Meta-IDS: Meta-Learning Automotive Intrusion Detection Systems with Adaptive and Learnable

Hong-Quan Wang

wanghongquan22@mails.ucas.ac.cn

University of Chinese Academy of Sciences

Jin Li

Chinese Academy of Sciences

Dong-Hua Huang

Beijing DualPi Intelligent Security Technology Co., LTD.

Yao-Dong Tao

Beijing DualPi Intelligent Security Technology Co., LTD.

Research Article

Keywords: Intrusion Detection System (IDS), Meta-Learning, Meta-SGD, Controller Area Network (CAN), Low-Volume Attacks, Vehicular Network Security

Posted Date: March 8th, 2024

DOI: https://doi.org/10.21203/rs.3.rs-3999020/v1

Additional Declarations: No competing interests reported.

Meta-IDS: Meta-Learning Automotive Intrusion Detection Systems with Adaptive and Learnable

Hong-Quan Wang[1,2], Jin Li[2], Dong-Hua Huang[3], Yao-Dong Tao[3,4*]

[1*]School of computer science and technology, University of Chinese Academy of Sciences, Yanqihu East Road, Beijing, 101408, Beijing, China.

[2]Shenyang Institute of Computing Technology, Chinese Academy of Sciences, Nanping East Road, Shenyang, 110168, Liaoning, China.

[3]Beijing DualPi Intelligent Security Technology Co., LTD., Xinjiekouwai Avenue, Beijing, 100088, Beijing, China.

[4]Beijing Jiaotong University, Shangyuan Village, Beijing, 100044, Beijing, China.

*Corresponding author(s). E-mail(s): taoyaodong@dualpi.com;
Contributing authors: wanghongquan22@mails.ucas.ac.cn;
lijin@sict.ac.cn; huangdonghua@dualpi.com;

Abstract

In the rapidly evolving landscape of vehicular communications, the widespread use of the Controller Area Network (CAN) in modern vehicles has revealed significant security vulnerabilities. However, existing Intrusion Detection Systems (IDS) struggle to adapt to varied attack scenarios and precisely detect low-volume attacks. In this paper, we introduce a novel IDS that employs meta-learning via the Meta-SGD algorithm, enhancing adaptability across a diverse spectrum of cyber threats, called Meta-IDS. Specifically, our methodology includes a bi-level optimization technique where the inner level focuses on optimizing detection accuracy for specific attack scenarios, and the outer level adjusts meta-parameters to ensure generalizability across different scenarios. For modeling low-volume attacks, we devise the Attack Prominence Score (APS), identifying subtle attack patterns with a threshold of APS > 7, allowing for precise differentiation of these attacks. The extensive experiment results show that the proposed method facilitates efficient tuning and rapid adaptation for different modeling paradigms in few-shot scenarios. The detection performance is exceptional, with F1-scores reaching 100% across most attack scenarios, including

low-volume attacks. Also, the real-time vehicle-level evaluations demonstrate its adaptability for the vehicular networks.

Keywords: Intrusion Detection System (IDS), Meta-Learning, Meta-SGD, Controller Area Network (CAN), Low-Volume Attacks, Vehicular Network Security

1 Introduction

Recently, the Internet of Vehicles (IoV) has been widely criticized for the vulnerability of its communication protocols, which made several security threats, such as authentication hijacking, message tapping, and availability destruction [1]. These are mainly caused by the Controller Area Network(CAN), considered the de facto standard for in-vehicle communication [2], [3]. It is vulnerable to attacks due to its weak security mechanisms, such as the lack of authentication, message broadcast mechanism, the absence of encryption, and its ID-based arbitration mechanism [1]. Although some traditional security mechanisms(e.g., cryptographic techniques, authentication) have been proposed for secure communication and are widely used in conventional Ethernet, they are not applicable in the environment of the in-vehicle network which has strong hard timing constraints [4].

The Intrusion Detection System (IDS) is regarded as an effective method to protect in-vehicle networks because of its low overhead for the network system and adaptive capacity for different in-vehicle environments [5]. Recently, the Machine Learning-based IDS is considered to be a detection method with high accuracy and efficiency [6]. Although the supervised learning-based IDS has a low false alarm rate (FAR), the kind methods have a weak ability to detect unknown attacks [7], [8], [9]. In contrast, the unsupervised learning-based IDS can identify various anomaly traffic through a deviation between normal behavior and abnormal behavior, but with a relatively high false positive rate (FPR) [10], [11], [12]. Besides, many works design the attack model by injecting a high volume of bursty anomaly messages, which could cause false-high performance [13], [14], [15]. In fact, the attack usually maintains invisible and low volume. Therefore, existing methods either lack the ability to quickly adapt to the current environment and detect attacks well in it or cannot cope with the circumstances in the attack traffic is insidious.

To fill the gaps aforementioned, we propose a novel detection model for automotive, named Meta-IDS, a lightweight intrusion detection system with adaptive and learnable in this paper. Specifically, we utilize the learning rate-trainable meta-learning technique to train an initial model. When IVN confronts a new attack, the initial model can adapt it and keep promising performance because every parameter in the model has a trainable learning rate. On the one hand, The proposed model can identify insidious attack messages with training on few-shot samples under different circumstances to be quickly adaptive to different attack detection scenarios. Moreover, Meta-IDS has demonstrated that this paradigm is with excellent generalizability on various previous IDSs.

Our main contributions are summarized as follows:

1. We propose an adaptive and learnable automobile IDS called Meta-IDS that exploits ***an optimization-based meta-learning technique for intrusion detection for automobiles***.
2. We employ ***a learning rate-trainable meta-learning method*** to develop an anomaly detection model with superior adaptability to diverse attack scenarios, surpassing traditional transfer methods in speed. Additionally, we introduce the ***Attack Prominence Score (APS)*** to accurately model and identify low-volume and insidious attacks in complex scenarios, marking a significant advancement in anomaly detection strategies.
3. We provide a comparison result on multiple automotive IDS combined with the proposed learning paradigm, which shows promising generalizability and incremental performance improvement.
4. Our empirical results demonstrate that Meta-IDS significantly outperforms baseline defenses in detection performance, while also meeting industry requirements for resource cost.

The rest of this paper is organized as follows: section 2 introduces the related work. Section 3 presents the vulnerabilities in IVN, as well as the attack scenarios and IDS deployment. Section 4 describes proposed Meta-IDS, which is then analyzed and evaluated in section 5, respectively. Finally, we draw our conclusion in section 6.

2 Related WORK

2.1 intrusion detection system for automotive

As the safety of automotive passengers and pedestrians has become an issue of paramount concern, there is a large amount of study is being carried out both in industry and academia to address this issue [16], [17]. IDSs, which play a great role in it, are mainly categorized into two categories signature-based and anomaly-based. The former detects a potential intrusion behavior by matching monitored events against a database of attack signatures, which focuses on known attacks and needs to be updated regularly. However, it's inconvenient to update it frequently in people's real life. Also, it may encounter problems as soon as attack patterns deviate from the original specification. Thus, the signature-based IDSs cannot handle the task at hand well. In contrast, the anomaly-based methods mainly overcome the aforementioned drawbacks, which learn behavior variations from automotive traffic so that filter the attack message.

Many traditional approaches have been proposed in previous works to protect the security of IVNs. Song et al. [14] proposed an interval-based approach for attack detection with the fact that messages in automotive networks usually have a fixed time interval or frequency. They monitor the intervals between messages and compare them with normal time intervals between messages to generate anomaly signals. Marchetti et al. [18] constructed an ID sequence-based IDS that emphasizes transitions between arbitration IDs. A graphs-based IDS is present, which could record the normal behavior of the network system[19]. Wang et al. [20] proposed an IDS based on ID-based entropy analysis of CAN messages. However, these methods have some common problems,

with limited adaptation to various attacks and relatively weak performance. On the other hand, machine learning (ML) has been introduced with its capability to respond to such challenges in time.

2.2 machine-learning-based intrusion detection systems for in-vehicle networks

The traditional machine-learning-based approaches have been widely adopted for intrusion detection. Avatefipour et al. [21] proposed a modified One-Class Support Vector Machine(OCSVM) based intrusion detection model. Kalutarage et al. [22] model message sequence using so-called N-grams distributions and utilizes benign data for training and threshold estimation to detect anomalies. However, these approaches require high computational resources. Levi et al. [23] trained a Hidden Markov Model to learn normal vehicle behavior to detect anomalies. Although they overcame low computational resources by moving the detection mechanism to the cloud, it needs extra effort to transmit important data extracted by configurable rules to the cloud and maintain a backend. Moulahi et al. [24] presented a comparative detection performance on Random Forest(RT), Decision Tree(DT), Support Vector Machine(SVM), and Multilayer perception(MLP) based methods. Fenzl et al. [25] proposed an IDS based on decision trees modeled through genetic programming.

Recently, Deep Learning(DL)-based methods have been proposed with outstanding performance. Ma et al. [26] proposed a supervised learning model, by combining a Gated Recurrent Unit(GRU)-based network and a low complexity feature extraction algorithm to detect automotive network attacks. Ale et al. [27] developed an IDS using Deep Bayesian Learning (DBL) to detect and analyze automobile hacking attacks. Xiao et al. [28] presented a new RNN-based IDS using optimized LSTM and GRU architectures along with a simplified attention model. Shi et al. [29] proposed an unsupervised learning model named Temporal Convolutional Network-Based Intrusion Detection System(TCNIDS), by combining a temporal convolutional network and a word embedding model. Desta et al. [30] proposed an IDS by training individual LSTM models for each ID and then aggregating them to generate a unified anomaly signal. Ashraf et al. [31] proposed an LSTM autoencoder-based scheme to design an IDS that used a statistical feature extraction technique for capturing contextual features from network traffic. Yang et al. [32] proposed a novel transfer learning and ensemble learning-based IDS for CAN attack detection and evaluated the model's performance on two general network intrusion datasets, namely the Car-Hacking dataset [33] and the CICIDS2017 dataset [34].

2.3 Literature comparison

There is a finding in the aforementioned discussion that ML-based methods show outstanding performance compared to signature-based ones. However, most of them are only designed for a specific network environment as they are trained on a benchmark dataset that was generated under a designed circumstance and lack generalization ability for tasks from various attack scenarios. [21], [22], [24], [25], [26], [29]. Additionally, the huge computational resources and time cost needed by them to train a model

Table 1: Comparison between various state-of-the-art intrusion detection schemes for automotive

Literature	Aafvas	Pfid	Vloat	Vlont
Avatefipour et al. [21]	No	Low	High	High
Kalutarage et al. [22]	No	High	High	High
Moulahi et al. [24]	No	Low	High	High
Fenzl et al. [25]	No	Low	High	High
Ma et al. [26]	No	High	High	High
Ale et al. [27]	No	Low	High	High
Xiao et al. [28]	No	High	High	High
Shi et al. [29]	No	Low	High	Low
Desta et al. [30]	No	Low	High	Low
Ashraf et al. [31]	No	Low	High	Low
Yang et al. [32]	Yes	High	High	High
proposed Meta-IDS	Yes	High	Low	Low

Aafvas: Yes, indicates that the IDS proposed in the literature is adaptive for various attack scenarios; No, indicates that the IDS proposed in the literature is just adaptive for a few specific attack scenarios

Pfid: High, indicates that the FPR of the proposed IDS in the literature is lower than 1%, and Recall, Accuracy, and F1-score are all at least 99% Low, indicates that the FPR of the proposed IDS in the literature is at least 1%, or one of Recall, Accuracy, F1-score lower than 99%;

Vloat, Vlont: High, indicates that the volume of corresponding traffic in the dataset is at least 1000; Low, indicates that the volume of corresponding traffic in the dataset is lower than 1,000

on a high-volume dataset cannot be ignored. Most importantly, the trained model may not cope with low-volume attacks as it relies on the learning of high-volume normal and attack traffic samples in the task's dataset. However, in real-world applications, the hacker may launch attacks in very varying scenarios which may not be limited to the previous training dataset represented and the computational resources are rather limited in vehicular environments. The IDS proposed in [32] is the only research that considers adaptive for different attack scenarios. Unfortunately, its computational cost is rather high and it also faces the problem that the dataset it trained on has a high volume. Thus, there still should be an IDS designed to rapidly adapt to different attack scenarios and has the ability to detect attacks that are stealthy and with comparatively low volume. Our proposed IDS aims to address these challenges.

To summarize, Table 1 clearly compares 7 existing works with the proposed Meta-IDS based on the following dimensions, which are an IDS that could fill the aforementioned gaps of previous works should focus on Adaptive ability for various attack scenarios(Aafvas), Performance for intrusion detection(Pfid), the Volume level of attack and normal traffic(Vloat, Vlont) in evaluation dataset.

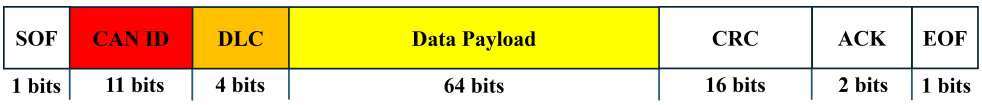

Fig. 1: The configuration of the CAN data frame.

3 Preliminary

3.1 CAN Protocol

The CAN protocol is the predominant network technology enabling communication between electronic control units(ECUs) in modern automotive vehicles. The CAN network utilizes a bus topology where ECUs connect to a common backbone and can transmit and receive messages related to various vehicle functions such as steering and engine RPM. This architecture allows the different ECU nodes distributed throughout the vehicle to communicate efficiently by sending data packets over the shared CAN bus. In the CAN bus, every ECU transmits its data messages to all nodes in the system using a standardized frame format. A schematic diagram illustrating the typical configuration of a CAN frame is provided in Fig. 1. Every CAN frame contains a distinct CAN identifier (ID) that can be 11 bits long for standard CAN or 29 bits for extended CAN. Since messages are broadcast over the bus, the ID allows each ECU to filter and select only messages relevant to it. However, the lack of authentication allows ECUs to spoof the IDs of other components when sending messages over the bus, which is called the spoofing attack [35]. Because multiple messages may be transmitted simultaneously on the bus, the ID also functions as a priority code determining which ECU gains access first. CAN prioritize messages with lower ID values, assigning them higher priority access to the bus. By persistently transmitting frames with very low IDs, an attacker could launch a Denial-of-Service(DoS) attack and dominate bus access [36]. Note that the adversary has the capability to hijack the ECUs and bus with more insidious attacks, especially in low-volume scenarios. Moreover, they have not satisfied existing attack modes, and have explored some transferring attacks that fast switch between different attacks, which our proposed model concentrated on. A CAN frame is comprised of several components in addition to the ID: start of frame (SOF), data length code (DLC), data payload, cyclic redundancy check (CRC), acknowledgment (ACK), and end of frame (EOF). The ID uniquely identifies the frame, while the SOF indicates the start, DLC specifies the data length, Data field contains the message data, CRC is used for error checking, ACK confirms receipt, and EOF marks the end [37].

For conciseness, this paper omits some finer field specifics not pertinent to the presented work. While the CAN standard is public information, actual message payload semantics differ across vehicle manufacturers and remain unpublished proprietary knowledge. Therefore, we have developed an automated methodology to extract spatial or temporal features from CAN messages without requiring predefined payload semantics. This allows analysis without access to manufacturer-specific definitions.

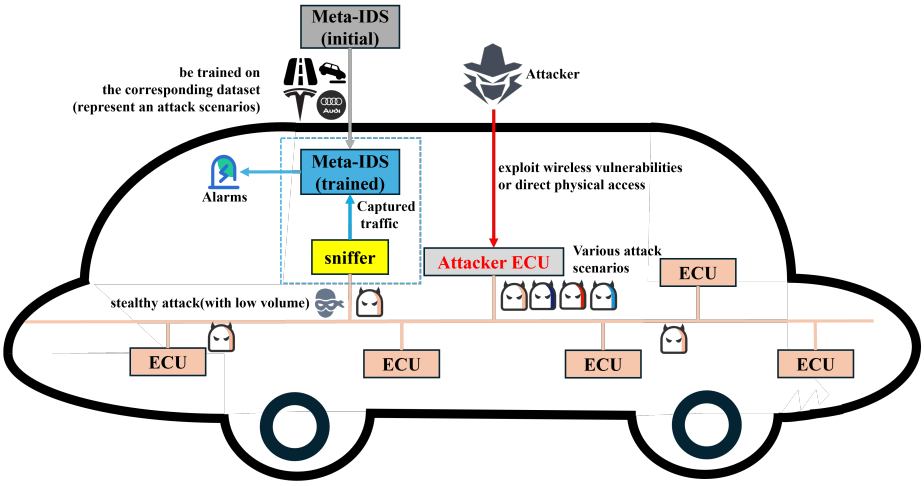

Fig. 2: The attack model and IDS deployment scenario.

3.2 Attack Model and IDS Deployment Scenario

In Fig. 2, we show the attack model and IDS deployment in automotive for our proposed method. This analysis assumes an adversary that can gain control of one or more ECUs through remote exploitation of wireless vulnerabilities (e.g., telematics unit [38])or direct physical access (e.g., via OBD-II [39]). A compromised ECU falls fully under the adversary's control and becomes a rogue attacker ECU. The adversary can then either suspend the ECU's normal functionality or reprogram it to inject arbitrary CAN messages. In spoofing attacks, the adversary could deceit message IDs to impersonate other ECUs and transmit falsified data payloads over the bus. In this circumstance, the adversary may launch attacks not only with high volumes which are so obvious that can be easily detected, but also with low volumes which are so stealthy that can avoid conventional detection methods. The straightforward method is to inject 1 malicious packet per second, which is usually buried after the other normal traffic. The false data is crafted to slowly modify the engine's operating profile over time. By transmitting stealthy low-rate forgeries, the attacker aims to degrade performance and cause wear without triggering alerts. Furthermore, the attack scenario can also be various. For instance, the adversary may not only attempt a DoS attack by flooding the CAN bus with high volumes of malicious diagnostic requests during routine city driving when traffic is steady but also attempt to overwhelm CAN communications by injecting a flood of forged frames during aggressive highway driving. Moreover, the differences in frame field specifications for automobiles of different manufacturers may also make attack scenarios various.

To secure automotive networks, the proposed Meta-IDS can be started up within the IVN after being rapidly trained on various datasets specified to various attack scenarios. The IDS is placed on the CAN bus to monitor all broadcast traffic. Since every CAN message gets transmitted to all nodes, the IDS will receive each frame as it is broadcast on the bus. By analyzing the live CAN traffic, the IDS can detect

potential attacks even though faced with various attack scenarios and attack traffic with low volume. If any malicious activity is identified, the IDS can trigger alarms to notify nodes across the IVN.

Protecting vehicles from intrusions, all packets transmitted on the protected vehicular network are captured using packet sniffing (e.g., NetFlow) and analyzed by the proposed IDS before reaching vehicle systems [40]. For example, if an adversary launches a Denial-of-Service (DoS) attack by flooding malicious traffic, the IDS can detect this by processing the sniffer's captured data. The IDS would then trigger alarms and block the attacker's access, shielding the vehicle from compromise.

3.3 Neural Networks

The neural network has been effective in detecting cyber-attacks through modeling important features of time-series messages on IoV. Convolutional Neural Networks (CNNs) play a significant role in spatial feature extraction from time series data, such as messages transmitted over the CAN bus. The CNN uses a 1-D convolution process, where a filter or kernel systematically moves across the input data-similar to a sliding window. This operation enables CNN to perform a content analysis of each CAN message, where the captured feature map highlights critical aspects in the data sequence. Thus, this component has been applied to recognize intrusions or unusual activity within the automotive, especially fuzzy and masquerade attacks [15], [8].

Long Short-Term Memory (LSTM) networks are usually employed to address the challenges associated with long-term dependencies in sequential data. The gate control cells, including the forget gate, input gate, and output gate, collaboratively decide which information should be retained, updated, or discarded. Thus, this component can capture temporal features in the context of CAN messages. This selective memory process is vital for identifying and responding to complex intrusion patterns that evolve in vehicular networks, ensuring that the network remains robust against periodicity attacks (e.g., DoS attack) [41], [42].

However, existing research has predominantly overlooked the detection of low-volume attacks and the adaptability to varying attack traffic scenarios. Our proposed Meta-IDS effectively addresses the gaps in intrusion detection systems by introducing a learning paradigm that leverages the strengths of neural networks for feature extraction and handling sequential data dependencies. It extends the capabilities of neural networks to detect low-volume intrusions and more attack scenarios based on meta-learning. This advancement not only contributes to the field of cybersecurity in intelligent transportation systems but also sets a new benchmark for future research in intrusion detection methodologies.

4 Methodology

In this section, we propose the overall pipeline of the defense mechanism, including the data processing, bi-level optimization, and model structures in detail. Fig. 3 depicts the functionality of our presented framework.

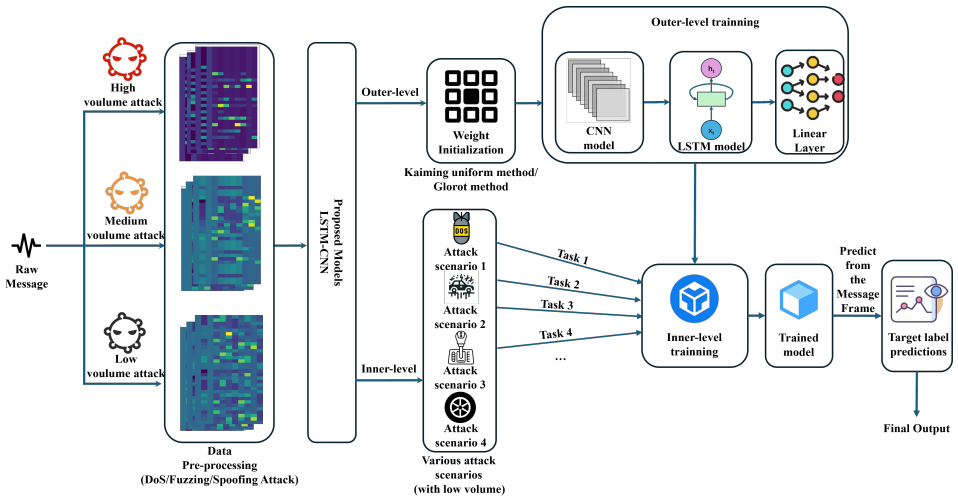

Fig. 3: An overview of the proposed framework.

4.1 Data Preprocessing

The framework we developed was rigorously tested using the CAN Car-Hacking dataset [33], which includes a series of CAN messages recorded from a real vehicle during specific message injection attacks. This dataset not only contains standard messages but also features three distinct categories of attack injections: DoS, fuzzy, and spoofing attacks (e.g., Gear and RPM). Our analysis centered on these diverse attack scenarios, and the subsequent sections detail the outcomes of this evaluation.

The dataset comprises a series of CAN messages, of which each accompanied by a specific label. The structural components of a message include a recorded timestamp, CAN ID, data length, and D[0] to D[7], which represent the actual data payload. In this dataset, all messages that are part of the injection attacks are marked with a 'T', indicating an attack, whereas messages not involved in these attacks are labeled with an 'R', signifying their normal status. To adapt the model input, preprocessing is a necessary step. The process began with the zero-filling method to address and fill any missing data points within the messages. The CAN IDs present in each message, originally in hexadecimal format, were converted to decimal to maintain consistency across the dataset. Given the 'timestamp' attribute's propensity to correlate with periods of cyber-attack simulations, which may lead to bias, it was replaced by the interval between consecutive messages. In addition, we utilize the Synthetic Minority Oversampling Technique (SMOTE) to address the imbalance typically found between normal and attack samples within the dataset. This method augments the dataset by generating synthetic samples from the minority class, calculated as follows.

$$X_n = X + \text{rand}(0,1) \times (X_i - X), \tag{1}$$

Where X is a minority class sample, X_i is a sample randomly chosen from the k-nearest neighbors of X, and X_n is the synthetic instance created. Finally, all features

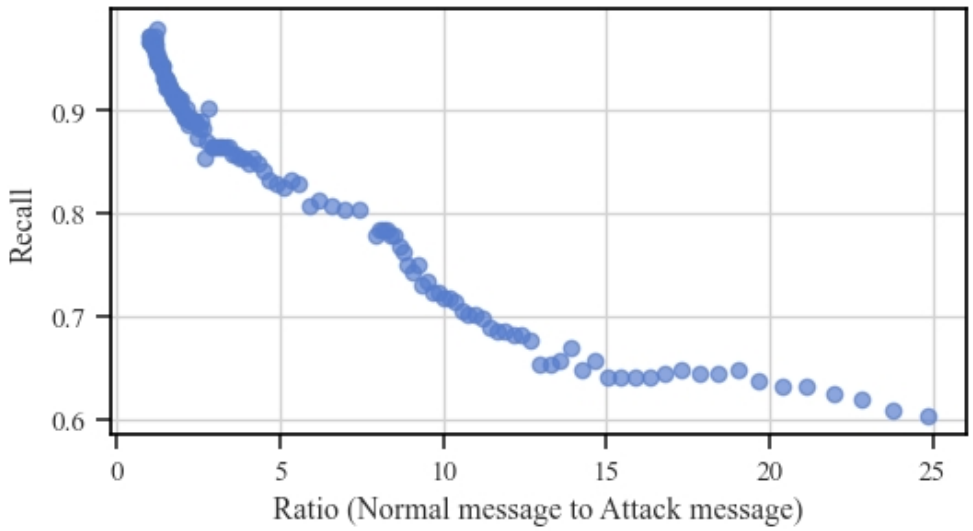

Fig. 4: Impact of Normal-to-Attack Message Ratio on Anomaly Detection Recall Rate.

of each message in the dataset were normalized using the Z-score method, which standardizes the features to a mean of 0 and a standard deviation of 1. The normalization formula is given by

$$x_n = \frac{x - \mu}{\sigma}, \tag{2}$$

where μ denotes the original feature value, μ is the mean of the feature values, and σ is their standard deviation. These preprocessing steps were essential in curating a dataset conducive to training a robust anomaly detection model.

4.1.1 Insidious Attack Modelling

The proposed method should detect low-volume attacks, which have always been neglected by previous works. Therefore, we play the attack role to launch attacks with fewer intrusion messages and generate a multitude of tasks, essentially a variety of attack scenarios to tailor the objective for bi-level optimization-based meta-learning. Specifically, we divided the data about the four principal attack scenarios into separate subsets. Then, normal and anomalous messages for each attack scenario are segregated and evenly distributed across 4,096 split datasets. After that, we generated 16,384 tasks, which are specifically configured to serve as individual tasks for the bi-level optimization-based meta-learning, enabling the model to learn from a diverse range of attack patterns and scenarios. This extensive preparation ensures that the meta-learning algorithm has a broad and representative sample of tasks to learn from, which is crucial for its ability to generalize and adapt to new, unknown attacks.

To accurately quantify the volume of attack traffic within a CAN frame stream, we introduce a metric named Attack Prominence Score (APS), which is defined as the ratio of normal to attack messages that correspond to these recall rate thresholds, calculated as follows.

$$APS = \frac{\sum_{i=0}^{n} \mathbb{I}(y_i = T)}{N} = \begin{cases} \text{High}, & \text{if } > \sigma \\ \text{Low}, & \text{if } < \delta \\ \text{Middle}, & \text{else} \end{cases}, \tag{3}$$

where σ and δ are the recall rate thresholds to determine the level of attack volume. Specifically, the APS is designed to gauge whether the traffic volume is indicative of a high-volume, medium-volume, or low-volume attack. The underlying principle of this metric is the assumption that the more concealed the attack traffic within a flow, the harder it is for the model to learn and identify its features. Consequently, this results in a higher false alarm for anomalous messages during intrusion detection. To search the threshold of APS, we introduce a neural network architecture constructed from a simple yet effective model (e.g., the combination of LSTM and a linear layer). This model was trained and evaluated on a dataset randomly chosen from a pool of preprocessed subsets, which had been divided into training, validation, and test sets. The model's performance was rigorously assessed by progressively decreasing the number of attack samples in the training set. Analysis of the model's recall rate was conducted by observing changes in the recall as the ratio of normal to attack samples varied in the training data.

As shown in Fig. 4, there is a clear trend that the recall rate declines as the ratio of normal to attack messages increases. Thus, we established a recall rate of $\delta = 80\%$ as the threshold for differentiating between medium and low attack volume, and $\sigma = 95\%$ for distinguishing high from medium volume. Accordingly, a normal to abnormal message ratio of 7 signifies the threshold between medium and low volume, while a ratio of 1.5 serves as the threshold between high and medium volume.

4.2 Bi-Level Optimization-based Meta Learning

In our study, we present a bi-level optimization framework based on the Meta-SGD algorithm to advance the IDS for vehicular networks. This approach not only identifies the best initial parameters for the models but also optimizes the learning rates. This dual focus is essential for our IDS to rapidly adapt to the wide array of attack patterns that characterize vehicular network traffic.

4.2.1 Outer-Level Optimization

At the outer level, our optimization is focused on adjusting the meta-parameters that govern the learning process. This includes the high-level adjustments that affect the learning trajectory for our neural network model when encountering new types of attacks or anomalies within the network traffic. The objective at this level is to optimize the model's performance across a variety of tasks, reflecting diverse attack scenarios in vehicular networks. This way, the model becomes adept at quickly adapting to new, unseen threats with minimal additional training.

The global optimization aim at this level is to discover a set of meta-parameters that are well-suited for task-specific adaptations but also robust enough to act as a solid foundation for the model across different tasks. The proposed model updates both the initial parameters and the learning rates by minimizing the meta-objective function, which is defined as follows:

$$\Theta^*, \alpha^* = \arg\min_{\Theta,\alpha} \sum_{\tau \in \mathcal{T}} \mathcal{L}_\tau(f_\Theta - \alpha \nabla \mathcal{L}_\tau(f_\Theta)), \tag{4}$$

where Θ^* represents the optimal initial parameters, and α^* denotes the set of optimal learning rates. The symbol $\mathcal{T}$ denotes the set of tasks, and $\mathcal{L}_\tau$ is the loss function associated with task τ. The function f_Θ is the predictive model parameterized by Θ, and $\nabla \mathcal{L}_\tau(f_\Theta)$ represents the gradient of the loss function with respect to the model parameters. The optimization is performed over both the initial parameters Θ and the learning rates α, aiming to find the combination that minimizes the sum of task-specific losses, thereby enhancing the model's adaptability across a diverse set of attack scenarios. This strategic optimization ensures that the IDS maintains its performance and robustness when deployed in real-world vehicular networks.

Algorithm 1 Bi-Level Optimization-based Meta-Learning Based on Meta-SGD

Randomly initialize meta-parameters Θ and learning rates α.
while not converged on average loss of batches of tasks **do**
    Initialize total gradients for meta-parameters $\Delta\Theta \leftarrow 0$ and for learning rates $\Delta\alpha \leftarrow 0$.
    Sample batch of tasks $\mathcal{B} = \{\tau_1, ..., \tau_J\}$ from task distribution $\mathcal{P}(\mathcal{T})$.
    for each task $\tau_j \in \mathcal{B}$ **do**
        Split data for task τ_j into support set $\mathcal{S}_{\tau_j}$ and query set $\mathcal{Q}_{\tau_j}$.
        Evaluate task-specific loss on support set $\mathcal{L}_{\tau_j}(f_\Theta, \mathcal{S}_{\tau_j})$.
        Compute gradients on support set $\nabla_\Theta \mathcal{L}_{\tau_j}(f_\Theta, \mathcal{S}_{\tau_j})$.
        Update task-specific parameters: $\Theta'_{\tau_j} \leftarrow \Theta - \alpha \nabla_\Theta \mathcal{L}_{\tau_j}(f_\Theta, \mathcal{S}_{\tau_j})$.
        Evaluate task-specific loss on query set $\mathcal{L}_{\tau_j}(f_{\Theta'_{\tau_j}}, \mathcal{Q}_{\tau_j})$.
        Compute gradients with respect to parameters on the query set $\nabla_\Theta \mathcal{L}_{\tau_j}(f_{\Theta'_{\tau_j}}, \mathcal{Q}_{\tau_j})$.
        Compute gradients with respect to learning rates on the query set $\nabla_\alpha \mathcal{L}_{\tau_j}(f_{\Theta'_{\tau_j}}, \mathcal{Q}_{\tau_j})$.
        Sum batch gradients: $\Delta\Theta \leftarrow \Delta\Theta + \nabla_\Theta \mathcal{L}_{\tau_j}(f_{\Theta'_{\tau_j}}, \mathcal{Q}_{\tau_j})$.
        Sum batch gradients: $\Delta\alpha \leftarrow \Delta\alpha + \nabla_\alpha \mathcal{L}_{\tau_j}(f_{\Theta'_{\tau_j}}, \mathcal{Q}_{\tau_j})$.
    end for
    Update meta-parameters using total batch gradients:
    $\Theta \leftarrow \Theta - \beta\Delta\Theta$.
    Update learning rates using total batch gradients:
    $\alpha \leftarrow \alpha - \gamma\Delta\alpha$.
end while
return Optimized meta-parameters Θ^* and learning rates α^*.

4.2.2 Inner-Level Optimization

Following the outer-level meta-parameter optimization, the inner-level optimization involves the rapid and task-specific adaptation of the model. Using the optimized learning rates, our defense framework fine-tunes the model parameters for each distinct task, which corresponds to a particular type of network attack.

Specifically, in the Inner-Level optimization of our Meta-Learning framework, each task τ_j extracted from the distribution $P(\mathcal{T})$ is subjected to an expedited learning process, tailor-made to enhance the model's performance on that specific task. This process is underpinned by the task-specific parameters Θ', which are fine-tuned from the initial meta-parameters Θ.

For each task τ_j, the Inner-Level learning commences with the calculation of the task-specific loss $L_{\tau_j}(f_\Theta, S_{\tau_j})$ on the support set S_{τ_j}. Following this, we compute the gradient of this loss with respect to the parameters $\nabla_\Theta L_{\tau_j}(f_\Theta, S_{\tau_j})$, which serves as the basis for the parameter updates. These updates are applied to derive the task-adapted parameters Θ'_{τ_j}, according to the equation 5, where α is the task-specific learning rate.

$$\Theta'_{\tau_j} = \Theta - \alpha \nabla_\Theta L_{\tau_j}(f_\Theta, S_{\tau_j}), \tag{5}$$

The updated parameters Θ'_{τ_j} are then employed to evaluate the model on the query set Q_{τ_j}, resulting in a task-specific loss that reflects the model's adaptability after the rapid learning step. It is this loss on the query set that is then used to compute further gradients, which influence the meta-parameters in the Outer-Level optimization. The strategic use of both support and query sets during this Inner-Level optimization ensures that the model not only learns the specifics of the task at hand but also generalizes well across all tasks in the task distribution $P(\mathcal{T})$.

In Algorithm 1, we present the process of the bi-Level optimization-based meta-learning. The combination of outer and inner-level optimizations in our bi-level framework allows the IDS to demonstrate superior adaptability and detection accuracy. It effectively learns to identify and respond to new and evolving threats, a crucial feature in the ever-changing domain of network security.

4.3 Model Structures

In our investigative framework, we have integrated four distinct neural network configurations to serve as the foundational model structures: These include (1) a single-layer LSTM, designated as M1; (2) a combination of a CNN preceding a single-layer LSTM, identified as M2; (3) a double-layered stacked LSTM, referred to as M3; and (4) a more complex arrangement where a CNN precedes a two-layer stacked LSTM, known as M4. Each LSTM layer across the models is designed with recurrent units to extract the temporal feature effectively from CAN messages. For the CNN-LSTM based models, the spatio-temporal features of CAN messages are modeled. To finalize the architectures, a trio of dense layers is incorporated, with the neuron count across these layers to map the valuable features to label space. The model inputs comprise time intervals between successive IDs, CAN ID, DLC, and the data fields, encompassing a sequence

width of 11, which is optimal for capturing the nuances across all potential anomaly patterns within the traffic.

The aforementioned model acts as a foundation model. Its meta-parameters are trained upon various tasks that encompass a breadth of attack scenarios in vehicular networks during outer-level optimization. Concurrently, in the inner-level optimization, the model undergoes rapid adaptation for each specific task. This involves fine-tuning the task-specific parameters to minimize the loss for that task, thereby enabling the model to quickly learn the peculiarities of each type of attack. we consider the cross-entropy as the optimization function. The architecture combines CNN to capture spatial features indicative of anomalies within individual messages, and LSTM layers to model the temporal patterns across sequences of messages. This part of the proposed scheme is presented in Fig. 3.

5 Performance Analysis

In this section, we present the experiment setting and performance analysis. The experimental results are devoted to proving the effectiveness of Meta-IDS and its superiority over previous DL-based IDS by task-adaption and low-volume attack detection.

5.1 Experiment Setting

To develop the proposed Meta-IDS, the preprocessing and Meta-learning algorithms were implemented using Pandas [43], Scikit-learn [44], and PyTorch [45] libraries in Python.[1] The experiments were carried out on a server with AMD EPYC 7542(32-Core, 2.90GHz), 256 Gigabytes(GB) of memory, and an NVIDIA GeForce RTX 3090 GPU for model training and testing and a Raspberry Pi 4 machine with a BCM2711 64-bit CPU and 4GB of memory for vehicle-level model evaluation. To evaluate our performance, we report the accuracy, precision, recall, F1-score, and CPU resource usage on the vehicle-level machine for the considered scenarios and models. The metrics are given by the following formulas:

$$Accuracy = \frac{\mathcal{TP} + \mathcal{TN}}{\mathcal{TP} + \mathcal{TN} + \mathcal{FP} + \mathcal{FN}}, \tag{6}$$

$$Precision = \frac{\mathcal{TP}}{\mathcal{TP} + \mathcal{FP}}, \tag{7}$$

$$Recall = \frac{\mathcal{TP}}{\mathcal{TP} + \mathcal{FN}}, \tag{8}$$

$$F1 - score = \frac{2 * Precision * Recall}{Precision + Recall}, \tag{9}$$

where a True Positive (TP) refers to an attack sequence that is correctly identified as such. A False Positive (FP) occurs when a genuine sequence is mistakenly classified as an attack. A False Negative (FN) happens when an attack sequence is incorrectly

[1] The code for the major modules is available at:https://github.com/335659554/Meta-IDS

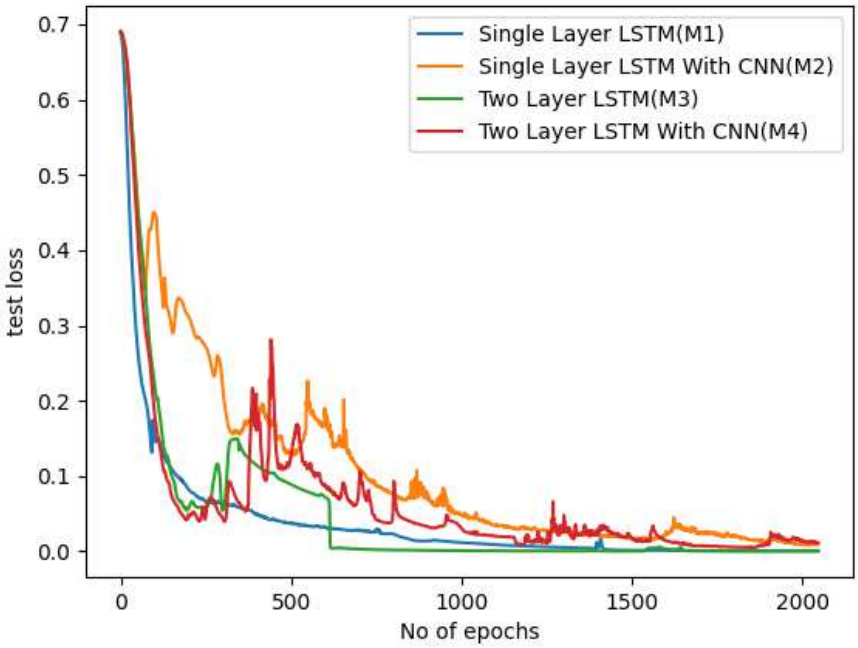

Fig. 5: An overview of the proposed framework.

classified as genuine. Finally, a True Negative (TN) refers to a normal sequence that is correctly identified as genuine. Besides, the false alarm is a significant factor for DL-based IDS consideration, in which frequent alarms affect normal driving or neglect potential attack messages. Thus, we define the false positive rate (FPR) and false negative rate (FNR) to report the performance of Meta-IDS in terms of false alarms, calculated as follows.

$$FNR = \frac{\mathcal{FN}}{\mathcal{TP} + \mathcal{FN}}, \tag{10}$$

$$FPR = \frac{\mathcal{FP}}{\mathcal{TN} + \mathcal{FP}}, \tag{11}$$

5.2 Optimization Analysis

Our foundation models undergo a rigorous bi-level optimization meta-learning process, tailored to enhance their learning capabilities across a spectrum of intrusion detection tasks. These tasks are derived from the preprocessing stage, with 80% (13,108 tasks) allocated for training. This training equips the models with the meta-parameters necessary for rapid adaptation to diverse attack scenarios. The remaining 20% (3,276 tasks) serve as a testbed to evaluate the models' quick adaptation skills.

Table 2: Performance on various attack scenarios for M1 to M4

	M1			
Metric	DoS	Fuzzy	Gear spoofing	RPM spoofing
Accuracy(%)	100.0 (0.0)	100.0 (0.0)	98.44 (0.0022)	100.0 (0.0)
Precision(%)	100.0 (0.0)	100.0 (0.0)	96.66 (0.0100)	100.0 (0.0)
Recall(%)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)
F1-score(%)	100.0 (0.0)	100.0 (0.0)	98.00(0.0036)	100.0 (0.0)
FPR(%)	0.0 (0.0)	0.0 (0.0)	2.27 (0.0046)	0.0 (0.0)
FNR(%)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)

	M2			
Metric	DoS	Fuzzy	Gear spoofing	RPM spoofing
Accuracy(%)	99.95 (0.0)	99.95 (0.0)	99.94 (0.0146)	100.0 (0.0)
Precision(%)	99.91 (0.0)	100.0 (0.0)	99.81 (0.0)	100.0 (0.0)
Recall(%)	100.0 (0.0)	99.86 (0.0)	100.0 (0.0)	100.0 (0.0)
F1-score(%)	99.96 (0.0)	99.93 (0.0)	99.91 (0.0)	100.0 (0.0)
FPR(%)	0.11 (0.0)	0.0 (0.0)	0.09 (0.0)	0.0 (0.0)
FNR(%)	0.0 (0.0)	0.14 (0.0)	0.0 (0.0)	0.0 (0.0)

	M3			
Metric	DoS	Fuzzy	Gear spoofing	RPM spoofing
Accuracy(%)	100.0 (0.0)	75.71 (0.0101)	100.0 (0.0)	100.0 (0.0)
Precision(%)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)
Recall(%)	100.0 (0.0)	27.54 (0.0464)	100.0 (0.0)	100.0 (0.0)
F1-score(%)	100.0 (0.0)	39.37 (0.0527)	100.0 (0.0)	100.0 (0.0)
FPR(%)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)
FNR(%)	0.0 (0.0)	72.46 (0.0464)	0.0 (0.0)	0.0 (0.0)

	M4			
Metric	DoS	Fuzzy	Gear spoofing	RPM spoofing
Accuracy(%)	99.95 (0.0)	99.46 (0.0)	95.93 (0.0132)	100.0 (0.0)
Precision(%)	99.91 (0.0)	100.0 (0.0)	88.89 (0.0988)	100.0 (0.0)
Recall(%)	100.0 (0.0)	98.56 (0.0003)	88.89 (0.0988)	100.0 (0.0)
F1-score(%)	99.96 (0.0)	99.27 (0.0)	88.89 (0.0988)	100.0 (0.0)
FPR(%)	0.11 (0.0)	0.0 (0.0)	0.87 (0.0006)	0.0 (0.0)
FNR(%)	0.0 (0.0)	1.44 (0.0003)	11.11 (0.0988)	0.0 (0.0)

We employ a task-specific split within each inner-level optimization, designating 80% of the data for training and the remaining 20% for testing. The depicted meta-learning curves illustrate the progression of our models' performance, as reflected by the average test loss across all tasks for tests. This loss decreases with increasing epochs, evidencing the successful tuning of meta-parameters and the models' improved adaptability to new and varied attack scenarios, as shown in Fig. 5.

5.3 Comparative Analysis of the Fundamental Models

This section presents a comparative analysis of the performance of four fundamental meta-models, each subjected to a rigorous training regimen through our bespoke meta-learning optimization framework. The models were assessed across four primary

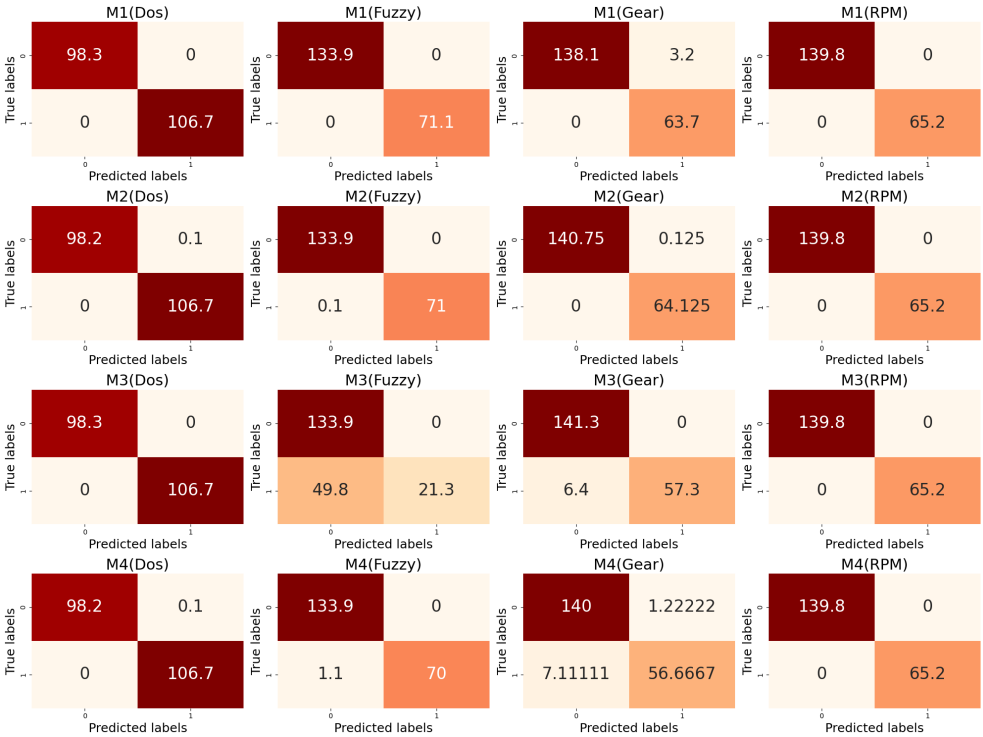

Fig. 6: Confusion matrices for various attack scenarios.

attack vectors within our benchmark dataset. Ten independent tasks, representing unique distributions of attack scenarios for each attack type, were derived by randomly selecting 10 continuous samples, each sample spanning 1024 time steps.

For each task, datasets were partitioned into training and testing sets with an 80:20 split to evaluate the models' rapid adaptability and intrusion detection capabilities across diverse attack scenarios. The mean performance metrics - Accuracy, Precision, Recall, F1-score, False Positive Rate (FPR), and False Negative Rate (FNR) - and their variances are computed over 10 independent experiments for each attack type and are tabulated as shown in Table 2.

For the DoS attack scenarios, M1 and M3 report perfect accuracy, precision, recall, and F1-score of 100%, while M2 shows slightly lower accuracy and precision at 99.95% and 99.91% respectively. M4 closely matches M2 with an accuracy of 99.95% and precision of 99.91%. M2 and M4 have a minor FPR of 0.11%.

In the Fuzzy attack scenarios, M1 and M2 again demonstrate perfect or near-perfect performance across all metrics. However, M3 significantly underperforms in recall (27.54%) and F1-score (39.37%), suggesting a high rate of missed detections (FNR of 72.46%). M4 maintains high performance but with a slight decrease in accuracy compared to M1 and M2.

Gear spoofing presents more challenging scenarios, M1's accuracy slightly drops to 98.44%, with precision at 96.66% and an F1-score of 98.00%. M2 maintains its strong performance with an accuracy of 99.94% and precision of 99.81%. M4 shows a more significant drop, with accuracy at 95.93% and precision at 88.89%, indicating difficulties in accurately detecting gear spoofing attacks.

Finally, in the RPM spoofing scenarios, all models perform exceptionally well, with M1, M2, M3, and M4 all achieving perfect or near-perfect metrics across the board.

Upon reviewing the specific performance metrics for each model across different attack scenarios, it becomes evident that M1 stands out due to its consistently high performance. With perfect scores in accuracy, precision, recall, and F1-score across most scenarios, M1 exhibits unparalleled effectiveness in detecting various types of cyber-attacks. While other models like M2, M3, and M4 show strong capabilities in certain areas, they each have scenarios where their performance slightly dips. Specifically, M1's ability to maintain a 0% false positive rate (FPR) and false negative rate (FNR) across almost all scenarios highlights its superiority. Given the data, M1 emerges as the best model overall, offering robust and reliable intrusion detection across a wide range of attack vectors.

The confusion matrices drawn in Fig. 7 show that Model M1 achieves perfect performance in DoS, Fuzzy, and RPM spoofing scenarios with zero false positives or negatives. M2, while also strong, exhibits a slight imperfection with a minimal false positive in Gear spoofing. M3 demonstrates perfect accuracy in DoS and RPM spoofing but struggles with Fuzzy attacks, as indicated by false negatives. M4 mirrors this high accuracy in DoS and RPM spoofing but has a noticeable number of false positives and negatives in Gear spoofing, suggesting areas for improvement. Across all scenarios, M1's consistent precision in predicting true attacks and non-attacks solidifies its robustness within the Meta-IDS framework.

Upon examination of the experimental results presented in Table. 2, a pivotal conclusion emerges regarding the adaptability of intrusion detection systems (IDS) within the context of rapid learning, particularly when few-shot learning is employed to enable the system to swiftly adjust to various attack scenarios. It is observed that ***an increase in the complexity of the model does not necessarily translate to enhanced performance***. In scenarios characterized by a minimal amount of traffic data available for the meta-model to perform inner-level adaptation, the augmentation of model complexity might, counterintuitively, lead to a degradation in model efficacy. This suggests that within the confines of rapid adaptation where data is scarce, streamlined models may be more advantageous, emphasizing the importance of model design that aligns with the constraints and objectives of few-shot learning.

5.4 Adaption Capability on Volume

Our study extends beyond the initial adaptability evaluation to examine the performance of our Meta-IDS framework under low-volume attack scenarios. These scenarios are especially challenging due to the subtler signatures of malicious activity, which are harder to distinguish from normal behavior.

To assess the adaptability of each meta-model (M1, M2, M3, and M4) that underwent bi-level meta-learning training, we initially extracted ten sample segments from

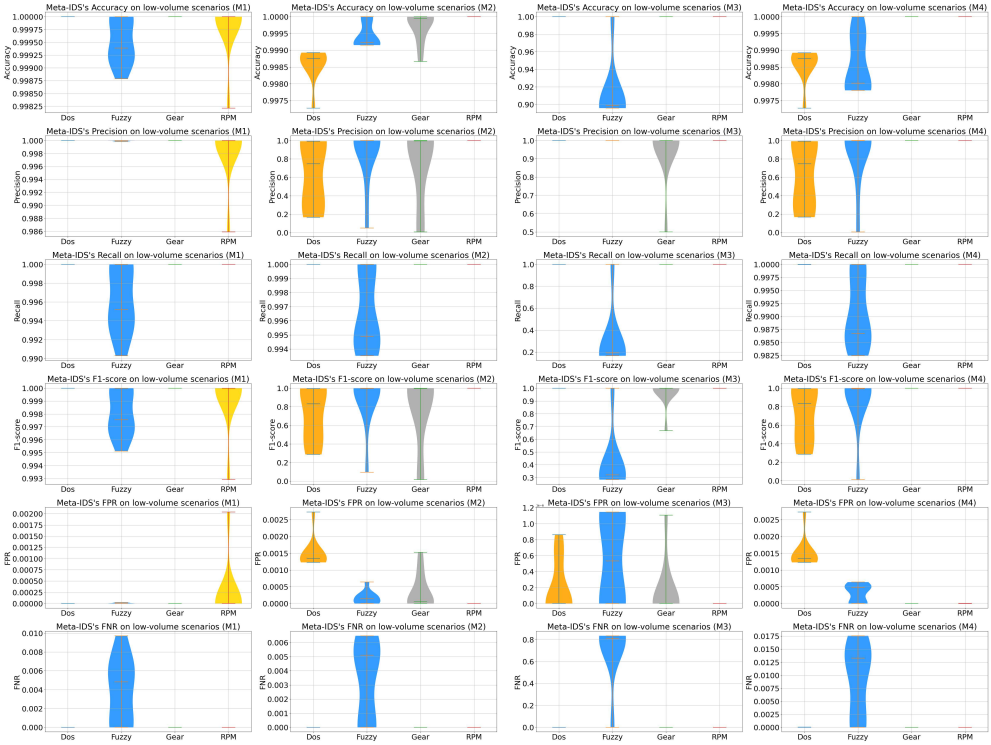

Fig. 7: Violin plots for various low-volume attack scenarios.

each attack dataset, representing ten distinct attack scenarios. Each model then underwent an inner-level optimization on the training dataset of one sample segment to adapt to the corresponding attack scenario. Consequently, for each type of attack, we derived ten M_{inner} models for our experiments.

In a rigorous evaluation designed to mimic the real-world sparse nature of intrusions, we randomly chose one M_{inner} model per attack type to undergo testing against ten low-volume attack sample segments. These low-volume segments conform to the criteria we defined in Section 4.1.1, presenting a stringent test of detection capabilities.

The violin plots in Fig. 7 provide a comprehensive visualization of the performance metrics for models M1 through M4 within our Meta-IDS framework across various low-volume attack scenarios. These plots highlight the models' accuracy, precision, recall, F1-score, and error rates, allowing us to discern not only average performance but also the variability and consistency of each model. M1 exhibits narrow distributions in accuracy and precision, indicating high and consistent performance, especially in DoS and Gear scenarios. M2 and M4 show broader distributions in the FPR and FNR for the Gear and Fuzzy scenarios, reflecting a degree of variability in performance. M3, while generally consistent, indicates potential for improvement in accuracy across Fuzzy attacks.

Table 3: Performance comparison with existing approaches for individual attack types

	DoS					
Metric	NB[7]	DT[7]	MLP[7]	CNN[7]	Song et. al[15]	**Meta-IDS**
Precision(%)	74.32	85.30	88.06	91.26	100.0	**100.0**
Recall(%)	98.67	97.64	97.95	98.44	99.89	**100.0**
F1-score(%)	84.78	91.05	92.74	94.71	99.95	**100.0**
	Fuzzy					
Metric	NB[7]	DT[7]	MLP[7]	CNN[7]	Song et. al[15]	**Meta-IDS**
Precision(%)	16.62	81.04	86.17	94.14	99.95	**100.0**
Recall(%)	99.99	96.36	97.77	99.22	99.65	**100.0**
F1-score(%)	28.50	88.04	91.60	96.61	99.80	**100.0**
	Gear Spoofing					
Metric	NB[7]	DT[7]	MLP[7]	CNN[7]	Song et. al[15]	**Meta-IDS**
Precision(%)	82.55	84.17	88.25	93.87	**99.99**	98.44
Recall(%)	97.66	96.64	98.03	98.69	**99.89**	96.66
F1-score(%)	89.47	89.98	92.88	96.22	**99.94**	98.00
	RPM Spoofing					
Metric	NB[7]	DT[7]	MLP[7]	CNN[7]	Song et. al[15]	**Meta-IDS**
Precision(%)	81.14	81.66	92.86	95.29	99.99	**100.0**
Recall(%)	93.07	94.29	93.99	96.05	99.94	**100.0**
F1-score(%)	86.70	87.52	93.42	95.67	99.96	**100.0**

5.5 Comparison With Existing Works

We compare the results of our proposed methods with four other techniques on the metrics of Precision, Recall, and F1-Score on the Car-Hacking Dataset in Table 3 for each attack type. The comparison is shown with existing works which had implemented Naive Bayes(NB), Decision Trees(DT), Multilayer perceptron (MLP), CNN in [7], and an IDS using deep convolutional neural network(DCNN) proposed by Song et. al[15]. In Table 3, we can observe that our approach receives the best F1-score for all types of attacks, which means that our approach demonstrates the ability to effectively detect intrusions while also minimizing false alarms.

As shown in Table 3, the Meta-IDS method demonstrates outstanding performance across various attack types, often outperforming existing methods in precision, recall, and F1-score. However, in the context of Gear Spoofing attacks, while Meta-IDS shows superior recall, its precision is slightly outmatched by Song et al.'s approach, which presents an opportunity for further enhancement. Crucially, it also exhibits a notable capacity to detect low-volume attacks, which is a critical advancement over existing works, ensuring both precision and breadth in our cybersecurity measures.

5.6 Vehicle-Level Model Evaluation

To verify the real-world applicability of our Meta-IDS framework, we have rigorously evaluated the M_{inner} models, each representing a version of the Meta-IDS that has

Table 4: Model evaluation on a vehicle-level system

System Component	Avg Test Time(ms)	Model Space(MB)
Z-score	0.189	0.001
Detection model	4.433	0.019
Total	**4.623**	**0.020**

been fine-tuned through inner-level optimization for a distinct attack type. These evaluations were carried out on a Raspberry Pi 4, a computing platform akin to an in-vehicle environment, to ascertain the operational feasibility of each M_{inner} model within the computational constraints of vehicular systems.

In preparation for real-world deployment, the Meta-IDS was benchmarked against the rigorous real-time requirements of critical vehicle safety services. The U.S. Department of Transportation mandates that essential functions, such as collision and attack warnings, must maintain a latency no greater than 10 to 100 ms [46]. Our system is engineered to process each packet within the vehicular network promptly, ensuring that alarm generation times remain below the 10 ms threshold. Each packet is rapidly processed via Z-score normalization and evaluated through our detection model, which has been finely tuned through an inner-level optimization process. As substantiated by the results in Table 4, the Meta-IDS boasts an average processing time of just 4.623 ms per packet on the CAR-Hacking dataset—significantly below the latency requirement. Furthermore, the memory footprint for the normalization and detection processes is exceptionally light, consuming only 0.001 MB and 0.019 MB respectively, which is negligible compared to the more than 1 GB RAM available on vehicle-level machines like the Raspberry Pi 4. This exemplary performance illustrates the Meta-IDS's potential for seamless integration into vehicular networks, delivering robust security without compromising on efficiency or speed.

5.7 Discussion and limitation

This research propels forward the field of vehicular network security by introducing Meta-IDS, an intrusion detection system that utilizes Meta-SGD, an optimized meta-learning approach with an adaptable learning rate. Our study is pioneering in its focus on low-volume attack scenarios, which are often overlooked yet represent a critical vulnerability in vehicular networks.

Our findings demonstrate that Meta-IDS achieves excellent performance in adapting to and detecting intrusions in various attack scenarios, characterized by both regular and low-volume traffic. The adaptability of Meta-IDS is particularly notable, as it showcases the system's capacity to rapidly acclimate to the unique patterns of each attack type, a significant advancement in the domain of cybersecurity.

Despite its strengths, Meta-IDS, like any system grounded in supervised learning, faces challenges in detecting attacks that diverge from learned patterns. This is particularly true for sophisticated cyber threats that manifest as novel, low-volume attacks, which may not trigger the learned detection mechanisms effectively. To address this limitation, future research must explore unsupervised or semi-supervised

learning strategies, potentially integrating adversarial training to enhance the model's generalization capabilities.

Moreover, the successful deployment of Meta-IDS in a real-world setting necessitates the collaboration with automotive industry stakeholders. Such partnerships would enable the integration of semantic analysis and manufacturer-specific insights into the IDS, further refining its accuracy and effectiveness.

In conclusion, while Meta-IDS marks a substantial leap in intrusion detection for vehicular networks, it is not without areas for development. Enhancing its capabilities to detect novel and sophisticated attack vectors remains a crucial area for future work. Additionally, optimizing computational efficiency for deployment in resource-constrained vehicular systems will be essential to ensure that Meta-IDS can operate effectively in real-time environments.

6 Conclusion

In this work, we introduced Meta-IDS, a groundbreaking intrusion detection system tailored for vehicular networks that harnesses the power of meta-learning with Meta-SGD to learn and adapt swiftly to a variety of attack scenarios. Distinguished by its vehicle-level evaluation, Meta-IDS has been meticulously validated on a Raspberry Pi 4, mimicking the computational constraints and operational realities of in-vehicle environments. Our approach not only demonstrated superior detection capabilities across known attack scenarios but also advanced the field by focusing on low-volume attacks, which are notoriously difficult to detect and have been largely neglected in previous studies.

The Meta-IDS framework excelled in fast adaptation and detection accuracy, operating well within the latency thresholds critical for vehicular safety services. With an average processing time of 4.623 ms per packet and a negligible memory footprint, the system proved suitable for real-time detection, ensuring robust security without compromising on computational efficiency.

7 ACKNOWLEDGMENT

This work was supported in part by the Intelligent Driving Operating System Basic Software Project (CEIEC-2023-ZM02-0106) of the Ministry of Industry and Information Technology of China.

References

[1] Rajapaksha, S., Kalutarage, H., Al-Kadri, M.O., Petrovski, A., Madzudzo, G., Cheah, M.: AI-Based Intrusion Detection Systems for In-Vehicle Networks: A Survey. ACM Computing Surveys **55**(11), 1–40 (2023) https://doi.org/10.1145/3570954

[2] Avatefipour, O., Malik, H.: State-of-the-Art Survey on In-Vehicle Network Communication CAN-Bus Security and Vulnerabilities. International Journal of Computer Science and Network **6**(6), 720–727 (2017)

[3] Aliwa, E., Rana, O., Perera, C., Burnap, P.: Cyberattacks and Countermeasures for In-Vehicle Networks. ACM Computing Surveys **54**(1), 1–37 (2022) https://doi.org/10.1145/3431233

[4] Han, M.L., Kwak, B.I., Kim, H.K.: Event-Triggered Interval-Based Anomaly Detection and Attack Identification Methods for an In-Vehicle Network. IEEE Transactions on Information Forensics and Security **16**, 2941–2956 (2021) https://doi.org/10.1109/TIFS.2021.3069171

[5] Wang, Q., Qian, Y., Lu, Z., Shoukry, Y., Qu, G.: A Delay based Plug-in-Monitor for Intrusion Detection in Controller Area Network. In: 2018 Asian Hardware Oriented Security and Trust Symposium (AsianHOST), pp. 86–91. IEEE, Hong Kong (2018). https://doi.org/10.1109/AsianHOST.2018.8607178

[6] Wu, F., Li, T., Wu, Z., Wu, S., Xiao, C.: Research on Network Intrusion Detection Technology Based on Machine Learning. International Journal of Wireless Information Networks **28**(3), 262–275 (2021) https://doi.org/10.1007/s10776-021-00520-z

[7] Lo, W., Alqahtani, H., Thakur, K., Almadhor, A., Chander, S., Kumar, G.: A hybrid deep learning based intrusion detection system using spatial-temporal representation of in-vehicle network traffic. Vehicular Communications **35**, 100471 (2022) https://doi.org/10.1016/j.vehcom.2022.100471

[8] Desta, A.K., Ohira, S., Arai, I., Fujikawa, K.: Rec-CNN: In-vehicle networks intrusion detection using convolutional neural networks trained on recurrence plots. Vehicular Communications **35**, 100470 (2022) https://doi.org/10.1016/j.vehcom.2022.100470

[9] Zhang, J., Li, F., Zhang, H., Li, R., Li, Y.: Intrusion detection system using deep learning for in-vehicle security. Ad Hoc Networks **95**, 101974 (2019) https://doi.org/10.1016/j.adhoc.2019.101974

[10] Khan, I.A., Moustafa, N., Pi, D., Haider, W., Li, B., Jolfaei, A.: An Enhanced Multi-Stage Deep Learning Framework for Detecting Malicious Activities From Autonomous Vehicles. IEEE Transactions on Intelligent Transportation Systems **23**(12), 25469–25478 (2022) https://doi.org/10.1109/TITS.2021.3105834

[11] Berger, I., Rieke, R., Kolomeets, M., Chechulin, A., Kotenko, I.: Comparative Study of Machine Learning Methods for In-Vehicle Intrusion Detection. In: Katsikas, S.K., Cuppens, F., Cuppens, N., Lambrinoudakis, C., Antón, A., Gritzalis, S., Mylopoulos, J., Kalloniatis, C. (eds.) Computer Security vol. 11387, pp. 85–101. Springer International Publishing, Cham (2019). https://doi.org/10.1007/978-3-030-12786-2_6

[12] Zhu, K., Chen, Z., Peng, Y., Zhang, L.: Mobile Edge Assisted Literal Multi-Dimensional Anomaly Detection of In-Vehicle Network Using LSTM. IEEE

Transactions on Vehicular Technology **68**(5), 4275–4284 (2019) https://doi.org/10.1109/TVT.2019.2907269

[13] Song, H.M., Kim, H.K.: Self-Supervised Anomaly Detection for In-Vehicle Network Using Noised Pseudo Normal Data. IEEE Transactions on Vehicular Technology **70**(2), 1098–1108 (2021) https://doi.org/10.1109/TVT.2021.3051026

[14] Song, H.M., Kim, H.R., Kim, H.K.: 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), pp. 63–68. IEEE, Kota Kinabalu, Malaysia (2016). https://doi.org/10.1109/ICOIN.2016.7427089

[15] Song, H.M., Woo, J., Kim, H.K.: In-vehicle network intrusion detection using deep convolutional neural network. Vehicular Communications **21**, 100198 (2020) https://doi.org/10.1016/j.vehcom.2019.100198

[16] Yahiatene, Y., Rachedi, A., Riahla, M.A., Menacer, D.E., Nait-Abdesselam, F.: A blockchain-based framework to secure vehicular social networks. Transactions on Emerging Telecommunications Technologies **30**(8), 3650 (2019) https://doi.org/10.1002/ett.3650

[17] Tandon, R., Verma, A., Gupta, P.K.: D-BLAC: A dual blockchain-based decentralized architecture for authentication and communication in VANET. Expert Systems with Applications **237**, 121461 (2024) https://doi.org/10.1016/j.eswa.2023.121461

[18] Marchetti, M., Stabili, D.: Anomaly detection of CAN bus messages through analysis of ID sequences. In: 2017 IEEE Intelligent Vehicles Symposium (IV), pp. 1577–1583. IEEE, Los Angeles, CA, USA (2017). https://doi.org/10.1109/IVS.2017.7995934

[19] Islam, R., Refat, R.U.D., Yerram, S.M., Malik, H.: Graph-Based Intrusion Detection System for Controller Area Networks. IEEE Transactions on Intelligent Transportation Systems **23**(3), 1727–1736 (2022) https://doi.org/10.1109/TITS.2020.3025685

[20] Wang, Q., Lu, Z., Qu, G.: An Entropy Analysis Based Intrusion Detection System for Controller Area Network in Vehicles. In: 2018 31st IEEE International System-on-Chip Conference (SOCC), pp. 90–95. IEEE, Arlington, VA (2018). https://doi.org/10.1109/SOCC.2018.8618564

[21] Avatefipour, O., Al-Sumaiti, A.S., El-Sherbeeny, A.M., Awwad, E.M., Elmeligy, M.A., Mohamed, M.A., Malik, H.: An Intelligent Secured Framework for Cyberattack Detection in Electric Vehicles' CAN Bus Using Machine Learning. IEEE Access **7**, 127580–127592 (2019) https://doi.org/10.1109/ACCESS.2019.2937576

[22] Kalutarage, H.K., Al-Kadri, M.O., Cheah, M., Madzudzo, G.: Context-aware

Anomaly Detector for Monitoring Cyber Attacks on Automotive CAN Bus. In: ACM Computer Science in Cars Symposium, pp. 1–8. ACM, Kaiserslautern Germany (2019). https://doi.org/10.1145/3359999.3360496

[23] Levi, M., Allouche, Y., Kontorovich, A.: Advanced Analytics for Connected Car Cybersecurity. In: 2018 IEEE 87th Vehicular Technology Conference (VTC Spring), pp. 1–7. IEEE, Porto (2018). https://doi.org/10.1109/VTCSpring.2018.8417690

[24] Moulahi, T., Zidi, S., Alabdulatif, A., Atiquzzaman, M.: Comparative Performance Evaluation of Intrusion Detection Based on Machine Learning in In-Vehicle Controller Area Network Bus. IEEE Access **9**, 99595–99605 (2021) https://doi.org/10.1109/ACCESS.2021.3095962

[25] Fenzl, F., Rieke, R., Dominik, A.: In-vehicle detection of targeted CAN bus attacks. In: Proceedings of the 16th International Conference on Availability, Reliability and Security, pp. 1–7. ACM, Vienna Austria (2021). https://doi.org/10.1145/3465481.3465755

[26] Ma, H., Cao, J., Mi, B., Huang, D., Liu, Y., Li, S.: A GRU-Based Lightweight System for CAN Intrusion Detection in Real Time. Security and Communication Networks **2022**, 1–11 (2022) https://doi.org/10.1155/2022/5827056

[27] Ale, L., King, S.A., Zhang, N.: Deep Bayesian Learning for Car Hacking Detection. arXiv (2021)

[28] Xiao, J., Wu, H., Li, X.: Internet of Things Meets Vehicles: Sheltering In-Vehicle Network through Lightweight Machine Learning. Symmetry **11**(11), 1388 (2019) https://doi.org/10.3390/sym11111388

[29] Shi, D., Xu, M., Wu, T., Kou, L.: Intrusion Detecting System Based on Temporal Convolutional Network for In-Vehicle CAN Networks. Mobile Information Systems **2021**, 1–13 (2021) https://doi.org/10.1155/2021/1440259

[30] Desta, A.K., Ohira, S., Arai, I., Fujikawa, K.: MLIDS: Handling Raw High-Dimensional CAN Bus Data Using Long Short-Term Memory Networks for Intrusion Detection in In-Vehicle Networks. In: 2020 30th International Telecommunication Networks and Applications Conference (ITNAC), pp. 1–7. IEEE, Melbourne, VIC, Australia (2020). https://doi.org/10.1109/ITNAC50341.2020.9315024

[31] Ashraf, J., Bakhshi, A.D., Moustafa, N., Khurshid, H., Javed, A., Beheshti, A.: Novel Deep Learning-Enabled LSTM Autoencoder Architecture for Discovering Anomalous Events From Intelligent Transportation Systems. IEEE Transactions on Intelligent Transportation Systems **22**(7), 4507–4518 (2021) https://doi.org/10.1109/TITS.2020.3017882

[32] Yang, L., Shami, A.: A Transfer Learning and Optimized CNN Based Intrusion Detection System for Internet of Vehicles. In: ICC 2022 - IEEE International Conference on Communications, pp. 2774–2779. IEEE, Seoul, Korea, Republic of (2022). https://doi.org/10.1109/ICC45855.2022.9838780

[33] Seo, E., Song, H.M., Kim, H.K.: GIDS: GAN based Intrusion Detection System for In-Vehicle Network. In: 2018 16th Annual Conference on Privacy, Security And Trust (PST), pp. 1–6. IEEE, Belfast (2018). https://doi.org/10.1109/PST.2018.8514157

[34] Sharafaldin, I., Habibi Lashkari, A., Ghorbani, A.A.: Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization:. In: Proceedings of the 4th International Conference on Information Systems Security And Privacy, pp. 108–116. SCITEPRESS - Science and Technology Publications, Funchal, Madeira, Portugal (2018). https://doi.org/10.5220/0006639801080116

[35] Iehira, K., Inoue, H., Ishida, K.: Spoofing attack using bus-off attacks against a specific ECU of the CAN bus. In: 2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC), pp. 1–4. IEEE, Las Vegas, NV (2018). https://doi.org/10.1109/CCNC.2018.8319180

[36] Bozdal, M., Samie, M., Aslam, S., Jennions, I.: Evaluation of can bus security challenges. Sensors **20**(8) (2020) https://doi.org/10.3390/s20082364

[37] GmbH, R.B. (ed.): CAN Specication Version 2.0. Bosch, Stuttgart (1991)

[38] Miller, C., Valasek, C.: Remote exploitation of an unaltered passenger vehicle. Black Hat USA **2015**(S 91), 1–91 (2015)

[39] Koscher, K., Czeskis, A., Roesner, F., Patel, S., Kohno, T., Checkoway, S., McCoy, D., Kantor, B., Anderson, D., Shacham, H., Savage, S.: Experimental Security Analysis of a Modern Automobile. In: 2010 IEEE Symposium on Security And Privacy, pp. 447–462. IEEE, Oakland, CA, USA (2010). https://doi.org/10.1109/SP.2010.34

[40] Zhou, D., Yan, Z., Fu, Y., Yao, Z.: A survey on network data collection. Journal of Network and Computer Applications **Volume 116**, 15 (2018) https://doi.org/10.1016/j.jnca.2018.05.004

[41] Tariq, S., Lee, S., Kim, H.K., Woo, S.S.: Detecting In-vehicle CAN Message Attacks Using Heuristics and RNNs. In: Fournaris, A.P., Lampropoulos, K., Marín Tordera, E. (eds.) Information and Operational Technology Security Systems vol. 11398, pp. 39–45. Springer International Publishing, Cham (2019). https://doi.org/10.1007/978-3-030-12085-6_4

[42] Balaji, P., Ghaderi, M.: NeuroCAN: Contextual Anomaly Detection in Controller Area Networks. In: 2021 IEEE International Smart Cities Conference (ISC2),

pp. 1–7. IEEE, Manchester, United Kingdom (2021). https://doi.org/10.1109/ISC253183.2021.9562830

[43] team, T.: Pandas-Dev/Pandas: Pandas. Zenodo (2023). https://doi.org/10.5281/zenodo.8092754

[44] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)

[45] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: PyTorch: An Imperative Style, High-Performance Deep Learning Library, pp. 8024–8035. Curran Associates, Inc., Vancouver, Canada. (2019). http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[46] Abualhoul, M.Y., Shagdar, O., Nashashibi, F.: Visible Light inter-vehicle Communication for platooning of autonomous vehicles. In: 2016 IEEE Intelligent Vehicles Symposium (IV), pp. 508–513. IEEE, Gotenburg, Sweden (2016). https://doi.org/10.1109/IVS.2016.7535434