J Inno Sci Eng, 2025, 9(2):259-267

DOI:https://doi.org/ 10.38088/jise.1717959

Research Article

Lightweight CNN-Based Intrusion Detection for Automotive CAN Bus in Light Commercial Vehicles

Emre TÜFEKCİOĞLU [1,2*] **Cemal HANİLÇİ** [1] **Hakan GÜRKAN** [1]

[1] *Department of Electrical and Electronics Engineering, Bursa Technical University, 16310 Bursa, Türkiye*

[2]*KARSAN Otomotiv Sanayi ve Ticaret A.Ş, 16280 Nilüfer / BURSA, Türkiye*

ARTICLE INFO

Received Date: 11/06/2025
Accepted Date: 7/07/2025

Cite this paper as:
Tüfekcioğlu, E., Hanilci C. & Gürkan H. (2025). Lightweight CNN-Based Intrusion Detection for Automotive CAN Bus in Light Commercial Vehicles. *Journal of Innovative Science and Engineering.* 9(2), 259-267.

*Corresponding author: Emre Tüfekcioğlu
E-mail:emre.tufekcioglu@karsan.com.tr

Keywords:
CAN Bus
Intrusion Detection System,
Cybersecurity,
Vehicular Communication,
Deep Learning,
CNN,
Artificial Intelligence,
Anomaly Detection,
Light Commercial Vehicles

© Copyright 2025 by
Bursa Technical University. Available online at http://jise.btu.edu.tr/

The works published in Journal of Innovative Science and Engineering (JISE) are licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

ABSTRACT

With the rapid advancement of digitalization and automation, modern vehicles, especially in the light commercial segment, have evolved into complex, interconnected platforms resembling mobile computing systems. This transformation has increased the dependency on in-vehicle communication networks and, as a result, exposed them to a wider range of cybersecurity threats. A fundamental aspect of the proposed method is the use of a lightweight CNN model specific for deployment in embedded automotive environments with limited computational resources and optimized for efficiency. Operating on low-power hardware platforms such as edge ECUs, the tiny device developed in this study works effectively unlike conventional deep learning architectures seeking high processing power and memory. Despite its minimal computational footprint, the model is capable of accurately distinguishing between legitimate and spoofed communication traffic, as well as detecting a variety of attack forms that target different CAN protocol components. The performance metrics of the model further highlight its effectiveness, achieving a ROC AUC Score of 0.9887, an Accuracy of 0.9887, a Precision of 0.9825, a Recall of 0.9952, and an F1-Score of 0.9888. Particularly for real-time on-vehicle intrusion detection systems, this harmony between performance and efficiency makes the strategy especially important. Just as importantly is the introduction of a specifically produced hybrid dataset, which is fundamental for system evaluation and training. The dataset aggregates synthetic generated attack scenarios with real-world spoofing, injection, and denial-of- service (DoS) conditions using actual CAN traffic acquired from a J1939-compliant light commercial vehicle. Standard 11-bit identities combined with industrial communication protocols help the dataset to reflect real-world vehicle dynamics across several ECUs under various scenarios. The model can learn fine-grained patterns often missed by conventional rule-based or manually engineered approaches by means of the image-like transformation of CAN messages—preserving bit-level and temporal information. In intelligent transportation systems, the lightweight CNN architecture and the strong dataset combine to create a scalable and deployable IDS framework that can improve in-vehicle cybersecurity.

1. Introduction

Mobility is evolving rapidly in today's data-driven automotive landscape. Modern vehicles now function as networked computing platforms that integrate electric propulsion, partial autonomy and cloud-based services. These conveniences, however, enlarge the attack surface. Controller Area Network (CAN) is still the backbone of in-vehicle communication because of its real-time guarantees and low cost, yet it provides neither encryption nor message authentication. Consequently, adversaries can inject,

spoof or flood messages that putting vehicle in danger safety-critical control units. High-profile incidents such as the Jeep Cherokee hack and the recent CAN-injection thefts of Toyota models demonstrate the tangible risks. Against this backdrop, research interest has shifted toward lightweight, deep-learning-based intrusion-detection systems (IDS) that can run on embedded hardware. Recent studies explore depth-wise separable CNNs (DSC-KAN), transformer architectures enhanced with federated learning and unsupervised GAN-based models. Building on these advances, we propose an IDS that transforms raw CAN frames into 8 × 13 binary images and classifies them with a compact CNN, achieving high accuracy (98.9 %) while fitting the resource constraints of light commercial vehicles.

The development of a lightweight, embedded-suitable Intrusion Detection System (IDS) for CAN networks in electric light commercial vehicles is the main topic of this work. The proposed architecture converts CAN messages into binary matrices by means of an image-based feature extraction technique coupled with a small CNN. Under different traffic conditions and bus load levels, this design helps the model to learn spatial and temporal patterns in real traffic and correctly identify spoofed or imposter communications.

To train and evaluate the system, a custom dataset was constructed using CAN traffic collected from a production-grade electric vehicle developed by Karsan. The dataset includes both genuine and adversarial scenarios, covering injections, spoofing, and DoS attacks. To the best of our knowledge, this is the first study to implement an IDS on CAN traffic conforming to the J1939 protocol, which is widely used in commercial vehicle architecture but remains underexplored in the literature [5,6].

The CAN protocol, originally developed by Bosch in the late 1980s, is widely adopted for its efficiency and simplicity in enabling communication among electronic control units (ECUs) without a centralized controller. However, it was not designed with external connectivity in mind, leaving it vulnerable to unauthorized access and manipulation [5,6]. Real-world incidents have demonstrated that attackers, whether through physical or remote access, can exploit these vulnerabilities to compromise vehicle safety [1,2]. For instance, falsified brake commands may lead to unintended vehicle behavior, posing serious risks to passenger safety [2,3]. To address these challenges, the automotive cybersecurity community has increasingly turned to artificial intelligence and machine learning techniques. Deep learning models, particularly CNNs, have shown promise in identifying complex patterns within CAN traffic, enabling adaptive intrusion detection beyond the capabilities of traditional rule-based systems [5,6,10].

This study introduces a scalable and intelligent IDS system that dynamically detects threats by learning from authentic communication sequences. Designed for deployment in embedded automotive environments, the system aims to enhance cybersecurity in next generation connected and autonomous vehicles by offering a novel detection mechanism, a reproducible dataset, and robust empirical performance metrics.

2. Related Work

Intrusion Detection Systems (IDS) intended for Controller Area Network (CAN) communications have garnered significant interest in recent years, especially with the development of autonomous, connected, and electric vehicles. Although conventional rule-based detection techniques offer simplicity, they are insufficient in adapting to complex and evolving cyberattack patterns. Thus, recent literature has moved towards methodologies based on deep learning (DL), particularly those employing convolutional neural networks (CNNs) [7-10] and recurrent models like LSTM [6]. More recent research includes lightweight depth wise separable CNNs that maintain accuracy with fewer parameters [11], transformer-based federated learning approaches that preserve data privacy across vehicles [12], unsupervised GAN-driven detectors that require no labelled data [14] and graph convolutional methods that model sequential frame dependencies [19].

Other recent efforts have explored unsupervised IDS based on attention autoencoders [13], deep embedded clustering approaches [14], ensemble multi-class classifiers for in-vehicle networks [16], comprehensive CNN/LSTM/GRU comparisons that evaluate model complexity and accuracy trade-offs [17], triple-attention architectures coupled with global optimization [18] and fully unsupervised detection frameworks tailored for CAN traffic [20]. Recent studies propose highly efficient architectures such as depth wise separable CNNs [11] and LETNN-based models that approximate self-attention with Toeplitz matrices [15], demonstrating real-time inference on automotive-grade microcontrollers.

These contributions together position our work as a practical, scalable, protocol-aware intrusion detection

system fit for contemporary intelligent transportation systems. While Wu et al [6] created a byte-level lightweight CNN fit for embedded systems [6], Shahriar et al. [5], proposed ConvIDS, a CNN architecture meant to identify payload anomalies in sequential CAN IDs [5]. Image-based representations of CAN messages were first presented by Marchetti and Stabili [4], allowing CNNs to extract spatial features free from handcrafted features [4]. These studies show three important constraints addressed in this work, even if they reflect major developments in CAN Bus security:

First, most current works use datasets gathered under simulated or limited environments that fail to reflect the high-volume and heterogeneous traffic observed in modern electric vehicle (EV) infrastructures. Characterized by high message density and continuous ECU interaction, our study focuses on CAN traffic obtained from a light commercial EV, so reflecting a more realistic and stress-intensive communication environment.

Second, past IDS datasets and models sometimes ignore industry-specific protocols and stress standard 11-bit CAN identifiers. Conversely, our hybrid dataset includes J1939-compliant messages, so enabling detection capabilities spanning heavy-duty and commercial vehicle networks where this protocol is extensively used.

Third, many deep learning-based IDS systems are computationally demanding and inappropriate for embedded deployment notwithstanding their accuracy. We developed small and resource-efficient CNN architecture especially targeted for low-power edge devices.

These contributions together position our work as a practical, scalable, protocol-aware intrusion detection system fit for contemporary intelligent transportation systems.

3. Proposed Intrusion Methods

3.1. Attack Scenarios

Though in-car communication mostly uses the Controller Area Network (CAN) Bus protocol, it lacks the necessary security elements, such as message authentication, encryption, and access control [1-5]. This weakness makes it an easy target for different cyberattacks. This work tested four different attack techniques in a controlled laboratory environment using both simulated intrusion attempts and real CAN traffic. The assaults were injected using a Raspberry Pi 5 and MCP2515 module, while normal traffic was tracked with CANoe software and a Vector VN1630 interface. Table 1 provides a comparative overview of various attack types with respect to their complexity and visibility characteristics

Table 1: Comparisons of CAN Bus Attack Types by Complexity, Visibility

Attack Type	Complexity	Visibility	References
Message Injection	Low	Medium	[4,5]
Spoofing	Medium	Low	[2,3,6]
Replay	Medium	Low	[2,6]
Masquerade	High	Very Low	[5]
DoS	Low	High	[1-4]
Fuzzy Attack	Medium	Medium	[1]
ID Flooding	Medium	High	[3]

Message Injection, one of the most basic attack forms in CAN systems is message injection. It sends illegal frames using current message IDs on the bus. Rewording or copying accurate ECU messages helps to produce these hostile messages influencing vehicle behavior. To remain invisible in the network, an assailant could send modified throttle, RPM, or break status messages with suitable IDs and frequency [4,5].

In ***Spoofing attack,*** under incorrect source identities, messages are sent allowing spoofing attacks to pass for real ECUs. This kind of attack uses the broadcast part and the absence of authentication in CAN communication. Utilizing real communications and changing their content before retransmission enabled us to spoof, or copy, the anti-lock braking system or transmission control module in our tests. [2,3,6].

At ***Denial-of-Service (DoS)*** Attack, typically utilizing low identification values to dominate the arbitration process, DoS attacks seek to flood the CAN bus with a high frequency of messages. Permanent high-priority frame delivery allows attackers to delay or obstruct authorized transfer, consequently possibly displaying important safety mechanisms useless. This was accomplished in our system by repeatedly low-ID messages like 0x000 at high speeds [1-4].

In ***Replay Attacks***, essential CAN communications are recorded during normal operation and then transmits in replay attacks. Replayed frame structure and timing seem reasonable, hence traditional rule-based systems find it difficult to identify such assaults. For instance, while the car is unattended, one

can replay a previously captured message for door unlocking or headlight activation [2,6].

Masquerade Attacks, Masquerade assaults, a sophisticated form of spoofing, include an attacker delivering fake messages implied to disable or control a real ECU and seize control of its communication abilities. Since these attacks can avoid systems depending simply on message frequency or ID, they are extremely dangerous. Our technology replicated masquerade behavior by replacing approved messages with material controlled by attackers [5].

As part of ***fuzzy attacks***, transmission of random or controlled CAN packets with arbitrary identities and contents is involved. One can aim to affect ECUs by identifying latent properties or inducing unexpected system responses. Continuous generation and injection of random IDs and 8-byte payloads let us evaluate how resilient the system was under fuzzing conditions [1].

Arbitration Abuse (ID Flooding), Lower ID value frames are given priority under the CAN arbitration system. Attackers might use this weakness by constantly sending low-ID messages to control the bus, so preventing or blocking communication with ECUs having higher IDs. This specific flooding method monitored the transmission latency of diagnostic or infotainment information by fast injection of IDs, such 0x001 or 0x003 [3].

3.2. Custom Dataset

This study introduces a large-scale, hybrid custom dataset designed to support the training and evaluation of deep learning-based Intrusion Detection Systems (IDS) for Controller Area Networks (CAN) in electric light commercial vehicles. The dataset adheres to the J1939 protocol and was collected from a production-grade electric vehicle developed by Karsan, ensuring both industrial relevance and real-world protocol compliance.

Data collection was carried out in two coordinated phases. In the first phase, authentic in-vehicle CAN traffic was recorded using a Vector VN1630 interface in conjunction with Vector CANoe, a widely used simulation and analysis platform. Raw messages were logged directly via the OBD-II port from multiple Electronic Control Units (ECUs), including those responsible for engine, transmission, braking, and dashboard instrumentation. To ensure diversity in signal frequency and bus load, data was collected under varied driving conditions such as idling, acceleration, deceleration, and highway cruising. Timestamping with millisecond precision allowed for accurate sequence modeling.

In the second phase, cyberattack scenarios were simulated in a controlled environment. Using a Raspberry Pi 5 with an MCP2515 CAN transceiver, various adversarial actions—such as spoofing, message injection, denial-of-service (DoS), and replay attacks—were injected into the network. These intrusions were executed during both static and dynamic states of the vehicle. Custom Python scripts facilitated precise control over payload manipulation and timing, and all injected frames were clearly labeled to enable supervised learning and post-analysis.

To ensure consistency with previous work, the preprocessing pipeline was inspired by the framework of Marchetti and Stabili [4]. Steps included labeling, cleaning, time alignment, and class balancing. The final dataset contains 4,792,115 CAN frames formatted as time-ordered sequences, each including message ID, Data Length Code (DLC), and raw payload bytes.

Table 2: Data Split Summary for the CAN Bus Dataset

Set	Total Samples	Attack Samples	Normal Samples
Training	3204680	130999	3073681
Validation	1373435	56142	1317293
Test	80000	40000	40000

Design choices for dataset construction were informed by prior research. For example, stealthy payload manipulation strategies were guided by Shahriar et al. [5] and Wu et al. [6], while the selection of attack types aligns with works by Avatefipour et al. [2] and Alfardus and Rawat [1]. These references emphasize the importance of detecting not only syntactically anomalous frames but also semantically manipulated messages that mimic legitimate traffic.

By combining real-world CAN traffic with realistic adversarial scenarios, this dataset offers a reproducible, high-fidelity foundation for developing and evaluating robust IDS solutions in embedded automotive environments.

3.3. Proposed Binary Image Generation Algorithm

To transform CAN Bus messages into a visual representation suitable for convolutional neural network (CNN)-based intrusion detection models, a

structured preprocessing algorithm was developed.

Algorithm 1: Pseudocode for Binary Image Generation from CAN Messages

Input: Input folder with CAN message CSV files
Output: 8×13 grayscale images representing CAN messages

1. Open or create a log file in the output folder
2. **foreach** CSV file in the input folder **do**
   2.1. Load the CSV file as a table of CAN messages
   2.2. foreach row in the table **do**
      a. Extract **timestamp**, **CAN_ID**, **DLC**, **data bytes**
      b. Convert **CAN_ID** to binary (11 or 29 bits)
      c. Convert **DLC** to 4-bit binary
      d. Convert each data byte to 8-bit binary
      e. Concatenate all binary segments into one binary stream
      f. if binary stream length < 104 bits **then**
      g. Pad with zeros to reach 104 bits
      h. Reshape the 104-bit binary stream into an 8 × 13 binary matrix
      i. Convert binary matrix to a grayscale image (0 = black, 1 = white)
      j. Save the image using a unique name (e.g., `timestamp_CANID.png`)
      k. Log the image path and source message info to the log file
3. Close the log file

The initial samples and their matrix representations are also logged for traceability and debugging. By converting sequential CAN messages into spatially structured visual inputs, this approach supports the use of computer vision techniques in the context of CAN Bus intrusion detection [5], [6].

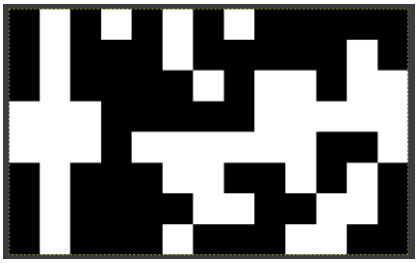

Figure 1: Binary Image Example

Finally, depending on their source filenames, the images are stored into organized output folders. This company helps with deep learning pipelines to process future batches. The approach allows the use of computer vision methods to CAN Bus intrusion detection by converting temporal and semantic message features into a visual domain. Figure 1 shows a visual representation of the produced binary image.

3.4. Proposed Model Architecture

The proposed model adopts a compact convolutional neural network (CNN) architecture designed to process binary-encoded CAN messages represented as 8×13 grayscale images. The structure and meaning of each CAN frame are shown in this way. It includes the message ID, the data length code (DLC), and the data payload, all of which are organized into a fixed-length binary matrix. The model's architecture is kept simple on purpose to keep the performance for detecting different types of attacks high while reducing the amount of work that needs to be done. An input layer accepting 8x13 single-channel grayscale images starts the architecture. Two consecutive convolutional layers transverse this input. Using ReLU activation and a 3x3 kernel size, the first convolutional layer employs 16 filters. To lower spatial dimensions, a max pooling operation with a 2x2 pool size follows. Following another max pooling layer, the second convolutional layer uses 32 filters, also with a 3x3 kernel size and ReLU activation. The feature maps are flattened into a one-dimensional vector following the convolutional blocks and then fed through a fully connected dense layer comprising 32 units and ReLU activation. One uses a dropout layer with a rate of 0.2 to enhance generalization and stop overfitting. A further dense layer with sixteen units follows this to act as a compact feature representation layer. Applied before the last output layer is still another dropout layer with a higher dropout rate of 0.4. One sigmoid-activated single neuron in the output layer generates a binary prediction showing whether the input message is authentic or hostile. Table 3 presents a summary of the proposed architecture for the deep learning model.

Table 2: Model Summary Table

#	Layer Type	Input Shape	Output Shape
1	Input	(None,8,13,1)	(None,8,13,1)
2	Cast	(None, 8, 13, 1)	(None, 8, 13, 1)
3	Conv-2D	(None, 8, 13, 1)	(None, 8, 13, 16)
4	Max Pooling 2D	(None, 8, 13, 16)	(None, 4, 6, 16)
5	Conv-2D	(None, 4, 6, 16)	(None, 4, 6, 32)
6	Flatten	(None, 4, 6, 32)	(None, 768)
7	Dense	(None, 768)	(None, 32)
8	Dropout	(None, 32)	(None, 32)
9	Dense	(None, 32)	(None, 16)
10	Dropout	(None, 16)	(None, 16)
11	Cast	(None, 16)	(None, 16)
12	Dense	(None, 16)	(None, 1)

Because mixed-precision training causes the TensorFlow runtime to automatically insert implicit casting operations—also known as Cast layers—which are then included into the model. While maintaining float32 precision where needed to preserve numerical stability, mixed-precision computation lets some operations run in float16 form for enhanced speed and low memory use. Cast layers thus show in the computation graph to translate data types at particular model points. These layers are required for hardware-level compatibility and optimization but have no bearing on the logical framework of the network.

With less than 200,000 trainable parameters and a lightweight architecture, which fits for deployment on embedded systems with limited memory and processing capability, The model is simple yet can learn minute changes in CAN traffic patterns including structural abnormalities and payload manipulations. The architectural fit with image-based CAN data enables it to exploit spatial correlations in the binary format, so enhancing detection robustness without depending on recurrent or temporal components.

3.5. Experimental Setup and Results

Emphasizing computational efficiency, generalization, and effective attack detection, the training configuration of the proposed CNN model has been optimized to perform effectively under real-world constraints. Designed for binary classification, that is, to tell whether a CAN Bus frame is authentic or spoofed —the model was trained with binary cross-entropy loss function. Using an Adam optimizer with a learning rate of 1e-4 ensures stable convergence.

Using TensorFlow's mixed-precision training, which lets some operations run in float16 precision and so lowers memory usage, training was accelerated. Combining the optimizer with a loss scaling method helped to preserve numerical stability under this precision-aware approach. Especially, model behavior is unaffected by runtime implicit casting operations introduced in mixed-precision execution.

Early stopping and model checkpointing were used to avoid overfitting and guarantee generalization. Training stopped when the validation loss stopped improving, and based on validation accuracy the best-performing model was preserved.

Class weighting was used to correct class imbalance in the training data, so raising sensitivity to rare attack samples.

Training, validation, and test sets formed out of the dataset. Training comprised specifically 3,073,681 authentic and 130,999 imposter samples; validation comprised 1,317,293 authentic and 56,142 attack samples. There was final evaluation using 40,000 authentic and 40,000 attack samples on a balanced test set. From binary-encoded CAN frame fields, every sample was preprocessed into an 8×13 grayscale image.

For a maximum of 16 epochs with a batch size of 64 the model was trained on a high-performance workstation furnished with an Intel Core i7-12650H CPU, 16 GB RAM, and an NVIDIA GeForce RTX 4070 Laptop GPU.

Table 3: Confusion Matrix

Confusion Matrix			
True Label	0	39291	709
	1	192	39808
		0	1
		Predicted Label	

Following training, the efficacy of the proposed model was carefully evaluated using a balanced test set. Essential classification metrics—namely Accuracy (98.87%), Precision (98.25%), Recall (99.52%), F1-Score (98.88%), and ROC-AUC (98.87%)—were employed to assess its detection performance. These values, derived from the Confusion Matrix given in Table 4, highlight the model's high reliability and consistent predictive capability. The results line up with detection performances reported in previous studies by Avatefipour et al. [2] and Jichici et al. [3], showing the model's strong capability to differentiate between genuine and counterfeit CAN traffic.

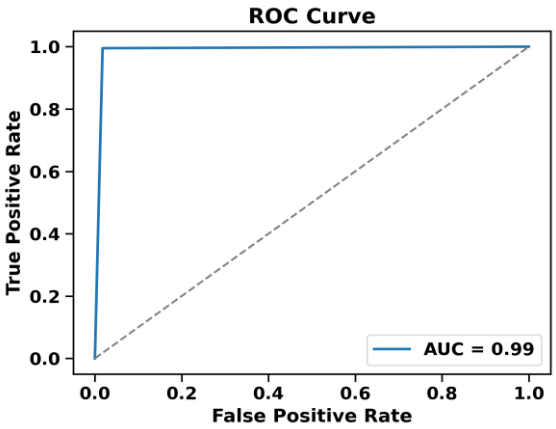

Figure 2: ROC Curve

The ROC-AUC value of 98.87% points out the discriminative efficiency of the proposed CNN-based architecture. Figure 2 shows the ROC curve, which visually represents the trade-off between true positive and false positive rates at different thresholds.

Moreover, Figure 3 illustrates the model's learning behavior during training, showcasing the accuracy progression across epochs. Furthermore, Figure 4 illustrates the loss curve, providing insight into the model's convergence and generalization patterns during the training and validation phases. Figure 5 and 6 illustrate the PCA and t-SNE feature embeddings, respectively, providing insights into the model's classification of distinct classes within the feature space.

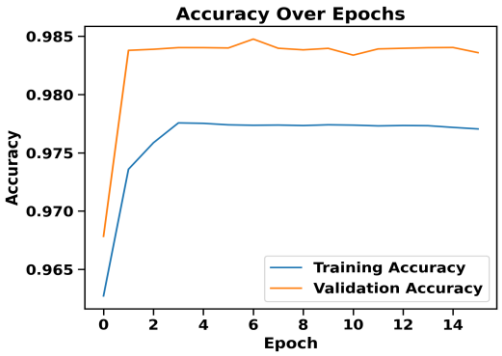

Figure 3: Accuracy Over Epochs

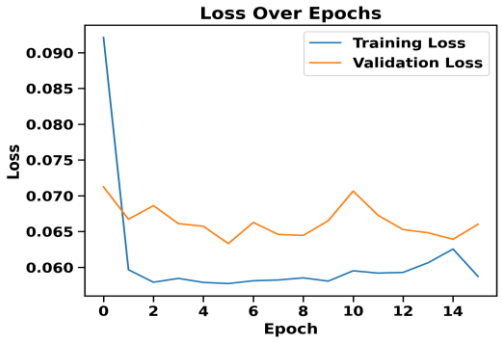

Figure 4: Loss Over Epochs

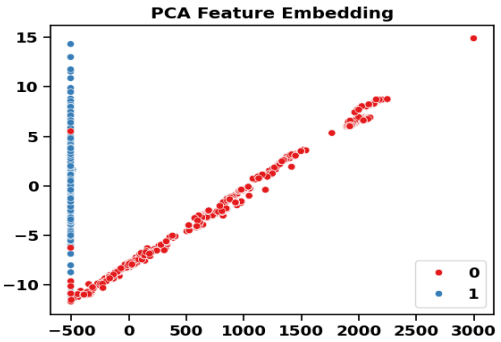

Figure 5: PCA Feature Embedding

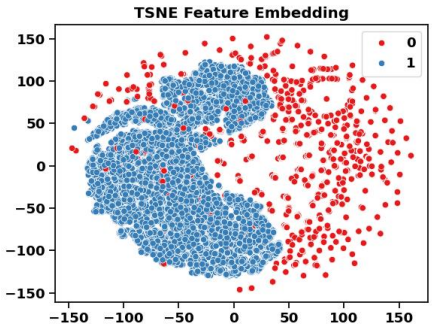

Figure 6: TSNE Feature Embedding

4. Conclusion

Specifically developed for Controller Area Network (CAN) Bus cybersecurity in commercial vehicles, this work presented a lightweight, image-based intrusion detection system. By converting binary-encoded CAN frames into 8×13 grayscale images, the proposed CNN model successfully extracted spatial patterns, allowing it to learn from structured message components such as the ID, DLC, and payload. Designed for embedded computing platforms with limited resources, the architecture offered an effective and low-latency solution suitable for real-time applications—thus addressing the operational constraints highlighted in past work [1,5,6] .

A hybrid dataset was constructed by combining real-world CAN traffic—collected from a J1939-compliant light commercial vehicle—with synthetically generated attack messages injected via low-cost embedded hardware. This empirical approach focused on realism by using authentic vehicle communication logs and deliberately timed intrusions, in contrast to studies based solely on generative augmentation or virtual simulation environments. The resulting dataset preserved both the temporal and semantic structure of CAN traffic, aligning with industry best practices as emphasized in prior studies [2,3].

With 98.87% accuracy, 0.9952 recall, and a ROC-AUC of 0.9887, experimental evaluation showed the effectiveness of the proposed model, verifying its strong detection capacities against spoofing, injection, and denial-of- service (DoS) attacks. Furthermore, applied to the feature embeddings of the model were Principal Component Analysis (PCA), t-distributed Stochastic Neighbor Embedding (t-SNE), and cosine similarity analysis, so exposing obvious separability between real and spoofed samples. In line with trends seen in recent literature, these results confirmed the relevance of compact CNN

architectures for resource-constrained edge cybersecurity systems and validated the discriminative quality of the learned representations [4].

Unlike many academic methods depending just on pre-processed datasets or theoretical assumptions, this work offered a totally reproducible, end-to-end solution including data acquisition, binary image generation, model training, and performance evaluation under real-world conditions. It also showed how reasonably priced embedded platforms like the Raspberry Pi can replicate attack scenarios, so enabling real-data-based research more freely independent of outside generative tools.

Based on empirical validation and ready for use, this work offers a practical, efficient, and deployable CAN Bus intrusion detection system, supporting the larger vision of intelligent and safe transportation systems.

The dataset will be expanded to support multi-vehicle and fleet-level evaluations in next work, and extra model improvements including temporal learning modules such recurrent neural networks or transformer-based architectures. Moreover, distributed and cooperative cybersecurity methods including federated learning will be investigated to enable safe model training between several vehicle systems without endangering data privacy.

Article Information

Financial Disclosure: The author (s) has no received any financial support for the research, authorship or publication of this study.

Authors' Contribution: Concept: E.T., C.H., H.G.; Design: E.T., C.H., H.G.; Supervision: E.T., C.H., H.G.; Resources: E.T., C.H., H.G.; Data Collection: E.T., C.H., H.G.; Analysis: E.T., C.H., H.G.; Literature Search: E.T., C.H., H.G.; Writing Manuscript: E.T., C.H., H.G.; Critical Review: E.T., C.H., H.G.

Conflict of Interest/Common Interest: No conflict of interest or common interest has been declared by the authors.

Ethics Committee Approval: This study does not require ethics committee permission or any special permission

Acknowledgments
We would want to sincerely thank you to Karsan for their essential contribution in allowing the data collecting process. Particularly, we would want to thank Mr. M. Alper BALIM, E&E Vehicle Software Engineering Administrator, and Mr. Nurettin ÖZEKMEKCİ, Manager of E&E Design Engineering, whose priceless direction, insights, and ongoing support have greatly helped this research to be successful. We also want to thank the whole Karsan R&D team for their kind cooperation, test vehicle preparation, and guarantee of access to the required tools and systems for experimental use. Their dedication and transparency made it feasible to create an industrial-grade, realistic dataset and to do all stages of this work under reasonable, dependable surroundings.

References

[1] Alfardus, A., & Rawat, D. B. (2021, 1–4 Aralık). Intrusion detection system for CAN bus in-vehicle network based on machine learning algorithms.In Proceedings of the 2021 IEEE 12th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON) (pp. 944–949).IEEE. https://doi.org/10.1109/UEMCON53757.2021.9666745

[2]Avatefipour, O., Al-Sumaiti, A. S., El-Sherbeeny, A. M., Awwad, E. M., Elmeligy, M. A., Mohamed, M. A., & Malik, H. (2019). An intelligent secured framework for cyber-attack detection in electric vehicles' CAN bus using machine learning. IEEE Access,7,127,580-127 https://doi.org/10.1109/ACCESS.2019.2937576

[3] Jichici, C., Groza, B., Ragobete, R., Murvay, P.-S., & Andreica, T. (2022). Effective intrusion detection and prevention for the commercial vehicle SAE J1939 CAN bus. IEEE Transactions on Intelligent Transportation Systems, 23(10), 17 425–17 439. https://doi.org/10.1109/TITS.2022.3151712

[4] Marchetti, M., & Stabili, D. (2017, 11–14 Haziran). Anomaly detection of CAN bus messages through analysis of ID sequences [Conference paper]. In Proceedings of the 2017 IEEE Intelligent Vehicles Symposium(IV)(pp.1577–1583).IEEE. https://doi.org/10.1109/IVS.2017.7995934

[5] Shahriar, M. H., Xiao, Y., Moriano, P., Lou, W., & Hou, Y. T. (2023). CANShield: Deep-learning-based intrusion detection framework for controller area networks at the signal level. IEEE Internet of Things Journal, 10(24), 22111–22127. https://doi.org/10.1109/JIOT.2023.3303271

[6] Wu, S., Li, S., & Sun, W. (2023, 6–8 Aralık). ConvIDS: A convolutional LSTM-based intrusion detection model for in-vehicle CAN bus [Conference paper]. In Proceedings of the 2023 IEEE 6th International Conference on Automation, Electronics & Electrical Engineering (AUTEEE) (pp. 285–290). IEEE. https://doi.org/10.1109/AUTEEE60196.2023.10408040

[7] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278–2324.https://doi.org/10.1109/5.726791

[8] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, & K. Q. Weinberger (Eds.), Advances in Neural Information Processing Systems 25 (pp. 1097–1105). Curran Associates. (https://papers.nips.cc/paper_files/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html)

[9] Yamashita, R., Nishio, M., Do, R. K. G., & Togashi, K. (2018). Convolutional neural networks: An overview and application in radiology. Insights into Imaging, 9(4), 611–629. https://doi.org/10.1007/s13244-018-0639-9

[10] Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017, 21–23 Ağustos). Understanding of a convolutional neural network [Conference paper]. In Proceedings of the 2017 International Conference on Engineering & Technology (ICET) (pp. 1–6). IEEE. https://doi.org/10.1109/ICEngTechnol.2017.8308186

[11] Zhao, W., Yang, Y., Hu, H., Chen, Y., & Yu, F. (2025). A lightweight intrusion detection approach for CAN bus using depthwise separable convolutional Kolmogorov–Arnold network. Scientific Reports, 15, 17550. https://doi.org/10.1038/s41598-025-02474-1

[12] Zhang, Y., Song, J., Sun, Y., Gao, Z., & Hu, Z. (2025). Federated two-stage transformer-based network for intrusion detection in non-IID data of controller area networks. Cybersecurity, 8(1), 29. https://doi.org/10.1186/s42400-024-00329-2

[13] Wei, P., Wang, B., Dai, X., Li, L., & He, F. (2023). A novel intrusion detection model for the CAN bus packet of in-vehicle network based on attention mechanism and autoencoder. Digital Communications and Networks, 9, 14–21. https://doi.org/10.1016/j.dcan.2022.04.021

[14] Shi, J. Z., Dong, L., Jiang, X., & Jin, X. (2024). IDS-DEC: A novel intrusion detection for CAN bus traffic based on deep embedded clustering. Vehicular Communications, 49, Article 100830. https://doi.org/10.1016/j.vehcom.2024.100830

[15] Shi, D., Xu, M., Qin, Z., & Zhong, Y. (2025). Local-enhanced Toeplitz neural network for in-vehicle network intrusion detection. Journal of King Saud University – Computer and Information Sciences, 37, Article 8. https://doi.org/10.1007/s44443-025-00006-5

[16] Gou, W., Zhang, H., & Zhang, R. (2023). Multi-classification and tree-based ensemble network for the intrusion detection system in the Internet of Vehicles. Sensors, 23(21), 8788. https://doi.org/10.3390/s23218788

[17] Rai, R., Grover, J., Sharma, P., & Pareek, A. (2025). Securing the CAN bus using deep learning for intrusion detection in vehicles. Scientific Reports, 15, 13820. https://doi.org/10.1038/s41598-025-98433-x

[18] Yang, H., & Effatparvar, M. (2025). A deep learning based intrusion detection system for CAN vehicle based on combination of triple attention mechanism and GGO algorithm. Scientific Reports, 15, 19462. https://doi.org/10.1038/s41598-025-04720-y

[19] Devnath, M. K. (2023). GCNIDS: Graph convolutional network-based intrusion detection system for CAN bus. arXiv Preprint, arXiv:2309.10173. https://doi.org/10.48550/arXiv.2309.10173

[20] Kabilan, N., Ravi, V., & Sowmya, V. (2024). Unsupervised intrusion detection system for in-vehicle communication networks. Journal of Safety Science and Resilience, 5, 119–129. https://doi.org/10.1016/j.jnlssr.2023.12.004