Wavelet Transform Based PID Sequence Analysis for IDS on CAN Protocol

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

[1] Convergence Security, Soonchunhyang University, Asan, Korea
{arupreza,munkhdelgerekh}@sch.ac.kr
[2] Department of Information Security Engineering, Soonchunhyang University, Asan, Korea
{catalyst32,brightprice,yim}@sch.ac.kr

Abstract. Due to the increasing complexity of the group of software and hardware components used in automobiles, current threats continue to hit the onboard network day after day. These additional components highlight the difficulties of developing compelling and responsive security solutions. To differentiate and defend automotive systems against deleterious exercises, a few intrusions detection systems (IDS) have been developed. Deep learning is one of the greatest options for detecting malicious packets where recurrent and convolutional neural network is vastly implemented. However, feature escalation is equally necessary for training a model. To safeguard automotive systems, we used an RNN-based LSTM algorithm as an intrusion detection system and wavelet conversion were used for feature escalation. This research emphasizes a depiction of vulnerabilities, highlights threat models, and makes it simple to recognize known threats that are displayed within the CAN.

1 Introduction

In the latest years, communication and automobile technology advanced by using the blessings of the internet of things (IoT) and day-by-day communication medium and new functions integrated into cars. The IoT includes sensors, smart devices, cloud stations, and so forth. These devices are linked via numerous communication protocols physically and exchange data inside the network. With the help of new components and smart terminals, the intelligent transportation system buildup and ended up a critical application and it is connected smart vehicles electronically with street infrastructure, mobile gadgets, and the internet. Current automobiles opened many entry points for hackers with Bluetooth, mobile communique, gateways, telematics, and multiple ECUs [1]. Ethernet in cars is currently receiving quite a little attention and is utilized in the latest automobiles to over-transmit huge amounts of data with high bandwidth and very low latency and jitter [2]. This paradigm shift has extended the attack surface and safety researchers have a scope of research and present-day automobiles are no longer secure from cyberattacks [3]. Cars use a variety of electronic devices and software programs. Presently, motors are geared up with more than 70 or one hundred electronic control units (ECUs) and near approximately 2500 data to transmit internally [4] with dependable inner communique

L. Barolli (Ed.): IMIS 2022, LNNS 496, pp. 85–96, 2022.
https://doi.org/10.1007/978-3-031-08819-3_9

over CAN buses and buses including C-CAN, M-CAN, and B-CAN media [5]. It's far geared up with numerous ECUs connected in parallel. Security researchers have supplied numerous strategies to reduce cyber threats to in-vehicle networks including the entropy-based approach that measures and finds out the abnormality in CAN traffic by the usage of self-data [6]. They overlook the semantic features and are concerned about statistical functions. However semantic features should consider for more efficient and powerful intrusion detection, and network protection analysis. The time interval-based IDS method can define frequency periods of CAN messages [7]. Our preceding research represents how data label differs for exceptional sorts of assault scenario [8].

Section 2 demonstrates an entire portrayal of CAN identity and represents which way it generates data set and previous studies. Different types of attack scenarios are explained here Sect. 3. Here data generation process, experimental setup, and data conversion method are described. Section 4 mentioned RNN LSTM structure. After that in Sect. 5 our experimental result stated. Sooner or later, the future plan and conclusion are given in Sect. 6.

2 Background and Motivation

2.1 Background

CAN ID is an identifier for the CAN data frame and payloads contained with this identity. Signal information is diagnosed with the aid of this identity. Due to the broadcast nature, CAN ID no longer includes which node getting this massage. But the payload which is 8 bytes hexadecimal number represents the actual commands. CAN massage is unique for each car version even for automobiles of the identical manufacturer [9]. Producers preserve those records immensely confidential but reverse engineering and data analysis can make them partially accessible (Fig. 1).

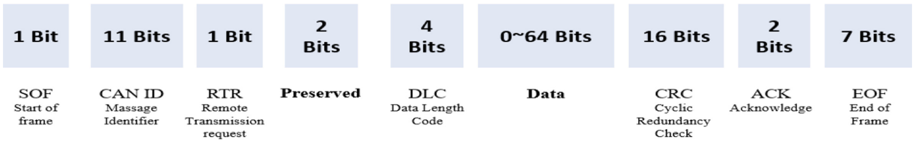

Fig. 1. Data frame of CAN protocol

The controller area network broadcast messages and all connected nodes receive the messages. CAN bus frame consists of 4 sorts [10]: data frame, error frame, overload frame, and remote frame. (1) Data frame: This is the only frame used for payload transfer. (2) Remote frame: This frame is solely used to request payload transfer. When an ECU gets a remote frame, it replies instantly. (3) Error frame: This frame defines and examines if there is an error. (4) Overload frame: This frame is used to postpone the commencement of the next message if there is an overload.

Now we describe the payload of CAN massage which is mainly a signal produced by ECUs. For communicating with each other all eight bytes are not used by ECUs. Those have few segmentations like Constants: Constant values remain unchanged over time.

Multi-Values: A few bytes responsible for the payload of the command are called multi-values [11], reported 2–3 changing values within these types of signals. An instance of a 2-value field will be responsible for the selected door for open and closed. Counters: counters are indicators that behave as cyclic counters inside a selected range. These indicators could function as additional syntax checks or be intended to order longer sign records at the destination ECU(s). Whilst this value is an internal positive range, it turns agitated, just like the speed value. Now we describe the payload of CAN massage which is mainly a signal produced by ECUs. Checkcodes: Except for the CRC-15 field at the end of each CAN body, the payload also can contain extra checkcodes, typically as the last signal within the payload.

2.2 Previous Work and Assault Scenario

In this phase, we talk about the essence of the related work concerning network anomaly detection of in-car systems. The number one to demonstrate the assault injection through wi-fi communication in-car systems. Koscher et al. become an investigation the security of contemporary vehicles [12]. By way of the usage of CARSHARK device, they collect all kinds of records and due to the broadcasting manner on CAN, attackers can easily have an effect on CAN communication. Open Car Testbed and Network Experiments (OCTANE) is a kind of device-generated data similar to the car packet. In this paper [13], the author implements this device and investigated various styles of assaults on related automobiles which provided an overview of the artificial neural network (ANN) - based IDS to protect against cyber attacks on modern-day vehicles systems. Javed AR et al. proposed the viability of evaluation approaches to detect intruders within the vehicle network [14]. They converted records points into a vector sequence to be fed into the CNN layers. DBN-based IDS for the CAN BUS IDS was provided with the aid of Kang et al. [15]. It provides an unsupervised deep belief network (DBN). LSTM model applied by Taylor et al. [16] Their concept is based on the prediction of the following statistics of the CAN bus network whilst acknowledging that the statistics originate from the senders. Kleberger et al. mentioned protection features architecture and afterward mentioned the problems and solutions [17]. The wireless attack was initiated with the aid of Woo et al. [18]. They stated the connected vehicle environment, numerous forms of assault models, and protection necessities. Luo J-N et al. [32] proposed an authentication mechanism for in-vehicle networks where they replaced CRC fields with MAC. Khan et al. investigated SDN-based false data injection into the brake-associated ECUs. They developed a fake data assault dataset and used LSTM to hit upon the attack, and they completed a detection rate of 87% [19]. Cloud-primarily based attack on the robotic vehicle made by Loukas et al. [20]. They used multiple machine learning classifiers. Researchers in [21] Jaynes et al. used a proprietary machine-learning set of policies to classify CAN bus messages. Their consequences display that the k-nearest-neighbor (k-NN) set of rules is more entire with 86.00% accuracy. Lee et al. [10] developed an IDS for reading request/response messages in the CAN bus, primarily based on offset ratio and time interval reviews.

In this section, we are discussing our implemented assaults fuzzing, DoS and replay.

Fuzzing Attack. In this elegance of assaults, the attacker makes use of any spoofing identity that includes subjective data to compose the message. In this situation, all hubs will receive an unusually useful spoofed message which is randomly created packet IDs. In this case, all nodes acquire anomalous anomalies [22]. To release a fuzzing assault, an attacker ought to first watch car internal massage and pick a goal. The fuzzing assault is basically produced at a slower rate than the DoS assault [23]. Be that as it could, it is plausible to carry out a fuzzing assault at a higher rate.

Dos Attack. High priority packets are injected by using an attacker in a short time interval on the bus and keep the bus busy. Typically, the attacker uses one or more excessive-priority IDs to generate DoS assault. Consequently, specific low-precedence nodes cannot get proper access to the network. All hubs share a single bus, increasing higher volume of data at the bus can create a delay or deny entries of valid packets example mentioned in [24]. The DoS attack can purpose a vehicle not to answer the driving force's commands on time.

Replay Attack Class. At first, attackers carefully observed the running massage and processed all the massage. Each payload contains important CAN message management. Therefore, the vehicle may experience manageable damage or surprising behavior. Replay attacks are one of the most important hard attacks to discover [25]. An attacker intercepts network traffic and excludes data originating from one or more randomly selected target node IDs. The attacker saves these data along with the actual packet access time. It is used later to accurately mimic or replay by injecting these packets into the network.

3 Data Description

3.1 Data Collection and Setup

Most of the researcher's interfering device is connected to an OBDII connector after that the analyzer can find available PID (parameter ID). But our process is different. We collect data from the internal gateway where all integrated ECUs are connected. In the gateway, we tap interfering devices specifically on CAN High and CAN Low and collect row data. The main difference is between the OBUII port, and our method that, in the OBUII port all ECUs are not connected therefore all types of data cannot be captured on the other hand in the internal gateway all ECUs data can be captured. Raw CAN traffic includes both diagnostic response messages and regular CAN traffic messages. Here, we analyze the raw CAN traffic and extract candidates that have the same value as the diagnostic response. The CAN specifications vary by vehicle model, but the PID is the same as defined in the J1979 standard [9]. Vehicle status data can be collected using standard PIDs, regardless of vehicle model. As an interfacing device, we use the PEAK CAN system. For our investigation, we use python Jupyter and Keres with TensorFlow as the backend. We conduct our experiment with Intel(R) Core (TM) i9-10900K CPU @ 3.70 GHz 3.70 GHz and NVIDIA GeForce RTX 2080 super.

3.2 Wavelet Transform and Feature Extraction

Wavelets are a popular tool for computational harmonic analysis. A notable feature is the functionality to carry out a multiresolution analysis [26]. Wavelets are absolutely suited for defining multiresolution capabilities. Those sparse representation belongings are key to the first-rate general performance of wavelets in applications including data compression and denoising. PyWavelets is a Python package that specifies some of the n-dimensional discrete wavelet transforms similarly to the 1D continuous wavelet transforms. All multidimensional modifications are implemented in Python via software that is separable from 1D transformations. PyWavelets had been designed for use with the aid of scientists working on a ramification of packages inclusive of time series evaluation, signal processing, image processing, and medical imaging [27]. The peaks in the frequency spectrum propose the maximum occurring frequencies within the signal. The bigger and sharper a height is, the greater commonplace a frequency is in a signal. The location (frequency-value) and height (amplitude) of the peaks within the frequency spectrum may be used as entering for classifiers. This easy technique works pretty properly for plenty of class issues. Wavelets had been able to classify the human activity recognition dataset with a 91% accuracy [28]. Due to the fact, that most of the indicators we see in real life are non-desk bound in nature. Approximately ECG indicators, the stock marketplace, device or sensor data, and so on, and many others, in actual-lifestyles problems, begin to get complex whilst all are coping with dynamic systems. A much higher technique for studying dynamic indicators is to use the Wavelet transform. The wavelet redesign has an excessive choice in each frequency and the time domain. It does no longer only inform us which frequencies are observed in a signal but additionally, at which period those frequencies have passed off. PyWavelets begin at the beginning of the character and slowly flow into wavelets closer to the end of the signal.

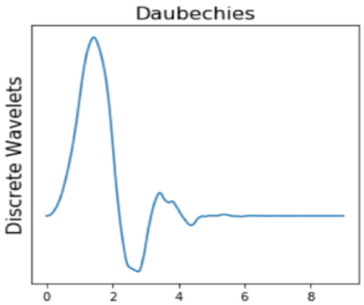

Fig. 2. Daubechies wavelet multiplier.

This process is also known as convolution. Wavelet transform is that there are numerous one-of-a-kind families (types) of wavelets. We would select a selected wavelet's own family which suits best the capabilities we're seeking out in our signal. Each kind of wavelet has a different shape, smoothness, and compactness and is beneficial for a particular data. A wavelet can be complex or actual. If it is complex, it's also divided into a real element representing the amplitude and an imaginary detail representing the phase.

Here, we use one family of wavelets known as ‘Daubechies’ shown in Fig. 2. It is an orthogonal wavelet that needs time-frequency localization. Wavelet coefficient shows approximation coefficients vector and detailed coefficients vector.

Here is used CAN ID and 8 bytes of hexadecimal data where every single byte is a PID. In CAN protocol single PID or combine PID carry out the functional massage and ECUs get command and apply all of the operations through this PID. As a result, PID changing pattern played an important role in detecting CAN vulnerability. A total of 1200000 data was captured. This data set was split by 10000 and shuffled for making the data set more challenging. The deep-learning algorithm cannot deal with characters and CAN ID and PID is character categories. For that reason, by using a label encoder, we give a numerical name for every character and convert the data set into the spectrum frequency domain and this wavelet conversion represents which time which frequency exists. In our data set, there is used label 2 which returns the double-level discrete wavelet transform (DWT) of the vector of x and as a result, it gives the coefficient cA1, cD2, cD1. cA represents the approximation coefficients vector and cD detail coefficients vector. cA1 and cD2 give three features and cD1 gives five features as a result total features of this data set are nine.

4 Deep Learning Model and Architecture

In RNN, LSTM is the most popular neural network that included special units referred to as memory blocks within the recurrent hidden layer. Memory blocks are self-connected cells, and they save the initial state of the network after it modifies the flow of the input in large multiplication units named gates. Each memory blocks have an entry port and exit port. The input gate controls the flow that enters the activation into the memory cell. The output gate controls the output flow of cell activations into the rest of the network. Afterward, in forget gate, all the memory block is connected [29]. The internal state of the cell is the forgetting gate scale, and it adds it as input to the cell by the cell’s self-repeating connection.

Table 1. Deep learning algorithm parameters

Parameter	Values
Activation function	tanh
Activation function output	Softmax
Output layer cells	4
Optimizer	RMSprop
Learning rate	0.0001
Batch size	128
Loss function	CategoricalCrossentropy
Dropout	0.2
Encoder	Label encoder

This adaptively forgets or resets the cell's memory. In addition, for learning the right timing of the outputs, the modern LSTM structure consists of peephole connections from its inner cells to the gates inside the identical cell [17]. Deep LSTM RNNs are built by stacking multiple LSTM layers. Note that LSTM RNNs are already deep neural network architectures withinside the experience that they will be taken into consideration as a feed-in advance neural network unrolled in time wherein every layer shares the same version parameters. Like a deep neural network models inputs undergo a couple of non-linear.

However, features at a particular point in time can most effectively be processed by using a single non-linear layer earlier than contributing to the output at that factor in time. Because of this, depth has a special that means in deep LSTM RNN. Inputs to the network at a particular time step do not propagate perfectly on the time LSTM layers and same as at additionally through multiple LSTM layers. There are distinctive thoughts about deep layers in RNNs permit the network to learn at different time scales over the input [18]. Deep LSTM RNNs provide some other advantages over standard LSTM RNNs: They are able to make higher use of parameters by using dispensing them over the gap using multiple layers. As an example, the model doesn't increase the memory size of the model, it remains approximately the same number of parameters. Our LSTM model consists of four hidden layers where used activation function as tanh and output activation function as softmax included with four neurons. Every hidden layer represents 120 neurons with dropout and batch normalization. Optimizer RMSprop with a learning rate of 0.0001 gave a better result. Table 1 mentioned the detailed brief.

5 Experiment Result and Performance Evaluation

There are so many patterns already found after data analysis. In specific CAN ID, the data injection time gap is almost the same. For example, in fizzing and DoS attack a set of CAN ID used for injecting high-frequency data. Another finding is, that here is used BMW car data set, where 56 CAN ID operates all of the internal commands in the car. But when the fuzzing attack appeared 1496 CAN ID was generated and in the DoS attack, the ID number was totally inverse from fuzzing. The number is 41, in a replay attack number of CAN ID remains the same as anticipated. After wavelet conversion data is arranged in a manner it can specify the variation of the data, the x-axis represents time, and the y-axis is frequency. The pattern between normal and attack mode can be differentiated easily. Though normal and replay attack have small difference and their frequency amplitude are identical -200 to 400 but in replay attack amplitude spikes define accurately this is not a normal data set, whereas normal data follow a periodic manner shown in Fig. 3(a).

Fuzzing attack frequency range between -1000 to 1500 which is higher than normal mode and in the replay there is missing the smoothness of frequency amplitude represented in Fig. 3(b). DoS attack frequency amplitude is lower than normal Fig. 3(c), the range is -100 to 200. In this experiment, the LSTM model is used, and LSTM needs 3-dimensional data as input, and we used here a time-series pattern because CAN massage is a time series data. At a time, it will take ten data sets and train the model. We reshape the data into 1200000, 10, 11, and feed the model. Our overall accuracy is 99.98% and

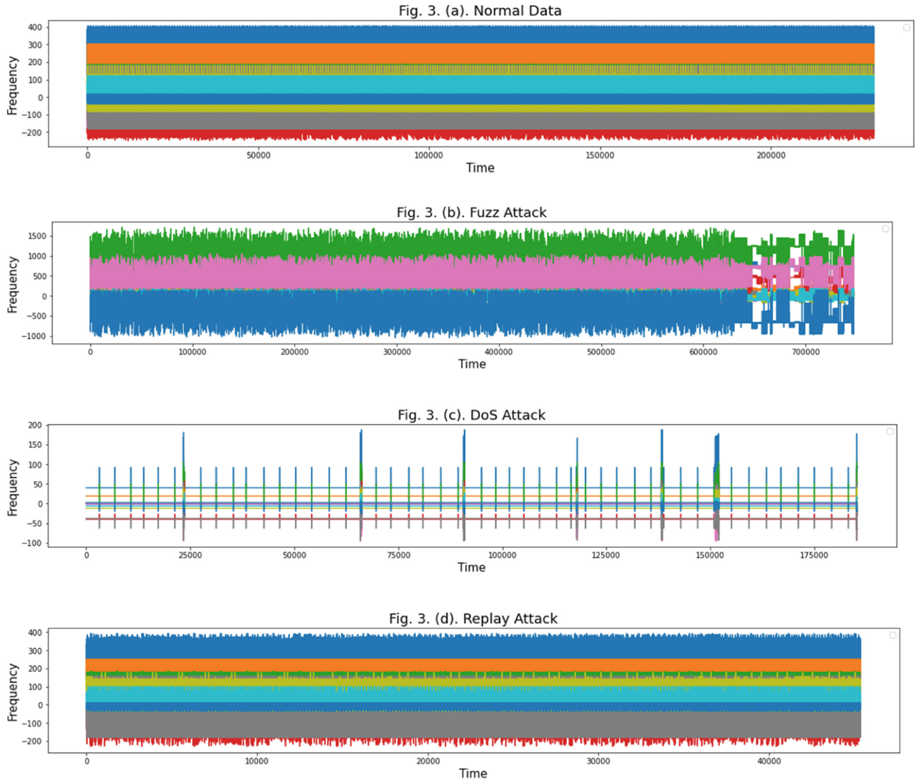

Fig. 3. The portrayal of normal and attack scenarios (a) Normal, (b) Fuzzing, (c) DoS, (d) Replay after wavelet conversion.

our ROC score is 0.9985341. Table 2 and confusion matrix Fig. 4(a) gives an overview of the result evaluation.

Table 2. Performance evaluations for implemented model.

Data type		Precision	Recall	F1-score	Total test data
Normal		0.99	1.00	1.00	11487
Fuzzing		1.00	1.00	1.00	36808
DoS		1.00	1.00	1.00	9231
Replay		1.00	0.97	0.98	2474
Overall score	Accuracy	NA	NA	1.00	60000
	Macro avg	1.00	0.99	1.00	60000
	Weighted avg	1.00	1.00	1.00	60000

The model can accurately identify Fuzzing and DoS attack, the performance rate almost is 100% because the Fuzzing and DoS attack has their own characteristics, and wavelet conversion identifies characteristics perfectly. In a replay attack, our model identifies a few amounts of normal data as a replay attack, but the overall performance is 97%. The reason is assaulter initiates a replay attack by actual data of the victim's car.

Figure 4(b) represents the ROC score for the multiclass classification. ROC curve is a performance measurement mechanism for classification problems at various threshold settings. ROC is a probability curve and AUC represents the degree or measure of separability. It tells how much the model is capable of distinguishing between classes. In our ROC curve normal, fuzzing and DoS attack are classified perfectly because everyone has their own pattern but replay attack AUC is 98% because of the similarity with normal data.

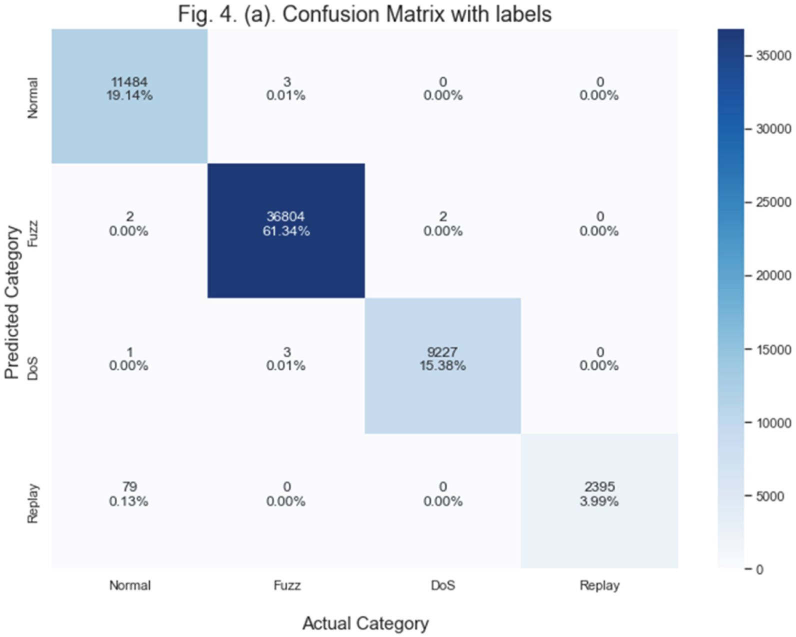

Fig. 4. Evaluation score (a) Confusion matrix, (b) ROC score.

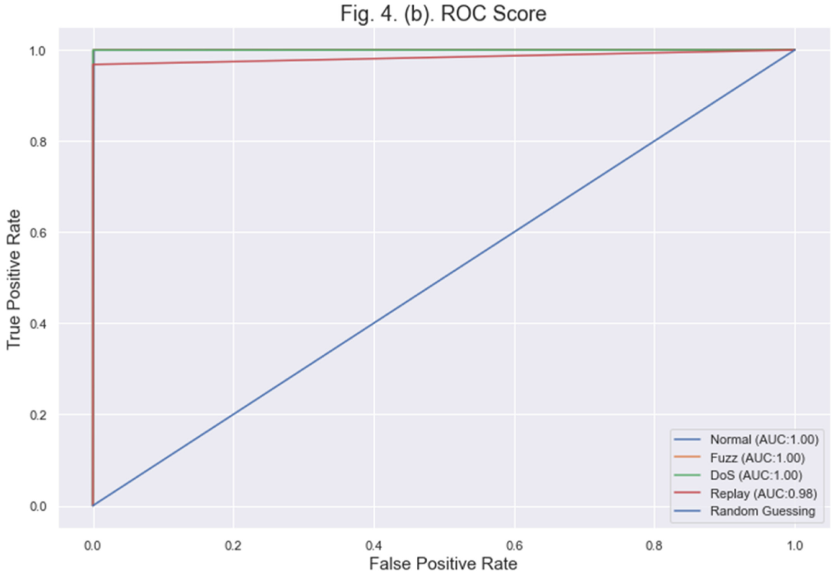

Fig. 4. continued

6 Conclusion

Even though the automobile trying-out era has made exquisite progress, there is still a lack of relevant vehicle safety assessment tools inside the market. Existing testing tools additionally have problems. For that reason, a research scope opened for the researcher. We are employed here in an LSTM model. Other researchers have employed LSTM before, but their input data is different. Here we tried to find out data changing patterns in the frequency domain for different types of attacks and used core data from a registered car. In this investigation, we used statistical data to detect malicious data for BMW model car and it is perfectly identifying the malicious. Assaulters use different types of attack, but they cannot maintain normal mode frequency patterns and wavelets specify all of the details. Which make our IDS specialty that can differentiate normal and abnormal situation for every single portion. Our future goal is by analyzing all features of CAN massage to make a universal IDS that can be employed for all types of cars.

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

References

1. Kelarestaghi, K.B., Foruhandeh, M., Heaslip, K., Gerdes, R.: Intelligent transportation system security: impact-oriented risk assessment of in-vehicle networks. IEEE Intell. Transp. Syst. Mag. **13**(2), 91–104 (2021). https://doi.org/10.1109/MITS.2018.2889714
2. Carnevale, B., Fanucci, L., Bisase, S., Hunjan, H.: MACsec-based security for automotive ethernet backbones. J. Circuits Syst. Comput. **27**(05), 1850082 (2018). https://doi.org/10.1142/S0218126618500822
3. Checkoway, S., et al.: Comprehensive experimental analyses of automotive attack surfaces. In: Proceedings of the 20th USENIX Security Symposium, pp. 77–92 (2011)
4. Hassija, V., Chamola, V., Saxena, V., Jain, D., Goyal, P., Sikdar, B.: A survey on IoT security: application areas, security threats, and solution architectures. IEEE Access 82721–82743 (2019). https://doi.org/10.1109/ACCESS.2019.2924045
5. 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.) IMIS 2020. AISC, vol. 1195, pp. 234–243. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-50399-4_23
6. Muter, M., Asaj, N.: Entropy-based anomaly detection for in-vehicle networks. In: 2011 IEEE Intelligent Vehicles Symposium (IV), pp. 1110–1115, June 2011. https://doi.org/10.1109/IVS.2011.5940552
7. Song, H.M., Kim, H.R., Kim, H.K.: Intrusion detection system based on the analysis of time intervals of CAN messages for in-vehicle network (2016). https://doi.org/10.1109/ICOIN.2016.7427089
8. Islam, M.R., Oh, I., Batzorig, M., Kim, S., Yim, K.: A concept of IDS for CAN protocol based on statics theory. In: Barolli, L. (ed.) BWCCA 2021. LNNS, vol. 346, pp. 294–302. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-90072-4_32
9. Song, H.M., Kim, H.K.: Discovering CAN specification using on-board diagnostics. IEEE Des. Test **38**(3), 93–103 (2021). https://doi.org/10.1109/MDAT.2020.3011036
10. Lee, H., Jeong, S.H., Kim, H.K.: OTIDS: a novel intrusion detection system for in-vehicle network by using remote frame. In: 2017 15th Annual Conference on Privacy, Security and Trust (PST), pp. 57–5709. IEEE (2017). https://doi.org/10.1109/PST.2017.00017
11. Markovitz, M., Wool, A.: Field classification, modeling and anomaly detection in unknown CAN bus networks. Veh. Commun. **9**, 43–52 (2017). https://doi.org/10.1016/j.vehcom.2017.02.005
12. Koscher, K., et al.: Experimental security analysis of a modern automobile (2010). https://doi.org/10.1109/SP.2010.34
13. Haas, R.E., Moller, D.P.F., Bansal, P., Ghosh, R., Bhat, S.S.: Intrusion detection in connected cars. In: 2017 IEEE International Conference on Electro Information Technology (EIT), pp. 516–519, May 2017. https://doi.org/10.1109/EIT.2017.8053416
14. Javed, A.R., Rehman, S.U., Khan, M.U., Alazab, M., Reddy, T.: CANintelliIDS: detecting in-vehicle intrusion attacks on a controller area network using CNN and attention-based GRU. IEEE Trans. Netw. Sci. Eng. **8**(2), 1456–1466 (2021). https://doi.org/10.1109/TNSE.2021.3059881
15. Kang, M.-J., Kang, J.-W.: Intrusion detection system using deep neural network for in-vehicle network security. PLoS ONE **11**(6), e0155781 (2016). https://doi.org/10.1371/journal.pone.0155781
16. Taylor, A., Leblanc, S., Japkowicz, N.: Anomaly detection in automobile control network data with long short-term memory networks. In: 2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA), pp. 130–139, October 2016. https://doi.org/10.1109/DSAA.2016.20

17. Kleberger, P., Olovsson, T., Jonsson, E.: Security aspects of the in-vehicle network in the connected car (2011). https://doi.org/10.1109/IVS.2011.5940525
18. 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. 1–14 (2014). https://doi.org/10.1109/TITS.2014.2351612
19. Khan, Z., Chowdhury, M., Islam, M., Huang, C.-Y., Rahman, M.: Long short-term memory neural networks for false information attack detection in software-defined in-vehicle network, June 2019. http://arxiv.org/abs/1906.10203
20. Loukas, G., Vuong, T., Heartfield, R., Sakellari, G., Yoon, Y., Gan, D.: Cloud-based cyber-physical intrusion detection for vehicles using deep learning. IEEE Access **6**, 3491–3508 (2018). https://doi.org/10.1109/ACCESS.2017.2782159
21. Jaynes, M., Dantu, R., Varriale, R., Evans, N.: Automating ECU identification for vehicle security (2017). https://doi.org/10.1109/ICMLA.2016.53
22. Lee, H., Choi, K., Chung, K., Kim, J., Yim, K.: Fuzzing CAN packets into automobiles. In: 2015 IEEE 29th International Conference on Advanced Information Networking and Applications, pp. 817–821, March 2015. https://doi.org/10.1109/AINA.2015.274
23. Nowdehi, N., Aoudi, W., Almgren, M., Olovsson, T.: CASAD: can-aware stealthy-attack detection for in-vehicle networks, September 2019. http://arxiv.org/abs/1909.08407
24. Murvay, P.-S., Groza, B.: DoS attacks on controller area networks by fault injections from the software layer. In: Proceedings of the 12th International Conference on Availability, Reliability and Security, pp. 1–10, August 2017. https://doi.org/10.1145/3098954.3103174
25. Hoppe, T., Kiltz, S., Lang, A., Dittmann, J.: Exemplary automotive attack scenarios: trojan horses for electronic throttle control system (ETC) and replay attacks on the power window system, VDI Berichte, pp. 165–183 (2007)
26. Mallat, S.: A Wavelet Tour of Signal Processing. Elsevier, Amsterdam (2009)
27. Lee, G., Gommers, R., Waselewski, F., Wohlfahrt, K., O'Leary, A.: PyWavelets: a python package for wavelet analysis. J. Open Source Softw. **4**(36), 1237 (2019). https://doi.org/10.21105/joss.01237
28. Taspinar, A.: A guide for using the wavelet transform in machine learning (2018). https://ataspinar.com/
29. Gers, F.A., Schraudolph, N.N., Schmidhuber, J.: CrossRef List. Deleted DOIs, vol. 1 (2000). https://doi.org/10.1162/153244303768966139
30. Gers, F.A., Schmidhuber, J., Cummins, F.: Learning to forget: continual prediction with LSTM. Neural Comput. **3**, 115–143 (2000). https://doi.org/10.1162/089976600300015015
31. Hermans, M., Schrauwen, B.: Training and analyzing deep recurrent neural networks. In: Advances in Neural Information Processing Systems, 2013. Appendix: Checklist of Items to be Sent to Conference Proceedings Editors (see instructions at conference webpage), pp. 190–198 (2013)
32. Luo, J.-N., Wu, C.-M., Yang, M.-H.: A CAN-bus lightweight authentication scheme. Sensors **21**(21), 7069 (2021). https://doi.org/10.3390/s21217069