A Concept of IDS for CAN Protocol Based on Statics Theory

Md Rezanur Islam[1], Insu Oh[2], Munkhdelgerekh Batzorig[1], Seoyeon Kim[2], and Kangbin Yim[2](✉)

[1] Department of Smart Convergence Security, Soonchunhyang University, Asan, Korea
{arupreza,munkhdelgerekh}@sch.ac.kr

[2] Department of Information Security Engineering, Soonchunhyang University, Asan, Korea
{catalyst32,seoyeon56,yim}@sch.ac.kr

Abstract. Day after day, modern attacks continue to hit the onboard network due to creating complexity of the group of software and hardware components utilized in vehicles. These new components display challenges within the improvement of compelling and responsive security mechanisms. A few intrusion detection systems (IDS) have been proposed to distinguish and defend vehicle systems from pernicious exercises. Here, in this study we depict statistical-based analysis which applied to intrusion detection method to secure car systems, with a specific accentuation on Control Area Network (CAN). This study underlines a portrayal of vulnerabilities, highlight threat models, easily recognize known attacks that are shown within the CAN.

1 Introduction

In this cutting-edge period, there are progressively developed multiple features and functions which are now included in modern vehicles to make themself not only intelligent but also connected them very well. Due to these reasons, variety of electronic devices and software are applied in the vehicles. Presently, vehicles are using over 60 or 70 electronic control units (ECUs) together associated and dependable for inner communication through the CAN bus [1]. Manufacturers ceaselessly create connected cars to prepared with infotainment and telematics frameworks that utilized SIM cards and portable advances tools to sending and getting information on web associations on a steady premise.

Consequently, threats toward the security level of electronic control units in vehicles are rising gradually. For security purposes, the detection of vulnerabilities is the most preoccupation of car manufacturers, as a compromised ECU or improvised communication data that leading to genuine failures as well as mechanical issues. Several studies demonstrated that programmers (attackers) can effortlessly enter into cars to control their typical driving behavior [2, 3]. It is exceptionally troublesome for CAN-based communication program, to arrange how to ensure the rising attacks, such like message replays, injections, and modifications [4, 5].

In Byoungsoo Lee et al., 2006 [7], authors proposed an entropy-based peculiarity detection strategy for in-vehicle systems. According to this study, most of the detection

L. Barolli (Ed.): BWCCA 2021, LNNS 346, pp. 294–302, 2022.
https://doi.org/10.1007/978-3-030-90072-4_32

systems are based on the whole dataset that produced by the CAN protocol. In the present study, underline a particular CAN based ID level that responsible for speed, RPM, gear, emergency light, etc. Each CAN ID creates a particular data set and a one-of-a-kind of data pattern. Normal data shows a pattern and refinement between normal and injected data set gives high and simple data values to distinguish them easily. Although an 8-Bytes message gives a specific value for different functions, an abnormal data sets, generate different types of massages.

Section 2 demonstrate a comprehensive portrayal of CAN ID and investigate how it generate data sets, and computational analysis performed here. In Sect. 3 describes different types of assault scenarios on CAN protocol. The execution assessment of this study proposed plan displayed in Sect. 4. Finally, future plan and conclusion of this study composition are given in Sect. 5.

2 Background and Related Work

2.1 Background

The CAN is a high-integrity serial data communication innovation that created in the early 1980s by Robert Bosch GmbH for proficient communication between automotive applications. Figure 1 describes data outline format of CAN 2.0B in this protocol a single or double wire arrange with information rates up to 1 Mb/s [19]. Through this protocol, each module of the vehicle communicates among themselves, driving data transmit and received by this CAN protocol [7]. Multiple ECUs connected in a parallel way which acroases the CAN bus like C-CAN, M-CAN, B-CAN medium [8]. An overall operation generates such as speed, rpm, breaks data through C-CAN. B-CAN is a type of network that communicate messages to controlling body parts of the vehicle. M-CAN bus responsible to navigate media and entertainment information [9].

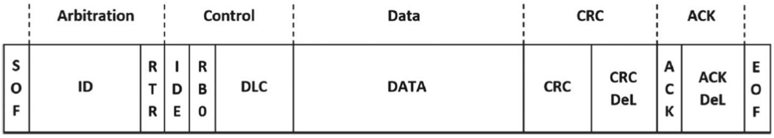


Fig. 1. Data frame format of CAN protocol [20].

CAN bus frame consist of 4 sorts [10]: data frame, error frame, overload frame, and remote outline. This is a standard structure of each of the frame that contains different areas: arbitration identifier, data, acknowledge, and a couple of others. The meaning of each field of a CAN frame described in Fig [21]. The start of frame (SOF) indicates the starting point of a CAN message with a dominant bit, and it inform all of the hubs to generating CAN message transmission. Arbitration comprises 11 bits and it can be expanded up to 29 bits organize. Additionally, control is known as the check field; gives data from the collector to check whether all planning packets are working effectively. Data field contains genuine information for CAN nodes to perform activities. It can be 8 bytes. CRC is known as cyclic redundancy code or security field, a 15-bit fault discovery instrument that checks for packet legitimacy. Moreover, acknowledge (ACK) known as the affirmation field which guarantees collector nodes to get the CAN packet accurately. At whatever point it identified a mistake amid to transmission handle, the transmitter would be notified promptly by the recipient to sending their data bundles once more. EOF field demonstrates the conclusion of the CAN data sets by a latent bit's flag. Presenting data frame utilized it for exchanging CAN packet information from CAN data outline. Consequently, CAN bus communicates with other hubs by transmitting the packets through the data frames. At whatever point the RTR (remote transmission request) bit flags present as dominant, it turns into a CAN data frame.

2.2 Related Works on Automotive Security

Müter et al., displayed the entropy-based appropriateness to identifying an irregularity within the CAN bus framework [13]. The entropy approach received the information-theoretic concept by measuring the coincidence that happened from a given dataset and utilized the gotten result as an IDS specification behavior profile. In this manner, the expanding number of attacks would be raising the number of entropies which indicated the interruptions that happened within the CAN bus. The authors tried to establish it practicality by utilizing the packet inclusion attacks, flooded by CAN bus network with DoS attacks, and exasperates that typically connected to the events. The result illustrated that low arbitrariness of the activity detail of the strategy may be recognized by any infringement from the normal behavior of the CAN bus systems.

A semi-supervised deep neural network (DNN) [22, 23] conveyed by Kang et al., which is the initial movement for the primary machine learning IDS. It is decoded first then the packet exchanged ECUs directly from a bitstream CAN bus lines. Since of the non-linearity of CAN packet highlights, the author proposed to confined Boltzmann machine (RBM) in preparing the extricated parameters [24]. The algorithm arranged in such a way that the normal and abnormal data divided by logistic values "1" and "0". Authors approved the demonstration utilizing spoofed tire weight observing framework (TPMS) packets in arrange to show the wrong values of TPMS indicate on the dashboard.

The statistical-based IDS strategy compares statistical perception with the priorly decided statistical perception. For instance, [14, 15] measurable properties like mean, variance, and standard deviation are used to finding out an unordinary behavior inside of the framework model. Univariate or multivariate time point arrangement is connected here. The statistical-based strategy utilized a rolling window time series to increase CAN bus activity. The univariate technique analyzes the CAN ID fields independently. In case of multivariate method investigations, it may be helpless in time intervals that is included in CAN ID, but it may not be viable for CAN packet substance.

3 Attack Scenarios

In the documents, in our review, there are four attack scenarios for embedded networks that are documented: Fuzzing (spoofing), DOS, Suspension and Replay attacks [3, 6, 16, 17]. Typically, through infotainment, navigation, and mobile network system to the

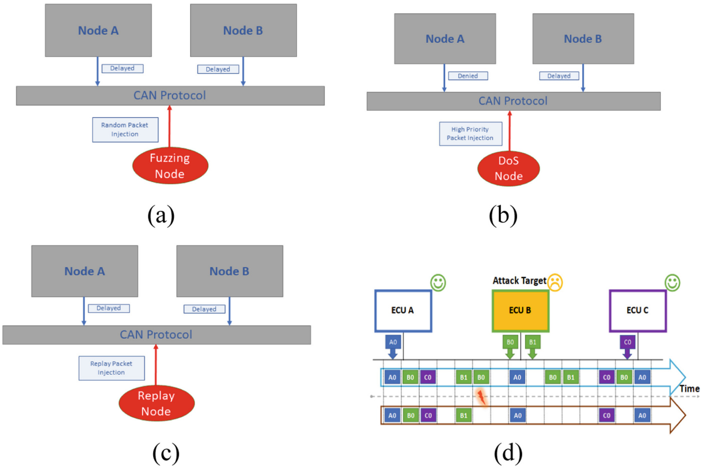


Fig. 2. (a, b, c, d [18]). Various type of attack scenario

first attacker accessing the vehicle's hardware that is connected to CAN. In the second step, he needs to send a message to the system.

When the rival injects a message that does not breach the CAN specification, then the attack cannot be detected by their algorithm. The frequency or interval is suddenly modified when attackers try to inject messages to execute a command to an ECU. While messages are being injected by attackers, ECUs still send their messages cyclically. In the end, the message rate on the network can be increased more than twice generally except for suspension and replay attacks (typically 20 to 100 times as much; this depends on the attacker's injection rate). There are two forms of CAN injection attacks. It injects CAN diagnostic messages, while another injects standard messages to label ECU messages.

Fuzzing Attack. In this attack class, aggressors make messages with arbitrarily spoofed IDs with subjective data, as appeared in Fig. 2 (a). In this case, all hubs would get spoofed abnormal utilitarian messages. In this case, all nodes would receive anomalous abnormal messages. To launch a fuzzing attack, an attacker must first watch and monitor messages in the vehicle and select target identifiers to produce unexpected behaviors. The fuzzing attack is ordinarily produced at a slower rate than the DoS attack [25]. Be that as it may, it is conceivable to perform a fuzzing attack at a higher rate.

Dos Attack. High priority packets injected by an attacker in a short time interval on the bus in as shown in Fig. 2 (b). Typically, the attacker uses one or more high-priority IDs to generate DoS attack images to occupy the CAN bus using malicious or infected ECUs (devices). Since all hubs share a single bus, expanding inhabitance on the bus can create a delay or deny entries of legitimate packets. The DoS attack can cause a vehicle not to reply to the driver's commands on time.

Replay Attack Class. Assailants can carefully watch data arrangements amid a particular time interim and infuse and rehashes the total or portion of these genuine message sequences. Each payload contains a substantial CAN control message. Consequently, it can cause conceivable harm or startling behavior to vehicles. A replay attack is one of the foremost challenging attacks to identify. The illustration of replay attack is appeared in Fig. 2 (c). To begin with, utilizing the OBD-II port, the attacker listens to the traffic on the network and filter out the data coming from one or more arbitrarily selected target node IDs. Attackers store this data with their exact packet entry time, which is utilized to imitate or replay this precisely at an afterward timestamp by infusing these packets into the network.

Suspension Attack. A suspension attack is basically executed by halting ECU from sending its messages. This may have two results, to begin with of all the usefulness regularly controlled by infected ECU will not be executed. Furthermore, there are ECUs that depend on approaching information to operate regularly will not get the command. From Fig. 2 (d) if ECU A requires data from ECU B and a suspension attack is executed on ECU B, ECU A will not work appropriately. In Fig. 2 (d) ECU B is completely compromised as a result of generating an attack it totally stops sending data [18].

4 Functions Entropy Based Data Analysis

4.1 Entropy Detection

The vehicle producer allocates the CAN bus ID, and the overall number of IDs is as it is utilized from 10 to 20%. For case, in a 2016 Ford Fusion, where 223 CAN IDs are utilized, speaking to 10.88 percent of all IDs [11]. When an attack occurred a large amount of data was generated on the system specifically on the fuzzing attack and the DoS attack. Our primary concern is almost the specific function of cars, for illustration speed, rpm, speed, controlling wheel, etc. have a particular CAN ID. The thought is by filtering function-wise CAN ID, we can calculate the sum and select a level of how much data is produced by a specific function CAN ID. Though it varies on different cars and models. In this entire data set, we implement four types of attacks. Fuzzing attack data collected from the unmodified registered cars and other three types of data collected from 4TU.Centre [12] At first, we compare three-set data with an injected data set, and it is collected on the same route and with the same driver.

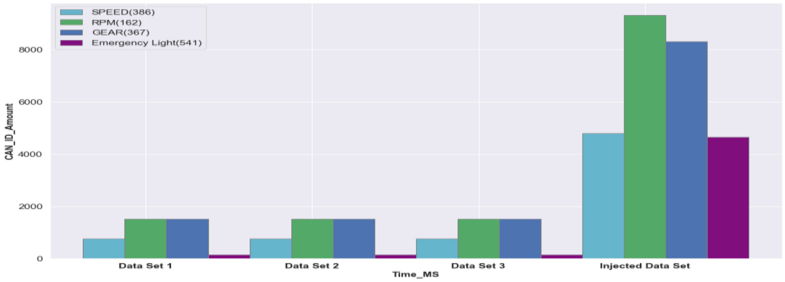
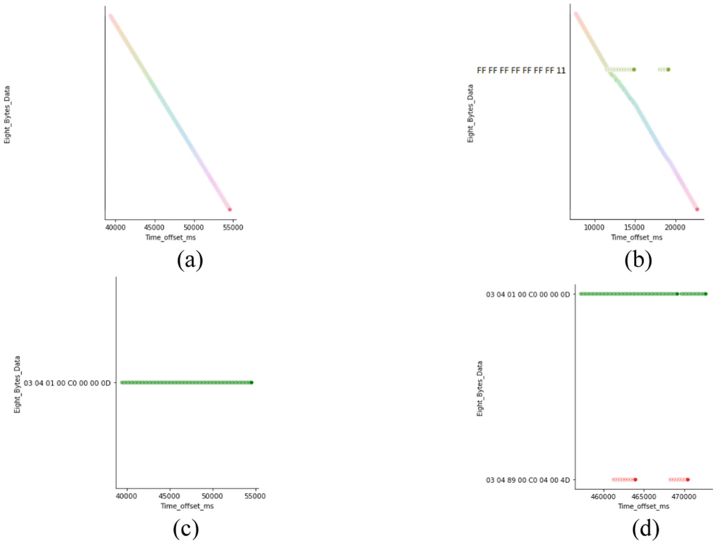


Fig. 3. An example of an attack case study with an emphasis on specific function

Figure 3 represents to a particular CAN ID more absolutely Speed, RPM, Gear, and Emergency light functions produced data where the X-axis is time and Y-axis is the sum of the data. The three sets of normal information taking after a design meaning create nearly the same amount of data completely different times but in case, we focus on injected data set the sum is more than three times higher. If we center on 8-Byte's hexadecimal massage, there moreover we can recognize between normal and anomalous states.

Figure 4 states a scatter plot of speed and emergency light information where a critical alter had been seen. In both functions, diverse sorts of the spoofed messages had found. In Fig. 4 (b). and Fig. 4 (d). the attack begins on 12000 ms to 14000 ms for speed, and for emergency light, it is on 462000 ms to 472000 ms. Comparing to normal data there has seen an excess of the spoofed message for attack time. As a result, the whole of the data at that particular time is high.



(a) (b) (c) (d)

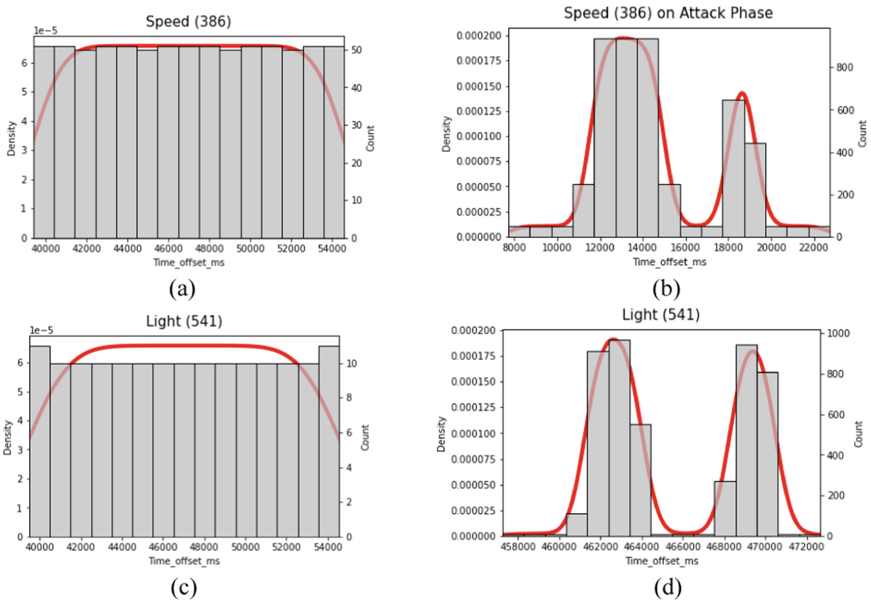
Fig. 4. (a, b, c, d). An example of an attack case study with emphasis on specific function.

4.2 Statistic Basis Data Analysis

In this section, we compared anomalous data with normal data by using the histogram. A histogram is a graphical representation that organized a bunch of data focusing on user-specified ranges. The distinction of four types of attack data with normal data will focus on histogram plots.

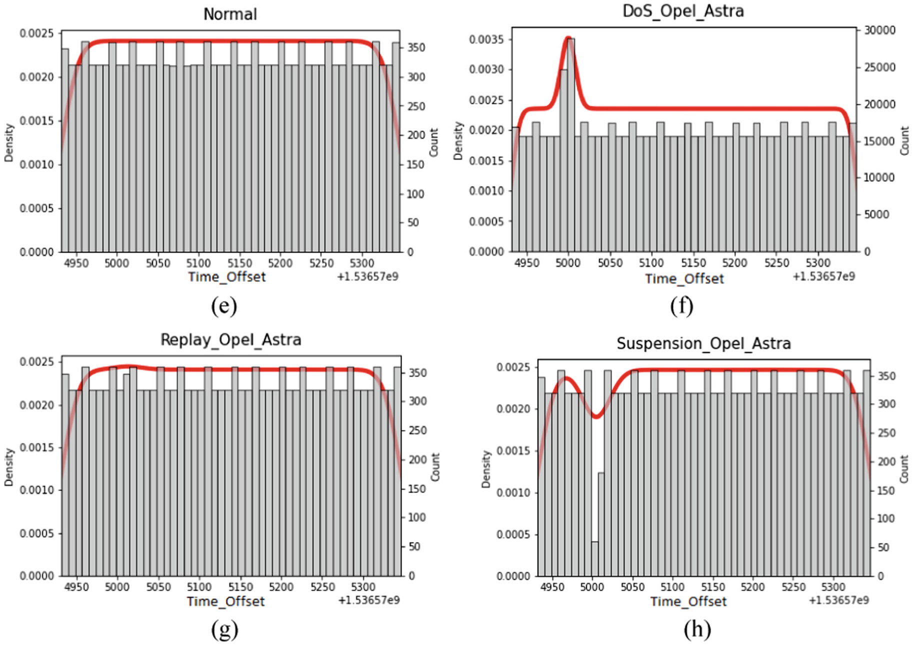
Figure 5 (b), and (d) appears a fuzzing attack histogram that appeared a noteworthy sum of vacillation amid to attack on speed, and emergency light. On the other hand, the normal data set speaks represent a periodic smoothness in Fig. 5 (a), and (c).

In the second part, we compared three types of attack DoS, Replay, and Suspension with normal operation time. This type of data was collected from 4TU.Centre [12] where



(a) (b) (c) (d)

Fig. 5. (a, b, c, d). A case study example of an attack on speed and emergency light



(e) (f) (g) (h)

Fig. 6. (a, b, c, d). A case study example of various attack they use Opel Astra model car. In normal data set has a unique CAN IDs: 85 and it flows a smooth periodic pattern described in Fig. 6 (e). The attack consists of injecting messages with CAN ID ‘000’ (highest priority) for 10 s and a total number of injected packets 40016 which was clearly shown in Fig. 6 (f) through a DoS attack. In replay attack scenario, attack consists of the injection CAN ID ‘1A1’ ten times faster and the total injection massage was 30 which shown in Fig. 6 (g) at the time offset 5000 to 5050. In spite of the fact that the changes are exceptionally slight, that can discernible. Finally, all of the suspension attack created to erasure of messages with CAN ID ‘1A1’ over a period of 10 s and the following graph clearly represent timing attack in Fig. 6 (h). Because of 5000 to 5020 time offset ECU did not perform.

5 Conclusion

This study design format constitutes by utilizing particular data function and depict histogram which effectively identify an assault. Moreover, when programmers tried to attack ECU based protocols, it effectively characterized CAN design-based messages, but programmer cannot be conceivable to know about the design of the specific operational histogram characteristics. The aim of this study was to find out a perfect CAN-based pattern that appropriates for all sorts of assaults likewise Fabrication attack, Masquerade attack, Conquest attack which is troublesome to identify. Within the moment develop an IDS framework by employing a deep learning algorithm.

Acknowledgments. This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No. 2021R1A4A2001810) and This work was supported by Institute for Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2019-0-01343, Regional strategic industry convergence security core talent training business).

References

1. Lu, Z., Wang, Q., Qu, G., Liu, Z.: BARS: a Blockchain- based Anonymous Reputation System for Trust Management in VANETs.arXiv:1807.06159 [cs.CR] (2018)
2. Koscher, A., et al.: Experimental Security analysis of a modern automobile. In: Security and Privacy (SP), 2010 IEEE Symposium on IEEE, 2010, pp. 447–462 (2010)
3. Miller, C., Valasek, C.: Adventures in automotive networks and control units. DEF CON **21**, 260–264 (2013)
4. Woo, S., Jo, H.J., Lee, D.H.: A practical wireless attack on the connected car and security protocol for in-vehicle can. IEEE Trans. Intell. Transp. Syst. **16**(2), 993–1006 (2015)
5. Petit, J., Shladover, S.E.: Potential cyberattacks on automated vehicles. IEEE Trans. Intell. Transp. Syst. **16**(2), 546–556 (2015)
6. Miller, C., Valasek, C.: Remote Exploitation of an Unaltered Passenger Vehicle, Black Hat USA (2015)
7. Lee, B.S., Park, M.K., Sung, K.G.: Developing an In-vehicle Network Education System Based on CAN (2006)

8. An, Y., Park, J., Oh, I., Kim, M., Yim, K.: Design and implementation of a novel testbed for automotive security analysis. In: Barolli, L., Poniszewska-Maranda, A., Park, H. (eds.) Innovative Mobile and Internet Services in Ubiquitous Computing. IMIS 2020. Advances in Intelligent Systems and Computing, vol. 1195. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-50399-4_23
9. A study on the implementation and analysis method of the connected car accident scenario model (KISA-WP-2018-002)
10. Lee, H., Jeong, S.H., Kim, H.K.: In 2017 15th Annual Conference on Privacy, Security and Trust (PST). OTIDS: A novel intrusion detection system for in vehicle network by using remote frame (Calgary, 2017), pp. 57–5709
11. Wang, Q., Lu, Z., Qu, G.: An entropy analysis-based intrusion detection system for controller area network in vehicles. In System-on-Chip Conference (SOCC), 2018 31st IEEE International. IEEE (2018)
12. 4TU.ResearchData. https://data.4tu.nl/
13. Müter, M., Asaj, N.: Entropy-based anomaly detection for in-vehicle networks. In: 2011 IEEE Intelligent Vehicles Symposium (IV), 5–9 June 2011, pp. 1110–1115 (2011). https://doi.org/10.1109/IVS.2011.5940552
14. Avalappampatty Sivasamy, A., Sundan, B.: A dynamic intrusion detection system based on multivariate Hotelling's T2 statistics approach for network environments. Sci. World J. 1–9 (2015)
15. Qayyum, A., Islam, M.H., Jamil, M.: In Proceedings of the IEEE Symposium on Emerging Technologies. Taxonomy of statistical based anomaly detection techniques for intrusion detection (Islamabad, 2005), pp. 270–276 (2005)
16. Cho, K.-T., Shin, K.G.: Fingerprinting electronic control units for vehicle intrusion detection. In: 25th USENIX Security Symposium (USENIX Security 16). Austin, TX: USENIX Association, 2016, pp. 911–927 (2016). https://www.usenix.org/conference/usenixsecurity16/technicalsessions/presentation/cho
17. Cho, K.T., Shin, K.G.:Viden: attacker identification on in-vehicle networks. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, ser. CCS 2017. New York, NY, USA: ACM, 2017, pp. 1109–1123 (2017). https://doi.org/10.1145/3133956.3134001
18. Nowdehi, N., Aoudi, W., Almgren, M., Olovsson, T.: CASAD: CAN-Aware Stealthy-Attack Detection for In-Vehicle Networks. CoRR abs/1909.08407 (2019)
19. BOSCH CAN (2004). www.can.bosch.com
20. Cho, K.T., Shin, K.G.: In 25th {USENIX} Security Symposium ({USENIX} Security 16). Fingerprinting electronic control units for vehicle intrusion detection (Austin, 2016), pp. 911–927 (2016)
21. Lokman, S.F., Othman, A.T., Abu-Bakar, M.H.: Intrusion detection system for automotive Controller Area Network (CAN) bus system: a review. EURASIP J. Wirel. Commun. Netw. **2019**, Article number: 184 (2019)
22. Deng, L., Yu, D.: Deep learning: methods and applications. Foundations and Trends®. Signal Process. **7**(3–4), 197–387 (2014)
23. Kang, M.J., Kang, J.W.: Intrusion detection system using deep neural network for in-vehicle network security. PLoS One **11**(6), e0155781 (2016)
24. Erhan, D., Bengio, Y., Courville, A., Manzagol, P.A., Vincent, P., Bengio, S.: Why does unsupervised pre-training help deep learning? J. Mach. Learn. Res. **11**(Feb), 625–660 (2010)
25. Tariq, S., Lee, S., Kim, H.K., Woo, S.S.:, CAN-ADF: The controller area network attack detection framework. Comput. Secur. **94**, 101857 (2020)