MIDS: Detecting Tampering Attacks on CAN Bus with Bidirectional Mamba

Anonymous Author(s)
Anonymous Institution

Abstract

The Controller Area Network (CAN) protocol is the primary communication standard for Electronic Control Units (ECUs) in modern vehicles. However, its lack of encryption and authentication mechanisms exposes it to various security vulnerabilities. In this paper, we introduce a novel type of ID tampering attack, distinct from traditional data tampering attack and well-known injection attacks. This new attack targets the CAN ID field, providing greater flexibility and a broader attack surface. Its covert nature makes it as difficult to detect as data modification tampering attacks, while potentially being even more damaging. To address this security gap, we propose the Mamba Intrusion Detection System (MIDS), designed to detect not only our new ID modification tampering attacks but also traditional data modification and injection attacks. To evaluate the effectiveness of MIDS, we collected over 100 million CAN messages from a Tesla Model 3, simulating a variety of attack scenarios, along with other publicly available datasets. Leveraging bi-directional detection technologies, MIDS outperforms state-of-the-art models by over 6% on the Tesla Model 3 dataset. Furthermore, it achieves exceptional F1 scores of 96.94% and 99.61% for detecting tampering and injection attacks, respectively, on public datasets.

1 Introduction

1.1 Background and Challenge

The Controller Area Network (CAN) is a serial communication protocol widely used in automotive and embedded systems. For example, CAN was commonly adopted as the standard communication protocol of Electronic Control Units (ECUs) in modern vehicles. However, the protocol's original design did not account for security, resulting in the absence of modern security mechanisms such as encryption, authentication, and integrity verification [30, 51]. Consequently, if an attacker exploits a vulnerability in a specific ECU, they could potentially compromise the entire CAN bus [3], allowing the execution of high-risk attacks.

In recent years, significant advancements have been made in detecting data tampering attacks [2, 7] and injection attacks [11, 20, 25, 33, 42, 43, 54] in the Controller Area Network (CAN). While these methods have demonstrated effectiveness, attackers can exploit a new class of attacks involving ID tampering, which pose a greater threat, provided that they have gained access to the network gateway. As a novel form of tampering, the ID tampering attack alters the CAN message's ID field, potentially compromising the integrity of all ECUs within the system. When traffic passes through the gateway, it may be intercepted and modified by attackers.

In this paper, we adopt a novel threat model that encompasses ID tampering, data tampering, and combined tampering attacks. As illustrated in Fig. 1, these attacks typically involve altering critical data (e.g. sensor readings or control commands) , in order to maliciously manipulate device behavior (e.g., directing an ECU to send spoofed commands). Such manipulations can lead to severe and potentially catastrophic consequences.

Previous research has predominantly focused on injection attacks, with limited attention given to tampering attacks, especially ID tampering attacks. Notably, there is a lack of comprehensive studies specifically addressing ID tampering, which remains an underexplored area of research. Our proposed threat model, which includes both ID and data tampering attacks, presents unique challenges in detection. These types of attacks are particularly difficult to identify because they generally do not alter traffic patterns—tampering does not add or remove data packets. As a result, traditional Intrusion Detection Systems (IDS), which mainly rely on traffic anomaly detection, may ineffective in identifying these attacks [7].

Furthermore, tampering attacks, including our novel ID tampering attack, are considered highly feasible. As highlighted in prior studies [9], attackers can carry out such tampering by compromising the gateway adjacent to the sender and then intercept and modify CAN messages, thereby manipulating data on the CAN bus. In summary, it is crucial to develop advanced feature extraction techniques and deep

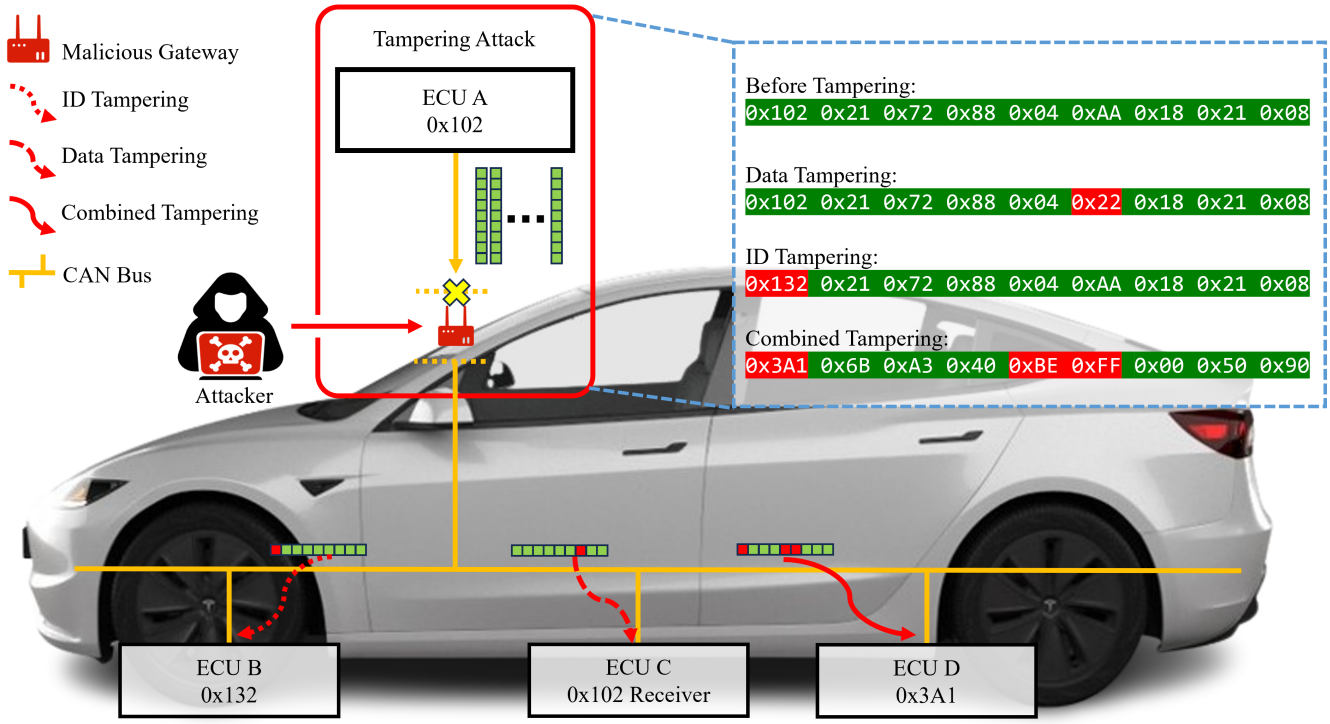

Figure 1: Overview of tampering attacks threat model. Attackers can exploit vulnerabilities in a weak gateway to initiate the entire tampering attack process. The compromised ECU, originally responsible for sending CAN frames with identifier ID 0x102, allows attackers to choose from three types of attacks: (1) Data tampering, (2) ID tampering, and (3) Combined 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 affects the receiver of ID 0x102 (shown as the ECU C). 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 ECU B and D), 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.

learning models capable of capturing the subtle, often undetectable characteristics of tampering attacks, including both ID and data tampering.

1.2 Contribution

In response to our new threat model, we propose the Mamba Intrusion Detection System (MIDS), a novel solution designed to detect both ID and data tampering attacks. We evaluate MIDS using an extensive dataset collected from a real-world Tesla Model 3 and publicly available data from the Internet, which includes instances of ID and data tampering attacks, as well as injection attacks. Both the dataset and the MIDS are made publicly available to promote transparency and facilitate further research.[1] [2] Our contributions are detailed as follows:

- Firstly, we introduce a novel threat model, as shown in Fig. 1, which encompasses both our new ID tampering attacks and data tampering attacks. Compared to previous works, these combined attacks are covert and difficult to detect because they do not alter traffic patterns. Additionally, ID tampering attacks have a broader attack surface, making them more flexible and potentially more harmful. The new threat model provides a realistic simulation of attacker behavior, presenting significant challenges for Intrusion Detection Systems (IDS) in detecting such attacks.

- Secondly, we propose MIDS, a novel deep learning model designed to detect both ID and data tampering attacks. MIDS integrates the Mamba architecture with Convolutional Neural Networks (CNNs), marking the first application of Mamba for detecting CAN tampering attacks. This design allows MIDS to efficiently handle long-sequence tasks while extracting local features with high precision. Additionally, the model is optimized to

[1] https://anonymous.4open.science/r/MIDS-796F-anonymous/

[2] https://drive.google.com/drive/folders/1I9uHpOG8W_Fb9ShoNn6pQB_5Xm83wMww?usp=sharing

capture the high-dimensional temporal characteristics of CAN signals, significantly improving its ability to detect covert tampering attacks. Experimental results demonstrate that MIDS outperforms existing methods across key performance metrics, including accuracy, recall, and F1 score.

- Thirdly, to evaluate the effectiveness of MIDS in detecting tampering attacks, we collected a comprehensive dataset of real-world CAN data from a Tesla Model 3, covering a range of driving scenarios, including highway driving, urban driving, parking, and braking. This dataset addresses the limitation of previous studies, such as the CrySyS and ROAD datasets, which did not include ID tampering attacks. The dataset contains ID tampering, data tampering, and combined tampering attacks of varying intensities. Additionally, we incorporated four publicly available datasets to further assess MIDS's ability to detect traditional injection attacks. To foster future research and collaboration, both our dataset and MIDS are publicly released, providing a solid foundation for advancing vehicular cybersecurity.

2 Related Work

2.1 CAN Attack Detection Model

In recent studies, researchers have focused on the CAN protocol's fixed ID frequency and the periodic message transmission by ECUs based on this frequency. [32] 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, [31, 35] introduced entropy-based anomaly detection methods. Overall, most early works relied on statistical features [4–6, 34, 44, 48] and protocol specifications [28] 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, [14] 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 capabilities in feature extraction and time-series data processing have introduced new approaches for CAN bus intrusion detection. Deep learning algorithms excel at extracting complex, high-level features from unlabeled data, making them highly effective for anomaly detection in CAN traffic. For instance, [24] applied Deep Belief Networks (DBN) to classify simulated data and trained Deep Neural Network (DNN) parameters to differentiate between normal and attack messages. [50] leveraged Long Short-Term Memory (LSTM) networks to detect anomalies by predicting the next value in CAN packets. [40] proposed an anomaly detection method based on Generative Adversarial Networks (GAN), using a generative model of CAN traffic to simulate attack data and enhance detection capabilities for previously unseen attacks. In some of the latest research, [26] introduced a detection model using ensemble learning, stacking numerous machine learning models to enhance the detection rate of injection attacks. [52] and [48] applied federated learning combined with other deep learning models; the former integrated LSTM, while the latter utilized DNN. Additionally, [1], [18], and [19] adopted CNN to predict CAN data.

For data tampering attacks detection, [2] proposed a security protocol that embeds partial Message Authentication Code (MAC) values within data frames and incorporates an incremental counter, significantly reducing bus utilization while improving tampering detection efficiency. In contrast, [7] employed an enhanced Isolation Forest algorithm (MS-iForest), introducing the concept of data quality assessment to improve sensitivity to local anomalies. This method not only enables rapid detection of data tampering but also demonstrates remarkable robustness and detection performance across multiple simulated and benchmark datasets.

However, while both approaches provide effective tampering detection strategies, they focus exclusively on data tampering attacks and each has notable limitations. Specifically, [2] struggles with high-frequency data scenarios, where the protocol's efficiency is compromised. On the other hand, [7] requires further improvements to handle imbalanced distributions of anomalous data, which negatively impacts its detection performance. Despite these limitations, these studies lay a solid foundation for enhancing CAN bus security and offer valuable insights for future research directions.

Table 1: Summary of CAN datasets

Name	Years	Rows	Attack Type	Reality	Vehicle Model
Simulated CAN [24]	2016	200,000	Injection	Simulated	N/A
HCRL CAN (OTIDS) [29]	2017	4,613,909	Injection & Benign	Real	KIA SOUL
HCRL Car-Hacking [40]	2018	17,558,462	Injection & Benign	Real	YF Sonata
AEGIS CAN [22]	2019	3,462,015	Benign	Real	Unknown
Bus-Off [47]	2019	189,083,068	Injection & Benign	Simulated	Volvo V40
TU CAN v2 [8]	2019	11,830,305	Injection & Benign	Real	Opel and Renault
ML350 CAN [38]	2019	730,519	Injection & Benign	Real	ML350
ReCAN [53]	2020	38,000,000	Benign	Real	5 Unknown Vehicles
SynCAN [15]	2020	42,958,391	Injection & Benign	Simulated	Unknown
HCRL A&D [23]	2020	8,694,507	Injection & Benign	Real	Avante CN7
Truck CAN Dataset [36]	2021	530,810,616	Benign	Real	Renault Euro VI
ROAD CAN Dataset [17]	2021	Unknown	Injection & Tampering	Real & Simulated	Unknown(2010s)
DAGA [46]	2022	200,000,000	Injection & Benign	Real & Simulated	N/A
Ventus [37]	2023	539,657,925	Injection & Benign	Simulated	N/A
CT&T [27]	2023	193,241,081	Injection & Benign	Real & Simulated	Multiple Chevrolet
CrySyS CAN [10]	2023	138,362,148	Injection & Tampering	Real & Simulated	Unknown
Ours	2024	108,053,935	Injection & Tampering	Real & Simulated	Tesla Model 3

2.2 CAN Attack Dataset

We conducted a comprehensive survey of existing CAN bus attack datasets, organizing and summarizing them in Table 1. Notably, these datasets are primarily designed for injection attacks rather than tampering attacks. Although ROAD and CrySyS include tampering attacks, they only focus on data tampering scenarios and tampering approaches are relatively monotonous, and the vehicles used in these datasets are not recent enough. This highlights the need for the development of a dataset specifically focused on tampering attacks.

As for injection attacks, 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: DoS, Fuzzing, Replay, and ID Spoofing. These attack types exhibit distinct characteristics and destructive effects during actual vehicle operation, providing a solid benchmark for research on vehicular network security.

Among the fourteen injection attack datasets we investigated, most were constructed based on similar threat model classifications and 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 were collected from real-world data targeting the KIA SOUL and YF Sonata models, respectively, and include both injection attack data and normal communication data. These datasets serve as crucial 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 replicate vehicle operational states and simulate injection attacks to reduce data collection costs. Other datasets, such as DAGA and CT&T, combine real-world and simulated data, enhancing diversity and applicability by integrating rich scene simulations with actual 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, although it only includes normal operational data. In contrast, datasets like HCRL A&D focus on specific attack types and provide more detailed annotations. Furthermore, newly released datasets, such as the Ventus and CrySyS CAN datasets, cover emerging threats, including tampering attacks, further enriching the research landscape.

3 Preliminaries

3.1 CAN Frame Structure

As illustrated in Fig. 2, a standard CAN frame consists of seven primary fields: Start of Frame (SOF): Marks the beginning of data transmission. Identifier Field: Manages priority-based access to the bus, preventing data frame conflicts during transmission. Control Field: Resolves data contention issues on the bus. Data Field: Carries the actual data payload, with a maximum capacity of 8 bytes. Cyclic Redundancy Check (CRC): Ensures data integrity. Acknowledgment Bit (ACK):

Confirms successful reception of the data frame by the receiver. End of Frame (EOF): Signals the completion of the data frame transmission. Tampering attacks typically modify the Identifier Field and Data Field, while injection attacks are usually executed by inserting a new CAN frame into the bus.

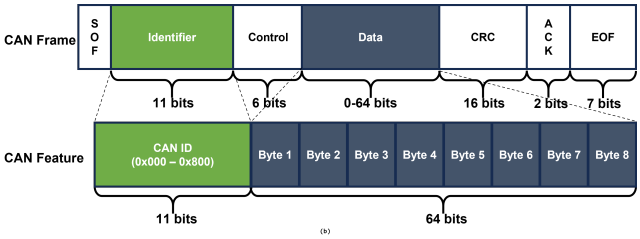

Figure 2: CAN frame structure

3.2 State-Space Models

State Space Models (SSMs) [13] are mathematical frameworks used to describe the behavior of dynamic systems. They are widely applied in areas such as control theory, signal processing, and computational neuroscience. In deep learning, SSMs are employed to handle sequential data, particularly to address long-range dependency issues. By introducing latent state variables, SSMs model the dynamic relationships between inputs and outputs. The core idea of SSMs is to represent the system's state through a set of latent variables, which capture the mapping between inputs and outputs.

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

The fundamental equations of SSMs can be expressed as Eq. 1 where $x(t)$ 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.

It is typically necessary to discretize the continuous-time model(e.g. SSMs) before applying them to discrete sequence data. 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. Using this convolution kernel, the model efficiently processes sequential data during both the training and inference stages.

3.3 Mamba

Mamba [12] is a novel Selective State Space Model (SSSM) that combines the structured SSMs 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 where f_A, f_B, and f_C are functions applied to the input $x(t)$.

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

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, described as Eq. 4:

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

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.

4 Methodology

4.1 Overall Framework

In this paper, we proposed MIDS, a novel IDS architecture based on the bidirectional Mamba model. MIDS is enhanced

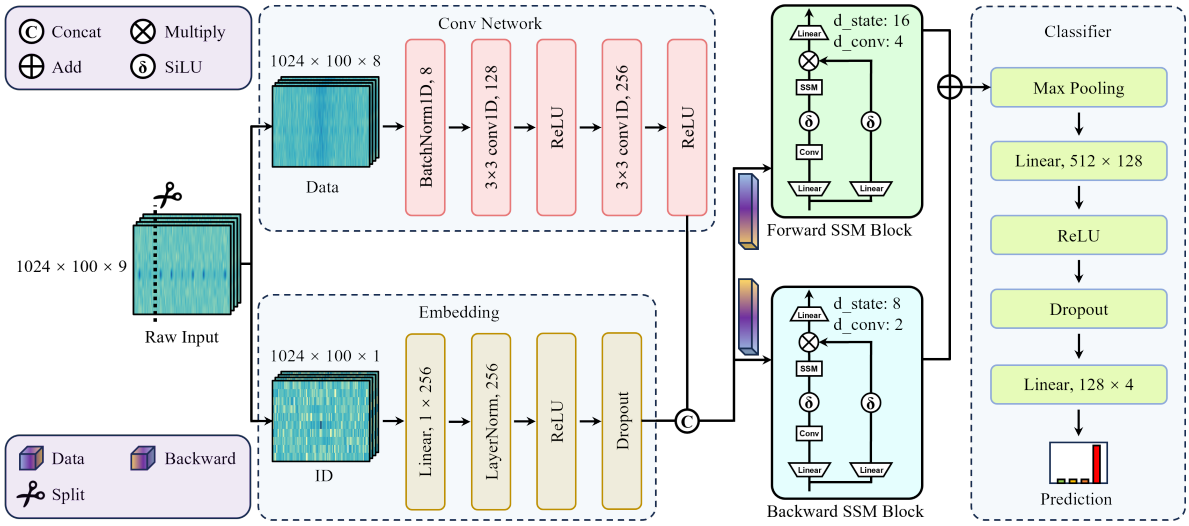

Figure 3: Model architecture of MIDS

to effectively detect various tampering attacks including ID and data tampering while ensuring efficiency and robustness against traditional injection attacks.

The design of MIDS follows a key insight: attacks on the CAN bus often manipulate either the identifier field (ID tampering) or the data field (data tampering), while in practice, these two dimensions of information exhibit complementary properties. The ID field encodes communication semantics and priority, whereas the data field carries the payload that reflects temporal correlations of sensor or control signals. Treating them jointly allows the IDS to capture both semantic anomalies in message scheduling and statistical inconsistencies in data dynamics.

Based on this principle, MIDS first decouples the input into two branches: the ID field and the data field. For the ID field, an embedding layer is employed to project discrete identifiers into a continuous space, thereby revealing latent structural relationships across IDs that cannot be captured by raw categorical representations. For the data field, temporal dependencies are extracted via convolutional layers, which are particularly effective in modeling local patterns and short-term fluctuations commonly observed in automotive signals.

After feature extraction, both representations are fused and processed by forward and backward Mamba-based state space modules. This bidirectional design reflects the observation that tampering may disrupt both past-consistent dynamics (detectable in forward modeling) and future-anticipatory dynamics (detectable in backward modeling). By combining the two, MIDS achieves a holistic understanding of sequence evolution, improving robustness against sophisticated adversaries who attempt to maintain local consistency while injecting subtle modifications.

Finally, a weighted fusion mechanism integrates forward and backward outputs to emphasize the most discriminative temporal cues. The classifier then outputs the attack category, distinguishing benign traffic from ID tampering, data tampering, or hybrid tampering. This framework ensures that MIDS not only achieves high detection accuracy but also remains computationally efficient, which is crucial for deployment in real-time automotive environments.

As shown in Fig. 3, MIDS is a dual-stream model that processes the ID and Data fields separately due to their distinct semantic roles. The input data is divided into ID and Data parts and processed independently to maximize feature extraction based on their unique characteristics. The ID field, which contains discrete identifiers, represents different types of CAN signals, while the Data field encodes the temporal sequences of these signals. Since the raw ID and Data fields extracted from a single CAN frame are often too short to capture sufficient contextual information, which may negatively affect detection accuracy, we address this limitation by grouping every 100 messages into a longer sequence format. This approach not only enriches the contextual information but also enhances the robustness of detection.

The Data field is processed using 1D-CNN to capture temporal features such as abrupt changes, periodic fluctuations, and local trends. The convolutional layers efficiently model 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 using an MLP-based embedding layer, which maps discrete IDs into dense vector representations. This embedding process captures latent relationships between IDs, such as distinguishing 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 further processed by the bidirectional Mamba module, which incorporates SSMs 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, provide 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 both 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, which distinguishes between normal signals, data tampering attacks, ID tampering attacks, and combined attacks.

MIDS effectively integrates dual-stream architecture, fine-grained data processing, bidirectional Mamba modeling, and global feature fusion, delivering high detection accuracy with low computational overhead. The Dual-stream architecture allows MIDS to detect ID and data abnormalities. While 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 offers a reliable and efficient solution for detecting complex CAN bus attacks in real-time scenarios.

4.2 Dataset Design and Analysis

Our dataset is divided into two categories as shown in the top section of Fig. 5: tampering datasets and injection datasets. Each category contains publicly available datasets: ROAD and CrySyS for data tampering attacks, and OTIDS and CT&T for traditional injection attacks, as listed in Table 1. Our new tampering attacks dataset which added ID tampering attacks is shown in the bottom section of Fig. 5. We collected CAN bus data from a Tesla Model 3, spending approximately 16 hours gathering vehicle operation data. To ensure the quality and relevance of the data, we designed various driving scenarios, tampering strategies, and tampering intervals, with detailed processing and annotation. As a result, we believe that this tampering attacks dataset effectively simulates diverse real-world scenarios in a comprehensive manner.

We designed three typical driving scenarios: standby mode, low-speed driving, and high-speed driving, and collected corresponding data for analysis. In the data visualization analysis (as shown in Fig. 4), we present the distribution of CAN bus data across these three scenarios. The three-dimensional distribution plots reveal distinct differences in data characteristics across the scenarios.

In standby mode, the data 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 standby mode, where signal variations are minimal and the system operates at a low activity level. In contrast, during low-speed driving, the distribution shifts toward green hues, signifying higher 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 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 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, this dataset can effectively assist IDS in detecting complex attack patterns and adapting to dynamic environments.Furthermore, the color variations in the data distributions emphasize the importance of multi-scenario coverage in enhancing IDS robustness. This multi-scenario dataset ensures the MIDS's adaptability across a wide range of operational states, thereby enabling reliable performance in real-world applications.

After collecting the tampering attacks dataset, we processed it with different tampering strategies and intervals to simulate various attack scenarios, ensuring the diversity of the experimental dataset. We designed three distinct tampering strategies to simulate different 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.Specifically, during the execution of different tampering strategies, CAN IDs 0x102 and 0x132 were selected as primary targets for tampering. These IDs correspond to critical

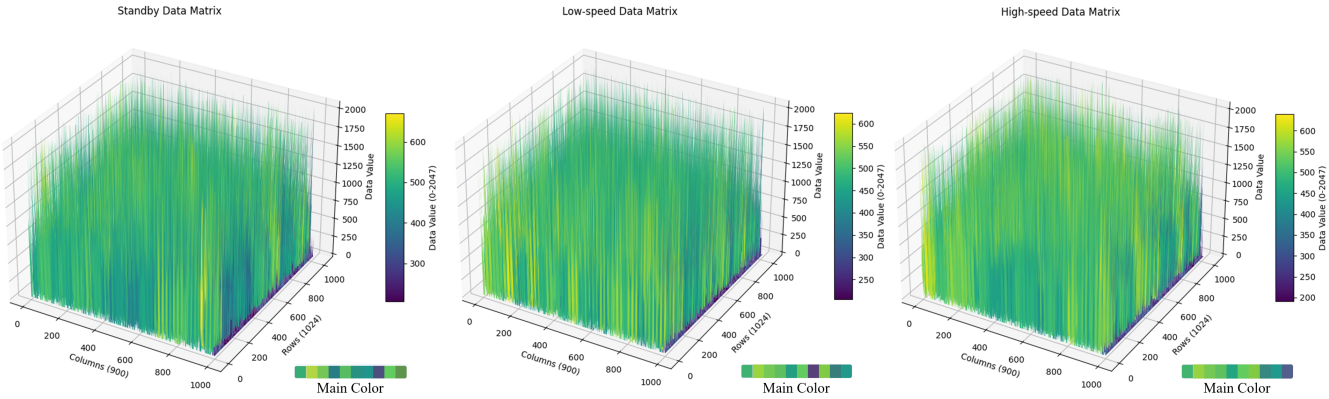

Figure 4: Data distribution visualization. The data originates from three distinct scenarios: standby state, low-speed driving state, and high-speed driving state (arranged from left to right). Each plot represents a batch of CAN bus messages (102,400 messages) from the experiment, approximately equivalent 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.

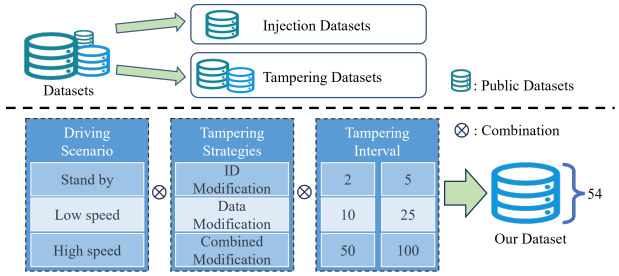

Figure 5: Dataset design

vehicle operation signals, and tampering with them could significantly affect vehicle behavior.

The tampering intervals reflect the attack intensity. We defined the tampering intervals as 2, 5, 10, 25, 50, and 100, based on typical CAN bus signal transmission rates. This approach allows a signal to be tampered with only after a specific number of occurrences, providing precise control over attack intensity. Low-frequency tampering represents covert attack scenarios, while high-frequency tampering simulates more direct and destructive attacks.

In summary, our tampering dataset incorporates a variety of scenarios, frequencies, and strategies to create robust testing conditions. These multi-factor processing steps ensure the broad applicability of the dataset and provide strong support for the optimization and validation of the IDS.

5 Evaluation

5.1 Experiment Setup

5.1.1 Experiment Environment

The experimental hardware environment for our tampering attacks dataset is illustrated in Fig. 6. The primary equipment includes a Tesla Model 3 testbed, a Peak CAN converter, a ZL-23-008 physical sensor, one Nvidia H100 GPU, and two Nvidia RTX4090 GPUs.

The software environment is based on the Ubuntu 22.04 operating system, with a data acquisition tool named TSMaster employed for the collection and recording of CAN bus frames.

5.1.2 Baseline Introduction

We first compare MIDS with state-of-the-art (SOTA) models in our tampering attacks dataset. The SOTA models we consider are as follows:

1. GIDS. GIDS utilizes a GAN model, where the generator is responsible for generating simulated attack data, while the discriminator is used to distinguish between normal traffic and attack traffic. GIDS provides real-time protection for intrusion detection in in-vehicle networks.

2. CanShield. CANShield focuses on raw signal-level information from CAN bus data. It utilizes CNN and LSTM to detect anomalies and intrusions. Experiments show that CANShield effectively identifies and classifies intrusions in CAN networks at the signal level.

3. CanBus-IDS. This model uses a Convolutional Adversarial Autoencoder for semi-supervised learning. It con-

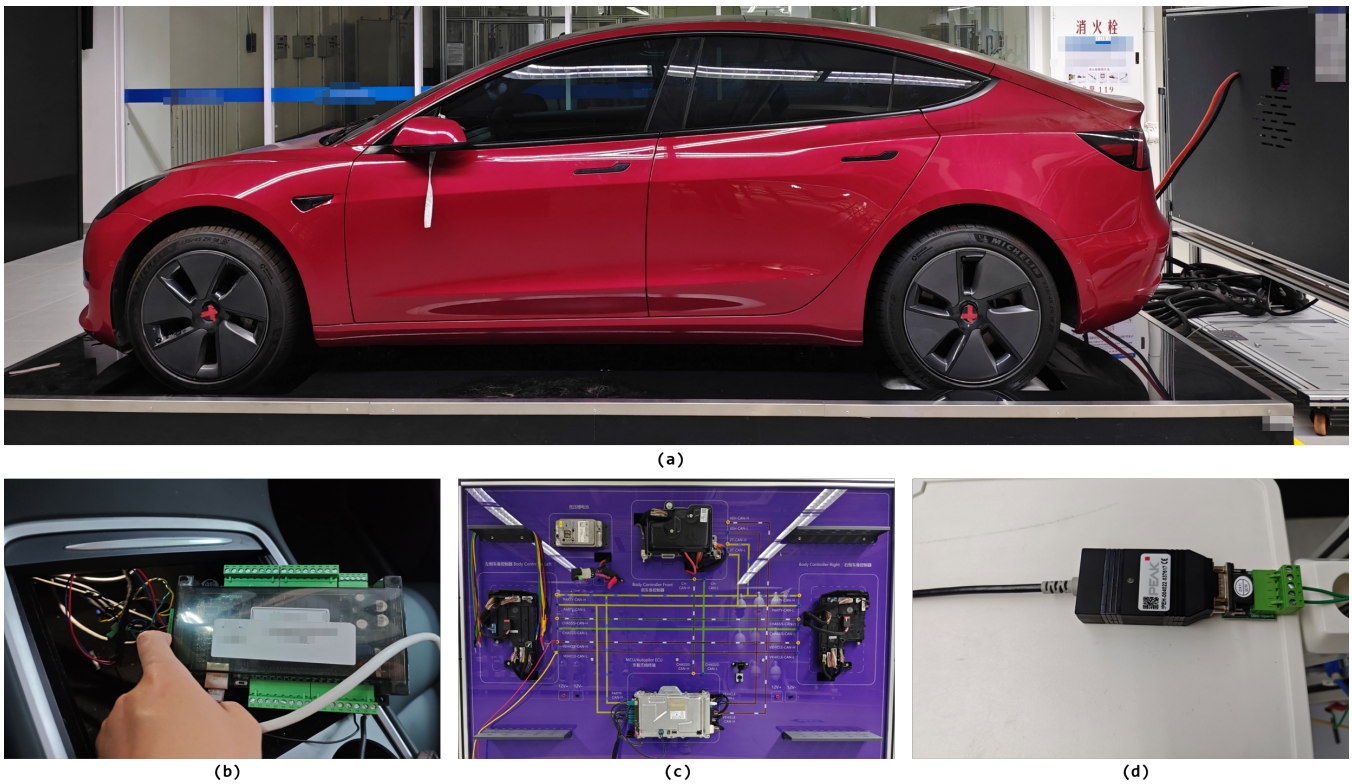

Figure 6: (a) An overview of the test bed, where a monitor positioned in front of the car simulates various driving scenarios. The physical vehicle, a Tesla Model 3, has its rear wheels suspended and free to spin on the test bed. (b) A sensor signal reception device that collects physical data, including 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.

sists of two main components: the encoder and the decoder. The encoder is a series of convolutional layers that process the input data and transform it into a lower-dimensional latent space. The decoder takes the compressed latent space representation and reconstructs it back into the original data format. Adversarial training is incorporated to enhance the model's ability to detect anomalies.

4. DCNN. DCNN utilizes a CNN model to detect attack traffic in the in-vehicle network. Experimental results show that the CNN-based intrusion detection system performs excellently in identifying attack traffic within the in-vehicle network, with high accuracy and low false positive rates. DCNN is capable of effectively detecting both known and unknown attack patterns.

5. CANTransfer. CANTransfer uses one-shot learning techniques in combination with CNN and LSTM, enabling the model to process spatial information while also modeling temporal dependencies, thereby enhancing the detection performance of network attacks. At the same time, one-shot learning allows the system to efficiently learn and detect intrusions across different attack patterns.

6. CanTransformer. CanTransformer utilizes the attention mechanism for intrusion detection in the CAN bus. Compared to traditional methods based on CNN or RNN, the Transformer model is better at capturing the long-term dependencies and complex relationships of attack patterns. The experiments show that this method achieves high accuracy and low false positive rates, effectively identifying anomalous behaviors in real-time data streams.

7. Foundational deep-learning methods. We have built foundational neural networks, including MLP and CNN, which serve as the basis for evaluation.

5.1.3 MIDS Configuration

Before the model training phase, we performed rigorous preprocessing of the CAN data and meticulously configured the training setup to ensure model stability and efficient convergence of the loss function. All hyperparameters used during training are detailed in Table 2. The Adam optimizer was employed for optimization, with a batch size of 1024 per iteration. Additionally, a 5-fold cross-validation approach was implemented to mitigate potential performance bias arising from differences in data distribution. Each fold consisted of 50 epochs to ensure that the model sufficiently captured the data characteristics.

Given the four-class classification task (three tampering strategies and no tampering) with significant class imbalance, we implemented a dynamic weighting strategy. This strategy assigned weights to each class based on its sample size, reducing the risk of the model overfitting to the majority class while improving its detection capabilities for the minority classes.

To comprehensively evaluate the model's performance, we adopted macro-weighted metrics, including Macro Precision, Macro Recall, Macro F1 Score, and Accuracy. At the final epoch of each fold, we recorded these four metrics and averaged them across the five folds to derive the final evaluation of the model.

5.1.4 Dataset

In accordance with our threat model, we primarily utilize the tampering dataset detailed in Section 4.2 for comparison in Sections 5.2, 5.3, and 5.4. To ensure the generalization ability of our model, we also evaluate its performance on a public dataset, as discussed in Section 5.5.

In implementing ID tampering attacks within our dataset, identifying critical CAN signals for tampering is essential. To achieve this, we employed fuzzing techniques, which led us to identify two key signals: the door status signal (ID 0x102) and the steering angle signal (ID 0x132). Manipulating these signals could potentially create unsafe scenarios, such as allowing the doors to open while driving or presenting incorrect steering information. These findings are further corroborated by the detailed descriptions of these signals in the Tesla Model 3's CAN database. Based on these analyses, we are confident that these signals have a significant impact on the vehicle's safety and operational integrity.

5.2 Evaluating of MIDS

We first evaluate MIDS on our tampering attack dataset. Fig. 7 illustrates the training process and results of MIDS. Subfigure (a) shows the accuracy and loss curves, which increase rapidly and reach approximately 80% by the 10th epoch. In the subsequent 40 epochs, the rate of increase slows down, eventually stabilizing at around 98%. The similarity between the training and validation curves suggests that the model is not significantly overfitting. This is mainly due to the large size of the dataset, which provides sufficient data for the model to generalize well.

Subfigure (b) shows the macro Precision, Recall, and F1 score throughout the training process, displaying the same trend as in Subfigure (a). Subfigure (c) presents the ROC curves for MIDS under different tampering strategies. As shown, MIDS achieves an AUC of over 0.998 in all categories, demonstrating its robustness in distinguishing tampering attacks.

Subfigure (d) compares MIDS's performance with that of SOTA models, while Subfigure (e) illustrates MIDS's performance on public datasets. A detailed discussion of these results is provided in Sections 5.3 and 5.5.

Finally, Subfigure (f) presents the confusion matrix for classification results across four types of tampering strategies: beginning tampering, ID tampering, data tampering, and combined tampering. MIDS demonstrates high accuracy in each category, further validating its effectiveness in detecting various tampering attacks.

5.3 Comparison with State-of-the-Art Model

It is important to note that, in order to preserve the model structure to the greatest extent, CANshield and CanBus-IDS adopt a binary classification approach (attack present vs. no attack), while the other baseline models use a four-class classification approach (no attack, ID attack, data field attack, and combined attack). Table 3 presents the comparison results between MIDS and other SOTA models. MIDS achieves the highest precision, recall, F1 score, and accuracy, owing to its dual-stream architecture and Mamba layer with bi-directional technology. The dual-stream architecture enables MIDS to process the ID and data fields separately, leading to a more comprehensive understanding of the semantics, while the bi-directional Mamba layer captures long-distance data relationships in both directions.

In detail, the MIDS model achieved the highest F1 score and accuracy, followed by CanTransformer in second place. This suggests that attention mechanisms are effective in capturing long-distance traffic patterns when detecting CAN tampering attacks, although Mamba still outperforms them. While CanTransformer also produced strong results, it highlights the effectiveness of combining CNN and LSTM architectures. The CanBus-IDS model achieved 100% recall but exhibited poor precision, illustrating the trade-off between sensitivity and specificity. The F1 score further suggests that models utilizing GANs may not be well-suited for tampering attack detection. Additionally, although the CNN and GIDS models achieved high accuracy, their low F1 scores can likely be attributed to class imbalance in the dataset. In summary, MIDS, with its more complex structure, demonstrates better performance in detecting tampering attacks. In contrast, simpler

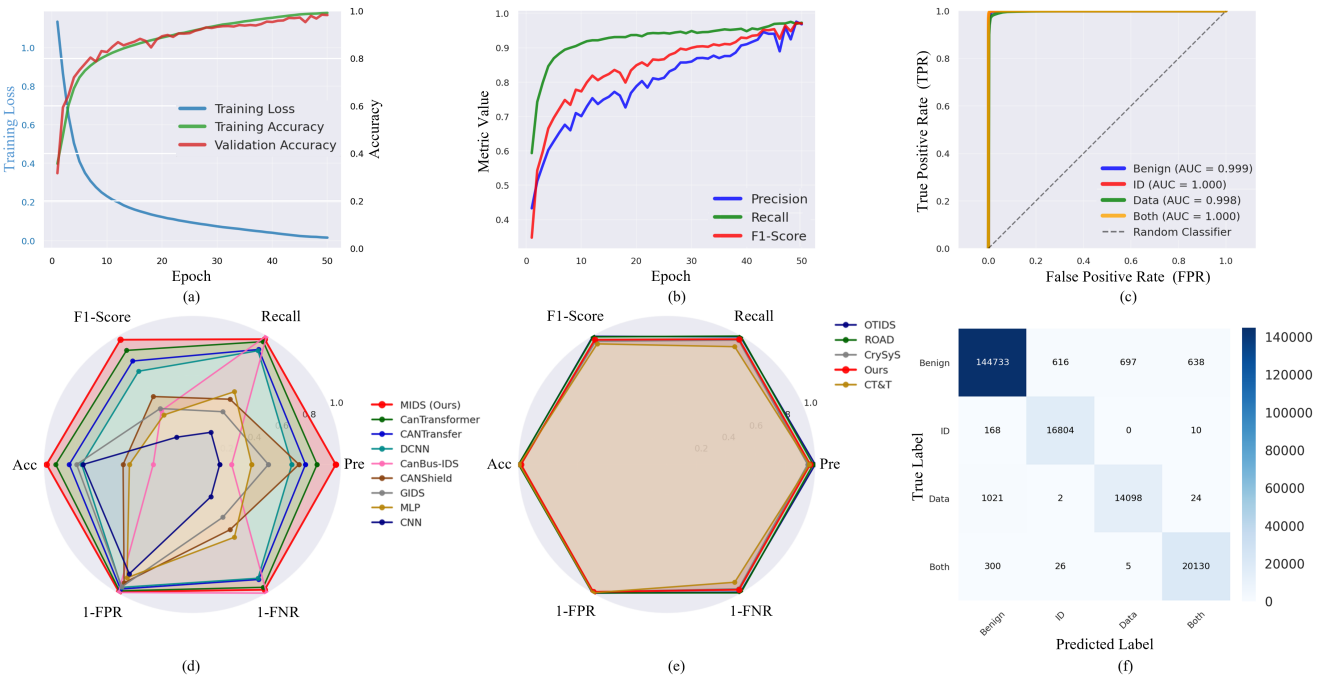

Figure 7: Overall model performance and comparisons

models like CNN and LSTM are less efficient in handling these attacks.

Table 2: MIDS' hyperparameters

Hyperparameters	Value
embedding_output_dim	256
mamba_hidden_state_dim	8
mamba_convolution_dim	4
mamba_feature_factor	2
conv_kernel_size	3
conv_padding_size	1
epoch	50
batch size	1024
k-fold	5
Train/Test split	4:1
optimizer	Adam

5.4 Ablation Study

We conducted an ablation study to evaluate the key components of the MIDS architecture by progressively removing or replacing specific modules and recording their impact on model performance. Experiments A1-A7 in Table 4 correspond to different ablation conditions, including the removal of individual modules, modification of module configurations, and the introduction of alternative designs. By comparing the performance of the complete MIDS architecture with the results from various ablation conditions, we aim to assess the contribution of each component to the overall system performance.

Experiments A1, A2, and A3 use parameter counts in a 2:4:1 ratio, and the results indicate that parameter count is a crucial factor affecting accuracy and F1 score. Experiments A4, A5, and A6 suggest that removing the bi-directional ability of Mamba, the partial conventional layer, and the embedding layer led to a 4.41%, 4.58%, and 4.70% reduction in F1 score, respectively. For each ablation experiment, we strictly maintained the same hyperparameter settings and experimental protocols as in the main MIDS experiment. In summary, Table 4 shows that the current architecture design yields optimal results. Each component of MIDS plays a crucial role, underscoring the rationality of the architecture and the effectiveness of the collaboration among its components.

5.5 Performance on Public Datasets

As shown in Table 5, we also evaluated the performance of MIDS on public datasets. MIDS achieved over 98.5% accuracy across all public datasets, demonstrating its strong intrusion detection capability. Moreover, compared to the public datasets, our dataset yielded the lowest accuracy, highlighting

Table 3: MIDS'performance on our dataset contrasted with SOTA models.

	Model Name	Used Layer/Technology	Precision	Recall	F1	Accuracy
SOTA	GIDS [39]	CNN, GAN	51.18%	41.00%	43.47%	77.96%
	CANShield [41]	CNN, LSTM	71.77%	50.69%	52.82%	46.68%
	CanBus-IDS [16]	CNN, GAN	26.41%	100.00%	41.78%	26.41%
	DCNN [45]	CNN	66.99%	88.40%	72.46%	74.54%
	CANTransfer [49]	CNN, LSTM, TL	76.15%	89.37%	80.38%	82.94%
	CanTransformer [21]	Attention	83.97%	95.44%	88.66%	92.09%
Foundational	MLP	-	39.94%	56.55%	38.48%	42.42%
	CNN	-	18.40%	25.00%	21.20%	73.59%
Ours	**MIDS**	CNN, Mamba	**96.55%**	**97.37%**	**96.94%**	**98.16%**

Table 4: Ablation study results

#	Description	Precision	Recall	F1	Accuracy	FPR↓	FNR↓
A1	**MIDS**	**96.55%**	**97.37%**	**96.94%**	**98.16%**	**1.00%**	**2.63%**
A2	MIDS_More_Param	89.10%	95.02%	91.84%	94.65%	2.30%	4.98%
A3	MIDS_Fewer_Param	86.26%	94.33%	89.91%	93.18%	2.75%	5.67%
A4	One_Directional_Mamba	89.49%	96.07%	92.53%	95.11%	1.92%	3.93%
A5	MIDS_Fewer_Conv	91.74%	93.09%	92.36%	95.22%	2.73%	5.73%
A6	MIDS_No_ID_Embed	90.40%	94.27%	92.24%	95.26%	2.26%	5.73%
A7	Only_Mamba	34.25%	42.24%	29.78%	49.42%	21.32%	57.76%

Table 5: MIDS' performance on public datasets

Dataset Name	Type	Precision	Recall	F1	Accuracy	FPR↓	FNR↓
Ours	Tampering	96.55%	97.37%	96.94%	98.16%	1.00%	2.63%
ROAD [17]	Tampering	98.63%	99.80%	99.21%	99.83%	0.17%	0.20%
CrySyS [10]	Tampering	94.90%	96.64%	95.76%	98.59%	1.03%	3.36%
OTIDS [29]	Injection	100.00%	99.24%	99.61%	99.63%	0.00%	0.76%
CT&T [27]	Injection	96.00%	91.51%	93.70%	99.24%	0.25%	8.49%

the high quality and complexity of our dataset.

6 Future Works

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.

7 Conclusion

In this paper, we proposed MIDS, a novel deep learning-based approach for detecting tampering and injection attacks on the CAN bus. By utilizing Mamba with bidirectional technology and a dual-stream architecture, MIDS effectively captures both local and long-range dependencies in CAN signals, achieving superior performance with an F1-score ranging from 93.70% to 99.96%.

We evaluated MIDS on a multi-scenarios dataset comprising over 100 million CAN messages from a Tesla Model 3, and using other publicly available datasets to demonstrate generalizability of MIDS. Extensive experiments and ablation studies confirm the robustness and efficiency of MIDS, showcasing its potential for real-world deployment. Future work will focus on expanding the dataset and adapting MIDS for emerging protocols, such as CAN FD, to address evolving security challenges.

Acknowledgments

We appreciate the support provided by an anonymous individual (due to the anonymous review request) during the research process.

Ethical Considerations

This work involves the development of a model for detecting CAN BUS attacks, with the use of collected CAN data and generated attack data to validate the effectiveness of the detection system. The following ethical considerations were taken into account during the research:

Firstly, the CAN data collected for this study was sourced from publicly available datasets and simulations that do not contain any personally identifiable information. All data used in this study has been anonymized to ensure that no sensitive information, including private vehicle data or proprietary business information, is exposed or misused. In accordance with privacy best practices, no data collected in this study could be traced back to any specific individual, vehicle, or company. Furthermore, the generated attack data was designed to simulate potential vulnerabilities in CAN systems without reflecting real-world incidents or exposing any specific organization's systems to harm. These simulated attacks are purely for academic research purposes and are not intended for use in actual environments without proper safeguards.

Secondly, the attack data generated for testing the model's effectiveness simulates realistic attack scenarios on CAN systems, but it does not target real-world systems or any specific operational environment. These attacks are designed solely to assess the resilience of the detection model and to understand potential vulnerabilities in theoretical or lab environments. While the research aims to improve security, it is critical to emphasize that the attacks generated are not intended for malicious use.There is an ethical responsibility to prevent the misuse of this data. Although the data and model are made available for academic research, it is imperative that researchers and practitioners use these resources responsibly. The research community must ensure that the methodologies developed are not employed in unauthorized or harmful ways, such as attacking real-world systems or infrastructure.

Thirdly, the availability of both the code and data enhances the reproducibility and transparency of the results. By making the code publicly accessible, we allow other researchers to validate and challenge the findings, fostering an open academic environment. It is vital that the work remains transparent so that any issues regarding data integrity, model bias, or potential limitations can be openly discussed within the community.

Open Science

To promote reproducibility, transparency, and verifiability of our research, we release the following artifacts associated with this work. All resources are hosted on anonymous repositories to preserve the double-blind review process.

1. **Dataset:** We construct a real-world CAN traffic dataset collected from a Tesla Model 3, which includes both benign traffic and manipulated traffic under tampering attack scenarios. *Availability:* The dataset is publicly accessible through an anonymous repository, and the download link is provided in the main body of the paper.

2. **Source Code:** The full implementation of the proposed MIDS framework is released, covering data preprocessing, model architecture, training, and evaluation procedures. *Availability:* The codebase is hosted on an anonymous repository, and the access link is included in the paper.

3. **Pre-trained Models:** Pre-trained models are not provided. Readers are encouraged to reproduce the models independently using the released source code and dataset.

4. **Experimental Configurations:** All hyperparameter settings and experimental configurations are explicitly integrated into the released source code. No additional configuration files are required.

5. **Evaluation Scripts:** We provide evaluation scripts that compute precision, recall, F1-score, and confusion matrices, ensuring faithful reproduction of the reported experimental results. *Availability:* The evaluation scripts are bundled with the released source code repository.

References

[1] Sudha Anbalagan, Gunasekaran Raja, Sugeerthi Gurumoorthy, R. Deepak Suresh, and Kapal Dev. Iids: Intelligent intrusion detection system for sustainable development in autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 24(12):15866–15875, 2023.

[2] Shunsuke Araki, Akiyoshi Tashiro, Ken'ichi Kakizaki, and Satoshi 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)*, pages 247–248, 2017.

[3] Stephen Checkoway, Damon McCoy, Brian Kantor, Danny Anderson, Hovav Shacham, Stefan Savage, Karl Koscher, Alexei Czeskis, Franziska Roesner, and Tadayoshi Kohno. Comprehensive experimental analyses of automotive attack surfaces. In *20th USENIX Security Symposium (USENIX Security 11)*, San Francisco, CA, August 2011. USENIX Association.

[4] Kyong-Tak Cho and Kang G. Shin. Fingerprinting electronic control units for vehicle intrusion detection. In *25th USENIX Security Symposium (USENIX Security 16)*, pages 911–927, Austin, TX, August 2016. USENIX Association.

[5] Wonsuk Choi, Hyo Jin Jo, Samuel Woo, Ji Young Chun, Jooyoung Park, and Dong Hoon Lee. Identifying ecus using inimitable characteristics of signals in controller area networks. *IEEE Trans. Veh. Technol.*, 67(6):4757–4770, 2018.

[6] Wonsuk Choi, Kyungho Joo, Hyo Jin Jo, Moon Chan Park, and Dong Hoon Lee. Voltageids: Low-level communication characteristics for automotive intrusion detection system. *IEEE Trans. Inf. Forensics Secur.*, 13(8):2114–2129, 2018.

[7] Xuting Duan, Huiwen Yan, Daxin Tian, Jianshan Zhou, Jian Su, and Wei 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*, 24(2):2122–2134, 2023.

[8] 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*, page 14, 2019.

[9] András Gazdag, Csongor Ferenczi, and Levente 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*, volume 2874 of *CEUR Workshop Proceedings*, Nov 6–8 2020. Open-access under CC BY 4.0.

[10] András Gazdag, Rudolf Ferenc, and Levente Buttyán. Crysys dataset of can traffic logs containing fabrication and masquerade attacks. 10, 2023.

[11] Bogdan Groza and Pal-Stefan Murvay. Efficient intrusion detection with bloom filtering in controller area networks. *IEEE Trans. Inf. Forensics Secur.*, 14(4):1037–1051, 2019.

[12] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *CoRR*, abs/2312.00752, 2023.

[13] Albert Gu, Karan Goel, and Christopher 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.

[14] Mee Lan Han, Byung Il Kwak, and Huy Kang Kim. TOW-IDS: intrusion detection system based on three overlapped wavelets for automotive ethernet. *IEEE Trans. Inf. Forensics Secur.*, 18:411–422, 2023.

[15] Markus Hanselmann, Thilo Strauss, Katharina Dormann, and Holger Ulmer. Canet: An unsupervised intrusion detection system for high dimensional can bus data. *IEEE Access*, 8:58194–58205, 2020.

[16] Thien-Nu Hoang and Daehee Kim. Detecting in-vehicle intrusion via semi-supervised learning-based convolutional adversarial autoencoders. *Veh. Commun.*, 38:100520, 2022.

[17] Samuel C Hollifield, Miki E Verma, Michael D Iannacone, Robert A Bridges, Bill Kay, and Frank L Combs. Poster: Real ornl automotive dynamometer (road) can intrusion dataset.

[18] Md Delwar Hossain, Hiroyuki Inoue, Hideya Ochiai, Doudou Fall, and Youki Kadobayashi. An effective in-vehicle can bus intrusion detection system using cnn deep learning approach. In *GLOBECOM 2020-2020 IEEE global communications conference*, pages 1–6. IEEE, 2020.

[19] Abdul Rehman Javed, Saif Ur Rehman, Mohib Ullah Khan, Mamoun Alazab, and Thippa 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*, 8(2):1456–1466, 2021.

[20] Mubark Jedh, Lotfi Ben Othmane, Noor Ahmed, and Bharat K. Bhargava. Detection of message injection

attacks onto the CAN bus using similarity of successive messages-sequence graphs. *CoRR*, abs/2104.03763, 2021.

[21] Hyunjun Jo and Deok-Hwan Kim. Intrusion detection using transformer in controller area network. *IEEE Access*, 12:121932–121946, 2024.

[22] Christian Kaiser, Andreas Festl, Gernot Pucher, Michael Fellmann, and Alexander Stocker. The vehicle data value chain as a lightweight model to describe digital vehicle services. In Alessandro Bozzon, Francisco José Domínguez Mayo, and Joaquim Filipe, editors, *Proceedings of the 15th International Conference on Web Information Systems and Technologies, WEBIST 2019, Vienna, Austria, September 18-20, 2019*, pages 68–79. ScitePress, 2019.

[23] Hyunjae Kang, Byung Il Kwak, Young Hun Lee, Haneol Lee, Hwejae Lee, and Huy Kang Kim. Car hacking and defense competition on in-vehicle network. In *Workshop on Automotive and Autonomous Vehicle Security (AutoSec)*, volume 2021, page 25, 2021.

[24] Kang J-W Kang M-J. Intrusion detection system using deep neural network for in-vehicle network security. *PLoS ONE 11(6): e0155781.*

[25] Muneeb Hassan Khan, Abdul Rehman Javed, Zafar Iqbal, Muhammad Asim, and Ali Ismail Awad. Divacan: Detecting in-vehicle intrusion attacks on a controller area network using ensemble learning. *Comput. Secur.*, 139:103712, 2024.

[26] Muneeb Hassan Khan, Abdul Rehman Javed, Zafar Iqbal, Muhammad Asim, and Ali Ismail Awad. Divacan: Detecting in-vehicle intrusion attacks on a controller area network using ensemble learning. *Computers & Security*, 139:103712, 2024.

[27] Brooke Lampe and Weizhi Meng. can-train-and-test: A new can intrusion detection dataset. In *2023 IEEE 98th Vehicular Technology Conference (VTC2023-Fall)*, pages 1–7, 2023.

[28] Ulf E. Larson, Dennis K. Nilsson, and Erland Jonsson. An approach to specification-based attack detection for in-vehicle networks. In *2008 IEEE Intelligent Vehicles Symposium*, pages 220–225, 2008.

[29] Hyunsung Lee, Seong Hoon Jeong, and Huy Kang 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*, pages 57–66. IEEE Computer Society, 2017.

[30] Chung-Wei Lin and Alberto 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*, pages 1–7. IEEE Computer Society, 2012.

[31] Mirco Marchetti, Dario Stabili, Alessandro Guido, and Michele 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*, pages 1–6. IEEE, 2016.

[32] C. Miller and C. Valasek. Adventures in automotive networks and control units.

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

[34] Pal-Stefan Murvay and Bogdan Groza. Source identification using signal characteristics in controller area networks. *IEEE Signal Process. Lett.*, 21(4):395–399, 2014.

[35] Michael Müter and Naim Asaj. Entropy-based anomaly detection for in-vehicle networks. In *IEEE Intelligent Vehicles Symposium (IV), 2011, Baden-Baden, Germany, June 5-9, 2011*, pages 1110–1115. IEEE, 2011.

[36] University of Turku. Can bus dataset collected from a heavy-duty truck, 5 2021. University of Turku.

[37] Francesco Pollicino, Dario Stabili, and Mirco Marchetti. Performance comparison of timing-based anomaly detectors for controller area network: A reproducible study. *ACM Trans. Cyber-Phys. Syst.*, 8(2), May 2024.

[38] Muhammad Sami. Intrusion detection in can bus, 2019.

[39] Eunbi Seo, Hyun Min Song, and Huy Kang Kim. GIDS: GAN based intrusion detection system for in-vehicle network. In Kieran McLaughlin, Ali A. Ghorbani, Sakir Sezer, Rongxing Lu, Liqun Chen, Robert H. Deng, Paul Miller, Stephen Marsh, and Jason R. C. Nurse, editors, *16th Annual Conference on Privacy, Security and Trust, PST 2018, Belfast, Northern Ireland, Uk, August 28-30, 2018*, pages 1–6. IEEE Computer Society, 2018.

[40] Eunbi Seo, Hyun Min Song, and Huy Kang Kim. GIDS: GAN based intrusion detection system for in-vehicle network. *CoRR*, abs/1907.07377, 2019.

[41] Md Hasan Shahriar, Yang Xiao, Pablo Moriano, Wenjing Lou, and Y. Thomas Hou. Canshield: Deep-learning-based intrusion detection framework for controller area networks at the signal level. *IEEE Internet of Things Journal*, 10(24):22111–22127, 2023.

[42] H. M. Song and H. K. Kim. Can signal extraction and translation dataset.

[43] Hyun Min Song, Ha Rang Kim, and Huy Kang 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*, pages 63–68. IEEE Computer Society, 2016.

[44] Hyun Min Song, Ha Rang Kim, and Huy Kang 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*, pages 63–68. IEEE Computer Society, 2016.

[45] Hyun Min Song, Jiyoung Woo, and Huy Kang Kim. In-vehicle network intrusion detection using deep convolutional neural network. *Veh. Commun.*, 21, 2020.

[46] Dario Stabili, Luca Ferretti, Mauro Andreolini, and Mirco Marchetti. DAGA: detecting attacks to in-vehicle networks via n-gram analysis. *IEEE Trans. Veh. Technol.*, 71(11):11540–11554, 2022.

[47] Dario Stabili and Mirco Marchetti. Detection of missing can messages through inter-arrival time analysis. In *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*, pages 1–7, 2019.

[48] Heng Sun, Mengsi Sun, Jian Weng, and Zhiquan Liu. Analysis of id sequences similarity using dtw in intrusion detection for can bus. *IEEE Transactions on Vehicular Technology*, 71(10):10426–10441, 2022.

[49] Shahroz Tariq, Sangyup Lee, and Simon S. Woo. Cantransfer: transfer learning based intrusion detection on a controller area network using convolutional LSTM network. In Chih-Cheng Hung, Tomás Cerný, Dongwan Shin, and Alessio Bechini, editors, *SAC '20: The 35th ACM/SIGAPP Symposium on Applied Computing, online event, [Brno, Czech Republic], March 30 - April 3, 2020*, pages 1048–1055. ACM, 2020.

[50] Adrian Taylor, Sylvain P. Leblanc, and Nathalie 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*, pages 130–139. IEEE, 2016.

[51] Qiyan Wang and Sanjay 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*, pages 13–18. IEEE, 2014.

[52] Tianqi Yu, Guodong Hua, Huaisheng Wang, Jianfeng Yang, and Jianling Hu. Federated-lstm based network intrusion detection method for intelligent connected vehicles. In *ICC 2022-IEEE International Conference on Communications*, pages 4324–4329. IEEE, 2022.

[53] Mattia Zago, Stefano Longari, Andrea Tricarico, Michele Carminati, Manuel Gil Pérez, Gregorio Martínez Pérez, and Stefano Zanero. Recan – dataset for reverse engineering of controller area networks. *Data in Brief*, 29:105149, 2020.

[54] Hengrun Zhang, Kai Zeng, and Shuai Lin. Federated graph neural network for fast anomaly detection in controller area networks. *IEEE Trans. Inf. Forensics Secur.*, 18:1566–1579, 2023.