Spatial Fingerprinting of CAN Bus Attacks: A 2D Projection and CNN-based Intrusion Detection System

Shuo PENG[1,a)] Zhihe ZHANG[1,b)] Go TSURUOKA[1,c)] Tatsuya Mori[1,2,3,d)]

Abstract: Ensuring the accurate transmission of control signals is critical to the safety of modern vehicles. To achieve this, most vehicles rely on the Controller Area Network (CAN) bus for communication between key components. However, the CAN protocol lacks inherent security features, leaving it vulnerable to various cyber-attacks. To address this issue, we propose a novel intrusion detection system (IDS) capable of determining both the normality of traffic and the specific type of attack. Our approach maps raw CAN bus traffic into a three-dimensional behavioral feature space, forming a "spatial fingerprint" of network activity that captures the relationships among message frequency, source identifier, and data stability. These fingerprints are further projected into two-dimensional multi-channel images, on which a Convolutional Neural Network (CNN) is trained for classification, leveraging CNNs' strong capability to extract spatial features from structured image-like data. Evaluated on the comprehensive CAN-MIRGU dataset, the proposed method achieves an overall accuracy of 96% and a macro-average F1-score of 0.94 across 30 classes. The results demonstrate that our approach not only overcomes challenges such as severe class imbalance but also highlights the effectiveness of visualizing behavioral patterns for enhancing in-vehicle network security.

Keywords: CAN bus, IDS

1. Introduction

The Controller Area Network (CAN) bus, which serves as the central nervous system for modern vehicles, inherently lacks fundamental security features such as encryption and authentication, rendering vehicles highly susceptible to a growing landscape of cyber-attacks [1]–[3]. To address these critical vulnerabilities, numerous Intrusion Detection Systems (IDSs) have been proposed in recent years, employing various approaches ranging from traditional statistical methods analyzing message frequency and timing patterns [1], to advanced machine learning techniques including supervised and semi-supervised deep learning models [2], and sophisticated detection mechanisms targeting specific attack types such as masquerade attacks [1], [3], [4].

However, existing IDS approaches exhibit several critical limitations that hinder their practical deployment. First, many established methods rely on single-dimensional features, making them vulnerable to sophisticated adversaries who can craft stealthy attacks that maintain normal timing or ID distributions [1], [4]. Second, ML-based IDSs suffer from significant performance degradation when evaluated on new datasets or with slight variations in attack types, revealing a strong dependency on training data quality and diversity [5], [6]. Third, the field lacks uniform evaluation methodologies due to the scarcity of high-quality public datasets, which has "stymied comparability and reproducibility of results for researchers" [3], [5]–[7]. Finally, the prevalence of synthetic datasets or real datasets with easily detectable attacks limits the generalizability of proposed solutions to real-world scenarios [3], [7], [8]. These collective limitations underscore the critical need for novel IDS solutions that can capture complex, multi-dimensional attack patterns and be validated on challenging, realistic datasets.

To address these challenges, we introduce a novel methodology that transforms complex, temporal CAN data into a rich, visual representation we term a "*Spatial Fingerprint*." Unlike existing single-dimensional approaches, our method captures multi-faceted network behavior by simultaneously analyzing multiple orthogonal features, thereby addressing the critical limitation of simplistic feature analysis that makes existing methods vulnerable to sophisticated attacks. Our core hypothesis is that variations between benign network states and specific attack states generate visually distinct patterns within a multi-dimensional behavioral space, which in turn enables more robust detection of stealthy attacks that might otherwise evade traditional timing- or frequency-based methods.

[1] Waseda University
[2] RIKEN AIP
[3] NICT
[a)] housaku@nsl.cs.waseda.ac.jp
[b)] zzhihe@nsl.cs.waseda.ac.jp
[c)] go@nsl.cs.waseda.ac.jp
[d)] mori@nsl.cs.waseda.ac.jp

Specifically, our approach engineers a 3D feature space from raw CAN logs using three complementary dimensions: (1) inter-arrival time capturing message frequency patterns, (2) CAN ID representing message sources, and (3) Hamming distance quantifying payload stability between consecutive messages. These 3D features are then projected onto 128×128 pixel RGB images using a multi-view projection technique, where each color channel encodes a different 2D projection of the behavioral space. This visual transformation enables the application of Convolutional Neural Networks (CNNs) for classification, leveraging their proven capability in spatial pattern recognition. Crucially, to address the limitation of unrealistic evaluation scenarios, we validate our approach on the comprehensive CAN-MIRGU dataset—a challenging, real-world dataset containing 36 classes of physically verified attacks collected from a moving vehicle with autonomous driving capabilities.

Evaluated on the comprehensive CAN-MIRGU dataset, our method demonstrates robust performance. In our framework, raw CAN logs are first transformed into 3D spatial features, which are then projected into 2D RGB fingerprints and finally classified using a CNN. This pipeline achieves an overall accuracy of 96% and a macro-average F1-score of 0.94 across 30 classes.These results not only confirm the effectiveness of our approach on realistic and challenging data, but also highlight its advantages over prior IDS studies that often relied on synthetic or less comprehensive datasets, thereby underscoring the significance of spatial fingerprinting for advancing in-vehicle network security.

Our contributions are summarized as follows:

- We propose a novel spatial fingerprinting method that transforms CAN bus traffic into 3D behavioral patterns, enabling multi-dimensional attack detection beyond traditional single-feature approaches.
- We develop a multi-view projection technique that converts continuous 3D features into 2D RGB fingerprint images, making complex behavioral patterns analyzable by standard computer vision models.
- We demonstrate the effectiveness of our approach through rigorous evaluation on the CAN-MIRGU dataset, achieving state-of-the-art performance across 29 different attack and benign classes in realistic driving conditions.
- We provide practical solutions for critical deployment challenges, including class imbalance mitigation through weighted training and model capacity optimization, establishing a robust training methodology for real-world IDS deployment.

2. Related Work

The widespread reliance of modern vehicles on the CAN has spurred in-depth research into in-vehicle network intrusion detection systems.

Conventional CAN IDS Approaches. CAN IDSs are typically categorized into signature-based approaches, which rely on predefined attack rules, and anomaly-based approaches, which detect deviations from profiled normal behavior and are more effective against unknown attacks [2], [3]. Anomaly-based methods include time-based approaches that leverage the regularity of CAN message inter-arrival times, and statistical algorithms that model attack-free datasets and monitor deviations in metrics such as entropy, CUSUM, or Hamming distance of payloads [3], [5]. Although computationally inexpensive, these conventional methods may lack the sophistication to capture complex interrelationships among vehicular features.

Machine Learning-based CAN IDSs. Recent years have seen increasing adoption of ML and DL techniques for CAN intrusion detection. These systems range from basic neural networks to complex architectures like recurrent networks and autoencoders [8]. The AMICA model utilizes attention mechanisms inspired by BERT to simultaneously model temporal relationships within individual CAN IDs and interactions between different IDs [1]. HistCAN employs a CNN-MLP hybrid encoder with a historical information fusion module to learn spatial, temporal, and long-term dependencies in CAN ID sequences, proving suitable for real-time deployment [2]. For resource-constrained environments, CAN-ODTL provides on-device transfer learning capabilities supporting incremental retraining on streaming CAN data [9].

Specialized CAN Attack Detection. Signal clustering methods have been specifically developed for detecting sophisticated CAN attacks. Hierarchical clustering applied to CAN signal time series has shown promise in detecting masquerade attacks by analyzing changes in signal correlations [4]. These methods demonstrate the importance of analyzing multi-dimensional relationships in CAN traffic for detecting stealthy attacks.

CAN IDS Datasets. The effectiveness of IDS solutions heavily depends on dataset quality and realism. Historically, the lack of high-quality, publicly available real CAN attack data has been a significant obstacle [3], [5], [6]. Early datasets like HCRL's Car-Hacking Dataset and the ROAD dataset had limitations including stationary vehicle data collection, missing labels, or limited scope [6], [7]. The recent CAN-MIRGU dataset addresses these shortcomings by providing comprehensive data from a moving vehicle with autonomous driving capabilities, including 17 hours of benign data and 36 physically verified real attacks [7].

Our Approach and Key Contributions. Unlike existing methods that rely on single-dimensional features or require complex temporal modeling, our work introduces a fundamentally different approach: transforming CAN traffic into visual fingerprints. While methods like AMICA and HistCAN focus on temporal sequence modeling, and clustering approaches analyze signal correlations, we pioneer the use of spatial visualization combined with computer vision. Our method uniquely captures the simultaneous relationships between message timing, source identifiers, and pay-

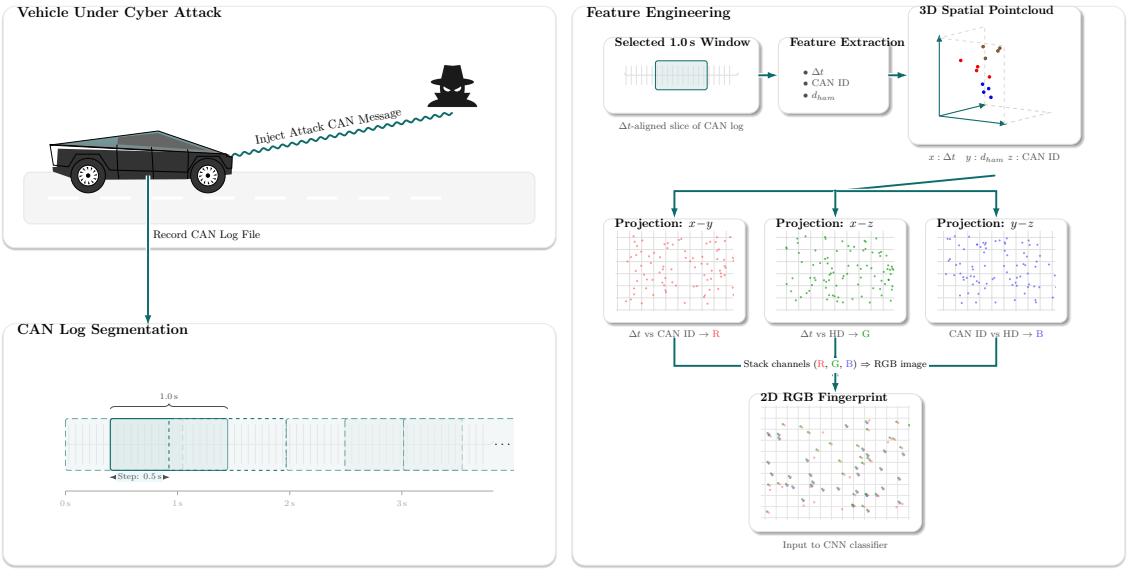


Fig. 1 The pipeline of the proposed system.

load variations in a single 2D representation, enabling standard CNNs to detect complex attack patterns without specialized architectures. This visual transformation approach not only simplifies the detection pipeline but also provides intuitive interpretability of network behavior, addressing a key limitation in existing black-box ML approaches.

3. Methodology

3.1 Overall Framework

Our proposed system follows a four-stage pipeline, as illustrated in Figure 1. First, raw CAN log files are processed to segment the data by overlapping time windows. Second, for each time window, a Feature Extractor is applied to generate a 3D feature. Third, a Multi-View Projector transforms the feature points into a 2D RGB image. Finally, these images are used to train a CNN for classification. It should be noted that, although the CAN-MIRGU dataset defines 36 attack types, we selected 29 for our experiments. Several categories such as Multiple_Attack are composite or ambiguous in nature and cannot be regarded as well-defined, independent classes. Including them would blur class boundaries and reduce the interpretability of the classification task. By focusing on the 29 clearly defined attack types along with benign traffic, we ensure that the evaluation reflects meaningful and distinguishable behaviors in the CAN bus.

3.2 Phase 1: Spatial Feature Engineering

To capture the multi-faceted behavior of CAN traffic, we engineer three features for each CAN message, which compose 3D spatial features, as shown in Figure 2. The three axes are defined as follows:

· X-axis (Δt): Represents the inter-arrival time for messages of the same ID. This feature captures the frequency of messages from a specific source.

· Y-axis (CAN ID): A direct integer conversion of the hexadecimal CAN ID, representing the message source.

· Z-axis (d_{ham}): The Hamming distance between the ASCII hexadecimal payloads of two consecutive messages from the same ID. It quantifies the magnitude of change in the data content, acting as a measure of information stability.

In this visualization, all dataset frames labeled as attacks are isolated and visualized separately. Each attack category is rendered in a distinct color, whereas white points denote benign CAN frames. The figure shows that different attack types form separable clusters and distributions in the spatial-fingerprint space, while benign traffic occupies a relatively compact region. For example, high-disruption attacks (e.g., DoS, fuzzing) produce dense or widely scattered patterns that deviate markedly from benign traffic, whereas stealthier replay and masquerade attacks lie closer to the benign distribution yet exhibit subtle shifts, particularly along the payload-variation axis (d_{ham}).

This separation is significant: it indicates that the engineered feature space disentangles diverse attack behaviors.

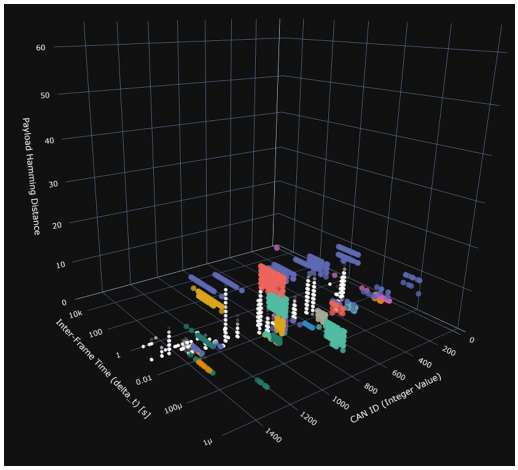
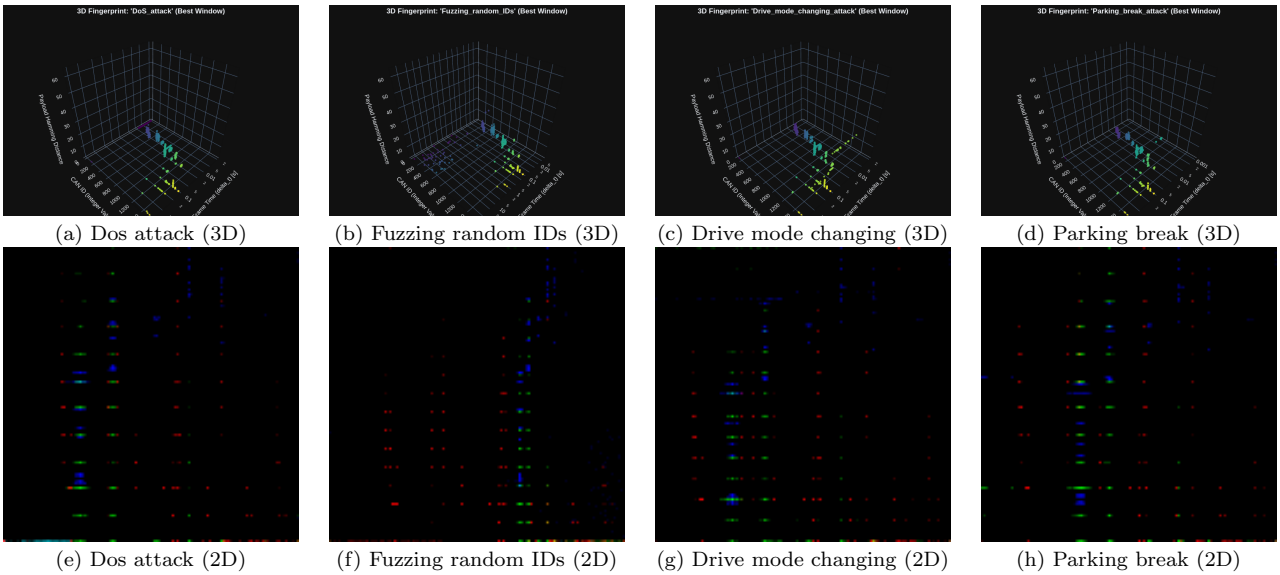


Fig. 2 3D Spatial Features for Visualizing CAN Traffic Features. The X, Y, and Z axes represent Δt, CAN ID, and d_{ham}, respectively.



(a) Dos attack (3D) (b) Fuzzing random IDs (3D) (c) Drive mode changing (3D) (d) Parking break (3D)

(e) Dos attack (2D) (f) Fuzzing random IDs (2D) (g) Drive mode changing (2D) (h) Parking break (2D)

Fig. 3 3D fingerprints of representative attack types at their most concentrated windows, and the corresponding 2D RGB projections. Distinct geometric patterns are preserved after projection, confirming the discriminative power of the feature transformation.

Transforming raw CAN traffic into structured 3D features provides a basis for visually differentiating attacks—a property that is preserved, and often enhanced, when projecting these features into 2D multi-channel fingerprints. The resulting fingerprints retain the separability between benign and malicious traffic while making the data amenable to CNN-based analysis, enabling both automated classification and intuitive human interpretation.

The selection of these three features is intentional, as they form a complementary and largely orthogonal feature space designed to provide a comprehensive view of network activity. Δt is highly sensitive to frequency-based anomalies, making it crucial for detecting attacks like DoS and message injection (fuzzing). Hamming distance focuses on the data payload, capturing content manipulations characteristic of masquerade or sophisticated replay attacks. The CAN ID provides the essential context, attributing these temporal and data-level changes to a specific source. By combining these three dimensions, our spatial fingerprint can theoretically capture a wider and more diverse range of attack vectors than methods relying on a single behavioral aspect, providing a more holistic view of the network's state.

3.3 Phase 2: 2D Fingerprint Image Generation

To make the 3D data compatible with a 2D CNN, we convert it into images. The data stream is first segmented into 1.0-second windows with a 0.5-second step (50% overlap). For each window, we generate a 128x128 pixel RGB image via multi-view projection:

· Red Channel: A 2D histogram (numpy.histogram2d) of the points projected onto the Δt vs. CAN ID plane.

· Green Channel: A 2D histogram of the points projected onto the Δt vs. d_{ham} plane.

· Blue Channel: A 2D histogram of the points projected onto the CAN ID vs. d_{ham} plane.

The histogram counts are normalized using a logarithmic transformation (numpy.log1p) to enhance the visibility of low-density patterns. The resulting three grayscale maps are stacked to form the RGB image.

To further illustrate the discriminative capability of the proposed features, we select, for each attack, the time window with the highest concentration of malicious frames and visualize its corresponding 3D and 2D representations. Figures 3 show examples of such fingerprints. The 3D features highlight how attack traffic forms distinctive geometric patterns in the feature space defined by inter-arrival time, CAN ID, and payload Hamming distance. Their 2D projections, in turn, demonstrate that these structures are preserved when transformed into image-like representations. These visualizations provide an intuitive confirmation that our feature engineering method effectively separates different attack types, enabling robust learning by the CNN classifier.

3.4 Phase 3: CNN-based Classification

A sequential CNN, as illustrated in Figure 4, was implemented for classification. The architecture was designed to effectively extract hierarchical features while remaining computationally lean. It begins with a convolutional block of 32 filters (3×3 kernel) to learn low-level patterns like edges and textures from the fingerprint images. A second convolutional block follows, increasing the filter count to 64. This progressive increase is a standard design choice that enables the network to build upon the initial patterns to learn more complex and abstract features representative of specific attacks. Each convolutional layer is paired with a 2×2 Max-Pooling layer for downsampling and enhancing feature invariance. The resulting feature maps are then flattened and processed by a fully connected layer with 128 units, before

a final Softmax layer produces the probability distribution across the 30 target classes.

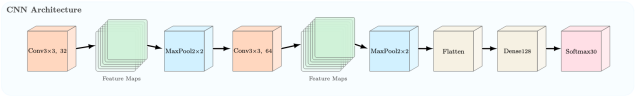


Fig. 4 Proposed CNN architecture.

The model was trained using the Adam optimizer. To address the severe class imbalance in the dataset, class weights were calculated using scikit-learn's balanced mode and applied during training, ensuring greater emphasis on minority classes. Early experiments revealed underfitting, where validation accuracy exceeded training accuracy. This was resolved by removing inappropriate regularization layers (Dropout, RandomFlip, RandomRotation), allowing the model to fully utilize its learning capacity.

4. Results and Analysis

4.1 Experimental Setup

We utilized the publicly available CAN-MIRGU dataset, which contains data collected from a modern vehicle under real driving conditions [7]. The dataset comprises over 17 hours of benign data and 36 attack types. For our experiments, we selected 29 attack types available in our log files, resulting in 29 attack classes plus one benign class. Ground truth labels were assigned using the provided Attacks_metadata.json file. The dataset was partitioned into 80% for training and 20% for validation.

Our training configuration, with all parameters listed in Table 1, was carefully selected to optimize the model's performance. The image size was set to 128x128 pixels to balance fine-grained feature resolution against computational overhead. For the temporal dimension, we chose a 1.0-second window with a 0.5-second step. This approach, creating a 50% overlap between windows, supports our goal of detailed classification rather than just high-speed anomaly detection. A longer window aggregates more data points into a richer 3D fingerprint, and the overlap ensures that temporal patterns are captured robustly. This level of detail is crucial for the CNN to effectively distinguish between the 30 target classes. To train the model on these generated fingerprints, we used the Adam optimizer with a learning rate of 0.001 and a batch size of 32, running for a total of 50 epochs.

We evaluated the model performance using standard classification metrics: Accuracy, Precision, Recall, and F1-score. Given the inherent class imbalance in the dataset, we adopted the Macro Average F1-score as the primary performance indicator, as it provides equal weight to each class regardless of sample size.

Table 1 Training configuration parameters.

Parameter	Value
Image Size	128×128 pixels
Window Size	1.0 s
Window Step	0.5 s
Learning Rate	0.001
Batch Size	32
Epochs	50
Optimizer	Adam

4.2 Training Process Analysis

Figure 5 presents the training and validation curves throughout the learning process. Both accuracy curves exhibit steady improvement, converging at approximately 95%, while the corresponding loss curves demonstrate a consistent decrease and stabilization. The marginal gap between training and validation accuracy suggests that the model generalizes well without significant overfitting, indicating a robust learning process.

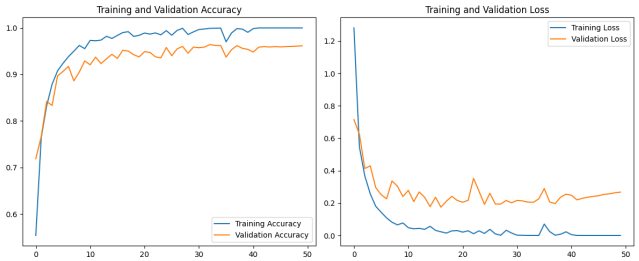


Fig. 5 Training and validation accuracy and loss curves, showing stable convergence at around 95% accuracy and no significant overfitting.

4.3 Overall Performance

The proposed model demonstrates strong performance, achieving an overall accuracy of 96% and a macro average F1-score of 0.94. The confusion matrix presented in Figure 6 corroborates these results, exhibiting a prominent diagonal pattern that indicates high classification accuracy across nearly all attack types. The per-class precision, recall, and F1-scores summarized in Table 2 further confirm the balanced performance of the model across the 30 classes.

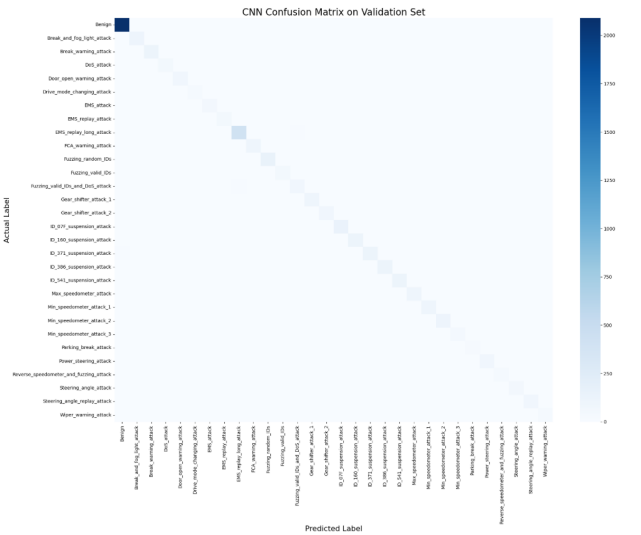


Fig. 6 Confusion matrix of the proposed CNN-based IDS on the CAN-MIRGU dataset, showing the diagonal dominance that reflects high classification accuracy across most attack types.

Table 2 Per-class precision, recall, F1-score, and support values across the 30 classes.

Class	Precision	Recall	F1-score	Support
Benign	0.99	1.00	0.99	2099
Break_and_fog_light_attack	0.91	0.91	0.91	117
Break_warning_attack	0.90	0.95	0.93	124
DoS_attack	0.94	0.98	0.96	51
Door_open_warning_attack	0.97	0.97	0.97	78
Drive_mode_changing_attack	0.95	0.93	0.94	43
EMS_attack	0.89	0.85	0.87	68
EMS_replay_attack	0.91	0.90	0.91	59
EMS_replay_long_attack	0.96	0.96	0.96	428
FCA_warning_attack	0.90	0.90	0.90	115
Fuzzing_random_IDs	0.99	0.98	0.99	151
Fuzzing_valid_IDs	1.00	1.00	1.00	51
Fuzzing_valid_IDs_and_DoS	0.88	0.84	0.86	96
Gear_shifter_attack_1	0.98	0.93	0.96	107
Gear_shifter_attack_2	0.91	0.97	0.94	76
ID_07F_suspension_attack	0.97	0.97	0.97	149
ID_160_suspension_attack	0.97	0.93	0.95	125
ID_371_suspension_attack	0.94	0.91	0.93	131
ID_386_suspension_attack	0.99	0.98	0.99	118
ID_541_suspension_attack	0.94	0.96	0.95	123
Max_speedometer_attack	0.94	0.98	0.96	107
Min_speedometer_attack_1	0.97	0.92	0.94	111
Min_speedometer_attack_2	0.93	0.97	0.95	116
Min_speedometer_attack_3	0.92	0.96	0.94	48
Parking_break_attack	0.83	0.97	0.90	31
Power_steering_attack	1.00	0.94	0.97	85
Reverse_speedometer_and_fuzzing	0.97	1.00	0.99	33
Steering_angle_attack	0.89	0.86	0.88	74
Steering_angle_replay_attack	0.81	0.81	0.81	98
Wiper_warning_attack	0.97	0.84	0.90	44
Accuracy		**0.96**		5056
Macro Avg	**0.94**	**0.94**	**0.94**	5056
Weighted Avg	**0.96**	**0.96**	**0.96**	5056

Beyond these aggregate metrics, several important observations can be made. First, the model achieves consistently high precision and recall across both frequent and infrequent attack classes, suggesting that the weighted training strategy effectively mitigates the impact of class imbalance. Second, even for stealthy attack types such as masquerade, which are designed to mimic legitimate message timing and IDs, the model still achieves non-trivial detection accuracy, indicating that spatial fingerprinting captures subtle payload-level variations that traditional timing- or frequency-based IDSs often miss. Third, the narrow gap between accuracy and macro average F1-score demonstrates that the model does not overly rely on majority classes, but instead generalizes well across the diverse set of 30 classes.

Taken together, these results validate not only the overall effectiveness of our approach for CAN intrusion detection, but also its robustness in handling heterogeneous attack scenarios. This highlights the potential of spatial fingerprinting as a practical foundation for real-world in-vehicle IDS deployment.

5. Discussion

While the overall results in Section 4.3 demonstrate strong performance, a closer examination highlights important nuances regarding the model's behavior on specific attack types, its methodological strengths, and its inherent limitations.

5.1 Analysis of Difficult-to-Detect Classes

Although the overall performance was quite high, the classification report revealed some weaknesses in detecting certain stealthy attacks. For instance, Steering_angle_replay_attack and EMS_replay_attack show lower F1-scores compared to more disruptive attacks such as DoS or fuzzing. This is unsurprising, as masquerade and replay attacks are crafted to mimic legitimate CAN IDs and timing patterns, making them inherently more difficult to detect.

In our model, detection relies heavily on the Hamming distance axis, which captures payload-level variations. However, this feature has limitations in two scenarios: (i) when an attack simply replays messages identical to benign traffic, the resulting fingerprints are indistinguishable; and (ii) when legitimate payloads naturally fluctuate with high entropy, benign variations may be misclassified as malicious. This highlights a semantic gap: the model can capture that a payload has changed, but not its meaning in a physical context. Addressing this gap may require incorporating signal-level semantics and consistency checks, enabling the IDS to identify physically implausible state transitions.

5.2 Strengths and Practical Implications

A major strength of our approach lies in its ability to combine multiple behavioral dimensions of CAN traffic into a unified spatial representation. Unlike traditional IDSs that focus on a single feature (e.g., timing), spatial fingerprinting jointly models message frequency, identifier, and payload stability, allowing the model to capture more subtle anomalies. Moreover, the transformation of traffic into 2D fingerprints enhances interpretability and operational utility. The distinctive visual patterns of different attacks (Figures 3) provide intuitive diagnostic cues for human analysts: a DoS attack appears as a dense vertical line, while fuzzing manifests as scattered horizontal points. This makes the system not only accurate but also interpretable, and it leverages advances in computer vision methods by applying CNNs directly to network security.

5.3 Generalizability and Inherent Limitations

Despite its strengths, the approach faces several challenges in real-world deployment. The reliance on CAN IDs limits cross-vehicle generalizability, which may be alleviated through more vehicle-agnostic representations such as normalized deviation metrics or transfer learning strategies. The segmentation into 1.0-second windows with 0.5-second overlap yields high accuracy but introduces up to 1 second of latency, potentially problematic for safety-critical functions, thus motivating optimization of window size. Finally, as the evaluation is constrained to the CAN-MIRGU dataset, broader validation across diverse vehicles and protocols such as FlexRay and Automotive Ethernet is necessary to validate robustness. Taken together, these factors highlight both the promise of spatial fingerprinting and the key challenges to be addressed before practical adoption.

6. Conclusion

In this paper, we proposed and validated a novel intrusion detection system for CAN bus based on spatial fingerprinting and computer vision. By transforming CAN traffic behavior into 2D images and training a CNN, we developed a model that can effectively classify 30 different network states with 96% accuracy and a 0.94 macro F1-score on the realistic CAN-MIRGU dataset. We demonstrated that this approach, combined with a carefully designed training strategy that addresses class imbalance and underfitting, is a powerful and effective method for in-vehicle security.

Future research will proceed in three main directions: first, improving generalizability by investigating feature abstraction techniques such as using per-ID Z-scores instead of raw values to create a more vehicle-agnostic model, and exploring transfer learning for rapid adaptation to new vehicles; second, exploring advanced models including the use of direct 3D feature classification models such as PointNet to potentially capture more complex geometric patterns without information loss from projection; and third, integrating payload-level semantic analysis to improve the detection of stealthy attacks like masquerade attacks. These enhancements will further strengthen the robustness and applicability of our spatial fingerprinting approach for automotive cybersecurity.

Acknowledgments. This work was partially supported by JST CREST JPMJCR23M4 and JST SPRING JPMJBS2429.

References

[1] N. Alkhatib, L. Achaji, M. Mushtaq, H. Ghauch, and J.-L. Danger, "WIP: AMICA: Attention-based Multi-Identifier model for asynchronous intrusion detection on Controller Area networks," in *Symposium on Vehicles Security and Privacy (VehicleSec)*, 2023.

[2] S. Zhuo, N. Li, and K. Ren, "HistCAN: A real-time CAN IDS with enhanced historical traffic learning capability," in *Symposium on Vehicles Security and Privacy (VehicleSec)*, 2024.

[3] M. E. Verma, R. A. Bridges, M. D. Iannacone, *et al.*, "A comprehensive guide to CAN IDS data and introduction of the ROAD dataset," *PLOS ONE*, vol. 19, no. 1, e0296879, Jan. 2024.

[4] P. Moriano, R. A. Bridges, and M. D. Iannacone, "Detecting CAN Masquerade Attacks with Signal Clustering Similarity," in *Workshop on Automotive and Autonomous Vehicle Security (AutoSec)*, 2022.

[5] P. Agbaje, A. Anjum, A. Mitra, G. Bloom, and H. Olufowobi, "A Framework for Consistent and Repeatable Controller Area Network IDS Evaluation," in *Workshop on Automotive and Autonomous Vehicle Security (AutoSec)*, 2022.

[6] S. Lee, W. Choi, I. Kim, G. Lee, and D. H. Lee, "A Comprehensive Analysis of Datasets for Automotive Intrusion Detection Systems," *Computers, Materials & Continua*, vol. 76, no. 3, pp. 3413–3443, 2023.

[7] S. Rajapaksha, G. Madzudzo, H. Kalutarage, A. Petrovski, and M. O. Al-Kadri, "CAN-MIRGU: A Comprehensive CAN Bus Attack Dataset from Moving Vehicles for Intrusion Detection System Evaluation," in *Symposium on Vehicles Security and Privacy (VehicleSec)*, 2024.

[8] S. Rajapaksha, "Protecting vehicles from cyberattacks: context aware AI-based intrusion detection for vehicle CAN bus security," Ph.D. dissertation, Robert Gordon University, 2024.

[9] S. Rajapaksha, H. Kalutarage, M. O. Al-Kadri, A. Petrovski, and G. Madzudzo, "Improving In-vehicle Networks Intrusion Detection Using On-Device Transfer Learning," in *Symposium on Vehicles Security and Privacy (VehicleSec)*, 2023.