CANGuard: Practical Intrusion Detection for In-Vehicle Network via Unsupervised Learning

WU ZHOU*, Didi Research America, USA
HAO FU*, Facebook, Inc., USA
SHRAY KAPOOR, Didi Research America, USA

Abstract Modern vehicles are becoming more advanced recently by incorporating new functionalities, such as V2X, more connectivity and autonomous driving. However, these new things also open the vehicle wider to the outside and thus pose more severe threats to the vehicle security and safety.

In this paper, we propose CANGuard, a vehicle intrusion detection system that learns in-vehicle traffic patterns and uses the patterns to detect anomaly in a vehicle network. CANGuard applies autoencoder, an unsupervised learning technique, on the raw CAN messages to learn efficient models of these data, and requires no expert to label CAN messages as needed in supervised approaches. Unlike another study that also uses unsupervised learning but can only detect attacks involving one single type of message, CANGuard can detect attacks involving multiple types of messages as well. Experiments with public data sets demonstrate that CANGuard has almost the same, at some case better, results as compared with state-of-art supervised approaches. Combined with its unsupervised nature and its capability to detect attacks involving multiple types of message, this proves CANGuard is more practical to be deployed in modern vehicle environments.

ACM Reference Format:
Wu Zhou, Hao Fu, and Shray Kapoor. 2021. CANGuard: Practical Intrusion Detection for In-Vehicle Network via Unsupervised Learning. In *The Sixth ACM/IEEE Symposium on Edge Computing (SEC '21), December 14–17, 2021, San Jose, CA, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3453142.3493514

1 INTRODUCTION

Modern vehicles are becoming more advanced while vendors incorporate more and more new functionalities, such as infotainment, telematics, and autonomous driving. However, these new components also open the vehicle wider to the outside and thus bring more severe threats. Recent events [3, 9] showed that attacks can come from different access points and reach internal units on the vehicle network to achieve various purposes (turning the gear, stopping the engine, etc.). Control area network is the dominant network used by modern vehicles and highly prone to a variety of attacks.

1.1 Controller Area Network and the Threats

Modern vehicle is a complex system composed of a number of electronic control units (ECUs), such as engine control, body control, etc. To work efficiently and safely, those ECUs need to communicate with each other to exchange critical data. Controller area network (CAN) is the major communication channel designed for this purpose. All ECUs connect to the CAN bus and send messages to the network such that other units can receive and take corresponding actions.

CAN message is composed of three parts. An identifier (ID) is used to specify what the message is used for. Payload contains the actual data to sent. A length field tells how long the payload is. For historical reasons, CAN bus is designed to be lightweight and lacks common security mechanism such as authentication. Any devices connected to the CAN can send messages to and receive messages from the network. There is no easy way to determine who sends the message. This, along with others, brings the following threats to the CAN bus.

DoS Attacks Denial of service (DoS) is to disable the working of the victim units. Other than DoS itself, attacker can use it as stepping stone for indirect purposes. For example, the masquerade attack discussed later uses DoS attack to bring down one victim first. On CAN bus, there can be two kinds of DoS: the first sends CAN messages with ID 0 that has the highest priority, the second sends large amount of messages with random IDs such that other ECUs exhaust all their resources in handling the attacking messages.

Spoofing Attack In spoofing attack, attackers send messages that seemingly come from another unit. This can mislead other units to get fake information at least, and even worse bring the vehicle into unsafe situation. Spoofing usually involves one unit sending message with ID assigned to another. Attackers usually achieve this by compromising one outside-facing unit and sending fake messages targeting some internal units.

Masquerade Attack The above attacks involve only one phase. Masquerade attack, however, needs first to disable one unit through DoS, and uses another unit to masquerade the disabled one. The necessity of the first step is because some units maybe very hard to manipulate so as to inject any messages as needed by attacking purpose.

1.2 CAN IDS

Network based intrusion detection system (NIDS) is a natural resort for people to explore for defending against these network threats. Fig 1 shows an overview diagram about how NIDS can be deployed. Usually, to communicate with outside, vehicles expose two types of access points: one is remote access point such as bluetooth and cellular interfaces, the other is physical one such as on-board diagnostic(OBD) unit designed for problem triage purpose. For effective defense, NIDS is usually deployed alongside the vehicle gateway, where it has access to all CAN messages flowing inside the vehicle network.

*These authors contributed equally to this work

SEC '21, December 14–17, 2021, San Jose, CA, USA

ACM ISBN 978-1-4503-8390-5/21/12…$15.00
https://doi.org/10.1145/3453142.3493514

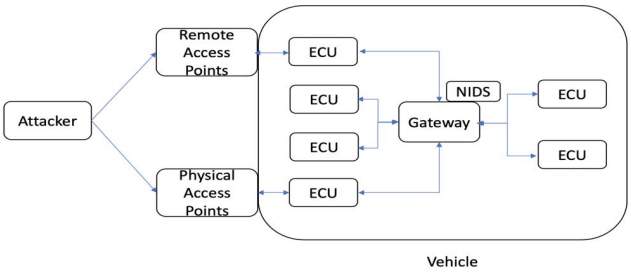

Fig. 1. Vehicle NIDS

To protect vehicles against potential threats, researchers proposed a variety of IDS to detect CAN attacks. Earlier solutions [7, 12] extracted different signatures based on the analysis of the detailed attack flow or the standard ECU specification. To save people from laborious analysis work, recent researchers [13, 15] developed machine learning and neural network based solutions to learn effective model to classify CAN packets as normal or attacks. However, these systems were supervised learning and required knowledgeable person to label the training data. There is also study of using unsupervised learning [14] to detect vehicle attacks as anomaly in the CAN message traffic. However, it only applies to CAN message stream with a unique identifier, and thus lacks the capability of detecting potential attacks that involve more than one type of messages.

For better defense against the evolving threats, we proposed CANGuard - an unsupervised learning framework to learn the efficient encoding of the normal CAN packets and use the trained model to detect anomaly in a target network. Unlike [14], CANGuard can also detect attacks involving multiple types of messages. Experiments with public data sets demonstrate that CANGuard has almost the same, at some case better, results as compared with state-of-art supervised approaches. Combined with its unsupervised nature, this proves CANGuard is more practical to be deployed in modern vehicle environments.

The rest of the paper is organized as follows. Section 2 reviews the related work. Section 3 presents the design of CANGuard, followed by the evaluation in Section 4. Finally, we conclude and discuss future work in Section 5.

2 RELATED WORK

In this section, we review prior of art researches relevant to CANGuard, particularly in the fields of automotive security and CAN intrusion detection system.

Automotive Security Koscher et al. [9] conducted one of the first experimental study of modern automobile security and showed that CAN bus can be exploited to damage vehicle safety. To address the threats, researchers proposed a variety of solutions. For example, Hartkopp et al. and Groza et al. [5] proposed to use message authentication to enhance CAN message security. Lemke et al. [10] proposed to apply firewall concept and use message signature to filter malicious traffic on CAN bus. They were effective in some cases, but cannot eliminate all potential CAN threats.

CAN IDS Hoppe et al. [7] used some special patterns, such as the increasing rate of cyclic CAN packet occurrences and obvious forged packet IDs in the CAN packets, to detect studied attacks. Muter et al. [12] extracted different attack signature based on the analysis of the standard ECU specification . Cho et al. [4] proposed to use some physical characteristics, such as the clock skew, to fingerprint particular ECUs. Those approaches usually required deep analysis of the general ECU specification, the individual ECU physical characteristics, or the anatomy of each attack process, and thus not scalable to new ECU modules and emerging CAN bus attacks.

To alleviate the laborious analysis cost of prior approaches, several IDSs based on machine learning have been proposed. Wu et al. [15] calculated information entropy of each time window to detect abnormal CAN messages. Avatefipour et al [2] leveraged one-class support vector machine for intrusion detection in CAN bus. These usually involve some types of feature engineering and need expert knowledge to manually identify a particular representation of that raw data that conveys characteristics uniquely seen to normal messages.

Recently, researchers proposed to incorporate automatic feature learning to counter vehicle attacks. Kang and Kang [8] constructed a deep belief network structure to build a classifier and evaluated it on a simulated dataset. Song et al. [13] proposed a system based on a deep convolutional neural network (DCNN) to protect the CAN bus of the vehicle. All those solution involved some sort of supervised learning technique, in contrast to the unsupervised nature of CANGuard. Taylor et al. [14] utilize long short-term memory network to predict anomaly in sequential CAN message stream in an unsupervised manner, but it only works well with message stream bearing one unique identifier.

3 CANGUARD

CANGuard is an unsupervised learning framework based on autoencoder technique and designed to handle both types of attacks: the dedicated IDS to detect attacks involving messages of only one ID, e.g., spoofing attacks, and the one-fits-all IDS to detect attacks involving more IDs, e.g., fuzzy DoS and masquerade attacks. Below we discuss CANGuard's architecture and each component.

3.1 Overall Architecture

Fig 2 shows the overall architecture of CANGuard IDS. Basically, CANGuard treats the input CAN message as a real-time input stream and the task as an anomaly detection problem. It works in two phases - the first to train an autoencoder model and the latter to predict anomaly using the trained model. Both phases start with preprocessing and feature mapper modules, with the input on left being the CAN message in normal environment and that on right the CAN messages from unknown environment. The preprocessing pipelines prepare the messages for the later efficient handling. Feature mappers convert these preprocessed messages into a series of input vectors, which will be fed into the later model training and prediction steps respectively at two sides.

The model CANGuard uses is autoencoder, an unsupervised learning framework. At the model training side, it accepts streams of CAN input vectors and tunes effective parameters for the later consumption. The model prediction side uses the generated model to calculate a score for each unknown CAN message. If the score is

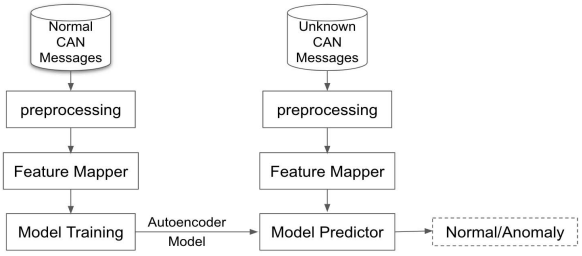

Fig. 2. CANGuard Overview

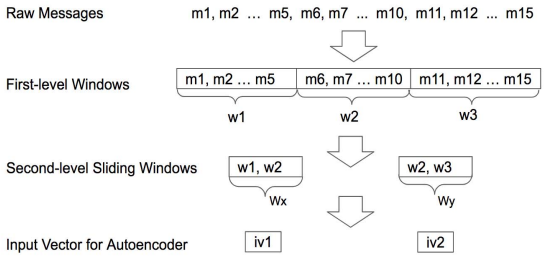

Fig. 3. Message Processing

greater than a threshold, CANGuard marks it anomaly, otherwise normal.

3.2 Message Preprocessing and Feature Mapper

To capture unique features that are characteristic of normal object and not prone to manipulation by attackers, statistical features are usually extracted. However, it is not efficient to compute statistics of long streams of input messages. Also the real-time requirement is not trivial to implement. CANGuard uses a message preprocessing pipeline to prepare the CAN message traffic for later steps.

As shown in Fig 3, CANGuard first divides the raw messages into separate windows of a pre-selected size. For example, window *w1* includes messages *m1* to *m5*, window *w2* includes messages *m6* to *m10*, and so on. In stage 2, a sliding window is applied to obtain windows of larger size from the smaller ones. For example, window *wx* includes *w1* and *w2*. Note that both *wx* and *wy* share information from messages m6 to m10.

For each second-level window in Fig 3, feature mapper generates one input vector which is composed of an encoded arbitration ID, a relative timestamp, and the payload. The encoded arbitration ID is designed to compress the raw message identifiers as included in the sliding window. The payload is generated by assembling these of the raw messages in the sliding window. The relative timestamp is used here to recognize different sequences of messages. For example, during normal operation each CAN message with the same identifier is commonly separated by a fixed interval. But when it was sent by an adversary, which usually works on a different system or clock, it is very possible that the messages are separated by a different interval. So even though the messages have totally the same sequence of payloads and arbitration identifiers, by including the relative timestamp as one feature autoencoder can still determine that this sequence of messages is abnormal.

3.3 Dedicated IDS Based on Autoencoder

Autoencoder is a type of artificial neural network recently applied to address anomaly detection task [1]. Fig 4 illustrates how CANGuard uses autoencoder to detect anomaly in CAN message stream. The major components include encoder(s), decoder(s) and loss function. The encoder is composed of a series of neural networks with weight and bias parameters, applied on the input vector to generate corresponding intermediate vector with a reduced dimension. The decoder has another set of neural networks with its own set of parameters, applied on the intermediate vector to reverse the encoding process. The output vector shares the same dimension as input vector and their difference will be measured by the loss function as reconstruction loss to indicate how well the model is doing in characterizing the unique features of the trained data set. CANGuard uses them to learn an optimum set of model parameters in the training, to detect anomaly in the prediction, and to determine the loss threshold in the middle.

The training phase accepts normal messages captured in controlled environment as input. The autoencoder model has a preselected layer of neural networks with an initial set of parameters. While applied on the input vectors, the generated reconstruction losses indicate how well the current set of parameters capture the unique features of normal CAN messages. With an optimization algorithm, training phase iteratively adjust the autoencoder parameters until the calculated loss values reach minimum. At the end of this process, the loss values for every input vectors are combined together to form a numeric series, which show the distribution of reconstruction losses over all normal message.

With the trained model, CANGuard predictor works in online mode as unknown CAN messages come in as live stream. To detect anomaly, the predictor goes through the same process to compute a reconstruction loss on each window of unknown messages and compares the new loss value to the threshold decided in the validation phase. Small loss value shows that the model can easily reconstruct the unknown messages and thus signals normal message. Larger loss value indicates that the model has difficulty in recovering the original messages and thus signals anomaly.

Different thresholds usually result in different false alarm and false negative rates. We have to consider their trade-off while selecting threshold. However, practical factors have big impact on how we weight these rates. For example, high false negative is not acceptable for life-threating risk, while high false alarm not tolerable in other situations. To gain flexibility and help deployment decide proper threshold, CANGuard uses a weighted mean of these two rates called F_β score:

$$F_\beta = \left(1 + \beta^2\right)\left(\frac{Precision \times Recall}{\beta^2 \times Precision + Recall}\right)$$

The value of β specifies the times of weight we assign to precision as to recall. At the end of the training phase, CANGuard slides along the distribution of reconstruction loss to find a minimum value of F_β score. The loss value on this point is selected as the final threshold.

CANGuard dedicates one autoencoder model to each message ID, i.e., messages with different IDs are fed into their own training and prediction pipelines to form a series of dedicated IDS. This is

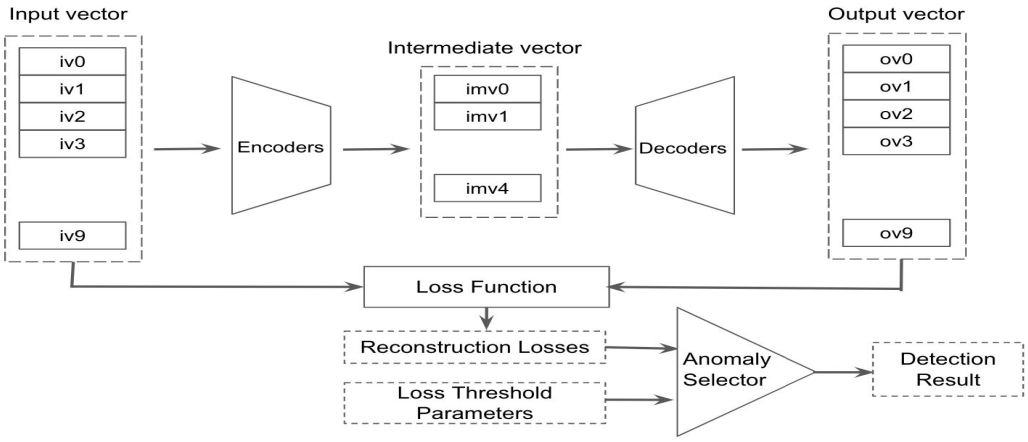

Fig. 4. Autoencoder model for Dedicated IDS

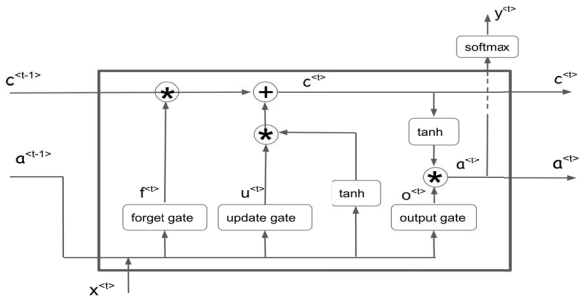

Fig. 5. LSTM

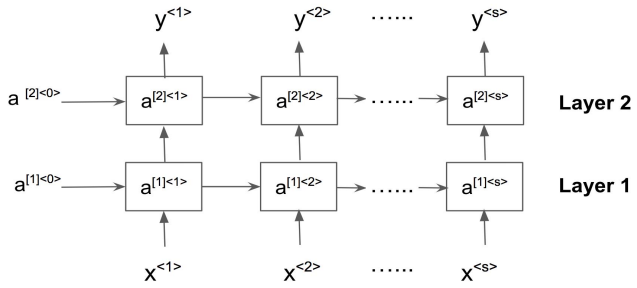

Fig. 6. Deep Network of LSTM

both effective and efficient in handling CAN threats involving only one ID. Section 4 shows some evaluation results for dedicated IDS. The tuning of one-fits-all model to detect anomaly involving more message types is still ongoing and left for future work.

3.4 One-fits-all IDS Using LSTM

Dedicated IDS uses sequential neural networks as model encoder-decoder to reconstruct the original messages and works well in deriving the representative characteristics of packets with unique identifier, thus able to detect anomaly in CAN attacks involving a single message type such as spoofing. However, there are more complicated attacks where multiple message channels are manipulated in a correlated way, e.g., fuzzy and masquerade attacks. Dedicated IDS will miss these attacks as each IDS will only check into the anomaly in one single channel. To address this deficiency, we propose a one-fits-all model, which use long short-term memory (LSTM) as model encoder-decoder to detect anomaly across multiple channels.

LSTM [6], shown in Fig 5, is an improved recurrent neural network architecture designed to solve the vanishing gradient problem. Instead of using traditional feedforward connections, LSTM has feedback connections, capable of processing data over a long time series and applied to a variety of tasks such as time series prediction, machine translation and speech recognition. With LSTM's time series prediction capability, Malhotra et al. [11] proposed to use the prediction error as an indicator of anomaly in the time series. More closely related to our work, Taylor [14] applied LSTM on the CAN message traffic to predict the sequential messages and use the error between the original messages and the predicted messages to detect anomaly in messages with a single identifier.

To detect anomaly across multiple channels, we propose to use LSTM in a different way. Same to the dedicated IDS as described by Fig 4, one-fits-all IDS bases its anomaly detection capability on the autoencoder model, which can correctly reconstruct normal packets with low loss rate and distinguish anomalous packets with a much higher loss rate. LSTM, instead of the sequential neural network, is used as the basic neuron of the encoder and decoder such that the correlation among packets with different identifiers can be effectively characterized by its memorization capability along the message sequence. Particularly, the one-fits-all IDS uses four layers of LSTM in the autoencoder architecture, with two layers for the encoder and decoder respectively. The deep neural network structure is depicted in Fig 6 and the evaluation result is presented in Section 4.

4 EVALUATION

In 2020, Song et al. [13] published four data sets to evaluate CAN IDS and used them to compare their supervised learning approach against some other machine learning solutions. Among these data sets, gear spoofing, RPM spoofing and zero identifier DoS only involve one type of CAN identifier. We use them to evaluate our

Table 1. Gear Spoofing

Algorithm	Precision	Recall	F1 score
ResNet	0.9999	0.9989	0.9994
SVM	1.0	0.9965	0.9982
Autoencoder	0.9998	0.9993	**0.9995**

Table 2. RPM Spoofing

Algorithm	Precision	Recall	F1 score
ResNet	0.9999	0.9994	0.9996
SVM	1.0	0.9977	0.9988
Autoencoder	0.9999	0.9994	**0.9996**

Table 3. Zero-Identifier DoS

Algorithm	Precision	Recall	F1 score
ResNet	1.0	0.9989	0.9995
SVM	1.0	0.9944	0.9972
Autoencoder	0.9997	0.9994	**0.9995**

Table 4. Fuzzy DoS

Algorithm	Precision	Recall	F1 score
ResNet	0.9995	0.9965	0.9980
SVM	0.9928	0.9555	0.9738
Autoencoder	0.9975	0.9991	**0.9993**

dedicated IDS. Fuzzy DoS involves multiple types of CAN identifiers and we use it to evaluate one-fits-all IDS. Each data set was created by first collecting CAN traffic from a real vehicle and then injecting fabricated messages in a controlled environment. The normal messages were collected for about forty minutes, while the injected messages to simulate corresponding attack lasted 3 to 5 seconds. In total there are 26 distinct arbitration IDs on the CAN bus at normal status.

In our setting, we set the model layer numbers to *4* for all the autoencoders we trained - both the dedicated IDS and one-fits-all IDS two encoders and two decoders. We set the size of the sliding window to *10* and the code length of the autoencoder to *5*. We used the learning rate of *1e-3*. After getting the performance result, we further compare them with two state-of-art approaches: reduced inception ResNet that reported the best result in the category of neural network approaches, and support vector machine (SVM) that reported the best result in the category of traditional machine learning approaches. While deciding the proper threshold, we set the value of β to 1 and assign same weights to false positive and false negative rates.

Table 1, 2, 3 and 4 show CANGuard's performance as compared with reduced inception ResNet and SVM. As shown from *F1 score* column, CANGuard reports the best result at detecting gear spoofing and shares the first position at detecting RPM spoofing and DoS. In the subdivided aspects of precision and recall, the results from CANGuard are also comparable with the best approaches. So overall, we believe CANGuard generated the same (or better) results in detecting the CAN threat that involves either one or multiple identifiers. Combined with its unsupervised nature, this proves that CANGuard can be a more practical IDS to be used in different vehicles.

5 CONCLUSION

In this paper, we propose CANGuard, a vehicle network intrusion detection system that learns in-vehicle traffic patterns and uses the learned patterns to detect anomalous behaviour in vehicle network. CANGuard applies autoencoder, an unsupervised learning technique, on the raw CAN messages to learn an efficient coding, and thus eliminates the need of expert labelling of these messages in prior supervised learning approaches. With the implementation and evaluation of dedicated IDS and one-fits-all IDS, we show that CANGuard achieves a same, in some case better, results as compared with the state of art supervised learning approach. This makes CANGuard a more practical solution to be deployed in a variety of vehicle environments.

REFERENCES

[1] J. An and S. Cho. Variational autoencoder based anomaly detection using reconstruction probability. *Special Lecture on IE*, 2(1):1–18, 2015.

[2] O. Avatefipour, A. S. Al-Sumaiti, A. M. El-Sherbeeny, E. M. Awwad, M. A. Elmeligy, M. A. Mohamed, and H. Malik. An intelligent secured framework for cyberattack detection in electric vehiclesâĂŹ can bus using machine learning. *IEEE Access*, 7:127580–127592, 2019.

[3] Z. Cai, A. Wang, and W. Zhang. 0-days & mitigations: Roadways to exploit and secure connected bmw cars. In *2019 BlackHat USA*, 2019.

[4] K.-T. Cho and K. G. Shin. Fingerprinting electronic control units for vehicle intrusion detection. In *25th {USENIX} Security Symposium ({USENIX} Security 16)*, pages 911–927, 2016.

[5] O. Hartkopp and R. M. SCHILLING. Message authenticated can. In *Escar Conference, Berlin, Germany*, 2012.

[6] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[7] T. Hoppe, S. Kiltz, and J. Dittmann. Applying intrusion detection to automotive it-early insights and remaining challenges. *Journal of Information Assurance and Security (JIAS)*, 4(6):226–235, 2009.

[8] M.-J. Kang and J.-W. Kang. Intrusion detection system using deep neural network for in-vehicle network security. *PloS one*, 11(6):e0155781, 2016.

[9] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, and S. Savage. Experimental security analysis of a modern automobile. In *2010 IEEE Symposium on Security and Privacy*, pages 447–462, 2010.

[10] K. Lemke, C. Paar, and M. Wolf. *Embedded security in cars*. Springer, 2006.

[11] P. Malhotra, L. Vig, G. Shroff, and P. Agarwal. Long short term memory networks for anomaly detection in time series. In *Proceedings*, volume 89, pages 89–94. Presses universitaires de Louvain, 2015.

[12] M. Müter, A. Groll, and F. C. Freiling. A structured approach to anomaly detection for in-vehicle networks. In *2010 Sixth International Conference on Information Assurance and Security*, pages 92–98. IEEE, 2010.

[13] H. M. Song, J. Woo, and H. K. Kim. In-vehicle network intrusion detection using deep convolutional neural network. *Vehicular Communications*, 21:100198, 2020.

[14] A. Taylor, S. Leblanc, and N. Japkowicz. 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)*, pages 130–139. IEEE, 2016.

[15] W. Wu, Y. Huang, R. Kurachi, G. Zeng, G. Xie, R. Li, and K. Li. Sliding window optimized information entropy analysis method for intrusion detection on in-vehicle networks. *IEEE Access*, 6:45233–45245, 2018.