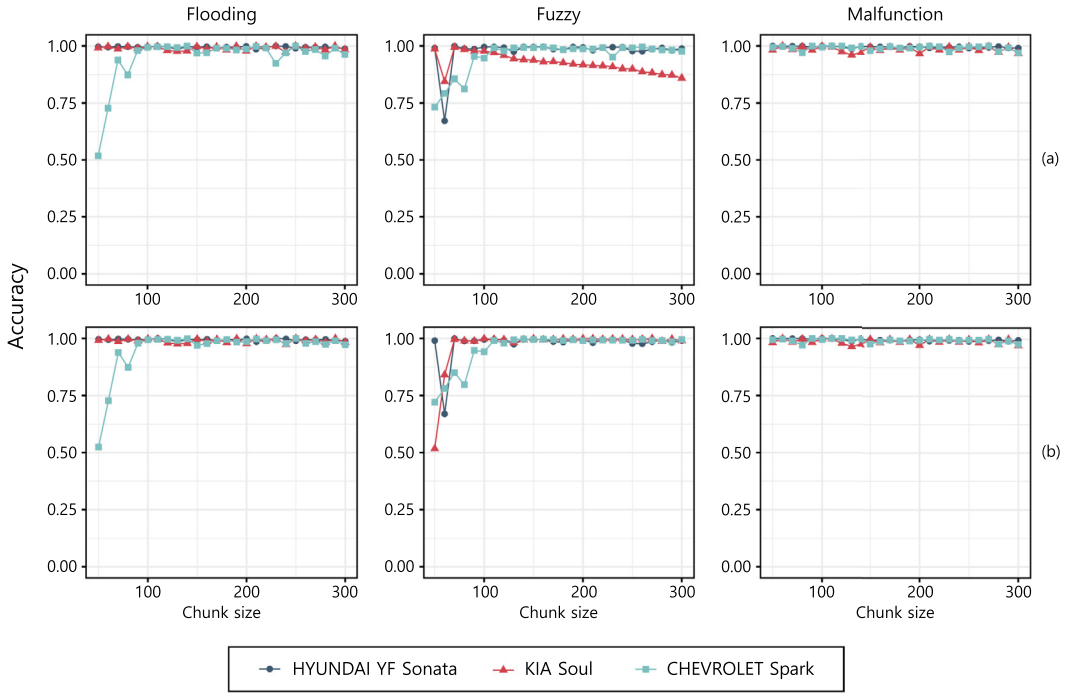

Fig. 8. Measurement of accuracy based on the type of vehicle, the three attack scenarios, and the periodic characteristics of CAN IDs. (a) all CAN IDs; (b) CAN IDs having a cycle less than 50. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

Table 2
Detection rate (Recall) of the proposed method[a].

Vehicle	Feature	Attack	Max	Min	Median	Avg
HYUNDAI YF Sonata	All CAN IDs	DoS	0.9920	0.9684	0.9868	0.9861
		Fuzzy	0.9911	0.8359	0.9817	0.9735
		Malfunction	1.0000	0.9988	0.9994	0.9995
	The CAN IDs with a short cycle	DoS	0.9917	0.9582	0.9865	0.9855
		Fuzzy	0.9909	0.8338	0.9815	0.9725
		Malfunction	1.0000	0.9987	0.9994	0.9994
KIA Soul	All CAN IDs	DoS	1.0000	0.9894	0.9972	0.9969
		Fuzzy	1.0000	0.7701	0.9983	0.9887
		Malfunction	1.0000	0.9979	1.0000	0.9998
	The CAN IDs with a short cycle	DoS	1.0000	0.9894	0.9972	0.9969
		Fuzzy	1.0000	0.0004	0.9973	0.9503
		Malfunction	1.0000	0.9960	1.0000	0.9996
CHEVROLET Spark	All CAN IDs	DoS	1.0000	0.0018	0.9966	0.9235
		Fuzzy	1.0000	0.0464	0.9892	0.8708
		Malfunction	1.0000	0.9906	1.0000	0.9983
	The CAN IDs with a short cycle	DoS	1.0000	0.0000	0.9963	0.9231
		Fuzzy	1.0000	0.0000	0.9892	0.8625
		Malfunction	1.0000	0.9906	1.0000	0.9983

[a] The values rounded to the four decimal places.

bines recall and precision. It is a useful indicator in the fields of information retrieval, pattern recognition, and machine learning. The recall indicates the rate of data detected by an algorithm among the entire attack data, and the precision indicates the rate of data equating to the attack data among the entire detected data. The F-measure is calculated using Eq. (3) and has a value between 0 and 1, where both the recall and the precision must be high to increase the F-measure. The closer the F-measure value is to 1, the better the performance of the algorithm.

$$\text{F-measure} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{3}$$

Fig. 9 shows the F-measure results based on the survival rate of all CAN IDs and CAN IDs with a short cycle. Row (a) reports the results of the F-measure with all CAN IDs extracted from the vehicle; and row (b) presents the results of the F-measure measured with CAN IDs having a cycle less than 50. The results of the F-measure with all CAN IDs and CAN IDs having a short cycle were similar, except for the case of the KIA Soul vehicle, where the F-measure decreased with increasing chunk size.

The F-measure for the flooding attack showed a high value of above 97% regardless of the chunk size with the exception of the CHEVROLET Spark. For the CHEVROLET Spark, the F-measure

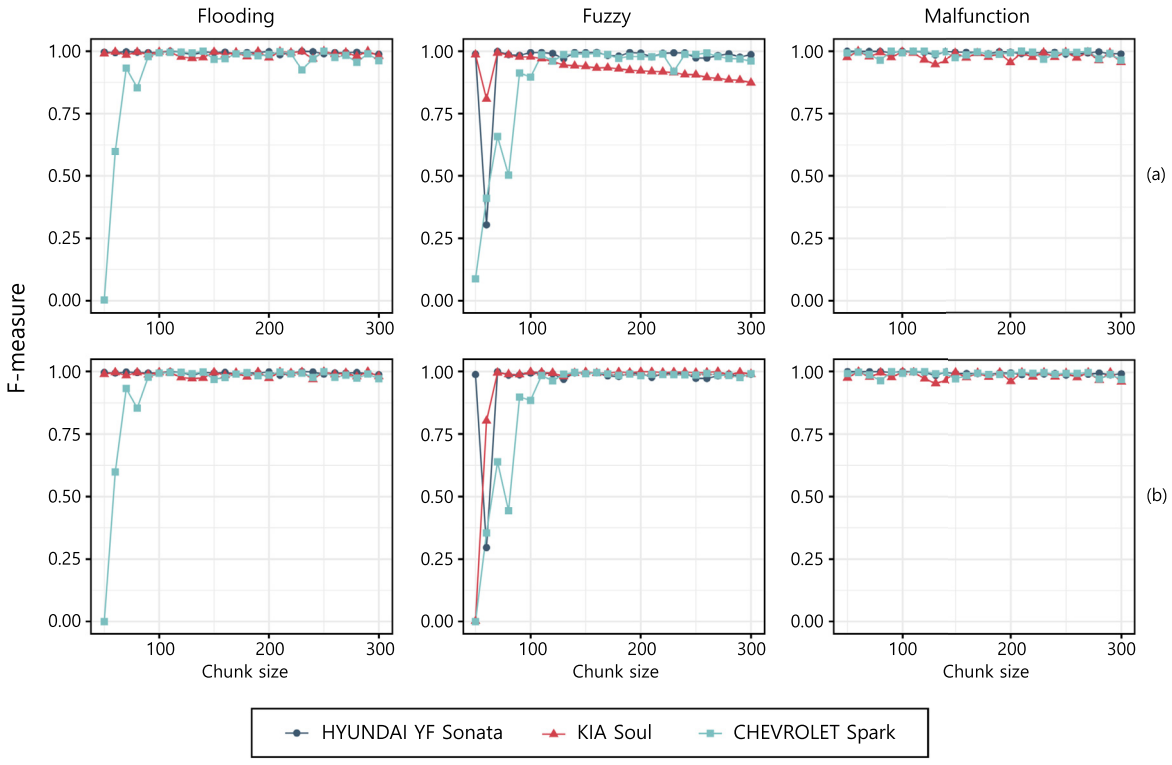

Fig. 9. F-measures based on the type of vehicle, the three attack scenarios, and the periodic characteristics of CAN IDs. (a) all CAN IDs; (b) CAN IDs having a cycle less than 50.

showed a high value at chunk sizes above 70; the rate of true-positives and false-negatives was unstable at chunk sizes below 70, so the F-measure was low. For the fuzzy attack, the F-measure at chunk sizes below 100 was very irregular, whereas, at chunk sizes above 100, the F-measure, except the KIA Soul, was high (the F-measure by the survival rate of all CAN IDs decreased with an increase of chunk size only in the KIA Soul). Finally, regardless of the type of vehicle, chunk size, and periodic characteristics of CAN IDs, the F-measure for the malfunction attack was above 95%. These results also demonstrate that our method can perform anomaly detection well with high F-measures for all vehicles used for evaluation.

- $T1(n)$: a polynomial for the anomaly detection module
- $T2(n)$: a polynomial for the survival rate operation module

5.2. Algorithm speed

In this section, we report the results of measuring the time complexity and detection speed of the anomaly detection algorithm by dividing the evaluation object into an individual CAN ID, all CAN IDs, and CAN IDs with a short cycle. The time complexity refers to the frequency of calling the command to activate an algorithm. The detection speed indicates the actual run-time measured in seconds.

Analyzing the time complexity is similar to measuring how long an algorithm takes to solve the problem as input n. The most important variable in time complexity is the n unit, which has a great effect on the given polynomial. Time complexity can be expressed using Big O (i.e., O(N)). The proposed algorithm embraces the two key modules, and the three evaluation objects described above were measured using these modules, so the time complexity was expressed in the same manner based on Big O. Even though the three evaluation objects were measured using the same modules, the number of CAN IDs applied to each evaluation object was different. Therefore, $T(n)$ was subordinate to constant variation of n. As shown in Table 3, the two modules were represented by a polynomial function $T(n)$ for constant values and measured the total steps of the input value n related to the number of CAN IDs.

Table 3
Time complexity of the detection algorithm for the HYUNDAI YF Sonata vehicle.

	$T1(n)$	$T2(n)$	Big O	Total steps
An individual CAN ID	$2(n^2)+3$	$3(n^3)+2n+8$	O(n^3)	1
All CAN IDs	$2(n^2)+3$	$3(n^3)+2n+8$	O(n^3)	19,683
CAN IDs with a short cycle	$2(n^2)+3$	$3(n^3)+2n+8$	O(n^3)	9261

To investigate whether the number of CAN IDs affected the detection speed, we measured the actual run-time of the two modules by the evaluation objects and by the type of vehicle. The calculation of the detection speed was performed on a laptop computer with an Intel Core i7-7500U CPU, 16.0 GB RAM, running on the Windows 10 Enterprise 64-bit edition. Fig. 10 shows the differences in detection speed depending on the type of vehicle and evaluation object. The total number of CAN IDs extracted from HYUNDAI YF Sonata, KIA Soul, and CHEVROLET Spark were different. The CHEVROLET Spark had a total of 83 CAN IDs, which was the most among the three tested vehicles. Accordingly, as shown in Fig. 10(right), we found that the detection speed of the CHEVROLET Spark vehicle measured with all CAN IDs was noticeably slower. We also analyzed the detection speed based on the evaluation objects. The results confirmed that the detection speed measured with an individual CAN ID was the fastest. The detection speed of an individual CAN ID versus CAN IDs with a short cycle did not show significant difference, although it can be seen that the gap of the detection speed grew with increasing numbers of CAN IDs and chunk size.

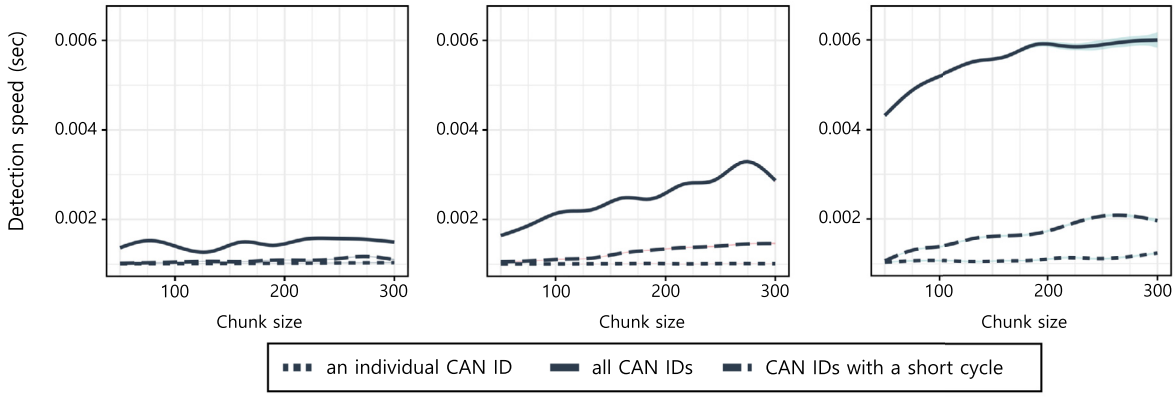

Fig. 10. Speed of detection of the anomaly detection algorithms for the type of vehicle and evaluation objects. HYUNDAI YF Sonata (left); KIA Soul (center); CHEVROLET Spark (right).

6. Discussion and conclusion

In the present paper, we developed and validated an anomaly intrusion detection method based on the survival analysis model. Because the intrinsic properties of CAN networks include several factors dependent on the particular vendor's technology, applying an intrusion detection technique without semantic knowledge concerning the CAN ID function is not easy. Nevertheless, the experimental results demonstrated that the normality and abnormality of the vehicle network could be identified with high accuracy and fast detection speed. Real-time analysis allows detection of anomalies in the CAN network immediately. Our method enables the vehicle to self-check the vehicle condition, and can promptly provide a driver with information needed to make decisions. Furthermore, as a general-purpose intrusion detection method, our method could be applied to all vehicles. However, it is not possible yet for the proposed method to actively intervene to return a vehicle to the normal condition. This challenge needs to be addressed in future research. In our own work, we continue to study algorithms not only for attack detection in general, but also for the selective detection of attack types. Our studies will be advanced with an active intrusion prevention system that responds by changing the manipulated CAN traffic caused by an attack.

Acknowledgements

This work was supported by Samsung Research Funding & Incubation Center for Future Technology under Project Number SRFC-TB1403-51.

References

[1] BBC News Technology, Hack attacks mounted on car control systems, http://www.bbc.com/news/10119492, 2010 (Accessed 10 July 2018).
[2] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, T. Kohno, Comprehensive experimental analyses of automotive attack surfaces, in: Proceedings of the 20th USENIX Conference on Security, USENIX Association, 2011, p. 6.
[3] Center for Information Security Technologies, Korea University, Hacking your car through a smart phone, https://www.youtube.com/watch?v=katVec_SwA4, 2012 (Accessed 10 July 2018), online.
[4] C. Miller, C. Valasek, Adventures in automotive networks and control units, DEF CON 21 (2013) 260–264.
[5] C. Miller, C. Valasek, Remote exploitation of an unaltered passenger vehicle, Black Hat USA 2015 (2015), 91 pp.
[6] S. Nie, L. Liu, Y. Du, Free-fall: hacking Tesla from wireless to CAN bus, https://www.blackhat.com/docs/us-17/thursday/us-17-Nie-Free-Fall-Hacking-Tesla-From-Wireless-To-CAN-Bus-wp.pdf, 2017 (Accessed 10 July 2018), online.
[7] C. Bernardini, M.R. Asghar, B. Crispo, Security and privacy in vehicular communications: challenges and opportunities, Veh. Commun. 10 (2017) 13–28, https://doi.org/10.1016/j.vehcom.2017.10.002.
[8] K. Etschberger, Controller Area Network: Basics, Protocols, Chips and Applications, Ixxat Press, 2001.
[9] K.T. Cho, K.G. Shin, Fingerprinting electronic control units for vehicle intrusion detection, in: USENIX Security Symposium, 2016, pp. 911–927.
[10] Robert Bosch GmbH, CAN Specification 2.0, http://esd.cs.ucr.edu/webres/can20.pdf, 1991 (Accessed 10 July 2018), online.
[11] Florian Hartwich, Bit time requirements for CAN FD, https://www.can-cia.org/fileadmin/resources/documents/proceedings/2013_hartwich_v2.pdf, 2013 (Accessed 10 July 2018), online.
[12] A. Taylor, N. Japkowicz, S. Leblanc, Frequency-based anomaly detection for the automotive CAN bus, in: World Congress on Industrial Control Systems Security, WCICSS, IEEE, 2015, pp. 45–49.
[13] M. Müter, N. Asaj, Entropy-based anomaly detection for in-vehicle networks, in: Intelligent Vehicles Symposium (IV), IEEE, 2011, pp. 1110–1115.
[14] M. Marchetti, D. Stabili, A. Guido, M. Colajanni, Evaluation of anomaly detection for in-vehicle networks through information-theoretic algorithms, in: 2nd International Forum on Research and Technologies for Society and Industry Leveraging a Better Tomorrow, RTSI, IEEE, 2016, pp. 1–6.
[15] H.M. Song, H.R. Kim, H.K. Kim, Intrusion detection system based on the analysis of time intervals of CAN messages for in-vehicle network, in: International Conference on Information Networking, ICOIN, IEEE, 2016, pp. 63–68.
[16] M. Marchetti, D. Stabili, Anomaly detection of CAN bus messages through analysis of ID sequences, in: Intelligent Vehicles Symposium (IV), IEEE, 2017, pp. 1577–1583.
[17] M. Markovitz, A. Wool, Field classification, modeling and anomaly detection in unknown CAN bus networks, Veh. Commun. 9 (2017) 43–52, https://doi.org/10.1016/j.vehcom.2017.02.005.
[18] A. Taylor, S. Leblanc, N. Japkowicz, Anomaly detection in automobile control network data with long short-term memory networks, in: International Conference on Data Science and Advanced Analytics, DSAA, IEEE, 2016, pp. 130–139.
[19] A. Ganesan, J. Rao, K. Shin, Exploiting Consistency Among Heterogeneous Sensors for Vehicle Anomaly Detection, Technical Paper, University of Michigan and Ford Motor Company, 2017.
[20] H. Li, L. Zhao, M. Juliato, S. Ahmed, M.R. Sastry, L.L. Yang, POSTER: intrusion detection system for in-vehicle networks using sensor correlation and integration, in: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, ACM, 2017, pp. 2531–2533.
[21] ESCRYPT, Automotive Intrusion Detection and Prevention System (IDPS) Continuous Protection as Part of the Automotive Security Lifecycle, Technical Paper, ESCRYPT, 2017, https://www.concarexpo.com/fileadmin/Redaktion/Dokumente/Praesentationen_ConCarForum_2017/24_escrypt_GmbH_-_Jan_Holle.pdf.
[22] Symantec, Building Comprehensive Security Into Cars, Symantec White Paper, 2017, https://www.symantec.com/content/dam/symantec/docs/white-papers/building-comprehensive-security-into-cars-en.pdf.
[23] D.G. Kleinbaum, M. Klein, Survival Analysis, vol. 3, Springer, 2010.
[24] D.R. Cox, Analysis of Survival Data, Routledge, 2018.
[25] T. Hoppe, S. Kiltz, J. Dittmann, Security threats to automotive CAN networks—practical examples and selected short-term countermeasures, Reliab. Eng. Syst. Saf. 96 (1) (2011) 11–25, https://doi.org/10.1016/j.ress.2010.06.026.
[26] K.T. Cho, K.G. Shin, Error handling of in-vehicle networks makes them vulnerable, in: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, ACM, 2016, pp. 1044–1055.
[27] K. Pazul, Controller area network (CAN) basics, Microchip Technology Inc., 1.
[28] W. Choi, K. Joo, H.J. Jo, M.C. Park, D.H. Lee, VoltageIDS: low-level communication characteristics for automotive intrusion detection system, IEEE Trans. Inf. Forensics Secur. 18 (8) (2018) 2114–2129, https://dx.doi.org/10.1109/TIFS.2018.2812149.

[29] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, et al., Experimental security analysis of a modern automobile, in: Symposium on Security and Privacy, SP, IEEE, 2010, pp. 447–462.

[30] T. Hoppe, J. Dittman, Sniffing/replay attacks on CAN buses: a simulated attack on the electric window lift classified using an adapted CERT taxonomy, in: Proceedings of the 2nd Workshop on Embedded Systems Security, WESS, 2007, pp. 1–6.

[31] T. Hoppe, S. Kiltz, J. Dittmann, Automotive it-security as a challenge: basic attacks from the black box perspective on the example of privacy threats, in: International Conference on Computer Safety, Reliability, and Security, Springer, 2009, pp. 145–158.

[32] P. Oehlert, Violating assumptions with fuzzing, IEEE Secur. Priv. 3 (2) (2005) 58–62.

[33] M. Sutton, A. Greene, P. Amini, Fuzzing: Brute Force Vulnerability Discovery, Pearson Education, 2007.