Universal Intrusion Detection System on In-Vehicle Network

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

[1] Department of Software Convergence, Soonchunhyang University, Asan, Korea
arupreza@sch.ac.kr

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

Abstract. The Controller Area Network (CAN) protocol is widely used in automotive and industrial applications for communication. However, the lack of authentication and encryption in CAN bus networks has made them vulnerable to cyberattacks. This study investigated the effectiveness of different intrusion detection models in accurately classifying attacks, such as Denial-of-Service (DoS) attacks, fuzzing, and replay attacks. A labeled dataset was created using a methodology that uses the CAN ID sequence, time gap, and hamming distance between hexadecimal strings of equal length. The resulting dataset was segmented and converted into heat maps that were input to deep learning models such as VGG-16, AlexNet, and ResNet-50. The study provides valuable insights for developing more robust security measures for in-vehicle networks. However, recent research has shown that intrusion detection systems need to be developed individually for each vehicle, taking into account the unique data characteristics of the vehicle. Therefore, this paper proposes to implement universal IDS by using three types of CNN architectures to find the best one that is suitable for all types of attacks with high accuracy.

1 Introduction

Controller Area Network (CAN) is a communication protocol widely used in the automotive and industrial sectors that allows electronic devices to communicate with each other and facilitate real-time data exchange and control. The multi-master bus characteristic of the CAN bus makes it very reliable and robust, allowing multiple devices to simultaneously transmit data on the bus without conflict [1]. However, with the increasing use of Internet of Things (IoT) and Internet of Vehicles (IoV) devices and connectivity, CAN bus networks have become increasingly vulnerable to cyberattacks. The openness and lack of authentication in CAN bus networks make them vulnerable to a range of attacks, with attackers able to manipulate CAN messages to gain control of vehicles or industrial processes [2]. Furthermore, the encryption and authentication in CAN bus networks makes it easier for attackers to access the network and compromise its integrity, which can have serious consequences.

To address these challenges, systems to detect intrusions into the in-vehicle network have gained significant attention. These systems can detect and respond to potential

L. Barolli (Ed.): IMIS 2023, LNDECT 177, pp. 78–85, 2023.
https://doi.org/10.1007/978-3-031-35836-4_9

cyberattacks, mitigate their impact on the integrity of the CAN bus network, and ensure secure communications between devices. However, recent research has shown the importance of developing a customized attack detection system for each vehicle. According to CAN DBC [3], each vehicle has unique data characteristics, even within the same manufacturer. Therefore, these IDS may not detect attacks when the vehicle is changed. Therefore, there is a need to develop a solution that creates an in-vehicle network intrusion detection system that can be deployed for each vehicle and takes into account the unique data characteristics of the vehicle. Specifically, the contributions of this research are as follows:

First, this study proposes a method for generalizing data from different vehicles. In particular, an approach is defined for addressing the challenge of data diversity in invehicle networks, which is a major obstacle to the development of a universal Intrusion Detection System (IDS). The proposed approach eliminates the need to develop IDSs for each vehicle individually.

Second, the proposed algorithm is able to classify attack-free and attack-relevant data independently by using data from other vehicles. This capability allows the proposed algorithm to effectively detect attacks even when the number of available data samples for a given vehicle is limited.

Finally, the proposed approach can be extended to the Roadside Unit (RSU) and the On-Board Unit (OBU) in the Vehicle-to-Everything (V2X) communication ecosystem. Through this extension, the security of V2X communication can be improved by using the proposed approach for V2X security implementation.

In our previous research [4], we performed an in-depth analysis of the Controller Area Network and now we have proposed a heat map-based IDS solution using CNN that can detect attacks independently. In this paper, we implement three types of CNN architectures to find the best one that is suitable for all types of attacks with high accuracy. In summary, implementing security measures such as encryption, authentication, and intrusion detection is crucial to protect CAN bus networks from cyberattacks and ensure their reliability and security. The development of universal intrusion detection systems can further improve the security of these networks by taking into account the unique characteristics of each vehicle. As the use of IoT and IoV devices continues to grow, it is critical to take proactive measures to protect CAN bus networks from cyber threats to ensure the safety of both vehicles and industrial processes.

2 CAN Specification

The specification of CAN was published by manufacturers in an article [1]. In a CAN network, data is transmitted in the form of messages consisting of two main components: the message identifier (CAN ID) and the payload data. The CAN ID is a unique identifier that specifies the priority and content of the message. The payload contains the actual information that is transmitted and can be between 0 and 8 bytes in size. When a message is transmitted, it is encapsulated in a CAN frame, which consists of seven main fields: the Start of Frame (SOF), Arbitration Field, Control Field, Data Field, CRC Field, Acknowledgment Field, and End of Frame (EOF). The Arbitration Field contains the CAN ID and determines the priority of the message. Messages with lower CAN IDs have higher priority. The Control Field contains information about the data length and the type of message, such as whether it is a remote frame or a data frame. In a remote frame, the CAN ID is used to request data from other nodes on the network, while in a data frame, the payload contains the actual data that is being transmitted. The CRC field is used for error detection and is calculated based on the data payload. It ensures that the transmitted data is error free and can be correctly interpreted by the receiving node. The Acknowledgment Field is used to confirm the successful reception of a message by the receiving node. If the message was received without errors, the receiving node sends an acknowledgement message back to the sending node. Finally, the End of Frame (EOF) field signals the end of the message and allows the receiving node to prepare for the next message on the bus. In summary, the CAN protocol uses unique message identifiers (CAN IDs) to determine the priority and content of messages transmitted on the network. Each message is encapsulated in a CAN frame that contains several fields, including the arbitration field, the control field, the data field, the CRC field, the acknowledgement field, and the end-of-frame field. This enables reliable and efficient communication between nodes in the network.

3 Related Works on Automotive Security

Zhu et al. [5] proposed a multidimensional IDS using multi-task LSTM for parallel computation on local terminals and mobile devices. Features include the addition of 64-bit data and CAN IDs time interval. The local part predicts the next data combination, while the mobile edge has two parallel LSTMs to improve performance and achieve 90% overall. A federated learning approach [6], proposed by J. Yang et al. proposed federated learning approach [6] achieved 94.85% accuracy using ID sequence as input to ConvLSTM. Federated learning is a distributed machine learning approach that allows multiple clients to jointly train a model without sharing their data with a central server. The model is trained locally on each client, and only model updates are sent to the central server for aggregation. The architecture proposed by Narasimhan et al. detects tampering with the incoming CAN features of a vehicle using autoencoder and clustering methods. The IDEC approach [7] is used to learn optimal features and cluster data using K-means. A modified version of the IDEC algorithm is proposed where the clustering loss function is not embedded and is more suitable for IDS [8]. This modified version uses an autoencoder architecture for initial pre-training and GMM for clustering attacks and normal traffic based on various sensor inputs with an accuracy rate of 80.1%. The authors of [9] analyzed the input sequence CAN ID for attacks using a bidirectional GPT network by computing the negative log likelihood (NLL) value. Each ID in the sequence is converted to an integer and compared to a predetermined threshold to determine whether or not it is an attack. The bidirectional processing allows contextual information to be captured. The NLL value provides an estimate of the model's confidence in its classification. The approach leverages the GPT network's ability to learn patterns in the data and make accurate predictions. The overall accuracy is 97.8% with bidirectional GPT. The GIDS system detects attacks on the on-board network using a two-step process [10] that converts uniquely encoded CAN data into CAN images. The first discriminator outputs a value between 0 and 1, and if it is above the threshold, the corresponding images are fed to the second discriminator, which also outputs a value between 0 and 1. By combining both discriminators, the system detects known and unknown attacks with high accuracy. Al-Jarrah et al. [11] presented a multi-model approach that uses LSTM and ConvLSTM to classify attacks. LSTM was used to input table data, while ConvLSTM used a recursion graph. The combination of the two models increased the accuracy by 2%, resulting in an overall accuracy of 95.1%. In this section, accuracy is measured based on the source data set. In short, data was collected from a specific vehicle to train the model, and the same vehicle was used to test the model. Therefore, the model was not generalized and cannot be implemented in other vehicles unless additional data from other vehicles are added.

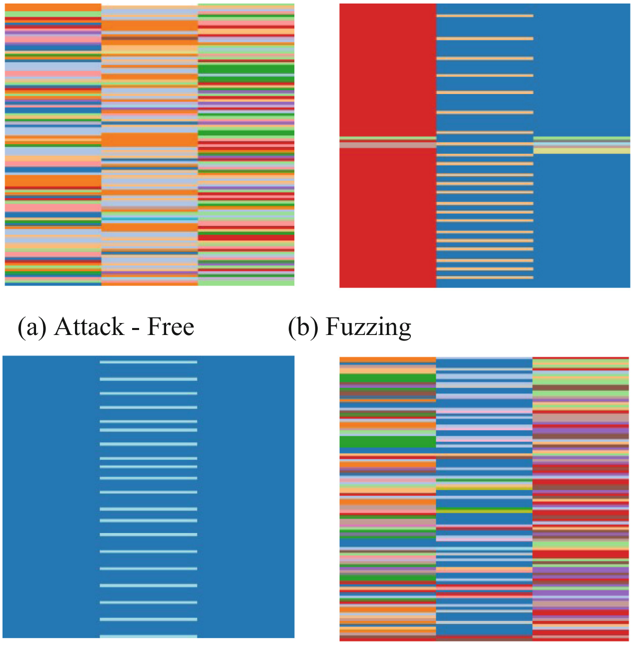
Our proposed model aims to train the model with data from different vehicles and test it on different vehicle models. In this way, the model can be used universally. Specifically, we trained the model with data from BMW and Kia and tested it with Tesla.

4 Data Pre-processing and Deep Learning Architectures

4.1 Data Pre-processing

In this study, we investigated different types of attacks, including Denial-of-Service (DoS) attacks, which can be categorized as low-speed attacks (L-DoS) or distributed attacks (D-DoS), and data injection attacks such as Fuzzing and replay attacks. To collect data for these attacks, our study employed a data collection method that used segments of 3–5 s. For DoS attacks, we injected 5000 and 1000 data points per segment, while for fuzzing attacks, we injected 100 and 500 data points per segment. For replay attacks, we injected two random data points. The results of our study shed light on the effectiveness of different intrusion detection models in accurately classifying these attacks and provide valuable insights for developing more robust security measures for in-vehicle networks.

When labeling a dataset, simplicity is crucial. In our approach, we split the dataset at the beginning and end times of an attack and compute the time interval and hamming distance between the resulting equal-length hexadecimal strings [8]. By treating each symbol as a 4-bit binary number and counting the number of distinct symbols at each position, we can determine the Hamming distance for this base-16 number system. This metric is suitable for error detection and correction, cryptography, and other data transmission and storage applications. We then segment the data into fixed-length segments, each containing 40 features, including CAN IDs, time interval, and Hamming distance for CAN ID. Using the scikit-learn library, we convert the hexadecimal CAN IDs into numeric values and scale them to a range between 0 and 1. The resulting segments are then converted into heatmaps that are input to Deep Learning models. Figure 1 shows the heatmap state for different states of the CAN data, where (a) represents attack-free data and (b, c, d) represent fuzzing, DoS, and replay, respectively. From this heatmap, it can be seen that attack-free and replay data have similarities, while fuzzing and DoS data have different characteristics.



(a) Attack - Free (b) Fuzzing

(c) DoS (d) Replay

Fig. 1. Heat-map decomposition according to ID sequence, time gap and hamming distance.

4.2 Deep Learning Architectures

VGG-16 has a simple architecture with 16 convolutional layers and three fully linked layers. It uses small 3 × 3 filters in all convolutional layers and has a fixed input size of 224 × 224. The VGG-16 architecture is known for its simplicity and accuracy in image classification tasks, but its small filters make it computationally intensive compared to other architectures [12]. AlexNet was the first CNN to win the ImageNet Challenge in 2012. It has five convolutional layers followed by three fully concatenated layers. It uses larger kernel sizes of 11 × 11 and 5 × 5 in the first convolutional layer and 3 × 3 in the other layers. AlexNet also uses overlapping pooling layers to reduce the feature map size. This architecture achieved a significant improvement in accuracy compared to previous models [13]. ResNet-50 is a much deeper CNN architecture with 50 layers that use residual connections. The residual connections allow the model to solve the vanishing gradient problem that can occur with very deep networks [14]. The residual connections also make it possible to train even deeper models with thousands of layers. ResNet-50 is computationally intensive thanks to the use of skip connections, which allow the model to learn identity mapping. This makes it possible to use a deeper architecture with fewer parameters than previous models, resulting in higher accuracy.

5 Result Evaluation

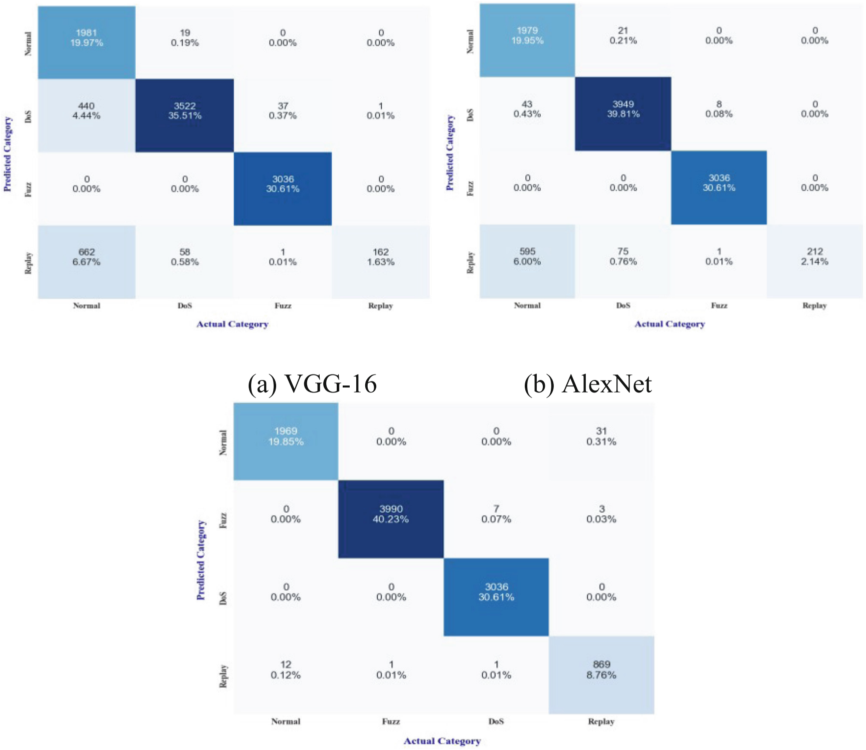
In this study, we implemented three well-known Convolutional Neural Network (CNN) models to determine their effectiveness in universally detecting intrusions into on-board networks. Specifically, we trained and tested these models with data from BMW and Kia vehicles, and then evaluated their performance on a Tesla vehicle with a different vehicle network architecture. Fuzzing and denial-of-service (DoS) attacks have unique characteristics that allow each model to accurately classify these types of attacks. However, the replay attack is more difficult to detect because the attacker uses the target's own dataset for anomalous injections. In particular, the ResNet-50 model has demonstrated its reliability in this regard by achieving an accuracy rate of nearly 99% in classifying attack-free, fuzzing, DoS, and replay attack scenarios. In contrast, the VGG-16 and AlexNet models achieved overall accuracy rates of 88% and 93%, respectively. Our results are summarized in the attached confusion matrix (Fig. 2), Table 1 and Table 2. Overall, the results highlight the superiority of the ResNet-50 model in accurately detecting replay attacks, indicating its potential for effective intrusion detection in on-board networks.

Table 1. Accuracy scores according to classifying category for different architecture models.

Types	VGG-16	AlexNet	ResNet-50
Normal	0.78	0.86	0.99
Fuzz	0.93	0.98	1.00
DoS	0.99	1.00	1.00
Replay	0.18	0.39	0.97

Table 2. Overall Accuracy score for different architecture models.

Types	VGG-16	AlexNet	ResNet-50
Accuracy	0.88	0.93	0.99
ROC	0.862	0.889	0.994



(a) VGG-16 (b) AlexNet

(c) ResNet-50

Fig. 2. Performance evaluation confusion matrix between different architecture models.

6 Conclusion

The development of a universal in-vehicle network intrusion detection system that works for all vehicle types, including mechanical, hybrid, and electronic vehicles, is the focus of this study. The main challenge in developing such a system is data generalization, which was addressed using heatmaps. The prototype of the proposed system achieved high accuracy and fast response times, indicating the potential for a reliable and effective solution to improve the security of vehicle network security. However, further experiments are needed to validate the performance of the system under different attack scenarios, including different levels of data injection. The study provides valuable insights for the development and implementation of robust invehicle network security measures. Overall, the proposed system has the potential to provide a more comprehensive and efficient solution for securing vehicle networks, contributing to the overall safety and security of the automotive industry.

Acknowledgments. This work was supported by the National Research Foundation of Korea (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. BOSCH CAN Specification Version 2.0 (1991)
2. Nardus, L., Miller, C., Valasek, C.: Remote Exploitation of an Unaltered Passenger Vehicle
3. Sunny, J., Sankaran, S., Saraswat, V.: A hybrid approach for fast anomaly detection in controller area networks. In: 2020 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS), pp. 1–6, December 2020. https://doi.org/10.1109/ANTS50601.2020.9342791
4. Islam, M.R., Oh, I., Yim, K.: CANTool an in-vehicle network data analyzer. In: 2022 International Conference on Information Technology Systems and Innovation (ICITSI), pp. 252–257, November 2022. https://doi.org/10.1109/ICITSI56531.2022.9970968
5. Zhu, K., Chen, Z., Peng, Y., Zhang, L.: Mobile edge assisted literal multi-dimensional anomaly detection of in-vehicle network using LSTM. IEEE Trans. Veh. Technol. **68**(5), 4275–4284 (2019). https://doi.org/10.1109/TVT.2019.2907269
6. Hussain, S., Ali Imran, M., Yang, J., Hu, J., Yu, T.: Federated AI-enabled in-vehicle network intrusion detection for internet of vehicles. Electronics **11**(22), 3658 (2022). https://doi.org/10.3390/ELECTRONICS11223658
7. Guo, X., Gao, L., Liu, X., Yin, J.: Improved deep embedded clustering with local structure preservation. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, pp. 1753–1759, August 2017. https://doi.org/10.24963/ijcai.2017/243
8. Narasimhan, H., Ravi, V., Mohammad, N.: Unsupervised deep learning approach for in-vehicle intrusion detection system. IEEE Consum. Electron. Mag. **12**(1), 103–108 (2023). https://doi.org/10.1109/MCE.2021.3116923
9. Nam, M., Park, S., Kim, D.S.: intrusion detection method using bi-directional GPT for in-vehicle controller area networks. IEEE Access **9**, 124931–124944 (2021). https://doi.org/10.1109/ACCESS.2021.3110524
10. Seo, E., Song, H.M., Kim, H.K.: GIDS: GAN based intrusion detection system for in-vehicle network (2018). https://doi.org/10.1109/PST.2018.8514157
11. Al-Jarrah, O.Y., El Haloui, K., Dianati, M., Maple, C.: A novel detection approach of unknown cyber-attacks for intra-vehicle networks using recurrence plots and neural networks. IEEE Open J. Veh. Technol. **4**, 271–280 (2023). https://doi.org/10.1109/OJVT.2023.3237802
12. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: 3rd International Conference on Learning Representations ICLR 2015 - Conference Track Proceedings, September 2014. http://arxiv.org/abs/1409.1556
13. Iandola, F.N., Han, S., Moskewicz, M.W., Ashraf, K., Dally, W.J., Keutzer, K.: SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 MB model size, pp. 1–13, February 2016. http://arxiv.org/abs/1602.07360
14. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778, June 2016. https://doi.org/10.1109/CVPR.2016.90