MIDS: Detecting Tampering Attacks on CAN Bus with Bidirectional Mamba

Qiqi Liu†‡, Yuyan Sun*†, Runhan Song†‡, Heng Zhang†‡, Limin Sun†
†Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
‡School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
liuqiqi@iie.ac.cn, sunyuyan@iie.ac.cn, songrunhan@iie.ac.cn,
zhangheng@iie.ac.cn, sunlimin@iie.ac.cn

***Abstract*—The Controller Area Network (CAN) protocol, despite its lack of encryption and authentication mechanisms, remains the primary communication standard among Electronic Control Units (ECUs) in modern vehicles. This widespread adoption renders CAN highly susceptible to security threats. While existing research predominantly addresses message injection attacks, less attention has been given to tampering attacks, which are both more covert and potentially more damaging. To bridge this gap, we propose a Mamba Intrusion Detection System (MIDS), an IDS specifically designed to detect and mitigate tampering attacks. Leveraging cutting-edge technologies, MIDS achieves remarkable performance, surpassing state-of-the-art models across all key evaluation metrics, with an F1 score of 0.9795. Its superior detection capabilities and practical applicability highlight its potential for deployment. We collected over 100 million CAN messages from a Tesla Model 3 to ensure robustness, covering diverse driving conditions and attack scenarios. MIDS and the dataset have been made publicly accessible.[1] [2]**

***Index Terms*—Controller Area Networks, Intrusion Detection Systems, Mamba, Tampering Attacks.**

I. Introduction

A. Background and Motivation

The Controller Area Network (CAN) is a serial communication protocol extensively utilized in automotive and embedded systems. In particular, Electronic Control Units (ECUs) in modern vehicles commonly adopt CAN as the standard for communication.However, the protocol's initial design did not consider network security, omitting modern security mechanisms such as encryption, authentication, and integrity verification [1]. As a result, if an attacker exploits a vulnerability in a specific ECU, they can potentially compromise the entire CAN bus [2], enabling the execution of high-risk attacks.

In recent years, substantial advancements have been made in the detection of injection attacks on CAN, including Denial of Service (DoS), fuzzing, and ID spoofing [3]–[10]. These detection approaches have demonstrated high accuracy in identifying various types of injection attacks. However, tampering attacks, which are less frequently mentioned, represent a covert and sophisticated form of cyberattack in which adversaries unlawfully **alter** system data. As shown in Fig. 1, such attacks typically involve altering critical data (e.g., sensor data or control commands) or maliciously interfering with device behavior (e.g., controlling an ECU to send spoofed commands) which cause severe consequences. Moreover, Tampering attacks exhibit greater stealth compared to injection attacks, primarily because they generally do not alter traffic patterns. This lack of noticeable anomalies in traffic makes traditional Intrusion Detection Systems (IDS) less effective in identifying such attacks [11]. In addition, tampering attacks are considered highly feasible. As highlighted in previous studies [11]–[14], attackers can execute tampering attacks by compromising the ECU, gaining control over the CAN bus, or manipulating data at the gateway adjacent to the sender.

Table I summarizes the performance of models effective in injection attack detection and compares their effectiveness in tampering attack detection. The injection attack dataset is sourced from [15], while the tampering attack dataset is constructed in this study. The results show that conventional machine learning and deep learning models, as well as existing methods for injection attacks, perform poorly in complex tampering scenarios. Therefore, developing advanced feature extraction techniques and deep learning models to capture the subtle characteristics of tampering attacks is essential.

B. Contribution

- This paper introduces Mamba Intrusion Detection System(MIDS), a novel deep learning model specifically designed for detecting CAN tampering attacks. MIDS integrates the Mamba architecture with Convolutional Neural Networks (CNN), marking the **first** application of Mamba to CAN tampering attack detection. This design enables MIDS to effectively handle long-sequence tasks while efficiently extracting local features. Additionally, the model is optimized to capture the high-dimensional temporal characteristics of CAN signals, significantly improving its ability to detect highly covert tampering attacks. Experimental results demonstrate that MIDS outperforms existing methods across key metrics, including accuracy, recall, and F1 score.
- Secondly, we collected a substantial amount of real-world CAN data from a Tesla Model 3, covering diverse driving scenarios such as highway driving, urban driving, parking, and braking. The dataset also simulates tampering attacks of varying types and intensities, including

[1] https://github.com/vrmei/CAN-Tampering
[2] https://drive.google.com/drive/folders/1I9uHpOG8W_Fb9ShoNn6pQB_5Xm83wMww?usp=sharing

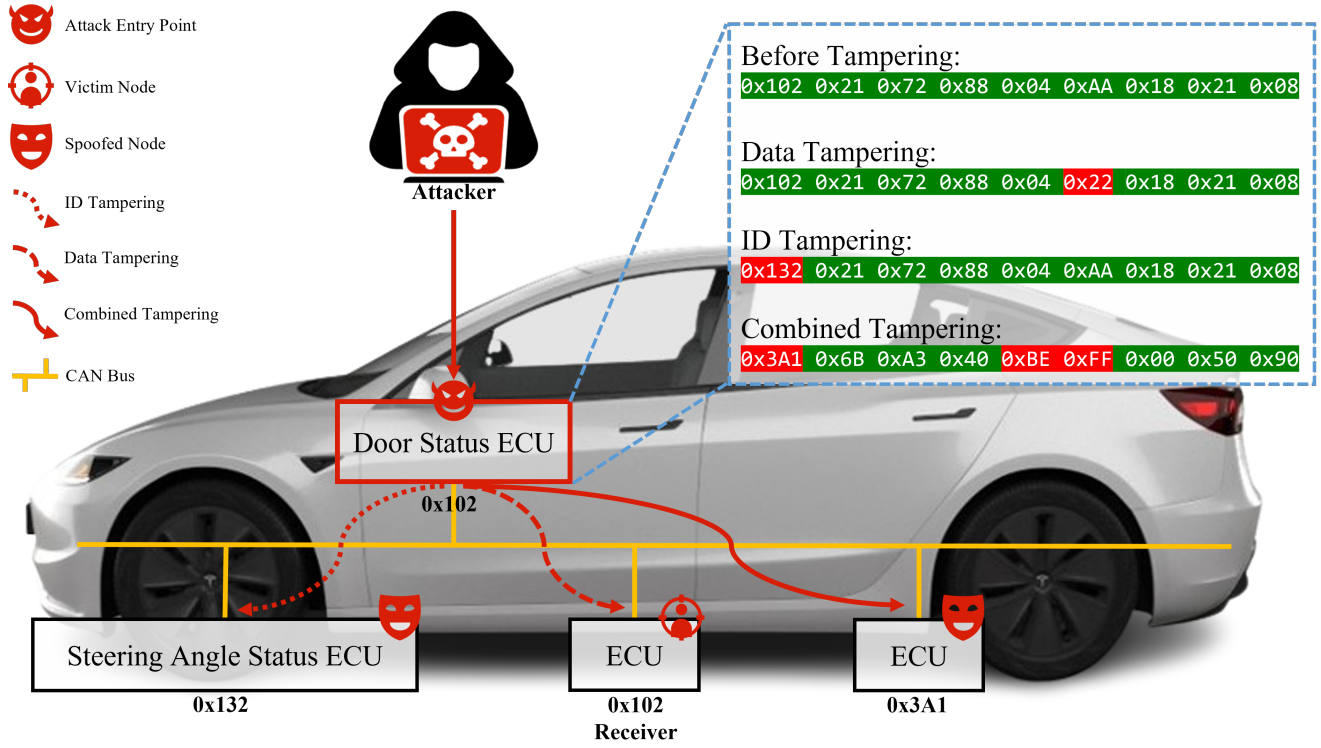

Fig. 1. Overview of tampering attack threat models. Attackers can exploit vulnerabilities in a weak ECU to initiate the entire tampering attack process. The compromised ECU, originally responsible for sending CAN frames with identifier ID 0x102, allows attackers to choose from three types of attacks: (1) Data tampering, (2) ID tampering, and (3) Both tampering. Each type leads to severe consequences but impacts different targets. In (1), the attacker directly tampers with the data field of the CAN frame being sent, which directly affects the receiver of ID 0x102 (shown as the victim node). This may result in scenarios such as a vehicle continuing to drive with its door open. In (2) and (3), the attacker tampers with the frame data to impersonate another ECU (e.g., sending IDs 0x132 or 0x3A1, shown as impersonated nodes), thereby sending malicious data that could cause abnormal steering angles, compromising the entire vehicle system's safety. Unlike injection attacks, tampering attacks generally do not significantly impact traffic distribution because the overall traffic flow in the system remains unchanged.

TABLE I
THE PERFORMANCE OF OUTSTANDING MODEL ARCHITECTURES FOR INJECTION ATTACKS ON TAMPERING ATTACK TASKS

Injection Results(DoS)	Precision	Recall	F1
Attention	99.64%	98.88%	99.26%
CNN [16]	99.94%	99.48%	99.71%
LSTM [17]	99.86%	99.28%	99.57%
MLP	99.91%	99.59%	99.75%
KNN	99.98%	99.98%	99.98%
Injection Results(Fuzzing)	Precision	Recall	F1
Attention	92.25%	87.82%	89.98%
CNN [16]	66.21%	99.41%	79.48%
LSTM [17]	96.94%	96.72%	96.83%
MLP	98.99%	97.68%	98.33%
KNN	79.56%	98.65%	88.08%
Injection Results(Gear)	Precision	Recall	F1
Attention	96.62%	98.31%	97.46%
CNN [16]	99.22%	99.14%	99.18%
LSTM [17]	96.94%	96.72%	96.83%
MLP	99.24%	99.08%	99.16%
KNN	98.51%	99.82%	99.16%
Tampering Results	Precision	Recall	F1
Attention	2.58%	25.00%	4.50%
CNN [16]	17.10%	25.00%	20.31%
LSTM [17]	63.62%	85.32%	69.44%
MLP	53.63%	73.28%	57.65%
KNN	40.61%	30.50%	30.86%

identifier ID tampering, data tampering, and combined tampering attacks. Through this dataset, we analyzed the effects of tampering attacks on vehicle states and control signals, as well as the system's responses under different levels of attack intensity. This comprehensive dataset serves as a foundation for evaluating the effectiveness of tampering attack detection methods.

- Thirdly, due to the diverse CAN bus protocols used by different vehicle models and the limited availability of publicly accessible datasets, there is a lack of generalized IDS research applicable across all vehicle types. To address this gap and advance automotive cybersecurity, we have made our dataset publicly available, including detailed CAN bus data records. This initiative aims to foster collaboration among researchers, promote progress in vehicular network security, and accelerate the development of robust automotive cybersecurity technologies. Moving forward, we plan to expand the dataset by incorporating data from additional vehicle models and communication protocols, such as CAN FD, to enhance the generalization and practical applicability of intrusion detection systems.

TABLE II
SUMMARY OF CAN DATASETS

Name	Years	Rows	Attack Type	Reality	Vehicle Model
Simulated CAN [18]	2016	200,000	Injection	Simulated	N/A
HCRL CAN (OTIDS) [19]	2017	4,613,909	Injection & Benign	Real	KIA SOUL
HCRL Car-Hacking [15]	2018	17,558,462	Injection & Benign	Real	YF Sonata
AEGIS CAN [20]	2019	3,462,015	Benign	Real	Unknown
Bus-Off [21]	2019	189,083,068	Injection & Benign	Simulated	Volvo V40
TU CAN v2 [22]	2019	11,830,305	Injection & Benign	Real	Opel and Renault
ML350 CAN [23]	2019	730,519	Injection & Benign	Real	ML350
ReCAN [24]	2020	38,000,000	Benign	Real	5 Unknown Vehicles
SynCAN [25]	2020	42,958,391	Injection & Benign	Simulated	Unknown
HCRL A&D [26]	2020	8,694,507	Injection & Benign	Real	Avante CN7
Truck CAN Dataset [27]	2021	530,810,616	Benign	Real	Renault Euro VI
ROAD CAN Dataset [28]	2021	Unknown	Injection & tampering	Real & Simulated	Unknown(2010s)
DAGA [29]	2022	200,000,000	Injection & Benign	Real & Simulated	N/A
Ventus [30]	2023	539,657,925	Injection & Benign	Simulated	N/A
CT&T [31]	2023	193,241,081	Injection & Benign	Real & Simulated	Multiple Chevrolet
CrySyS CAN [32]	2023	138,362,148	Injection & tampering	Real & Simulated	Unknown
Ours	2024	108,053,935	Injection & tampering	Real & Simulated	Tesla Model 3

II. Related Work

A. CAN Attack Detection Model

In recent studies, researchers have focused on the CAN protocol's fixed CAN ID frequency and the periodic message transmission by ECUs based on this frequency. [13] proposed a frequency distribution-based intrusion detection method, which utilizes the stability of specific CAN ID frequencies to monitor frequency variations and detect potential attacks. The core idea of this approach lies in identifying abnormal fluctuations in CAN ID frequency as potential indicators of malicious activities. Similarly, [33] [34] introduced entropy-based anomaly detection methods. Overall, most early works relied on statistical features [35]–[40] and protocol specifications [41] for detection.

Although these methods can identify anomalous behaviors to some extent, they have certain limitations, particularly when dealing with complex and previously unseen attack types. For instance, methods based on periodicity and protocol compliance often rely on the fixed patterns of CAN messages. However, attacks in real-world environments are often designed to flexibly evade these rules. Consequently, an increasing number of studies have shifted toward machine learning and deep learning-based techniques in an effort to address these challenges.

Machine learning methods, by building data-driven models, can identify complex attack behaviors without relying on explicit rules. Many studies have adopted statistical classification algorithms, such as Random Forest and Support Vector Machine, to develop intrusion detection systems for CAN buses. For instance, [42] combined Random Forest with the k-Nearest Neighbors algorithm to design an intrusion detector capable of identifying spoofing attacks in intelligent connected vehicles. Compared with traditional methods, machine learning approaches offer greater adaptability and flexibility, enabling them to address various unknown attack types effectively.

Furthermore, with the rapid development of deep learning, its strengths in feature extraction and time-series data processing have provided new approaches for CAN bus intrusion detection. Deep learning algorithms excel at extracting deep-level features from complex and unlabeled data, making them widely used in anomaly detection for CAN data. For instance, [18] utilized Deep Belief Networks (DBN) to classify simulated data and trained Deep Neural Network (DNN) parameters to distinguish between normal and attack messages. [17] employed Long Short-Term Memory (LSTM) networks to achieve anomaly detection by predicting the next value of CAN packets. [15] proposed an anomaly detection method based on Generative Adversarial Networks (GAN), using a generative model of CAN traffic to simulate attack data and improve detection capabilities for unknown attacks.

In some of the latest research, [43] introduced a detection model using ensemble learning, stacking numerous machine learning models to enhance the detection rate of injection attacks. [44] and [40] applied federated learning combined with other deep learning models; the former integrated LSTM, while the latter utilized DNN. Additionally, [45], [46], and [16] adopted CNN to predict CAN data.

For tampering attack detection, a task of particular interest, [47] proposed a security protocol embedding partial Message Authentication Code (MAC) values within data frames and incorporating an incremental counter, significantly reducing bus utilization while improving tampering detection efficiency. [48], on the other hand, employed an improved Isolation Forest algorithm (MS-iForest) that introduced the concept of data quality assessment to enhance sensitivity to local anomalies. This method not only rapidly detects data tampering but also demonstrates outstanding robustness and detection per-

formance across multiple simulated and benchmark datasets.

These two approaches provide effective tampering detection strategies from the protocol and algorithmic perspectives, respectively, yet each has its limitations. For example, [47] faces challenges in high-frequency data scenarios, while [48] requires improvements in handling imbalanced distributions of anomalous data. These studies lay a solid foundation for enhancing the security of CAN buses and provide valuable directions for future research.

B. Dataset

First, we conducted a comprehensive evaluation of existing CAN bus attack datasets, organizing and summarizing the information into Table II.

One of the most representative datasets for injection attacks is the dataset constructed by the Seo team from Seoul National University, which includes four types of attacks. This dataset covers typical CAN bus attack patterns, including DoS, Fuzzing, Replay, and ID Spoofing attacks. These attack types exhibit different characteristics and destructive effects during actual vehicle operation, providing a solid benchmark for research into vehicular network security.

Among the fourteen injection attack datasets investigated, most were constructed based on similar threat model classifications and expanded or adjusted according to real vehicle operation scenarios. For instance, the HCRL team released several injection attack datasets, such as the HCRL CAN (OTIDS) and HCRL Car-Hacking datasets. These datasets feature real-world data collection targeting KIA SOUL and YF Sonata models, respectively, and include both injection attack and normal communication data. Such datasets provide critical references for evaluating CAN bus attack detection algorithms.

Additionally, some datasets were generated through simulation, such as the Simulated CAN and SynCAN datasets. These datasets recreate vehicle operational states and construct injection attacks in a simulated environment to reduce data collection costs. Others, such as the DAGA and CT&T datasets, use a combination of real-world and simulated data. These datasets enhance diversity and applicability by incorporating rich scene simulations and real vehicle testing.

The datasets vary significantly in terms of data volume and coverage. For example, the Truck CAN Dataset, with over 500 million records, is one of the largest CAN bus datasets but only includes normal operational data. In contrast, datasets like HCRL A&D focus on specific attack types and provide more detailed annotations. Furthermore, recently released datasets, such as the Ventus and CrySyS CAN datasets, cover emerging threats like tampering attacks, further enriching the research landscape.

III. Preliminaries

A. CAN Frame Structure

As illustrated in Fig. 2, a CAN standard frame is composed of seven primary fields:

1) Start of Frame (SOF): Marks the beginning of data transmission, signaling the receiver to prepare for incoming data.
2) Identifier Field: Manages priority-based access to the bus, preventing data frame conflicts during transmission.
3) Control Field: Resolves data contention issues on the bus, ensuring proper communication.
4) Data Field: Carries the actual data payload, with a maximum capacity of 8 bytes.
5) Cyclic Redundancy Check (CRC): Ensures data integrity by verifying whether the data frame has been transmitted successfully.
6) Acknowledgment Bit (ACK): Confirms successful reception of the data frame by the receiver.
7) End of Frame (EOF): Signals the completion of the data frame transmission.

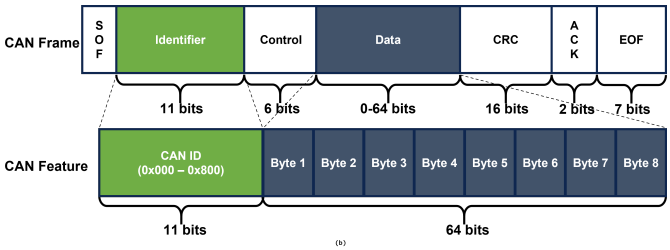

Fig. 2. CAN Frame Structure

B. State-Space Models

State Space Models (SSMs) [49] are a class of mathematical models used to describe the behavior of dynamic systems. They are widely applied in fields such as control theory, signal processing, and computational neuroscience. In deep learning, SSMs are utilized for handling sequential data, particularly addressing long-range dependency problems. By introducing latent state variables, SSMs model the dynamic relationships between inputs and outputs. The core idea is to represent the system's state as a set of latent variables, which are used to capture the mapping between inputs and outputs.

The fundamental equations of a SSM can be expressed as Eq. 1.

$$\begin{aligned} x'(t) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t). \end{aligned} \tag{1}$$

In this context, $x(t)$ represents the latent state of the system, $u(t)$ denotes the input signal, and $y(t)$ refers to the output signal. The matrices A, B, C, and D are the parameters of the model. These parameters describe how the input $u(t)$ influences the system state $x(t)$, and how the state $x(t)$ maps to the output $y(t)$. Specifically, A is the state transition matrix, which governs the dynamic behavior of the system; B is the input matrix, determining how the input affects the system state; and C is the output matrix, which maps the state to the output. If included, D represents the direct transmission matrix, which describes the immediate influence of the input on the output.

To apply SSMs to discrete sequence data, it is typically necessary to discretize the continuous-time model. Using the bilinear discretization method (Tustin method), SSMs can be transformed into a recursive form, allowing the latent state at each time step to be computed via state update equations. Furthermore, SSMs can also be represented in a convolutional form, enabling optimization through efficient convolutional computations. Specifically, SSMs can be expressed as the convolution of the input signal by unrolling the state transition equations. This formulation facilitates computation using efficient algorithms such as the Fast Fourier Transform (FFT):

$$y = K * u. \tag{2}$$

As shown in Eq. 2, K represents the convolution kernel of the SSM, and $*$ denotes the convolution operation. Through this convolution kernel, the model can efficiently process sequential data during both the training and inference stages.

C. Mamba

Mamba [50] is a novel Selective State Space Model (SSSM) that combines the structured state space model with selective state space mechanisms. Its primary objective is to address the issue of parameter rigidity in conventional SSMs. To overcome this limitation, Mamba introduces a dynamic selective mechanism. By treating the parameters of the SSM as functions of the input, the model dynamically adjusts its parameters based on the input context. This enables dynamic processing and selective forgetting of information, facilitating context-aware operations.

Specifically, Mamba introduces input-dependent parameterization for the parameters A, B, and C, allowing them to be dynamically adjusted based on the input, rather than remaining fixed. For instance, the parameters A, B, and C in the Selective State Space Model can be expressed as Eq. 3.

$$\begin{aligned} A(x(t)) &= f_A(x(t)), \\ B(x(t)) &= f_B(x(t)), \\ C(x(t)) &= f_C(x(t)), \end{aligned} \tag{3}$$

where, f_A, f_B, and f_C are functions applied to the input $x(t)$. These functions are typically implemented as linear transformations or more sophisticated nonlinear transformations, such as neural network layers.

In addition, the selective mechanism can selectively propagate information by adjusting the state update process. For example, the model's state update can be controlled through a gating mechanism, as follows:

$$g_t = \sigma(W_g x(t) + b_g) h_t = (1 - g_t) h_{t-1} + g_t x(t), \tag{4}$$

where, g_t is a gating variable that determines, through the gating mechanism, whether the current input $x(t)$ should be propagated to the state $h(t)$. This enables selective memory of the input, allowing the model to dynamically decide which information to retain or discard based on the input context.

IV. Methodology

A. Overall Framework

In this paper, we propose a novel IDS architecture based on the bidirectional Mamba model, as illustrated in Fig. 3. This architecture is designed to effectively detect various tampering attacks and improve the system's capability to handle complex attack patterns with enhanced efficiency and robustness.

The raw input data consists of the ID and Data fields extracted from CAN bus messages. Due to the short length of a single CAN message, contextual information is limited, which may reduce detection accuracy. To address this, we group every 100 messages into a long-sequence format, enriching contextual information and enhancing robustness in detection. The input data is then divided into two parts—ID and Data—each processed separately to maximize feature extraction based on their unique characteristics. The ID field, containing discrete identifiers, represents different types of CAN signals, while the Data field encodes temporal sequences of CAN signals.

The Data field is processed using a one-dimensional CNN to extract temporal features such as abrupt changes, periodic fluctuations, and local trends. The convolutional layers efficiently capture these temporal dependencies through localized operations, which are particularly effective for high-frequency CAN signals. This approach not only preserves critical information but also reduces computational complexity, making it suitable for real-time processing.

In contrast, the ID field is processed through an MLP-based embedding layer, which maps discrete IDs into dense vector representations. This embedding process captures latent relationships between IDs, such as differentiating control signals from sensor data, and reduces computational complexity. By encoding these discrete IDs into a low-dimensional continuous space, the model leverages semantic information for enhanced feature representation.

The extracted features from both fields are processed further by the bidirectional Mamba module, which incorporates Selective State Space Models (SSM) to capture long-term dependencies. The Forward SSM Block models the influence of past inputs on the current state, while the Backward SSM Block captures the impact of future data on current decisions. This bidirectional design enables the model to achieve a comprehensive understanding of sequential data, overcoming the limitations of unidirectional models. The SSMs, with their parameter selectivity and state transition mechanisms, offer improved numerical stability and flexibility in handling long-sequence data.

The outputs from the bidirectional Mamba module are fused through a weighted integration mechanism, generating a unified global feature representation. This fusion combines forward and backward sequence information, enhancing the model's ability to detect tampering attacks. Finally, the fused features are passed to a multi-class classifier that distinguishes between normal signals, data tampering attacks, ID tampering attacks, and combined attacks.

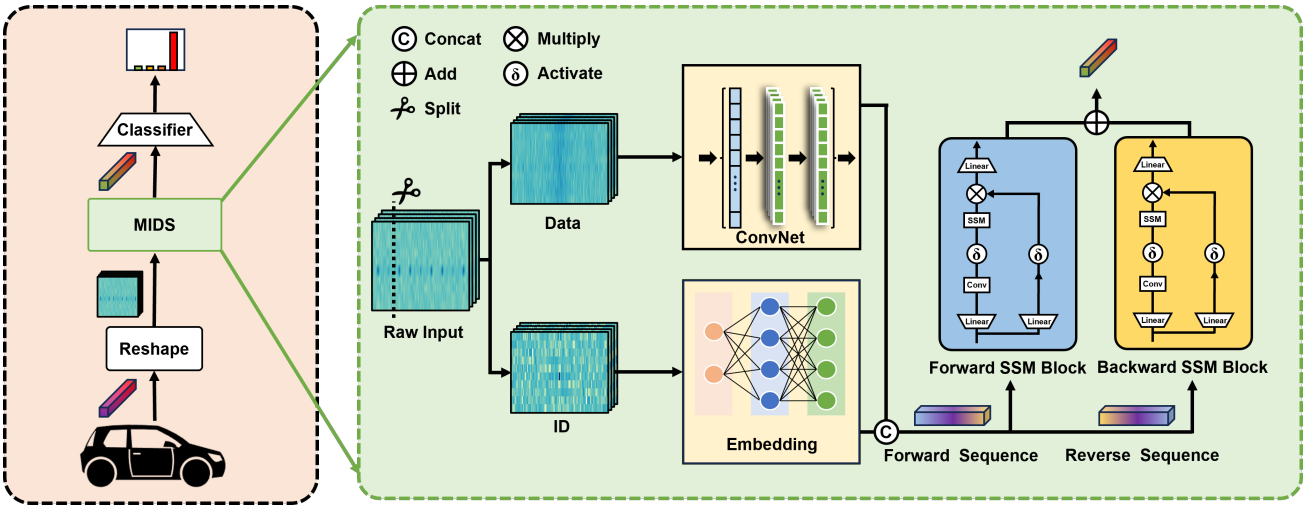

Fig. 3. MIDS Model Architecture: Designed for detecting tampering attacks on the CAN bus. The raw input data is split into the Data field and ID field, which are processed separately for feature extraction. Temporal features from the Data field are extracted using a Convolutional Neural Network (CNN), while the ID field is mapped to a continuous vector space through an embedding layer to capture latent relationships between IDs. The extracted features are fused and fed into forward and backward SSM modules, which model forward and backward sequence dependencies, respectively. A weighted mechanism is used to integrate bidirectional information. Finally, the fused features are passed to a classifier to predict attack types, such as data tampering or ID tampering.

This architecture effectively integrates fine-grained data processing, bidirectional Mamba modeling, and global feature fusion, delivering high detection accuracy with low computational overhead. The innovative combination of SSMs and the bidirectional Mamba model ensures robust handling of long-term dependencies and high-frequency data. As a result, the proposed IDS framework provides a reliable and efficient solution for detecting complex CAN bus attacks in real-time scenarios.

B. Dataset Analysis and Construction

We collected approximately 16 hours of vehicle operation data. This dataset consists of two main categories: normal operation data (free-attack) and tampering attack data. To comprehensively simulate diverse real-world scenarios, the data covers various operating modes and conditions, with detailed processing and annotation applied to ensure its quality and relevance.

For tampering attacks, we divided the data into three main scenarios based on different vehicle operating states: standby mode, low-speed driving, and high-speed driving. These three scenarios cover the typical states of a vehicle transitioning from stationary to dynamic operation. In the data visualization analysis (as shown in Fig. 4), we present the distribution of CAN bus data across these three scenarios: Standby, Low-speed, and High-speed.

The three-dimensional distribution plots reveal distinct differences in data characteristics across various scenarios. In standby mode, the distribution predominantly exhibits blue-purple hues, indicating low data values with minimal fluctuations, forming a dense and stable pattern. This reflects the characteristics of CAN bus signals in the standby state, where signal variations are negligible, and the system operates at a low activity level.

In contrast, during low-speed driving, the distribution shifts towards green hues, signifying increased data values and greater fluctuations. Although the distribution shows some regularity, it is less uniform than in standby mode. For high-speed driving, the distribution becomes more complex, with colors transitioning to yellow-green tones, representing significantly higher data values and substantial increases in fluctuation amplitude. These changes reflect the more dynamic and intense signal variations characteristic of high-speed operation.

These distribution differences provide valuable insights for analysis and help identify potential anomalies under various operating conditions. The stable data from the standby mode serves as a baseline for normal operation, while the more complex fluctuations observed in low-speed and high-speed scenarios simulate real-world conditions. By encompassing diverse scenarios, the IDS can effectively detect complex attack patterns and adapt to dynamic environments.

Moreover, the color variations in the data distributions underscore the importance of multi-scenario coverage in enhancing IDS robustness. This comprehensive approach ensures the system's adaptability across a wide range of operational states, enabling reliable performance in real-world applications.

After collecting the operational data, we processed it with multiple factors to simulate various tampering attack scenarios, ensuring the diversity of the experimental dataset. Specifically, two CAN IDs—0x102 and 0x3D9—were selected as primary targets for tampering. These IDs correspond to critical vehicle operation signals, where tampering could significantly impact vehicle behavior.

To simulate tampering attacks of varying intensity, we defined tampering frequencies as uneven values (2, 5, 10, 25, 50, and 100), based on common CAN bus signal transmission rates. This approach allows a signal to be tampered with

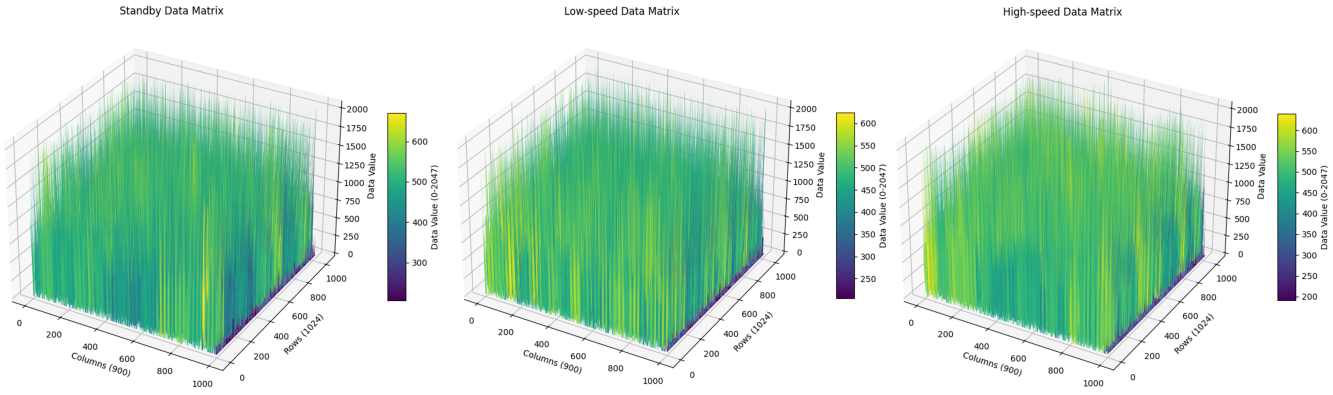

Fig. 4. The data distribution of the vehicle under different operating states is shown in the figure, with data originating from three distinct scenarios: standby state, low-speed driving state, and high-speed driving state (arranged from left to right). Each plot represents a batch of CAN bus messages (102,400 messages) from the experiment, approximately equivalent to one minute of vehicle operation, illustrating the distribution characteristics of data values under each state.It is evident from the figure that the predominant color transitions from low-value blue-purple tones to high-value yellow-green tones, reflecting the trend of data value changes corresponding to different operating states.

only after a specific number of occurrences, enabling precise control over attack intensity. Low-frequency tampering represents covert attack scenarios, while high-frequency tampering mimics direct and destructive attacks.

Furthermore, we designed three distinct tampering strategies to simulate diverse attack methods, each targeting different aspects of the CAN messages:

1) CAN ID Modification: This strategy alters only the CAN ID, simulating attackers impersonating legitimate signal sources to deceive the system.
2) Data Field Modification: This strategy modifies only the data field, simulating the impact of corrupted signal content on vehicle commands.
3) Combined Modification: This strategy alters both the CAN ID and data field simultaneously, representing more complex and concealed attack scenarios. Each strategy represents a specific attack pattern, ensuring comprehensive coverage of potential tampering methods.

In summary, our tampering dataset incorporates diverse scenarios, frequencies, and strategies to create robust testing conditions. Detailed design considerations are further described in Algorithm 1. These multi-factor processing steps ensure the dataset's broad applicability and provide strong support for the optimization and validation of the IDS.

Algorithm 1 Dataset building algorithm

Strategy: $\mathcal{S} = \{\mathcal{S}_{\text{ID}}, \mathcal{S}_{\text{Data}}, \mathcal{S}_{\text{ID+Data}}\}$
Frequency: $F = \{2, 5, 10, 25, 50, 100\}$
ID: $I = \{0x102, 0x3D9, (0x102, 0x3D9)\}$
DataFile: D

for $i \in I$ **do**
  $CAN_{\text{sig}} \leftarrow$ GET_CAN_SIG(D, i)
  for $f \in F$ **do**
    for $s \in \mathcal{S}$ **do**
      $D \leftarrow$ MODIFY_SIG$(CAN_{\text{sig}}, \text{frequency} = f, \text{modify} = s)$
    end for
  end for
end for
SAVE_DATASET(D)

function MODIFY_SIG(CAN_{sig}, f, s):
for $(index, frame) \in$ enumerate(CAN_{sig}) **do**
  if $(index + 1)\%f = 0$ **then**
    if $s = \mathcal{S}_{\text{ID}}$ **then**
      $frame.ID \leftarrow$ TAMPER$(frame.ID)$
    else if $s = \mathcal{S}_{\text{Data}}$ **then**
      $frame.data \leftarrow$ TAMPER$(frame.data)$
    else if $s = \mathcal{S}_{\text{ID+Data}}$ **then**
      $frame.ID \leftarrow$ TAMPER$(frame.ID)$
      $frame.data \leftarrow$ TAMPER$(frame.data)$
    end if
  end if
end for

V. Evaluation

A. Experiment Setup

Our experimental environment is illustrated in Fig. 5. The primary equipment includes a Tesla Model 3 test bed, a Peak CAN converter, and a ZL-23-008 physical sensor.

Using fuzzing techniques, we identified two critical CAN signals: the door status signal with ID 0x102 and the steering angle signal with ID 0x132. Manipulating these signals can result in unsafe scenarios, such as allowing the doors to open while driving or displaying incorrect steering prompts. Our experiments were primarily designed to investigate these two CAN messages. We not only found detailed descriptions of these signals in the Tesla Model 3's Database CAN(DBC) file but also demonstrated their potential impacts on the vehicle through controlled experiments in a laboratory environment. During the model training phase, we applied rigorous pre-

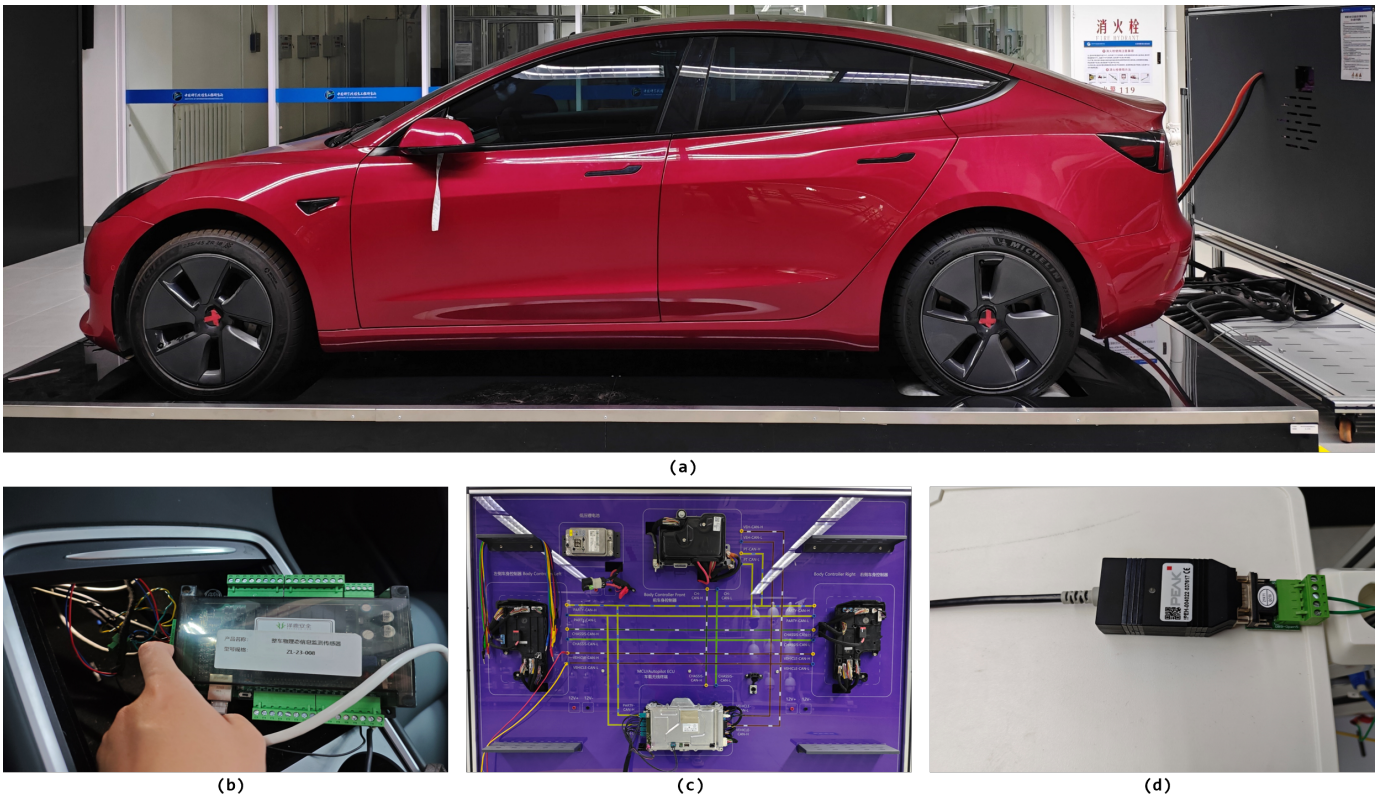

Fig. 5. (a) An overview of the test bed, where a monitor positioned in front of the car simulates various driving scenarios. The physical vehicle, a Tesla Model 3, has its rear wheels suspended and free to spin on the test bed. (b) A sensor signal reception device that collects physical data, including rear wheel speed and steering angle. The collected data are transmitted via a network to the host system of the monitor in (a) to emulate vehicle motion. This component of the work was conducted by another research group and is not detailed here. (c) Key ECU components and exposed network interfaces extracted from the Tesla Model 3, including the low-voltage battery, body controllers, and the in-car wireless terminal. (d) A USBCAN converter connected to the exposed CAN interface in (c), along with a laptop, is used to collect and process the data.

processing to the CAN data and carefully configured the training setup to ensure model stability and efficient loss convergence, with all hyperparameters used during training detailed in Table III. The batch size was set to 1024 for each training iteration, and a 5-fold cross-validation approach was employed to reduce performance bias caused by data distribution differences, with each fold comprising 50 epochs to ensure the model sufficiently captured the data characteristics. Given that this is a four-class classification task with significant class imbalance in the dataset, we implemented a dynamic weighting strategy. This strategy assigned weights to each class based on their sample sizes, preventing the model from overfitting to majority classes and improving its ability to detect minority classes. For optimization, we utilized the Adam optimizer and integrated a dynamic learning rate scheduler to ensure stable convergence in the later stages of training. To comprehensively evaluate the model's performance, we adopted Macro Average metrics as the final evaluation standard. These metrics included Macro Average Precision (assessing classification accuracy across all classes), Macro Average Recall (measuring the model's ability to retrieve samples from all classes), Macro Average F1 Score (the harmonic mean of Precision and Recall, reflecting overall performance), and Accuracy (evaluating overall prediction accuracy across all classes).At the final epoch of each fold, we recorded these four metrics and calculated their average across the five folds to obtain the final evaluation of the model's performance.

B. *Evaluating the Proposed MIDS Model*

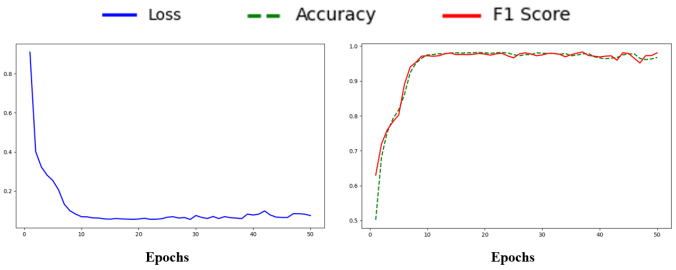

Fig. 6. Metrics of the MIDS Training Process.

Fig. 6 illustrates the variations in Loss, Accuracy, and F1-score during the training process of the proposed MIDS model. Meanwhile, Fig. 7 shows the confusion matrices obtained from 5-fold cross-validation, providing a detailed evaluation of classification performance across different attack classes.Fig. 8 provides a comprehensive comparison of the proposed MIDS

Fig. 7. Confusion matrix, where Class 0 represents normal data, Class 1 represents ID spoofing attacks, Class 2 represents data spoofing attacks, and Class 3 represents combined spoofing attacks.

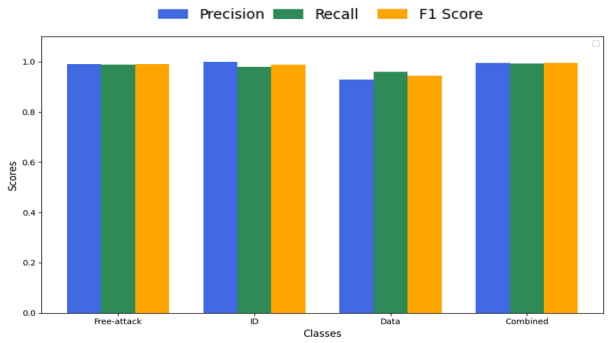

Fig. 8. Class-wise Performance Metrics Visualization: The bar chart illustrates the Precision, Recall, and F1 Score for each class ('Free-attack', 'ID', 'Data', 'Combined').

model's performance across different attack classes. It demonstrates the model's effectiveness by showcasing the key evaluation metrics, emphasizing its robustness and consistency in handling various classification tasks.

C. Comparison Experiment

In addition to the proposed MIDS model, we conducted experiments using models based on Attention mechanisms and other architectures. However, these alternative approaches demonstrated significantly poorer performance compared to the Mamba-based model. The superiority of the proposed model is clearly evident in Table IV, which provides a comparative analysis of the results.

D. Ablation Study

This section presents the design and results of the MIDS ablation experiments.

We conducted controlled experiments on the key components of the MIDS architecture by progressively removing or replacing certain modules and recording their impact on model performance. Experiments A1-A9 represent different ablation conditions, such as removing specific modules, modifying

TABLE III
MIDS HYPERPARAMETERS

Hyperparameters	Value
embedding_output_dim	256
mamba_hidden_state_dim	8
mamba_convolution_dim	4
mamba_feature_expansion_factor	2
conv_kernel_size	3
conv_padding_size	1
epoch	50
batch size	1024
k-fold	5
optimizer	Adam

TABLE IV
COMPARISON EXPERIMENT RESULTS

#	Description	Precision	Recall	F1	Accuracy
A1	**MIDS**	**97.88%**	**98.05%**	**97.95%**	**98.56%**
A2	Attention	79.67%	91.89%	83.64%	85.81%
A3	Attention+PE	15.54%	37.17%	18.36%	23.47%
A4	CNN+Attention	2.66%	25.00%	4.80%	10.63%
A5	CNN+LSTM	76.15%	89.37%	80.38%	82.94%
A6	CNN	66.99%	88.40%	72.46%	74.54%

module configurations, or introducing alternative designs. By comparing the results of the complete MIDS architecture with those under various ablation conditions, we aim to validate the contribution of each component to the overall performance. Specifically, we focus on performance metrics including Macro-Averaged Precision, Macro-Averaged Recall, Macro-Averaged F1-Score, and Accuracy.

In each ablation experiment, we strictly adhered to the same hyperparameter settings and experimental strategies as those used in the main MIDS experiment. The final results, as shown in Table V, were obtained by calculating the average of these four metrics across five folds.

The data in the table clearly demonstrate that the current architecture design achieves optimal results. Each component of MIDS plays a critical role, highlighting the rationality of the current architecture and the effectiveness of the collaboration among its components.

VI. CONCLUSION

This study introduces MIDS, a novel deep learning-based approach for detecting tampering attacks on the CAN bus. By integrating the bidirectional Mamba architecture with CNNs, MIDS effectively captures both local and long-range dependencies in CAN signals, achieving superior performance with an F1-score of 0.9795.

We collected over 100 million CAN messages from a Tesla Model 3, simulating diverse scenarios and tampering attacks, and made the dataset publicly available to support further research. Extensive experiments and ablation studies confirmed the robustness and efficiency of MIDS, demonstrating its

TABLE V
ABLATION STUDY RESULTS

#	Description	Macro Precision	Macro Recall	Macro F1	Accuracy
A1	**MIDS**	**97.88%**	**98.05%**	**97.95%**	**98.56%**
A2	MIDS_More_Parameters	96.97%	97.98%	97.38%	98.09%
A3	MIDS_Fewer_Parameters	96.70%	97.88%	97.26%	98.06%
A4	Unidirectional_Mamba	96.31%	97.84%	97.02%	97.87%
A5	MIDS_Fewer_Conv	35.87%	52.49%	37.54%	40.88%
A6	MIDS_No_Conv	41.25%	54.39%	42.38%	46.00%
A7	MIDS_Frequency_ID_Embedding	54.40%	65.81%	57.10%	68.87%
A8	MIDS_No_ID_Embedding	8.42%	25.00%	10.98%	33.70%
A9	Only_Mamba	19.99%	25.89%	12.89%	34.12%

potential for real-world deployment. Future work will focus on expanding datasets and adapting MIDS for new protocols like CAN FD to address evolving security challenges.

In future research, we plan to extend and refine the MIDS in the following aspects: First, we aim to explore the applicability of MIDS to more complex communication protocols, such as CAN FD and automotive Ethernet, to address the increasingly diverse communication demands in modern automotive networks. Second, we intend to validate the generalization capability of MIDS through experiments involving different vehicle models and multi-vehicle scenarios, thereby enhancing its adaptability to various vehicles and driving environments. Additionally, we will further optimize the model architecture to reduce computational complexity, ensuring that its performance meets the requirements of real-time onboard detection. Finally, to address potential emerging threats, we will integrate federated learning and transfer learning techniques, enabling MIDS to quickly adapt to and detect unknown threats in scenarios with limited data sharing. These efforts aim to provide stronger support for securing intelligent automotive networks.

ACKNOWLEDGMENT

We extend our sincere gratitude to Li Li, Zhiyuan Wang and Diehan Song for their significant contributions and constructive feedback, which have substantially enriched the quality of this work.

REFERENCES

[1] C. Lin and A. L. Sangiovanni-Vincentelli, "Cyber-security for the controller area network (CAN) communication protocol," in *2012 ASE International Conference on Cyber Security, Alexandria, VA, USA, December 14-16, 2012*. IEEE Computer Society, 2012, pp. 1–7. [Online]. Available: https://doi.org/10.1109/CyberSecurity.2012.7

[2] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, and T. Kohno, "Comprehensive experimental analyses of automotive attack surfaces," in *20th USENIX Security Symposium (USENIX Security 11)*. San Francisco, CA: USENIX Association, Aug. 2011. [Online]. Available: https://www.usenix.org/conference/usenix-security-11/comprehensive-experimental-analyses-automotive-attack-surfaces

[3] C. Miller and C. Valasek, "A survey of remote automotive attack surfaces," in *IOActive Labs Research*, 2014.

[4] M. H. Khan, A. R. Javed, Z. Iqbal, M. Asim, and A. I. Awad, "Divacan: Detecting in-vehicle intrusion attacks on a controller area network using ensemble learning," *Comput. Secur.*, vol. 139, p. 103712, 2024. [Online]. Available: https://doi.org/10.1016/j.cose.2024.103712

[5] H. Zhang, K. Zeng, and S. Lin, "Federated graph neural network for fast anomaly detection in controller area networks," *IEEE Trans. Inf. Forensics Secur.*, vol. 18, pp. 1566–1579, 2023. [Online]. Available: https://doi.org/10.1109/TIFS.2023.3240291

[6] H. M. Song and H. K. Kim., "Can signal extraction and translation dataset." [Online]. Available: https://ocslab.hksecurity.net/Datasets/can-signal-extraction-and-translation-dataset

[7] M. Jedh, L. B. Othmane, N. Ahmed, and B. K. Bhargava, "Detection of message injection attacks onto the CAN bus using similarity of successive messages-sequence graphs," *CoRR*, vol. abs/2104.03763, 2021. [Online]. Available: https://arxiv.org/abs/2104.03763

[8] R. Islam, R. U. D. Refat, S. M. Yerram, and H. Malik, "Graph-based intrusion detection system for controller area networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 3, pp. 1727–1736, 2022. [Online]. Available: https://doi.org/10.1109/TITS.2020.3025685

[9] 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 *2016 International Conference on Information Networking, ICOIN 2016, Kota Kinabalu, Malaysia, January 13-15, 2016*. IEEE Computer Society, 2016, pp. 63–68. [Online]. Available: https://doi.org/10.1109/ICOIN.2016.7427089

[10] B. Groza and P. Murvay, "Efficient intrusion detection with bloom filtering in controller area networks," *IEEE Trans. Inf. Forensics Secur.*, vol. 14, no. 4, pp. 1037–1051, 2019. [Online]. Available: https://doi.org/10.1109/TIFS.2018.2869351

[11] Q. Wang and S. Sawhney, "Vecure: A practical security framework to protect the CAN bus of vehicles," in *4th International Conference on the Internet of Things, IOT 2014, Cambridge, MA, USA, October 6-8, 2014*. IEEE, 2014, pp. 13–18. [Online]. Available: https://doi.org/10.1109/IOT.2014.7030108

[12] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, and S. Savage, "Experimental security analysis of a modern automobile," in *2010 IEEE Symposium on Security and Privacy*, 2010, pp. 447–462.

[13] C. Miller and C. Valasek, "Adventures in automotive networks and control units."

[14] A. Gazdag, C. Ferenczi, and L. Buttyán, "Development of a man-in-the-middle attack device for the can bus," in *Proceedings of the 1st Conference on Information Technology and Data Science Debrecen*, 2020, p. 115–130.

[15] E. Seo, H. M. Song, and H. K. Kim, "GIDS: GAN based intrusion detection system for in-vehicle network," *CoRR*, vol. abs/1907.07377, 2019. [Online]. Available: http://arxiv.org/abs/1907.07377

[16] A. R. Javed, S. Ur Rehman, M. U. Khan, M. Alazab, and T. Reddy, "Canintelliids: Detecting in-vehicle intrusion attacks on a controller area network using cnn and attention-based gru," *IEEE transactions on network science and engineering*, vol. 8, no. 2, pp. 1456–1466, 2021.

[17] A. Taylor, S. P. Leblanc, and N. Japkowicz, "Anomaly detection in automobile control network data with long short-term memory networks," in *2016 IEEE International Conference on Data Science and Advanced Analytics, DSAA 2016, Montreal, QC, Canada,*

October 17-19, 2016. IEEE, 2016, pp. 130–139. [Online]. Available: https://doi.org/10.1109/DSAA.2016.20

[18] K. J.-W. Kang M-J, "Intrusion detection system using deep neural network for in-vehicle network security." *PLoS ONE 11(6): e0155781*. [Online]. Available: https://doi.org/10.1371/journal.pone.0155781

[19] H. Lee, S. H. Jeong, and H. K. Kim, "OTIDS: A novel intrusion detection system for in-vehicle network by using remote frame," in *15th Annual Conference on Privacy, Security and Trust, PST 2017, Calgary, AB, Canada, August 28-30, 2017*. IEEE Computer Society, 2017, pp. 57–66. [Online]. Available: https://doi.org/10.1109/PST.2017.00017

[20] C. Kaiser, A. Festl, G. Pucher, M. Fellmann, and A. Stocker, "The vehicle data value chain as a lightweight model to describe digital vehicle services," in *Proceedings of the 15th International Conference on Web Information Systems and Technologies, WEBIST 2019, Vienna, Austria, September 18-20, 2019*, A. Bozzon, F. J. D. Mayo, and J. Filipe, Eds. ScitePress, 2019, pp. 68–79. [Online]. Available: https://doi.org/10.5220/0008113200680079

[21] D. Stabili and M. Marchetti, "Detection of missing can messages through inter-arrival time analysis," in *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*, 2019, pp. 1–7.

[22] G. Dupont, A. Lekidis, J. I. den Hartog, and S. Etalle, "Automotive controller area network (can) bus intrusion dataset v2." in *4TU.Centre for Research Data*, 2019, p. 14. [Online]. Available: https://doi.org/10.4121/UUID:B74B4928-C377-4585-9432-2004DFA20A5D

[23] M. Sami, "Intrusion detection in can bus," 2019. [Online]. Available: https://dx.doi.org/10.21227/24m9-a446

[24] M. Zago, S. Longari, A. Tricarico, M. Carminati, M. Gil Pérez, G. Martínez Pérez, and S. Zanero, "Recan – dataset for reverse engineering of controller area networks," *Data in Brief*, vol. 29, p. 105149, 2020.

[25] M. Hanselmann, T. Strauss, K. Dormann, and H. Ulmer, "Canet: An unsupervised intrusion detection system for high dimensional can bus data," *IEEE Access*, vol. 8, pp. 58 194–58 205, 2020.

[26] H. Kang, B. I. Kwak, Y. H. Lee, H. Lee, H. Lee, and H. K. Kim, "Car hacking and defense competition on in-vehicle network," in *Workshop on Automotive and Autonomous Vehicle Security (AutoSec)*, vol. 2021, 2021, p. 25.

[27] U. of Turku, "Can bus dataset collected from a heavy-duty truck," 5 2021, university of Turku. [Online]. Available: https://doi.org/10.23729/3160254e-85e9-4268-a636-5b3e54091706

[28] S. C. Hollifield, M. E. Verma, M. D. Iannacone, R. A. Bridges, B. Kay, and F. L. Combs, "Poster: Real ornl automotive dynamometer (road) can intrusion dataset."

[29] D. Stabili, L. Ferretti, M. Andreolini, and M. Marchetti, "DAGA: detecting attacks to in-vehicle networks via n-gram analysis," *IEEE Trans. Veh. Technol.*, vol. 71, no. 11, pp. 11 540–11 554, 2022. [Online]. Available: https://doi.org/10.1109/TVT.2022.3190721

[30] F. Pollicino, D. Stabili, and M. Marchetti, "Performance comparison of timing-based anomaly detectors for controller area network: A reproducible study," *ACM Trans. Cyber-Phys. Syst.*, vol. 8, no. 2, May 2024. [Online]. Available: https://doi.org/10.1145/3604913

[31] B. Lampe and W. Meng, "can-train-and-test: A new can intrusion detection dataset," in *2023 IEEE 98th Vehicular Technology Conference (VTC2023-Fall)*, 2023, pp. 1–7.

[32] A. Gazdag, R. Ferenc, and L. Buttyán, "Crysys dataset of can traffic logs containing fabrication and masquerade attacks," vol. 10, 2023. [Online]. Available: https://doi.org/10.1038/s41597-023-02716-9

[33] M. Müter and N. Asaj, "Entropy-based anomaly detection for in-vehicle networks," in *IEEE Intelligent Vehicles Symposium (IV), 2011, Baden-Baden, Germany, June 5-9, 2011*. IEEE, 2011, pp. 1110–1115. [Online]. Available: https://doi.org/10.1109/IVS.2011.5940552

[34] M. Marchetti, D. Stabili, A. Guido, and M. Colajanni, "Evaluation of anomaly detection for in-vehicle networks through information-theoretic algorithms," in *2nd IEEE International Forum on Research and Technologies for Society and Industry Leveraging a better tomorrow, RTSI 2016, Bologna, Italy, September 7-9, 2016*. IEEE, 2016, pp. 1–6. [Online]. Available: https://doi.org/10.1109/RTSI.2016.7740627

[35] 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 *2016 International Conference on Information Networking, ICOIN 2016, Kota Kinabalu, Malaysia, January 13-15, 2016*. IEEE Computer Society, 2016, pp. 63–68. [Online]. Available: https://doi.org/10.1109/ICOIN.2016.7427089

[36] P. Murvay and B. Groza, "Source identification using signal characteristics in controller area networks," *IEEE Signal Process. Lett.*, vol. 21, no. 4, pp. 395–399, 2014. [Online]. Available: https://doi.org/10.1109/LSP.2014.2304139

[37] W. Choi, H. J. Jo, S. Woo, J. Y. Chun, J. Park, and D. H. Lee, "Identifying ecus using inimitable characteristics of signals in controller area networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 6, pp. 4757–4770, 2018. [Online]. Available: https://doi.org/10.1109/TVT.2018.2810232

[38] W. Choi, K. Joo, H. J. Jo, M. C. Park, and D. H. Lee, "Voltageids: Low-level communication characteristics for automotive intrusion detection system," *IEEE Trans. Inf. Forensics Secur.*, vol. 13, no. 8, pp. 2114–2129, 2018. [Online]. Available: https://doi.org/10.1109/TIFS.2018.2812149

[39] K.-T. Cho and K. G. Shin, "Fingerprinting electronic control units for vehicle intrusion detection," in *25th USENIX Security Symposium (USENIX Security 16)*. Austin, TX: USENIX Association, Aug. 2016, pp. 911–927. [Online]. Available: https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/cho

[40] H. Sun, M. Sun, J. Weng, and Z. Liu, "Analysis of id sequences similarity using dtw in intrusion detection for can bus," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 10, pp. 10 426–10 441, 2022.

[41] U. E. Larson, D. K. Nilsson, and E. Jonsson, "An approach to specification-based attack detection for in-vehicle networks," in *2008 IEEE Intelligent Vehicles Symposium*, 2008, pp. 220–225.

[42] M. L. Han, B. I. Kwak, and H. K. Kim, "TOW-IDS: intrusion detection system based on three overlapped wavelets for automotive ethernet," *IEEE Trans. Inf. Forensics Secur.*, vol. 18, pp. 411–422, 2023. [Online]. Available: https://doi.org/10.1109/TIFS.2022.3221893

[43] M. H. Khan, A. R. Javed, Z. Iqbal, M. Asim, and A. I. Awad, "Divacan: Detecting in-vehicle intrusion attacks on a controller area network using ensemble learning," *Computers & Security*, vol. 139, p. 103712, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404824000130

[44] T. Yu, G. Hua, H. Wang, J. Yang, and J. Hu, "Federated-lstm based network intrusion detection method for intelligent connected vehicles," in *ICC 2022-IEEE International Conference on Communications*. IEEE, 2022, pp. 4324–4329.

[45] S. Anbalagan, G. Raja, S. Gurumoorthy, R. D. Suresh, and K. Dev, "Iids: Intelligent intrusion detection system for sustainable development in autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 12, pp. 15 866–15 875, 2023.

[46] M. D. Hossain, H. Inoue, H. Ochiai, D. Fall, and Y. Kadobayashi, "An effective in-vehicle can bus intrusion detection system using cnn deep learning approach," in *GLOBECOM 2020-2020 IEEE global communications conference*. IEEE, 2020, pp. 1–6.

[47] S. Araki, A. Tashiro, K. Kakizaki, and S. Uehara, "A study on a secure protocol against tampering and replay attacks focused on data field of can," in *2017 IEEE International Conference on Consumer Electronics - Taiwan (ICCE-TW)*, 2017, pp. 247–248.

[48] X. Duan, H. Yan, D. Tian, J. Zhou, J. Su, and W. Hao, "In-vehicle can bus tampering attacks detection for connected and autonomous vehicles using an improved isolation forest method," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 2, pp. 2122–2134, 2023.

[49] A. Gu, K. Goel, and C. Ré, "Efficiently modeling long sequences with structured state spaces," in *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. [Online]. Available: https://openreview.net/forum?id=uYLFoz1vlAC

[50] A. Gu and T. Dao, "Mamba: Linear-time sequence modeling with selective state spaces," *CoRR*, vol. abs/2312.00752, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2312.00752