Multi-Layer Perceptron-Based Intrusion Detection System for UAVCAN Networks

Seoyeon Kim*, Hyungchul Im†, Youngmin Jang†, and Seongsoo Lee*†
*School of Electronics Engineering
†Department of Intelligent Semiconductors
Soongsil University
Seoul, Republic of Korea
seoyeon0421@soongsil.ac.kr, tory@soongsil.ac.kr, ymjang21c@soongsil.ac.kr, *sslee@ssu.ac.kr

***Abstract*—In this work, we propose an intrusion detection system (IDS) designed to effectively detect various attacks that can occur in uncomplicated application-level vehicular computing and networking (UAVCAN) protocol. UAVCAN is an upper-layer communication protocol built on the controller area network (CAN) physical layer, allowing electronic control units (ECUs) within UAVs to exchange messages. Recently, various IDSs have been proposed for the CAN protocol because of its vulnerability to security threats. However, these existing IDSs were designed for in-vehicle CAN networks. As a result, they fail to fully reflect the message structure and communication characteristics of UAVCAN, making it difficult to apply them directly to UAVCAN environments. Therefore, we propose a UAV-specific IDS optimized for UAVCAN using a multilayer perceptron (MLP). The proposed model is designed to effectively reflect the characteristics of UAVCAN. We evaluate the proposed model using a dataset collected from a real drone equipped with a UAVCAN bus. Moreover, the same dataset is applied to existing in-vehicle CAN IDS models for performance comparison. The experimental results confirm that existing CAN IDS models fail to effectively detect attacks in the UAVCAN environment, whereas the proposed model achieves a high average accuracy of approximately 98.69%.**

***Index Terms*—UAVCAN, CAN, drone, intrusion detection system, multilayer perceptron**

I. Introduction

Recently, unmanned aerial vehicles (UAVs) have been widely used in various civil and commercial fields beyond military applications [1]. The electronic control unit (ECU), a key component of these UAVs, serves to control and manage various electrical subsystems. Multiple ECUs are interconnected via the controller area network (CAN) protocol, which provides low-latency communication for efficient system operation [2]. CAN is a representative in-vehicle network standard that has been widely adopted for its high transmission speed, reliability, and cost-effectiveness. It operates as a serial bus based on a broadcast communication mechanism. However, it lacks authentication and encryption capabilities, which makes it vulnerable to cyberattacks such as message injection.

These limitations have received considerable attention in the automotive industry, and numerous studies have been conducted to enhance CAN security [3]–[5]. In particular, various machine learning–based intrusion detection system (IDS) have been proposed [6]–[8]. Among them, DCNN is a representative supervised learning-based IDS utilizing a reduced Inception–ResNet architecture, which has demonstrated high detection performance [9]. An LSTM-based IDS model was introduced by Hossain et al., using CAN ID, DLC, and payload information as training data [10]. Similarly, HyDL-IDS combines CNN and LSTM architectures while adopting identical training data [11]. Seo et al. proposed a generative adversarial network (GAN)–based IDS, referred to as GIDS [12]. GIDS is a representative unsupervised-learning model for CAN security that is trained only on normal data and can detect previously unknown attacks.

Existing IDSs were designed for in-vehicle CAN. Consequently, these models are difficult to apply directly to uncomplicated application-level vehicular computing and networking (UAVCAN) systems and may exhibit degraded performance. UAVCAN constructs its data frames based on data structures defined by the data structure description language (DSDL), and the detailed frame composition is determined by the tail-byte and multi-frame handling mechanism. In short, UAVCAN extends the conventional in-vehicle CAN with additional mechanisms. Therefore, this study proposes a multilayer perceptron (MLP)-based IDS optimized for the UAVCAN environment.

The contributions of this work are described as follows:

- We propose a UAVCAN-specific IDS based on MLP (UM-IDS) that utilizes the CAN ID, DLC, and payload as bit-level inputs. Furthermore, the configurations of MLP layers are evaluated to determine the optimal number of hidden layers and design an efficient IDS architecture suitable for the UAVCAN environment.
- The proposed UM-IDS was evaluated using a publicly available dataset collected from an actual drone. Experimental results demonstrate that the proposed model achieved an average accuracy of 98.69%, a precision of 96.71%, and a recall of 99.70% for various attack types such as flooding, fuzzy, and replay.
- In addition, representative CAN-based IDS models were implemented and applied to the UAVCAN dataset to demonstrate the superiority of the proposed UM-IDS through performance comparison.

The rest of this work is organized as follows. Section II introduces the background for this study such as CAN and

979-8-3315-7896-1/26/$31.00 ©2026 IEEE

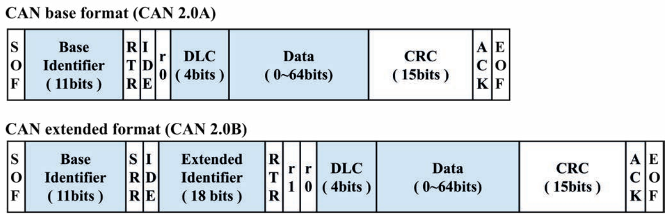

Fig. 1. Structure of a CAN frame.

UAVCAN protocols. Section III describes the data preprocessing procedure and the architecture of the proposed UM-IDS model. In Section IV, we present the experimental environment, experimental results, and a discussion of limitations. Finally, Section V concludes this work.

II. Background

A. Controller Area Network (CAN)

CAN is a representative in-vehicle communication protocol that enables the exchange of control signals and status information between multiple ECUs within a vehicle [13]. CAN-based networks are designed with real-time capability, reliability, and efficiency to support the core functions of the vehicle. Consequently, they have become a standard communication protocol in in-vehicle networks.

Fig. 1 shows the two formats of the CAN protocol: CAN 2.0A and CAN 2.0B. CAN 2.0A uses an 11-bit identifier (ID) to determine message priority. CAN 2.0B adopts an extended frame format with a 29-bit ID, providing a wider identifier space. During bus arbitration, a dominant bit (0) overrides a recessive bit (1), so the frame with the lowest ID wins.

B. Uncomplicated Application-level Vehicular Computing and Networking (UAVCAN)

UAVCAN is a lightweight communication protocol based on CAN 2.0B that provides stable and reliable communication for modern vehicles such as UAVs. This protocol adopts a distributed network architecture to prevent single point of failure. In addition, unlike conventional in-vehicle CAN systems, each node in the UAVCAN network has a unique identifier and exchanges data between nodes through two methods: message broadcasting and service invocation. The message formats used in these two transmission methods are defined by the DSDL. DSDL is a data schema that specifies a data type name and a unique identifier in each file. UAVCAN data types are predefined and embedded in the firmware of each node. Nodes encode and decode messages according to these definitions. Therefore, UAVCAN is an upper-layer protocol operating over the CAN bus that interprets and transmits messages according to the data types defined by DSDL.

In UAVCAN, message dissemination follows a publish–subscribe principle, enabling efficient exchange of data messages among network nodes. This process is classified into single-frame and multi-frame transmissions. Single-frame transmission is used when the entire data payload can be contained within a single CAN frame of up to 8 bytes. Multi-frame transmission is applied when the data size exceeds the transmission capacity of a single frame, in which case the payload is divided into multiple frames and sent sequentially. Meanwhile, service invocation operates as a peer-to-peer communication mechanism for node configuration and firmware updates. In this method, a server node receives a request from a client node and returns the processing result as a response.

C. MultiLayer Perceptron (MLP)

MLP is an artificial neural network architecture that includes one or more hidden layers between the input and output layers. It performs sequential nonlinear transformations by applying nonlinear functions between each layer. Through this process, the network can learn complex interactions among input features that cannot be represented by linear transformations alone. The output of the l hidden layer in a typical MLP can be expressed as follows:

$$h^{(l)} = \phi\big(W^{(l)}h^{(l-1)} + b^{(l)}\big), \qquad h^{(0)} = x \tag{1}$$

Here, $x \in \mathbb{R}^D$ denotes the input vector, while $W^{(l)}$ and $b^{(l)}$ represent the weight matrix and bias vector of the l-th layer, respectively. $\phi(\cdot)$ denotes the activation function (e.g., ReLU).

III. The proposed Uavcan Mlp-based IDS

A. Data Preprocessing

We use a preprocessing technique that converts UAVCAN frame data into a binary representation at the bit level. Each frame consists of an ID (29 bits), a DLC (4 bits), and a payload (64 bits), which are combined to form a total of 97 bits. The frame is then converted into a 1 × 97 binary image, where the bit sequence follows the order from the MSB to the LSB. This preprocessing step accurately preserves the CAN ID priority, DLC, and byte-level payload structure without any information loss. Consequently, the input dimension is fixed at 97, enabling direct utilization in the proposed UM-IDS. Finally, normal data are labeled as 0 and attack data as 1, allowing the model to distinguish between the two classes during training.

B. Model Design

As the proposed UM-IDS adopts an MLP-based architecture, its detection performance is highly influenced by the number of hidden layers. Therefore, this study aims to design a model that effectively distinguishes between normal and attack frames by optimizing the hidden-layer structure. Initially, the preprocessed 97-bit feature vector is fed into the input layer and subsequently propagated through the hidden layers. In the final output layer, a sigmoid function is applied to determine whether each frame is normal or an attack.

Fig. 2 illustrates the architecture of the proposed UM-IDS model. In this study, we compare six models, each with a different number of hidden layers (0, 1, 2, 3, 4, and 8), to evaluate their detection performance. The number of neurons in the hidden layers is configured from a maximum of 256 to a minimum of 2. A decay rate of 0.5 is applied so that

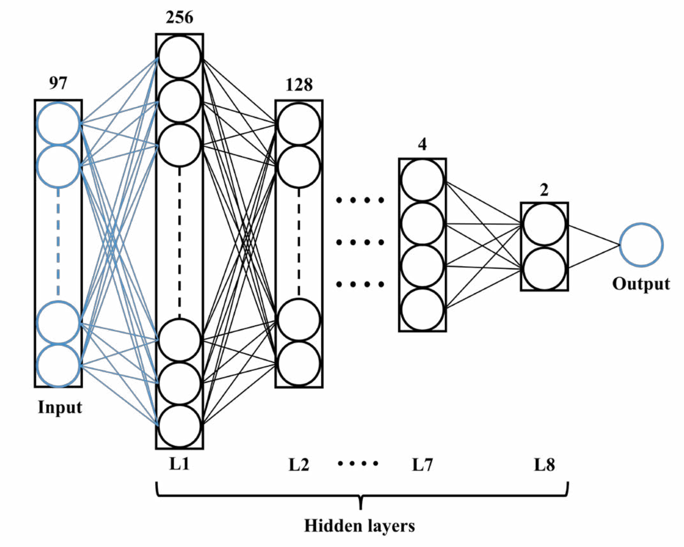

Fig. 2. Model architecture of UM-IDS

the number of neurons in each layer decreases by half as the number of hidden layers increases. For example, in the structure with eight hidden layers, the number of neurons gradually decreases from 256 in the first hidden layer to 128 in the second layer, and finally to 2 in the eighth layer. This gradual reduction structure enables hierarchical feature learning while efficiently controlling the model complexity. Furthermore, the ReLU activation function is employed for each hidden layer to allow the model to learn nonlinear relationships between inputs and outputs. At this stage, a dropout rate of 0.3 is set to prevent overfitting, thereby suppressing excessive dependence on specific inputs and improving the overall generalization performance.

IV. Experiments and Results

A. Experimental Settings

We trained and evaluated the UM-IDS model under the following experimental environment.

- **OS:** Windows 11
- **CPU:** Intel(R) Core(TM) i9-13900K @ 3.00GHz
- **GPU:** NVIDIA GeForce RTX 4090
- **RAM:** 128.0GB
- **Framework:** PyTorch 2.4.1

In addition, the hyperparameters were carefully tuned to ensure that the UM-IDS model achieved optimal performance during the experiments. The batch size was set to 32, and the training was conducted for a total of 50 epochs. The Adam optimizer was employed with a learning rate of 0.001. To prevent overfitting during training, an early stopping mechanism was applied.

B. Dataset

In this work, we used the UAVCAN dataset [14] provided by HCRL, as shown in Table 1. The dataset was collected by attaching a CAN shield and a Raspberry Pi 4 to a drone running the Pixhawk 4 (PX4) autopilot, and remotely injecting attacks from a PC over an SSH connection. Scenarios 1 and 2 correspond to flooding attacks, scenarios 3 and 4 to fuzzy attacks, and scenarios 5 and 6 to replay attacks. Scenarios 1–2 and 3–4 are distinguished by the attack-packet injection interval, whereas scenarios 5–6 are distinguished by the total data collection time. Each attack is defined as follows.

TABLE I
UAVCAN Attack Scenarios

Scenario	Attack Type	Interval (s)	Total Time (s)	Data Frame (N/A)
1	Flooding	0.0015	180	91,042 / 116,816
2	Flooding	0.0050	180	102,240 / 31,930
3	Fuzzy	0.0015	180	101,601 / 95,878
4	Fuzzy	0.0050	180	104,204 / 29,170
5	Replay	0.0050	210	129,996 / 50,612
6	Replay	0.0050	280	160,233 / 81,088

1) Flooding Attack
Flooding attack is a type of denial-of-service (DoS) attack that repeatedly transmits a large number of packets to exhaust the computational and communication resources of the target system. In UAV environments, this attack causes delays in normal communication between control and data exchange subsystems, making it impossible to maintain stable flight and mission execution.
2) Fuzzy Attack
Fuzzy attack is a type of attack that injects random values into the data field of message frames to induce abnormal behavior in ECUs. This attack can cause critical communication errors that disrupt flight control and may ultimately result in the drone becoming immobilized.
3) Replay Attack
Replay attack is a type of attack that deceives the receiver by copying and retransmitting legitimately transmitted data. Such attacks can cause the drone to repeatedly perform actions that may lead to potentially dangerous consequences.

C. Evaluation Metrics

The performance is evaluated based on four metrics—accuracy, precision, recall, and F1-score—using the proposed UM-IDS model. These metrics provide quantitative criteria for assessing the reliability and classification capability of the model. The metrics are computed from the confusion matrix using the counts of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). Each metric is calculated as follows:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{2}$$

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

TABLE II
PERFORMANCE OF UM-IDS ACROSS DIFFERENT HIDDEN LAYER NUMBERS

Number of Hidden layers	Attack Type	Accuracy	Precision	Recall	F1-score
No Hidden Layer	Flooding	0.9865 ± 0.0002	0.9699 ± 0.0005	1.0000 ± 0.0000	0.9847 ± 0.0003
	Fuzzy	0.9823 ± 0.0008	0.9765 ± 0.0015	0.9765 ± 0.0016	0.9765 ± 0.0011
	Replay	0.9340 ± 0.0016	0.8612 ± 0.0039	0.9399 ± 0.0011	0.8988 ± 0.0022
1 Hidden Layer (256)	Flooding	0.9865 ± 0.0002	0.9699 ± 0.0005	1.0000 ± 0.0000	0.9847 ± 0.0003
	Fuzzy	0.9985 ± 0.0001	0.9985 ± 0.0005	0.9975 ± 0.0004	0.9980 ± 0.0002
	Replay	0.9716 ± 0.0006	0.9295 ± 0.0016	0.9838 ± 0.0012	0.9558 ± 0.0010
2 Hidden Layer (256, 128)	Flooding	**0.9865 ± 0.0002**	**0.9699 ± 0.0005**	**1.0000 ± 0.0000**	**0.9847 ± 0.0003**
	Fuzzy	**0.9990 ± 0.0002**	**0.9990 ± 0.0004**	**0.9983 ± 0.0003**	**0.9986 ± 0.0002**
	Replay	**0.9753 ± 0.0004**	**0.9324 ± 0.0011**	**0.9928 ± 0.0004**	**0.9617 ± 0.0006**
3 Hidden Layer (256, 128, 64)	Flooding	0.9865 ± 0.0002	0.9699 ± 0.0005	1.0000 ± 0.0000	0.9847 ± 0.0003
	Fuzzy	0.9983 ± 0.0004	0.9983 ± 0.0012	0.9973 ± 0.0009	0.9978 ± 0.0006
	Replay	0.9723 ± 0.0009	0.9277 ± 0.0029	0.9883 ± 0.0013	0.9570 ± 0.0014
4 Hidden Layer (256, 128, 64, 32)	Flooding	0.9865 ± 0.0002	0.9699 ± 0.0005	1.0000 ± 0.0000	0.9847 ± 0.0003
	Fuzzy	0.9985 ± 0.0003	0.9986 ± 0.0006	0.9974 ± 0.0007	0.9980 ± 0.0004
	Replay	0.9718 ± 0.0012	0.9269 ± 0.0034	0.9876 ± 0.0014	0.9563 ± 0.0018
8 Hidden Layer (256, 128, ..., 4, 2)	Flooding	0.7497 ± 0.2410	0.5718 ± 0.4254	0.7000 ± 0.4583	0.6135 ± 0.4264
	Fuzzy	0.7996 ± 0.2003	0.3362 ± 0.4450	0.3980 ± 0.4874	0.3538 ± 0.4492
	Replay	0.5556 ± 0.2618	0.3403 ± 0.3203	0.6948 ± 0.4549	0.4272 ± 0.3315

$$F1-score=\frac{2\cdot Recall\cdot Precision}{Recall+Precision} \tag{5}$$

where TP denotes cases in which the model correctly classifies actual attack data as an attack, and TN refers to correctly identifying normal data as normal, while FP represents misclassifying normal data as an attack, and FN indicates misclassifying attack data as normal.

D. Experimental Results

For the experiments, each dataset was divided into training (70%), validation (15%), and testing (15%) sets. The performance evaluation was repeated using ten random seeds, with the mean performance and standard deviation calculated across all experiments. The standard deviation represents the degree of variation among experimental results, where a smaller value indicates that the model performs more consistently and stably. Therefore, we evaluated not only performance but also consistency and stability via standard-deviation analysis.

Table II presents the detection performance of the proposed UM-IDS according to the number of hidden layers for each attack type. As a result of the experiments, the model with two hidden layers achieved the highest performance, recording an average accuracy of 98.69% and an F1-score of 98.17%. In addition, the standard deviation values were less than 0.0012 for all performance metrics, indicating that this structure achieved the best balance between model complexity and generalization performance.

TABLE III
PERFORMANCE COMPARISON OF THE PROPOSED UM-IDS WITH EXISTING IDS MODELS

Attack Type	Models	Precision	Recall	F1-score
Flooding Attack	DCNN [9]	88.84	94.12	91.40
	LSTM-IDS [10]*	0	0	0
	HyDL-IDS [11]	97.08	100	98.52
	UM-IDS	**96.99**	**100**	**98.47**
Fuzzy Attack	DCNN [9]	86.12	94.10	89.93
	LSTM-IDS [10]*	0	0	0
	HyDL-IDS [11]	93.77	98.69	96.17
	UM-IDS	**99.90**	**99.83**	**99.86**
Replay Attack	DCNN [9]	85.34	94.82	89.83
	LSTM-IDS [10]*	0	0	0
	HyDL-IDS [11]	86.36	82.01	84.13
	UM-IDS	**93.24**	**99.28**	**96.17**

*Predicted all inputs as normal; ineffective as a UAVCAN IDS.

However, as the number of hidden layers increased to eight or more, the performance tends to degrade significantly. In particular, the model with eight layers showed remarkably lower performance for all attack types, and the standard deviation increased by at least 200 times and up to 400 times compared with other structures. This indicates that the variance of the training results was greatly enlarged, leading to a decrease in the stability and consistency of the model.

Table III presents a performance comparison between the proposed UM-IDS with the best-performing two-hidden-layer architecture and existing CAN-IDS models. The existing CAN IDS models showed relatively high detection performance under the flooding attack scenario of UAVCAN, but their performance decreased under the fuzzy and replay attack scenarios. In particular, the LSTM-IDS [10] failed to operate properly as an effective IDS in the UAVCAN environment. This result is mainly attributed to the fundamental differences in data structure and transmission mechanisms between CAN and UAVCAN. For example, in UAVCAN, a single message is segmented into multiple CAN frames for transmission. An attacker can exploit this mechanism to manipulate certain frames or alter the transmission order, thereby causing abnormal patterns within the same data type. Therefore, conventional CAN IDSs designed for single frame analysis have difficulty effectively detecting transmission pattern variations in UAVCAN. That is, conventional CAN IDS models have limitations because they fail to fully reflect the structural characteristics and transmission pattern diversity of UAVCAN, resulting in performance degradation. In contrast, the proposed UM-IDS is designed by considering these characteristics of UAVCAN and achieved superior detection performance in the UAVCAN environment.

E. Discussion and Limitation

In this work, we proposed UM-IDS, which explores the optimal network structure according to the number of hidden layers and achieves excellent attack detection performance in the UAVCAN environment. The experimental results demonstrated that excessively increasing the number of hidden layers can lead to performance degradation, while the model with two hidden layers achieved the highest detection performance. Furthermore, compared with existing in-vehicle CAN IDS models, the proposed UM-IDS achieved superior performance. This result experimentally demonstrates the necessity of a dedicated IDS specifically designed for UAVCAN environments. However, this study still has several limitations.

First, although the proposed model achieved higher detection accuracy than existing in-vehicle CAN IDS models, it has not yet achieved near-perfect performance. To ensure stable UAV operation in the presence of attacks, a model with higher detection performance is still required. Second, since the proposed UM-IDS can only perform binary classification, it is necessary to extend it to a multiclass model in the future to establish countermeasures for each attack type. Finally, since the proposed UM-IDS is based on supervised learning, it has a limitation in detecting unknown attacks. Therefore, additional techniques, such as threshold adjustment, should be applied to enable the detection of new attacks.

V. Conclusion

In this study, we propose a MLP-based IDS to enhance the security of UAVCAN. The proposed UM-IDS is specifically designed for the UAVCAN environment. It effectively reflects the multi-frame transmission characteristics and structural differences of UAVCAN that were not considered in conventional in-vehicle CAN IDS models. UM-IDS achieved detection accuracies of approximately 98.65%, 99.90%, and 97.53% under flooding, fuzzy, and replay attack scenarios, respectively. In contrast, existing in-vehicle CAN IDS models failed to achieve sufficient detection performance in the UAVCAN environment. These results demonstrate the necessity of a dedicated IDS specifically designed for UAVCAN. By comparing and analyzing the proposed UM-IDS with existing CAN-based IDS models, this study is expected to contribute to UAV security research based on the UAVCAN protocol. In future work, we will extend our model to an unsupervised learning-based IDS capable of detecting unknown attacks.

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] A. Sharma, P. Vanjani, N. Paliwal, C. M. W. Basnayaka, D. N. K. Jayakody, H.-C. Wang, and P. Muthuchidambaranathan, “Communication and networking technologies for UAVs: A survey,” *J. Netw. Comput. Appl.*, vol. 168, p. 102739, Oct. 2020.

[2] E. Aliwa, O. Rana, C. Perera, and P. Burnap, “Cyberattacks and countermeasures for in-vehicle networks,” *ACM Comput. Surv.*, vol. 54, no. 1, pp. 1–37, 2021.

[3] S. B. Park, H. J. Jo, and D. H. Lee, “G-IDCS: Graph-based intrusion detection and classification system for CAN protocol,” *IEEE Access*, vol. 11, pp. 39213–39227, 2023.

[4] S. Woo, H. J. Jo, and D. H. Lee, “A practical wireless attack on the connected car and security protocol for in-vehicle CAN,” *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 993–1006, Apr. 2015.

[5] H. M. Song, H. R. Kim, and H. K. Kim, “Intrusion detection system based on the analysis of time intervals of CAN messages for in-vehicle network,” in *Proc. Int. Conf. Inf. Netw. (ICOIN)*, Jan. 2016, pp. 63–68.

[6] 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, art. no. 100470, Mar. 2022.

[7] K. Agrawal, T. Alladi, A. Agrawal, V. Chamola, and A. Benslimane, “NovelADS: A novel anomaly detection system for intra-vehicular networks,” *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 11, pp. 22596–22606, Nov. 2022.

[8] S. Gao, L. Zhang, L. He, X. Deng, H. Yin, and H. Zhang, “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.

[9] 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.

[10] M. D. Hossain, H. Inoue, H. Ochiai, D. Fall, and Y. Kadobayashi, “LSTM-based intrusion detection system for in-vehicle CAN bus communications,” *IEEE Access*, vol. 8, pp. 185489–185502, 2020.

[11] W. Lo, H. Alqahtani, K. Thakur, A. Almadhor, S. Chander, and G. Kumar, “A hybrid deep learning based intrusion detection system using spatial–temporal representation of in-vehicle network traffic,” *Veh. Commun.*, vol. 35, Jun. 2022, Art. no. 100471.

[12] 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)*, Belfast, U.K., 2018, pp. 1–6.

[13] R. Bosch, “Can specification version 2.0,” *Rober Bousch GmbH, Postfach*, vol. 300240, p. 72, Sep. 1991.

[14] D. Kim, Y. Song, S. Kwon, H. Kim, J. D. Yoo, and H. K. Kim, “UAVCAN dataset description,” 2022, arXiv:2212.09268.