CNN-Based Intrusion Detection System for In-Vehicle Network Using Explainable AI

Semin Kim*, Jonggwon Kim†, Hyungchul Im†, and Seongsoo Lee*†
*School of Electronics Engineering
†Department of Intelligent Semiconductors
Soongsil University
Seoul, Republic of Korea
semin2371@soongsil.ac.kr, jonggwon@soongsil.ac.kr, tory@soongsil.ac.kr, *sslee@ssu.ac.kr

***Abstract*—Recently, various intrusion detection systems (IDSs) have been proposed to address the security vulnerabilities of the controller area network (CAN), which is widely used in modern in-vehicle networks. However, existing IDSs determine only whether an attack occurs at the input sequence level. Consequently, these models cannot identify which frames within a sequence correspond to the attack. To solve this problem, we propose the Frame-eXplainable IDS (FX-IDS). The proposed FX-IDS was designed by integrating a convolutional neural network (CNN)-based architecture with integrated gradients (IG), an explainable AI (XAI) technique. It computes the contribution of each CAN frame within the input sequence and precisely identifies the frames in which attacks occur by summing the contributions at the frame level. Experimental results show that FX-IDS achieved an average accuracy of 99.96%, a precision of 99.97%, and a recall of 99.92% in sequence-level detection. It also achieved 98.48% accuracy in identifying the locations of attack frames within input sequences. In addition, XAI-based visualization demonstrated that the decision rationale of the model can be intuitively interpreted, thereby validating both the reliability and explainability of the in-vehicle IDS.**

***Index Terms*—In-vehicle security, CAN, intrusion detection system, explainable AI.**

I. Introduction

With the rise of vehicle-to-everything (V2X) technologies, such as vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I), vehicles have become connected systems that continuously exchange data with their environment [1]. Moreover, the increasing incorporation of functionalities such as autonomous driving, advanced driver-assistance systems (ADAS), and infotainment has led to a growing number of electronic control units (ECUs) responsible for managing these systems. For the efficient communication among these ECUs, the controller area network (CAN) protocol, is widely used. However, since the CAN protocol lacks built-in security mechanisms, it is vulnerable to external attacks. For example, an attacker can access the CAN bus physically through the on-board diagnostics port (OBD-II) , or remotely via various wireless interfaces such as Wi-Fi, cellular networks, and bluetooth to carry out attacks [2]. This can lead to malfunctions in safety-critical functions such as braking and engine control.

Vehicle security is of utmost importance, as it goes beyond mere information leakage and is directly related to human safety [3]. Consequently, researchers have explored diverse countermeasures to mitigate CAN security vulnerabilities; in particular, numerous machine learning–based intrusion detection systems (IDSs) have been proposed [4]–[6]. For example, Song et al. proposed a deep convolutional neural network (DCNN) IDS [7]. The DCNN performed attack detection using 29 consecutive CAN frames. Similarly, Desta et al. proposed the Rec-CNN model, which uses recurrent plots of 128 CAN frames as input with a size of 128×128 [8]. Seo et al. proposed a generative adversarial network (GAN)–based IDS (GIDS) trained with unsupervised learning, which uses 64 consecutive CAN frames encoded as one-hot vectors as the model input [9]. Hoang et al. proposed a convolutional adversarial autoencoder (CAAE)–based IDS that employs the adversarial autoencoder (AAE), a technique that regularizes a vanilla autoencoder through adversarial training [10].

The aforementioned models classify an input sequence as an attack if it contains at least one attack frame. Consequently, when an attack is detected, they cannot identify which frames within the sequence are malicious. This results in an inefficient response, as normal frames within the same sequence are also flagged as attacks. In short, sequence-level decision models inflate frame-level false positives. In addition, since existing models cannot explain the basis for judgment, it is difficult to secure reliability in the actual vehicle security operation. To address these limitations, we propose a frame eXplainable IDS (FX-IDS) that integrates explainable AI (XAI) techniques into the proposed IDS model. The proposed FX-IDS applies integrated gradients (IG), an XAI method, to quantitatively compute the contribution of each CAN frame. This enables identification of attack frames within sequences and visualization of the contribution of each frame.

The major contributions of our study are as follows:

- To the best of our knowledge, this is the first work that applies XAI to pinpoint the specific frame that is an attack within the input sequence of in-vehicle network data.
- The proposed FX-IDS was evaluated on four attack scenarios DoS, Fuzzy, Gear spoofing, and RPM spoofing using an open dataset. Experimental results demonstrate that the model achieved detection performance within input sequences, with an average detection accuracy of 99.96%, a precision of 99.97%, and a recall of 99.92%.
- Additionally, the proposed model provides a visual inter-

Fig. 1. Structure of a standard CAN.

pretation of its decision basis through XAI-based contribution analysis. In particular, it enhances the reliability and explainability of frame-level attack detection.

The remainder of this paper is organized as follows. Section II provides the background for this study. Section III provides a detailed explanation of the data preprocessing method and the proposed FX-IDS model. In Section IV, we describe the experimental setup and results, followed by a discussion. Finally, Section V concludes this paper.

II. Background

A. Controller Area Network (CAN)

CAN is a serial communication protocol developed by Bosch for seamless data transmission and reception among ECUs within a vehicle [11]. This protocol is used for engine control, airbag systems, and vehicle diagnostics. Fig. 1 illustrates the standard CAN frame. CAN adopts a multi-master architecture, in which all nodes can act as masters and initiate message transmission whenever necessary. This can lead to collisions when multiple CAN nodes attempt to transmit simultaneously. To prevent this, CAN resolves bus access using an arbitration mechanism that prioritizes messages based on their identifiers. For example, messages with lower identifier values have higher priorities and gain transmission rights over those with higher identifiers.

The data field contains the actual information that the transmitting node intends to deliver to other nodes, and its length varies from 0 to 64 bits depending on the data length code (DLC). It includes data necessary for vehicle operation, such as device control commands and sensor measurements. Finally, the cyclic redundancy check (CRC) field is appended for transmission error detection. The receiving node recalculates the CRC of the frame using the same rule and checks whether it matches the CRC transmitted by the sender.

B. Integrated Gradients (IG)

Gradient-based XAI estimates the contribution of each input to the model output using gradient information. There are two fundamental axioms that underlie the attribution methods of gradient-based XAI. First, sensitivity states that if the output changes relative to a baseline, the gradient value of the corresponding input feature should not be zero. In other words, any input feature that the model actually responds to must be reflected as a non-zero contribution. Second, implementation invariance is the principle that if two networks produce the same output for the same input, the attributions for each input feature should be identical, regardless of differences in their internal structures. If either of the two axioms is not satisfied, the attribution may become sensitive to features that are not important to the model output. As a result, the reliability and consistency of XAI are diminished. Representative gradient-based XAI methods include IG, DeepLIFT, and Layer-wise Relevance Propagation (LRP).

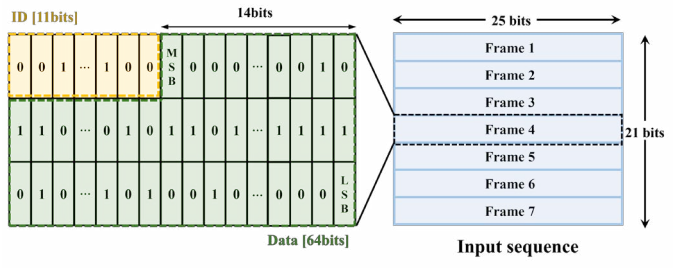

Fig. 2. Data preprocessing.

Among these, IG is an XAI method that mathematically quantifies how much each input feature of a deep learning model contributes to its prediction results, while satisfying the sensitivity and implementation invariance axioms [12]. IG calculates the cumulative contribution of each input feature to the model output $f(x)$ by integrating the gradients along the path from the baseline x' to the actual input x. It is defined as follows:

$$IG_i(x) = (x_i - x_i') \times \int_{\alpha=0}^{1} \frac{\partial f(x' + \alpha(x - x'))}{\partial x_i} \, d\alpha \quad (1)$$

where x_i denotes the input feature of the i-th dimension, and $\frac{\partial f}{\partial x_i}$ denotes the gradient of $f(x)$ with respect to the i-th dimension, indicating how sensitively the model output changes with respect to the input. That is, the influence of each feature is calculated by accumulating the changes in the output as the input gradually transitions from the baseline to the actual input. The integral of IG can be efficiently approximated through summation.

$$IG_i(x) \approx (x_i - x_i') \times \frac{1}{m} \sum_{k=1}^{m} \frac{\partial f\left(x' + \frac{k}{m}(x - x')\right)}{\partial x_i} \quad (2)$$

where m denotes the number of Riemann partition steps used to approximate the integral, which is typically set between 20 and 300. The Riemann approximation divides the path from the baseline to the input into m intervals and averages the gradients at each interval, allowing efficient implementation in most deep learning frameworks.

III. The Proposed Frame eXplainable IDS

A. Data Preprocessing

The proposed FX-IDS uses the ID and data fields of CAN frames as input features. Accordingly, an ID of 11 bits and a data field of 64 bits are combined to represent a single frame as a 75-bit vector. Each frame is converted into a 3×25 format. The first row is assigned an ID of 11 bits and data of 14 bits, while the second and third rows are each assigned 25 bits of data to form the entire frame including the ID and data fields. Subsequently, seven consecutive frames are combined

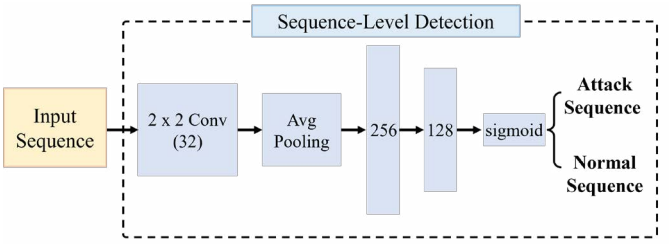

Fig. 3. Architecture of the proposed FX-IDS.

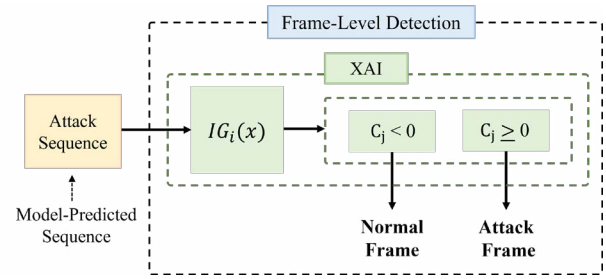

Fig. 4. Frame-level detection process of the FX-IDS.

to generate a final two-dimensional input sequence with a size of 21×25. This preprocessing procedure is illustrated in Fig. 2. All bits are arranged in an MSB-first order, and zero-padding is applied when the data field is less than 64 bits. Finally, an input sequence is labeled as an attack if it contains one or more attack frames.

B. Model Architecture

The proposed FX-IDS constructs the input sequence in a two-dimensional representation in order to distinguish clearly between normal and attack frames. Accordingly, as shown in Fig. 3, the proposed model employs a single convolutional layer to learn spatial correlations across frames, followed by two dense layers and a final output layer. The convolutional layer uses 32 kernels of size 2×2 to learn local spatial patterns among adjacent bits. The stride is set to 1 and the padding to 0 to minimize information loss, and the ReLU activation function is applied after the convolution operation to introduce nonlinearity. In addition, a 2×2 average pooling operation is used to reduce the size of the feature map by half and to average the values of adjacent regions. This allows the model to more stably reflect the differences in patterns between frames within the input sequence. The output, comprising 32 channels with a 10×12 spatial size, is subsequently flattened into a one-dimensional vector and fed into the dense layer. The dense layers and the final output layer are configured sequentially with 256, 128, and 1 nodes, respectively. Each layer calculates its output by applying weights and biases to the input values, followed by an activation function. The ReLU activation function is applied to each intermediate layer, while the sigmoid function is used in the final layer to output a value between 0 and 1. The output is classified as an attack if it is greater than or equal to 0.5, and as normal if it is less than 0.5. This output corresponds to sequence-level detection; if a sequence is classified as an attack, all frames within that sequence are regarded as attack frames.

TABLE I
OVERVIEW OF CAR HACKING DATASET

Attack type	Normal messages	Injected messages
DoS Attack	3,078,250	587,521
Fuzzy Attack	3,347,013	491,847
Gear Spoofing Attack	3,845,890	597,252
RPM Spoofing Attack	3,966,805	654,897

C. Frame-level Detection Process

Existing sequence-level detection IDS models cannot perform frame-level detection. However, the proposed FX-IDS can determine, for each frame within sequences, whether it is normal or an attack. It does so by performing IG-based contribution calculations on sequences classified as attacks. The proposed model applies (2) to sequences classified as attacks, and then calculates bit-wise contributions of the input sequence for frame-level detection. At this point, baseline x' is all set to 0. This is to clearly measure the effect of the distribution change of 1 on the model output in the attack frame, since the input is in a binary form consisting of 0 and 1.

The calculated bit-wise contributions are summed for each frame, considering that frame-level analysis is more intuitive than interpretation at a single-bit level. Through this process, the contributions are converted into frame-level contributions as follows.

$$C_j = \sum_{i=1}^{525} IG_i(x), \quad j = 1, 2, \ldots, 7 \tag{3}$$

where C_j represents the total contribution of the j-th frame. If the contribution C_j of the frame is greater than or equal to 0, the frame is determined as an attack frame. Conversely, if C_j is less than 0, it is determined as a normal frame.

IV. EXPERIMENTS AND RESULTS

A. Experimental Settings

The proposed FX-IDS is trained and evaluated under the following experimental environment.

- **OS**: Windows 11
- **CPU**: Intel(R) Core(TM) i7-12700K @ 3.60GHz
- **GPU**: NVIDIA GeForce RTX 3050
- **RAM**: 32.0GB
- **Framework**: PyTorch 2.4.1

In addition, the hyperparameters were carefully tuned to ensure that the FX-IDS model achieved optimal performance during the experiments. The batch size was set to 64, and training was conducted for a total of 10 epochs. The Adam optimizer was used with a learning rate of 0.001. For the IG-based contribution calculation, the Captum library was applied, and the number of Riemann partition steps, m, was set to 50.

TABLE II
SEQUENCE-LEVEL DETECTION PERFORMANCE OF THE FX-IDS

	Accuracy	Precision	Recall	F1 Score
DoS	99.93%	99.98%	99.78%	99.88%
Fuzzy	99.95%	99.94%	99.89%	99.91%
Gear	99.98%	99.96%	100%	99.98%
RPM	99.99%	99.99%	100%	99.99%

B. Dataset

In this study, we used the Car-hacking dataset provided by the hacking and countermeasure research lab (HCRL) [13]. The dataset was constructed by injecting attack messages into a real vehicle via the OBD-II port, and it comprises denial-of-service (DoS) attacks, fuzzy attacks, engine RPM spoofing attacks, and gear spoofing attacks. Table I shows the number of normal and injected attack frames for each attack type. A DoS attack exploits the arbitration mechanism of the CAN bus by continuously injecting high-priority messages, thereby occupying the bus and blocking the transmission of normal messages. A fuzzy attack injects random messages to cause ECU malfunctions, while a spoofing attack injects specific messages (e.g., engine RPM or gear) to manipulate the vehicle state.

C. Evaluation Metrics

In this study, since the dataset was imbalanced between normal and attack data, the model classification performance was evaluated using accuracy, precision, recall, and F1-score. In addition, the same metrics were applied to evaluate the frame-level classification performance, analyzing how accurately and consistently the model identifies attack frames within an input sequence. The calculation formulas for each evaluation metric are as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

$$F1\text{-}score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{7}$$

where true positive (TP) denotes the case in which the model correctly predicts actual attack data as an attack, and true negative (TN) corresponds to the case in which actual normal data are correctly predicted as normal. In contrast, false positive (FP) refers to the case in which actual normal data are incorrectly predicted as an attack, and false negative (FN) indicates the case in which actual attack data are incorrectly predicted as normal.

TABLE III
COMPARISON OF EXISITING IDS MODELS

Models	Detection Units	Precision	Recall	F1-Score
DCNN [6]	29 CAN frames	99.98%	99.84%	99.91%
Rec-CNN [7]	128 CAN frames	100%	99.91%	99.96%
CanNet [14]	16 CAN frames	99.91%	99.72%	99.69%
GIDS [8]	64 CAN frames	97.63%	99.44%	98.53%
FX-IDS (Ours)	**7 CAN frames ✓**	**99.97%**	**99.92%**	**99.94%**

✓ Each CAN frame can be identified as attack or normal.

TABLE IV
FRAME-LEVEL AVERAGE DETECTION ACCURACY ACROSS ALL ATTACK TYPES

	Accuracy	Precision	Recall	F1 Score
Average	98.48%	99.79%	96.52%	98.02%

D. Experimental Results

Table II shows the sequence-level detection performance of the proposed FX-IDS for each attack type. The proposed IDS achieved an average accuracy of 99.96% across all attack types, demonstrating outstanding detection performance. Furthermore, we compared the proposed model with existing IDS models as shown in Table III. Although the proposed model has the smallest sequence size, it achieved performance that was comparable to or better than other models. However, a fair comparison remains challenging because sequence lengths differ across models. To address this, a normalized frame-level evaluation is required, but existing models lack an architecture to distinguish individual frames within a sequence. For example, although Rec-CNN achieved the highest F1-score, it uses the largest sequence size of 128 frames as input. If any one of the 128 frames is malicious, the remaining 127 are likewise classified as attacks, which can induce significant operational errors. In contrast, the proposed FX-IDS is able to accurately identify attack frames within sequences by using XAI.

Table IV presents the frame-level detection performance of the proposed FX-IDS. The frame-level evaluation assesses whether the proposed model can accurately identify actual attack frames within sequences that are classified as attacks. This is conducted through IG-based contribution analysis. Based on the number of frames in each sequence, the contribution value C_j of each frame was calculated to determine the counts of TP, TN, FP, and FN. The experimental results confirmed that the proposed model can effectively distinguish between normal and attack frames within sequences. The performance is slightly lower than that of sequence-level detection. However, it is evaluated as an excellent result, considering that the proposed approach performs detection at a fine-grained level by independently determining each frame within sequences.

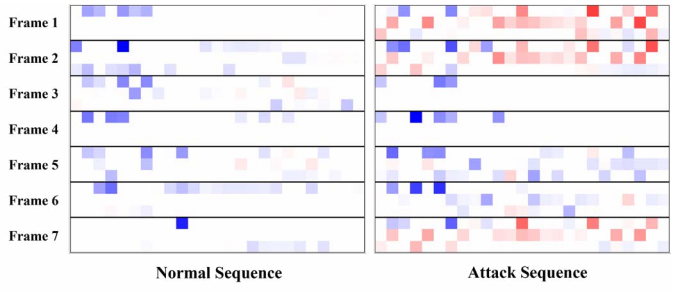

Fig. 5. Bit-wise contribution visualization.

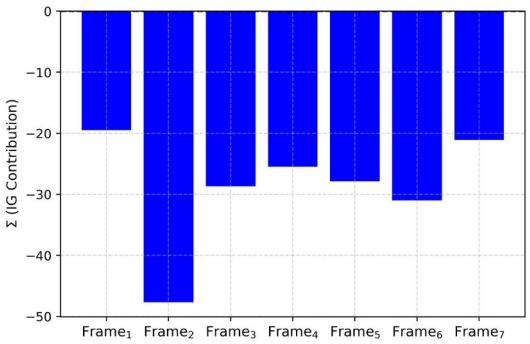

Fig. 6. Frame-level contribution visualization of a normal sequence.

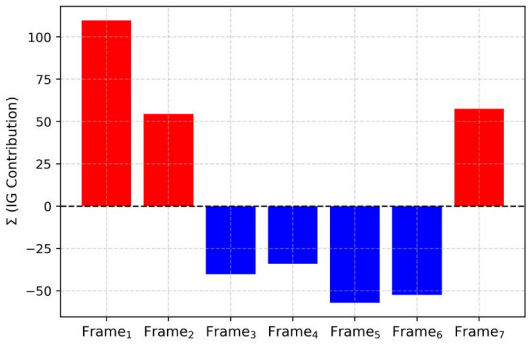

Fig. 7. Frame-level contribution visualization of a attack sequence.

E. XAI Visualization

As shown in Fig. 5, we visualized the bit-wise contribution of each sequence to explain the basis on which the proposed FX-IDS determines normal and attack sequences. Each bit has a contribution value, where red indicates a positive contribution and blue indicates a negative contribution. These represent the factors that contribute to the decision of the model toward an attack and normal classification, respectively. Areas where the color is deep indicate that the corresponding bits have a greater contribution to the decision of the model. In contrast, areas with pale or almost no color correspond to neutral features that have little to no influence on the prediction of the model. In the case of normal sequences, negative contributions (blue) are predominantly distributed across most bits, whereas in attack sequences, positive contributions (red) are concentrated in specific frame regions.

Fig. 6 and Fig. 7 show the results of the proposed FX-IDS visualizing frame-level contributions by summing the bit-wise contributions. These visualizations correspond to determining normal and attack sequences, respectively. In Fig. 6, negative contributions appear across all frames, indicating that the model consistently recognizes each frame as normal. In contrast, Fig. 7 shows prominent positive contributions in specific frames within the attack sequence. This suggests that these frames significantly influence the decision of the model to classify the input sequence as an attack. In fact, the sequence in Fig. 7 is composed of attack frames at the 1st, 2nd, and 7th positions, which coincide with the regions showing positive contributions. Therefore, the use of XAI enables a clear explanation of the rationale behind the model decision. This confirms that the proposed model can precisely identify attacks at the frame level within input sequences.

F. Discussion and Limitation

In this study, we proposed FX-IDS, which integrates XAI techniques to overcome the limitations of exising sequence-level detection-based IDS models. The FX-IDS model demonstrated excellent detection performance and was able to accurately identify the positions of attack frames. Furthermore, it not only provided detection results but also clearly explained the reasoning behind the model predictions through visualized representations. However, the proposed FX-IDS has several limitations. First, the IG-based contribution calculation requires high computational cost and memory resources. This makes it suitable for offline analysis environments but difficult to apply in real-time in-vehicle systems. Accordingly, future research should focus on improving the model structure by applying lightweight techniques to enable real-time operation. Second, since FX-IDS is a supervised learning-based model, it is limited in its ability to detect new types of attacks that have not been trained. Therefore, it should aim to expand the model into an unsupervised learning-based framework capable of detecting previously unknown attacks.

V. Conclusion

In this paper, we propose FX-IDS, which enables frame-level attack detection within input sequences using XAI. The proposed FX-IDS is structured by combining a single CNN-based architecture with IG. It is designed to make the model prediction rationale interpretable at both the sequence and frame levels. Experimental results show that FX-IDS achieved excellent performance compared to existing models in sequence-level detection. It was also able to accurately distinguish between normal and attack frames in frame-level detection. This suggests that the proposed approach can contribute to future IDS research by simultaneously enhancing the accuracy and reliability of CAN-based IDSs. In future work, we plan to apply lightweight approximation techniques that can replace IG computation in resource-constrained in-vehicle environments. Furthermore, we aim to develop an IDS capable

of detecting unknown attacks by integrating unsupervised learning with XAI.

ACKNOWLEDGMENT

This work was supported in part by the Korea Institute for Advancement of Technology (KIAT) grant, in part by the Korea Evaluation Institute of Industrial Technology (KEIT) funded by the Korea Government (MOTIE), in part by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant, and in part by the National Research Foundation of Korea (NRF) funded by the Korea Government (MSIT) (P0017011, RS-2022-00155731, RS-2023-00232192, RS-2025-02214672, 2021M3H2A1038042).

REFERENCES

[1] Z. Tan, N. Dai, Y. Su, R. Zhang, Y. Li, D. Wu, and S. Li, "Human–machine interaction in intelligent and connected vehicles: A review of status quo, issues, and opportunities," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 13954–13975, Sep. 2022.

[2] E. Aliwa, C. Perera, and O. Rana, "Cyberattacks and countermeasures for in-vehicle networks," *ACM Comput. Surv.*, vol. 54, no. 1, pp. 31–37, 2020.

[3] K. Koscher *et al.*, "Experimental security analysis of a modern automobile," in *Proc. IEEE Symp. Security and Privacy (SP)*, Oakland, CA, USA, 2010, pp. 447–462.

[4] Z. Deng, J. Liu, Y. Xun, and J. Qin, "IdentifierIDS: A practical voltage-Based intrusion detection system for real in-vehicle networks," *IEEE Trans. Inf. Forensics Security*, vol. 19, pp. 661–676, Oct. 2023.

[5] S. Jeong, S. Lee, H. Lee, and H. K. Kim, "X-CANIDS: Signal-aware explainable intrusion detection system for controller area network-based in-vehicle network," *IEEE Trans. Veh. Technol.*, vol. 73, no. 3, pp. 3230–3246, Mar. 2024.

[6] G. A. Al-Absi, Y. Fang, A. A. Qaseem, and H. Al-Absi, "DST-IDS: Dynamic spatial–temporal graph-transformer network for in-vehicle network intrusion detection system," *Veh. Commun.*, vol. 55, p. 100962, 2025.

[7] H. M. Song, J. Woo, and H. K. Kim, "In-vehicle network intrusion detection using deep convolutional neural network," *Veh. Commun.*, vol. 21, pp. 1-13, Jan. 2020.

[8] A. K. Desta, S. Ohira, I. Arai, and K. Fujikawa, "Rec-CNN: In-vehicle networks intrusion detection using convolutional neural networks trained on recurrence plots," *Veh. Commun.*, vol. 35, Jun. 2022, Art. no. 100470.

[9] E. Seo, H. M. Song, and H. K. Kim, "GIDS: GAN-based intrusion detection system for in-vehicle network," in *Proc. 16th Annu. Conf. Privacy, Secur. and Trust (PST)*, Aug. 2018, pp. 1-6.

[10] T.-N. Hoang and D. Kim, "Detecting in-vehicle intrusion via semi-supervised learning-based convolutional adversarial autoencoders," *Veh. Commun.*, vol. 38, Dec. 2022, Art. no. 100520.

[11] R. B. GmbH, "CAN Specification, Version 2.0," 1991.

[12] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *Proc. 34th Int. Conf. Mach. Learn. (ICML)*, Sydney, Australia, 2017, pp. 3319–3328.

[13] Hacking and countermeasure research lab. car-hacking Dataset. [Online]. Available: https://ocslab.hksecurity.net/Datasets/car-hacking-dataset

[14] Gao, S. Zhang, L. He, L. Deng, X. Yin, H. Zhang, H. "Attack detection for intelligent vehicles via CAN-Bus: A lightweight image network approach," *IEEE Trans. Veh. Technol.*, vol. 72, no. 12, pp. 16624–16636, Dec. 2023.

[15] D. Gunning and D. W. Aha, "DARPA's explainable artificial intelligence (XAI) program," *AI Magazine*, vol. 40, no. 2, pp. 44-58, 2019.

[16] H. Im and S. Lee, "TinyML-based intrusion detection system for in-vehicle network using convolutional neural network on embedded devices," *IEEE Embedded Syst. Lett.* , early access, Oct. 2024.