

Article

In-Vehicle Ethernet Network Intrusion Detection Using a Feature Masking Algorithm

Yue Jia, Yihu Xu , Yujing Wu , Mingkui Li, Xingming Li and Yanan Xu * 

Department of Electronic & Communication Engineering, Yanbian University, Yanji 133002, China; 2024050064@ybu.edu.cn (Y.J.); xuyh@ybu.edu.cn (Y.X.); yjwu@ybu.edu.cn (Y.W.); mk_li7@ybu.edu.cn (M.L.); 1244022019@ybu.edu.cn (X.L.)

* Correspondence: ynxu@ybu.edu.cn

Abstract

With the fast-paced advancement of intelligent connected vehicles (ICVs), In-Vehicle Ethernet has gradually become the core of in-vehicle network communication systems, thanks to its superior high-bandwidth data transmission performance. However, the open nature of In-Vehicle Ethernet and the intricacy of its communication protocols have brought significant challenges to the practice of network intrusion detection for this technology. To tackle the problem of network intrusion detection in In-Vehicle Ethernet, this study takes into account the data characteristics of the AVTP protocol as well as common network attack approaches. We put forward a novel intrusion detection method based on a feature mask algorithm, which is designed to enhance the overall security level of In-Vehicle Ethernet. Experimental results show that the proposed algorithm can detect 99.5% of abnormal data in In-Vehicle Ethernet. Compared with traditional anomaly detection algorithms such as the Bayesian algorithm and decision tree method, the proposed method achieves detection rate increases of 12.4% and 7.8% respectively. Compared with the state-of-the-art CNN and XGBoost algorithms, the proposed method yields a relatively modest improvement in detection rate, while reducing inference latency by 81.6% and 21.8% respectively. These findings effectively boost the network security performance of In-Vehicle Ethernet and provide a reliable foundation for safeguarding the network security of intelligent connected vehicles.

Keywords: in-vehicle Ethernet; network security; intrusion detection; AVTP protocol

1. Introduction

As intelligent connected vehicles (ICVs) and autonomous driving technologies advance at a fast pace, in-vehicle entertainment systems, advanced driver assistance systems (ADAS), and autonomous driving systems have been widely adopted in practice. This development trend has strongly promoted technological innovations in the intelligence, connectivity, and automation of ICVs [1]. When electronic control units (ECUs) of intelligent connected vehicles link with external devices and networks including smartphones, Wi-Fi, Bluetooth, OBD II diagnostic tools, navigation systems and cellular networks, they improve user driving experience yet expose vehicles to hacking attacks and information leakage risks. These risks pose a serious threat to passenger safety and public road security; thus, it is crucial to accurately identify vehicular network attacks, so as to guarantee the security and stability of the in-vehicle network [2].

Automotive Ethernet boasts high throughput, flexible scalability, and excellent compatibility with time-sensitive networking protocols, making it the core of in-vehicle network



Academic Editor: Theodore E. Simos

Received: 27 May 2026

Revised: 24 June 2026

Accepted: 25 June 2026

Published: 28 June 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and

conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

communication systems [3]. It is responsible for collecting data and transmitting real-time information from multimedia and sensing devices, including radars, cameras, and dashcams. In addition, it supports real-time communication between multiple electronic control units and ensures the smooth transmission of critical vehicle control data. With the accelerating networking process of intelligent connected vehicles, network anomalies in In-Vehicle Ethernet may result in the loss or abnormal transmission of critical data, which in turn endangers driving safety. Enhancing the real-time performance and reliability of in-vehicle network intrusion detection can offer solid theoretical support for the further development of intelligent connected vehicles.

Carrying out network intrusion detection on in-vehicle bus networks enables real-time identification of various network intrusion behaviors and fault information. Traditional research into network intrusion detection mostly relies on port access control mechanisms and attack signature databases. While methods based on static rules and preset patterns work effectively in specific scenarios, they struggle to handle increasingly complex variant attacks. Network intrusion detection in vehicular networks mainly encompasses approaches like deep learning and statistical methods. Among these, deep learning methods stand out for their strong ability to automatically extract features, gaining widespread attention in the field of in-vehicle network intrusion detection. Hagar et al. built network intrusion detection models using mainstream deep learning architectures such as convolutional neural networks (CNN) and long short-term memory (LSTM), focusing on exploring how these architectures can be applied to network traffic intrusion detection. Their experimental results confirmed that deep learning methods can effectively capture complex traffic patterns and have obvious advantages in improving detection accuracy [4]. Nevertheless, deep learning methods still face challenges from camouflage and injection attacks. At the same time, they are hard to deploy in resource-constrained in-vehicle environments and fail to meet the real-time requirements of in-vehicle networks [5,6]. In contrast, statistical methods have low computational overhead and high interpretability, making them highly suitable for real-time detection in in-vehicle networks. Abdullahi et al. identified statistical techniques as one of the main methods for DDoS detection in software-defined network environments and conducted an in-depth analysis of their detection capabilities and limitations [7]. However, traditional statistical methods rely on fixed thresholds, which restricts their adaptability to dynamic vehicular network environments. When attackers adjust traffic characteristics adaptively to simulate normal behavior, the detection performance drops sharply. Although existing deep learning and statistical methods each have their own advantages, they can barely meet the requirements of both detection accuracy and real-time performance at the same time [8]. To address these problems, this study focuses on the security requirements of the AVTP protocol and proposes a lightweight intrusion detection method for In-Vehicle Ethernet based on feature masking. Different from traditional filter-based feature selection that uses static hard thresholds, our method constructs probabilistic feature masks by jointly evaluating feature discriminability and redundancy, and forms a complete detection pipeline combined with adaptive parameter optimization. This method realizes adaptive selection of key features and suppression of redundant information, and effectively solves the problem that traditional methods struggle to balance detection accuracy and real-time performance in vehicular scenarios.

The structure of the remainder of this paper is organized as follows: Section 2 focuses on analyzing the data frame structure of the AVTP protocol in In-Vehicle Ethernet, as well as the various types of network attacks targeting this protocol; Section 3 elaborates on the design concept and specific implementation process of the proposed feature mask algorithm; Section 4 carries out experimental verification and performance evaluation by

using the In-Vehicle Ethernet intrusion detection dataset; Section 5 summarizes the entire research work of this paper and draws corresponding conclusions.

2. In-Vehicle Ethernet AVTP Protocol and Network Security Analysis

With the rapid advancement of intelligent connected vehicle technology, numerous devices such as advanced driver assistance systems, autonomous driving systems, in-vehicle entertainment systems and intelligent terminals have been equipped in vehicles. These devices not only greatly improve the driving experience and intelligence of vehicles, but also break the relatively closed operating environment of traditional in-vehicle bus networks. Consequently, the network security risks faced by in-vehicle networks have become increasingly prominent, with security vulnerabilities and hacker attacks emerging one after another [9]. The Audio Video Transport Protocol (AVTP) in In-Vehicle Ethernet is widely applied in typical scenarios such as autonomous driving and in-vehicle entertainment, due to its compatibility with various audio and video formats. Meanwhile, intelligent connected vehicles continuously improve their environmental perception capabilities by integrating multimodal sensor data, supporting core functions like environmental prediction, collision warning, and path planning [10]. However, the openness of the AVTP protocol and the complexity of its structure make it prone to network security threats during data transmission, such as data loss, transmission delays, and network intrusions. Therefore, it is necessary to enhance the overall security and stability of the in-vehicle bus network by ensuring the communication reliability of key data frames, reducing packet loss rates and transmission delays, and preventing the risk of data tampering.

2.1. In-Vehicle Ethernet AVTP Protocol

As a core protocol for real-time multimedia transmission in automotive Ethernet, the Audio Video Transport Protocol (AVTP) is primarily tasked with carrying high-bandwidth, low-latency audio and video data streams. To meet the high-quality audio and video communication requirements of autonomous driving and in-vehicle infotainment systems, AVTP is widely applied in high-speed full-duplex automotive Ethernet. It guarantees steady and uninterrupted data transmission at the physical and data link layers [11].

The in-vehicle AVTP protocol enables precise identification and synchronous control of multiple audio and video streams through key fields like Timestamp and Stream ID, supporting the orderly transmission and fine-grained scheduling of data flows. It offers structured support for the Quality of Service (QoS) of automotive Ethernet, thereby ensuring the prioritized delivery of critical audio and video data. Meanwhile, built on the time synchronization and deterministic scheduling mechanism of TSN, the AVTP protocol further enhances transmission timeliness and reliability [12]. In terms of protocol stack architecture, the AVTP protocol encapsulates data frames in line with the IEEE 802.3 Ethernet standard and collaborates with upper-layer protocols such as SOME/IP to form a layered cooperative communication system [13]. In summary, the AVTP protocol plays a pivotal role in guaranteeing the real-time, synchronized, and reliable transmission of in-vehicle audio and video communications, serving as an important foundation for the In-Vehicle Ethernet multimedia system.

Packets of the AVTP protocol are composed of a header, Stream ID, Presentation Time, Format and Packet Info, and AV Data. Specifically, the header defines the data type and sequence number, which helps detect packet loss; the Stream ID is used to uniquely identify each data stream; the Presentation Time specifies the exact moment when the receiver should deliver data to the application layer, thus ensuring audio and video synchronization. The AVTP protocol is compatible with multiple audio and video transmission formats, including 61883-6, 3550/AAF, and H.264. These formats are applicable to different in-vehicle

scenarios respectively: 61883-6 for in-vehicle entertainment audio transmission, 3550/AAF for high-resolution audio processing, and H.264 for audio-video synchronized transmission. The data frame format of the AVTP protocol in In-Vehicle Ethernet is illustrated in Figure 1.

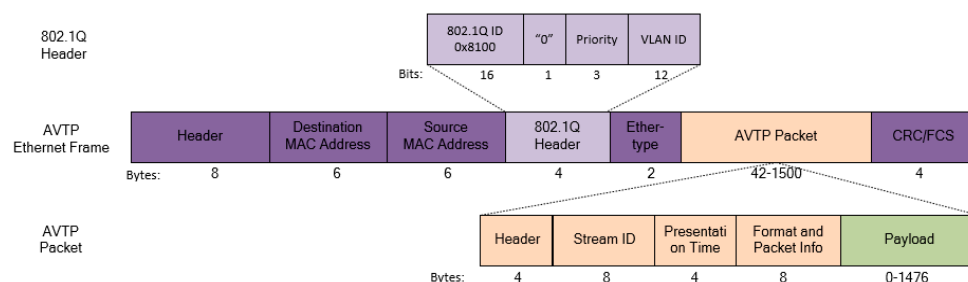


Figure 1. Data frame format of the AVTP protocol.

AVTP data transmission relies on the collaborative work of two core terminals, namely the Talker (sender) and the Listener (receiver). To optimize communication stability and data isolation, both terminals are generally configured within an identical Virtual Local Area Network (VLAN). The Talker terminal undertakes the collection and generation of original audio and video data, and encapsulates the processed data into standard AVTP frames. In addition to basic audio and video payloads, these frames also carry playback timing parameters and auxiliary control information, which are transmitted to the Listener through the In-Vehicle Ethernet link. During data transmission, network switching devices conduct priority queuing and traffic shaping for AVTP frames in compliance with the IEEE 802.1Qav protocol [14]. This mechanism grants audio and video streams higher link forwarding priority, effectively lowering transmission delay and network jitter. After receiving AVTP frames, the Listener terminal parses frame header information and extracts valid data payloads. Meanwhile, the whole network achieves high-precision time synchronization based on the IEEE 802.1AS protocol, enabling audio and video resources to be played strictly in accordance with preset timing sequences [15]. Such a complete workflow supports stable, low-latency and high-efficiency real-time multimedia data interaction for In-Vehicle Ethernet AVTP services.

2.2. Network Security Analysis of In-Vehicle Ethernet

In-Vehicle Ethernet features high bandwidth, low latency and service-oriented architecture. It serves as the core backbone network of intelligent connected vehicles to support advanced driver-assistance and autonomous driving functions, and ensures high-speed and reliable data transmission. Nevertheless, the increasingly open and complex network structure exposes automotive Ethernet to diverse cybersecurity threats such as spoofing attacks, replay attacks, denial-of-service attacks and malicious code injection. These attack methods not only tamper with key control information and disrupt the timing of in-vehicle communications, but also undermine the real-time performance and reliability of audio and video transmission, posing severe risks to driving safety and user privacy. To tackle these security challenges, extensive research efforts have been devoted to In-Vehicle Ethernet security in relevant fields.

Existing research on In-Vehicle Ethernet security has built a relatively comprehensive protection system. Althunayyan et al. categorize intrusion detection methods for vehicular networks into traditional detection and intelligent intrusion detection, noting that multidimensional performance evaluation is crucial for adapting to vehicular application scenarios [16]. In traditional cybersecurity research, three mainstream technical approaches are commonly adopted: firewall-based network access control, cryptography-driven data

protection, and intrusion detection-enabled real-time monitoring. These three approaches work together to form a comprehensive defense-in-depth system [17].

Firewall technology acts as a typical static defense measure at the network boundary, restricting unauthorized access through preconfigured network security rules and stateful inspection mechanisms. Luo et al. proposed a hierarchical in-vehicle network protection architecture, designing independent firewall modules for both CAN FD and In-Vehicle Ethernet. Specifically, the In-Vehicle Ethernet firewall adopts a stateful packet filtering mechanism to monitor the entire session process, thereby building a multi-level security barrier [18]. Cryptographic technology ensures the confidentiality and integrity of communication data by means of key management and identity authentication. Given the high real-time requirements and resource constraints of In-Vehicle Ethernet, Zhu et al. proposed a post-quantum enhanced session key agreement mechanism based on the NTRUEncrypt algorithm. This mechanism outperforms traditional ECDH and RSA algorithms in terms of execution efficiency [19]. Chen et al. developed a key management system for In-Vehicle Ethernet, integrating elliptic curve encryption (ECC), the Schnorr signature scheme, the AES-128 encryption algorithm, and the HMAC authentication method. This system not only realizes secure key distribution but also achieves encryption and authentication of multimedia data, ensuring both in-vehicle network security and real-time transmission requirements [20]. As a complement to static defense mechanisms, intrusion detection technology identifies anomalies and responds proactively by continuously monitoring network traffic behavioral characteristics, making it a core research focus in automotive Ethernet security [21]. Current related research mainly focuses on two aspects: protocol vulnerability analysis and intelligent detection algorithm design. At the protocol security level, the service discovery and subscription mechanisms of the In-Vehicle Ethernet SOME/IP protocol are susceptible to state hijacking and spoofing attacks, which may lead to unauthorized access to in-vehicle network resources and disrupt the normal operation of in-vehicle systems [22]. In In-Vehicle Ethernet, the Audio Video Transport Protocol (AVTP) undertakes real-time audio and video transmission tasks. By adopting bandwidth reservation and time-sensitive transmission mechanisms, it effectively improves the smoothness of multimedia data transmission and reduces latency. However, this protocol is also exposed to multiple security threats, including fake data stream injection and replay attacks. Jeong et al. proposed an intrusion detection method based on convolutional neural networks (CNNs) for the AVTP protocol, verifying its effectiveness in detecting replay attacks through a physical test platform [23]. Han et al. combined wavelet transformation with deep convolutional neural networks and proposed the TOW-IDS method. They built an in-vehicle network dataset covering normal traffic and five attack types, including frame injection, Precision Time Protocol (PTP) synchronization attacks, switch Media Access Control (MAC) flooding, CANDoS and CAN replay. They established an in-vehicle network dataset containing normal traffic and five attack types, namely frame injection, PTP synchronization attacks, switch MAC flooding, CANDoS and CAN replay. This dataset effectively enhances the performance of network intrusion detection [24]. da Luz et al. integrated the random forest algorithm with a pruned convolutional neural network (PCNN) to build a multi-stage deep learning architecture, which was validated under AEID and TOW-IDS dataset environments, further improving the effectiveness of network intrusion detection [25]. Bamber et al. developed a CNN-LSTM hybrid model based on spatiotemporal feature fusion technology, achieving 95% intrusion detection accuracy on the NSL-KDD dataset, which significantly improves the performance of network intrusion detection [26]. Zou et al. combined hierarchical clustering with a twin support vector machine to propose the HC-DTWSVM method, which achieved 81.21% detection accuracy on the UNSW-NB15 dataset. This method effectively mitigates the problem of

error accumulation in multi-classification tasks. Farrukh et al. designed the AIS-NIDS intelligent self-sustaining intrusion detection system, which realizes the recognition of unknown attacks and incremental learning through the collaboration of closed-set and open-set classifiers, ensuring the stability of the model during updates [27].

Apart from the above vehicle network intrusion detection schemes based on deep learning, tree models and traditional statistical methods, scholars have also proposed intelligent optimization and graph neural network detection frameworks for cyber-physical system security. Dakic et al. adopted modified Particle Swarm Optimization (PSO) to optimize machine learning classifiers for autonomous vehicle CAN traffic detection, yet iterative metaheuristic search introduces excessive latency that fails to meet the low-delay demands of in-vehicle Ethernet [28]. For power grid cyber-physical systems, Wang et al. designed a PSO-optimized attention temporal graph convolutional network to identify hidden dummy data injection attacks; however, stacked graph convolution layers bring massive parameter overhead and cannot be deployed on lightweight vehicle ECUs [29]. Both types of intelligent optimization-based detection methods achieve good detection accuracy but suffer high computational costs, which further highlights the necessity of our lightweight feature masking scheme for vehicle AVTP networks.

To secure the transmission of In-Vehicle Ethernet data, prevent data theft and tampering, and maintain the confidentiality and integrity of data frames, an in-depth analysis of network attack patterns and corresponding security defense systems is indispensable. On this basis, efficient network anomaly detection algorithms and targeted security defense strategies ought to be developed to comprehensively strengthen the overall security of In-Vehicle Ethernet.

3. Network Intrusion Detection Method Based on Feature Masking Algorithm

To address the network intrusion detection issue of In-Vehicle Ethernet, this paper puts forward an intrusion detection method for In-Vehicle Ethernet based on a feature mask algorithm. The overall architecture of this proposed detection method is illustrated in Figure 2. This intrusion detection architecture consists of three core modules, namely data preprocessing, network intrusion detection and adaptive parameter optimization. It effectively improves the detection capability for abnormal audio and video data, and offers an efficient and reliable solution to maintain secure and stable communication of In-Vehicle Ethernet.

3.1. Data Preprocessing

To tackle the issues of long byte length and high proportion of redundant information in raw AVTP protocol data frames, this study conducts data dimensionality reduction through feature extraction and standardization preprocessing. This approach preserves the core information required for anomaly discrimination while supplying high-quality input for the subsequent detection module.

In the automotive Ethernet AVTP protocol, each data frame is 438 bytes long, resulting in a relatively large data volume. To meet the real-time requirements of network intrusion detection, this study extracts 16-dimensional key information from AVTP data frames to construct a dimensionality-reduced data vector. This 16-dimensional vector comprises 3 dimensions of protocol type, 8 dimensions of data content, 4 dimensions of time statistics, and 1 dimension of data entropy. In this way, the key information that differentiates normal and abnormal data is retained, while the scale of data processing is significantly reduced. The specific process of feature extraction and standardization is detailed in Algorithm 1.

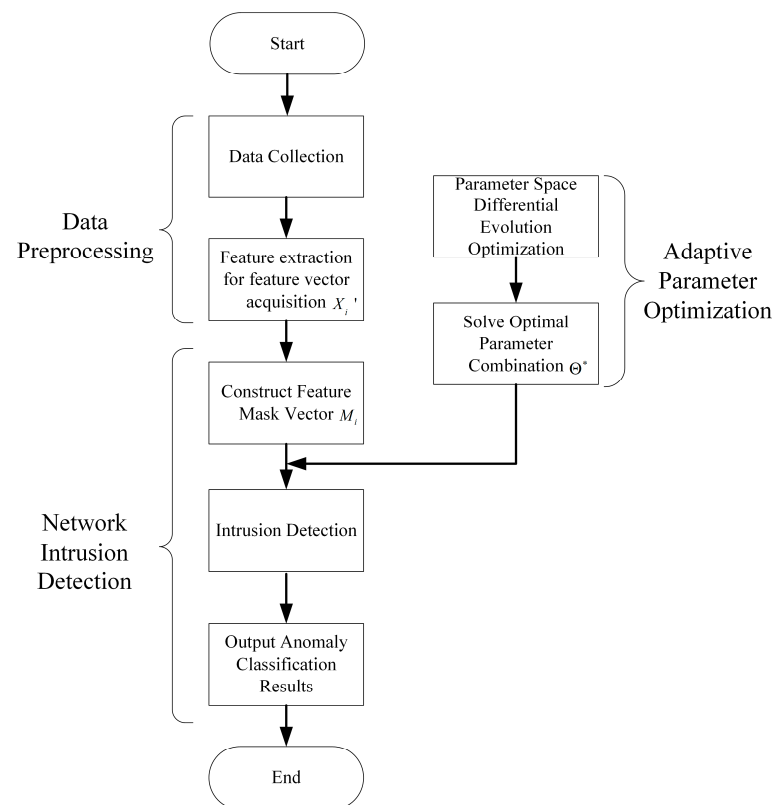


Figure 2. Network intrusion detection architecture.

After extracting features from each AVTP data frame, the data is represented as a 16-dimensional reduced-dimension vector X_i , as shown in Equation (1). Here, x_{ij} denotes the j -th feature of the i -th sample, where $j = 1, 2, \dots, 16$.

$$X_i = [x_{i1}, x_{i2}, \dots, x_{ij}] \quad (1)$$

In order to eliminate dimensional differences among different feature data, Z-score normalization is adopted for feature preprocessing. The normalized feature data x_{ij}' is defined in Equation (2). Here, x_{ij} denotes the original value of the j -th feature in the i -th sample, μ_j represents the mean of the j -th feature, and σ_j stands for the standard deviation of the j -th feature.

$$x_{ij}' = \frac{x_{ij} - \mu_j}{\sigma_j} \quad (2)$$

After Z-score normalization, a new dimension-reduced data vector X_i' is obtained, as defined in Equation (3).

$$X_i' = [x_{i1}', x_{i2}', \dots, x_{ij}'] \quad (3)$$

Once feature extraction and standardized preprocessing of AVTP protocol data frames are completed, this method specifically tackles the issues of high redundancy and large dimensional span in raw data. It captures the core feature information needed for anomaly detection while effectively achieving data dimensionality reduction, thus eliminating the interference of feature scale differences on the training process of network intrusion detection models. This work lays a solid data foundation for the stable training, high-precision inference, and lightweight deployment of the subsequent core intrusion detection algorithm based on feature masking.

Algorithm 1: AVTP Data Frame Feature Preprocessing Algorithm

Input: AVTP data frame set $F = \{f_1, f_2, \dots, f_N\}$, each frame has a length of 438 bytes

Output: The standardized set of dimensionality-reduced data vectors is denoted as

$X' = \{X'_1, X'_2, \dots, X'_N\}$, here $X'_i \in \mathbb{R}^{16}$

```

1: procedure EXTRACT_AND_STANDARDIZE(F)
2:   Compute  $\mu_j$  and  $\sigma_j$  for each feature dimension  $j = 1, \dots, 16$ 
3:   for  $i = 1$  to  $N$  do
4:      $X_i = \text{extract features}(f_i)$  // Extract 16-dimensional dimensionality-reduced data vectors.
5:     for  $j = 1$  to  $d$  do
6:        $x_{ij}' = (x_{ij} - \mu_j) / \sigma_j$  // Z-score normalization
7:     end for
8:      $X_i = (x'_{i1}, x'_{i2}, \dots, x'_{i16})$ 
9:   end for
10:  return  $X'$ 

```

3.2. Network Intrusion Detection

Data streams transmitted via the AVTP protocol in In-Vehicle Ethernet face notable network intrusion risks, including injection and replay attacks, which directly compromise driving safety. A key constraint, however, is that vehicles cannot accommodate high-performance computer server systems—this prevents the application of complex computer-based algorithms, such as those used for firewall protection, data encryption, identity authentication, and network maintenance. To address this and achieve lightweight, high-performance intrusion detection, this study proposes a feature masking-based intrusion detection algorithm, designed to identify AVTP abnormal traffic with high precision and low latency.

The detection process comprises three core stages: feature mask generation, anomaly probability assessment, and adaptive threshold optimization. The workflow of this proposed network intrusion detection algorithm is illustrated in Figure 3. By maintaining high detection accuracy, the algorithm also reduces false positive rates and inference latency, making it highly compatible with the application limitations of In-Vehicle Ethernet.

To gauge how effectively each feature distinguishes between normal and abnormal traffic, we compute the discrimination value $S(j)$ for the j -th feature, as specified in Equation (4). Specifically, $\mu_\alpha(j)$ refers to the average value of the j -th feature across all abnormal samples, while $\mu_n(j)$ denotes the average value of the same feature among normal samples. $\sigma_n(j)$ stands for the standard deviation of the j -th feature in normal samples. Here, ε represents an infinitesimal positive constant, which is introduced to avoid the denominator being zero in the calculation.

$$S(j) = \frac{\mu_\alpha(j) - \mu_n(j)}{\sigma_n(j) + \varepsilon} \quad (4)$$

In summary, $S(j)$ (feature discriminability) quantifies the feature's ability to distinguish normal AVTP traffic from attacks; a larger value indicates stronger classification utility.

To quantify the redundancy of each feature, we define $R(j)$ as follows: we take the maximum absolute value of the Pearson correlation coefficients between the j -th feature and all other features as the redundancy measure. A higher maximum correlation means the feature has higher redundancy and can be easily replaced by other features.

$$R(j) = \max_{k \neq j} |\gamma_{j,k}| \quad (5)$$

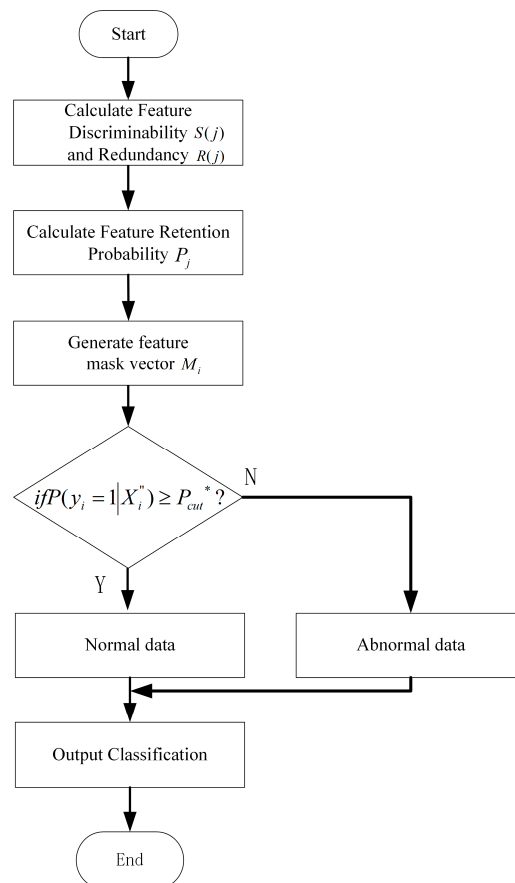


Figure 3. Flowchart of network intrusion detection.

Here, $\gamma_{j,k}$ stands for the Pearson correlation coefficient between feature j and feature k (where k represents all features except j), which is specifically defined in Equation (6). In this definition, $\text{Cov}(x_j, x_k)$ denotes the covariance between feature j and feature k , while σ_j and σ_k refer to their respective standard deviations.

$$\gamma_{j,k} = \frac{\text{Cov}(x_j, x_k)}{\sigma_j \sigma_k} \quad (6)$$

We integrate feature discriminability and redundancy to calculate the feature weight α_j , whose computational formula is shown in Equation (7).

$$\alpha_j = \frac{R(j)}{S(j) + R(j)} \quad (7)$$

By substituting Equation (7) into the retention probability expression and performing algebraic simplification, the retention probability P_j for each feature is obtained, as illustrated in Equation (8). The retention probability comprehensively reflects the discriminability and redundancy of the features.

$$P_j = \frac{S(j) \cdot |S(j) + R(j)|}{S^2(j) + S(j)R(j) + R^2(j)} \quad (8)$$

This probabilistic formula is designed to balance feature discriminability and redundancy: features with higher discriminability and lower redundancy will be assigned higher retention probability. The quadratic term in the denominator avoids the deviation caused by relying on a single indicator, and the whole calculation process only involves simple

algebraic operations with extremely low computational cost, which is fully compatible with the limited computing resources of vehicle-embedded ECUs.

Different from traditional filter-based feature selection that screens features via fixed static thresholds in a deterministic manner, our scheme generates feature retention probability by comprehensively measuring both feature discriminability and redundancy, and constructs binary masks through random sampling, which strengthens model robustness against volatile in-vehicle AVTP traffic. Unlike conventional filtering methods that only serve as isolated preprocessing steps separated from classifier training, the feature masking module in this work is tightly coupled with subsequent anomaly discrimination and adaptive parameter tuning modules to form an end-to-end joint optimization pipeline for vehicular intrusion detection. Moreover, this algorithm is custom-built for AVTP replay attack scenarios and limited-resource vehicle ECUs; by optimizing the decision threshold oriented to F2-score to guarantee high recall for driving safety, it strikes a balanced trade-off between detection accuracy and low inference overhead, delivering practical deployment advantages unavailable from universal general filtering algorithms.

Based on the retention probability of each feature, a binary feature mask vector M_i is constructed via a random selection strategy, as defined in Equation (9). Here, *rand* denotes a uniformly distributed random number within the interval $[0, 1]$, where $j = 1, 2, \dots, 16$.

$$M_i = [M_{i1}, M_{i2}, \dots, M_{i16}], \quad M_{ij} = \begin{cases} 1, & \text{rand} < P_j \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

Thus, the filtered and dimension-reduced data vector X_i'' is obtained, as defined in Equation (10). Here, $\odot$ denotes the element-wise multiplication operation.

$$X_i'' = X_i' \odot M \quad (10)$$

Taking the 16-dimensional filtered vector X_i'' as the input, we train a network anomaly detection model to establish a probabilistic characterization of network traffic behavior. In this model, the i -th output corresponds to the probability of abnormal network activity, as defined in Equation (11). Here, $f(\cdot)$ denotes the mapping function of the network anomaly detection model, and $\hat{y}_i$ represents the corresponding anomaly probability.

$$\hat{y}_i = f(X_i'') \quad (11)$$

To convert continuous probability values into discrete classification outcomes, conditional probability is introduced to express the confidence level of the network anomaly detection model, as defined in Equation (12).

$$P(y_i = 1 | X_i'') \quad (12)$$

The network intrusion decision function is given in Equation (13). Here, y_i denotes the final classification result of the i -th sample, where an output of 1 indicates an anomaly and 0 indicates normal behavior. p_{cut} stands for the decision threshold, with a value range of $(0, 1]$. $I(\cdot)$ denotes the indicator function, which outputs 1 if the condition within the parentheses is satisfied and 0 otherwise.

$$y_i = I(\hat{y}_i > p_{\text{cut}}) \quad (13)$$

By adopting the recall $R(p_{\text{cut}})$ and $F_2(p_{\text{cut}})$ scores to lower the missed detection rate of abnormal samples, the decision threshold is optimized adaptively. Under a given decision threshold, the calculation formulas for recall $R(p_{\text{cut}})$ and $F_2(p_{\text{cut}})$ are presented

in Equations (14) and (15), respectively. Here, $TP(p_{cut})$ denotes true positive, representing the number of abnormal samples correctly identified; $FN(p_{cut})$ denotes false Negative (FN) represents attack samples misclassified as normal. β is the weight coefficient of recall, which strengthens the model's capability to recognize attack samples.

$$R(p_{cut}) = \frac{TP(p_{cut})}{TP(p_{cut}) + FN(p_{cut})} \quad (14)$$

$$F_2(p_{cut}) = \frac{(1 + \beta^2) \cdot P(p_{cut}) \cdot R(p_{cut})}{\beta^2 \cdot P(p_{cut}) + R(p_{cut})} \quad (15)$$

Let P denote the set of candidate values composed of all decision thresholds within the interval $(0, 1]$. By traversing the set P , the optimal decision threshold p_{cut}^* is searched with the objective of maximizing the F_2 score, as defined in Equation (16).

$$p_{cut}^* = \arg \max_{p_{cut} \in P} F_2(p_{cut}) \quad (16)$$

Finally, by substituting the optimal decision boundary into the anomaly detection function, the final network intrusion detection result y_i is obtained, as shown in Equation (17). Leveraging the mask-selected features and the adaptive decision boundary method, the speed and accuracy of network anomaly detection are significantly improved.

$$y_i = \begin{cases} 1, & P(y_i = 1 | X_i'') > p_{cut}^* \\ 0, & \text{otherwise} \end{cases} \quad (17)$$

3.3. Adaptive Parameter Optimization

In actual vehicle operation scenarios, the traffic of In-Vehicle Ethernet shows obvious dynamic characteristics. As a result, network intrusion detection models relying on fixed parameters are unable to keep up with real-time changes in traffic distribution. This deficiency often causes the model to be trapped in local optima during the training process, making it hard to meet the security and reliability standards required by in-vehicle networks. To address this problem, this study proposes an adaptive parameter optimization method integrated with feature masks, which further enhances the detection accuracy of in-vehicle network intrusion detection and its adaptability to complex actual scenarios. For the dimensionality-reduced data vectors obtained after mask processing, an iterative population search strategy is adopted to optimize model parameters, which enables effective differentiation between normal and abnormal feature information. The flowchart of the adaptive parameter optimization process is shown in Figure 4.

In the adaptive parameter optimization phase, each set of parameter combinations is treated as an individual in the population. Taking the model's loss function as the fitness metric, the optimal parameter combination $\Theta_i = (\eta, D, T_m)$ is selected through multiple iterations. Ultimately, anomaly detection is completed based on these optimal parameters. Here, η denotes the learning rate, D represents the max tree depth, and T_m stands for the number of training epochs.

To evaluate the performance of different parameter combinations, the loss function $L(\Theta_i)$ is adopted as the fitness evaluation metric, as defined in Equation (18). A smaller loss function value corresponds to higher fitness and a superior parameter combination. Here, L_{cls} denotes the classification loss, L_{reg} represents the regularization term, and λ stands for the regularization coefficient.

$$L(\Theta_i) = L_{cls} + \lambda L_{reg} \quad (18)$$

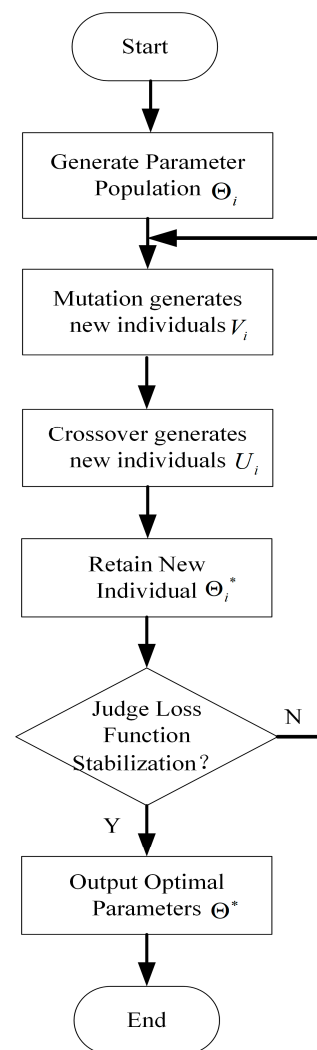


Figure 4. Flowchart of adaptive optimization.

In this paper, L2 regularization is adopted. We select L2 regularization for its low computation cost suited to embedded vehicle devices, and we do not use the Lagrangian method since it introduces extra constrained optimization overhead, conflicting with our lightweight target. L2 regularization can impose smooth constraints on network weights and suppress overfitting of the in-vehicle traffic model. The search range of the hyperparameter λ is set to $(0, 0.1]$. The upper bound of 0.1 is selected due to the limited feature dimensions and balanced sample distribution of the in-vehicle network. When $\lambda > 0.1$, the model parameters will be excessively penalized, leading to underfitting and a sharp drop in detection accuracy.

Construct the fitness function $\text{Fitness}(\Theta_i)$ based on the loss function, as defined in Equation (19). The optimization objective is to maximize the fitness value, enabling the adaptive parameter optimization algorithm to iteratively update parameters in the direction that enhances the performance of anomaly detection.

$$\text{Fitness}(\Theta_i) = -L(\Theta_i) \quad (19)$$

To maintain population diversity and prevent the adaptive parameter optimization algorithm from falling into local optima, a mutation operation is performed to generate new candidate individuals V_i , as shown in Equation (20). Here, F denotes the mutation factor, and $\Theta_1, \Theta_2, \Theta_3$ are three distinct individuals randomly selected from the population.

By leveraging the parameter differences among these selected individuals to apply random perturbations, entirely new mutated individuals are generated, which effectively expands the parameter search space.

$$V_i = \Theta_1 + F(\Theta_2 - \Theta_3) \quad (20)$$

To enhance the global exploration and local exploitation capabilities of the adaptive parameter optimization algorithm, a crossover operation is implemented. After acquiring the fused mutated individuals and parent individuals, new individuals U_i are generated, as illustrated in Equation (21). This operation not only retains the superior parameter characteristics identified during the search process but also maintains population diversity, thereby balancing the algorithm's global search capability and convergence speed. Here, CR denotes the crossover probability, and rand represents a random number uniformly distributed within the interval $[0, 1]$

$$U_i = \begin{cases} V_i, \text{rand} < \text{CR} \\ \Theta_i, \text{otherwise} \end{cases} \quad (21)$$

Through the processes of natural selection and population update, the optimal individual Θ_i^{t+1} in the population is selected, as defined in Equation (22). High-quality individuals are screened generation by generation based on their fitness values, which continuously guides the population to iteratively update toward the globally optimal parameter set. Here, Θ_i^t denotes the parent individual of the t -th generation, where t represents the current iteration number.

$$\Theta_i^{t+1} = \begin{cases} U_i, \text{Fitness}(U_i) > \text{Fitness}(\Theta_i^t) \\ \Theta_i^t, \text{otherwise} \end{cases} \quad (22)$$

After multiple rounds of iterative processes involving mutation, crossover, and selection operations, the globally optimal parameter combination Θ_i^* is ultimately obtained. Utilizing these optimal parameters, we train both the feature mask generation mechanism and the network anomaly detection model. This not only further enhances the recognition accuracy and generalization capability of the network anomaly detection model for abnormal AVTP protocol traffic but also ensures its real-time performance in resource-constrained vehicular environments. Ultimately, this achieves the dual optimization objectives of improving network anomaly detection accuracy and enhancing operational efficiency.

Adaptive parameter optimization helps acquire optimal parameters for the feature mask-based anomaly detection model. It remedies the defect of fixed-parameter strategies that cannot adapt to dynamic vehicular traffic environments. By virtue of an iterative population evolution mechanism, the network anomaly detection model can independently select the optimal parameter combination that matches the characteristics of AVTP traffic.

Furthermore, we embed an early stopping rule into the iterative optimization process. Specifically, we track the global optimal fitness value per round, and terminate the search in advance if the fitness shows no improvement for 10 consecutive rounds, indicating algorithm convergence.

The 10-round threshold is determined based on the latency-accuracy trade-off for in-vehicle embedded devices. A smaller threshold (3–5 rounds) causes premature convergence to local optima and reduces detection accuracy, while a larger threshold (≥ 20 rounds) brings redundant iterations and raises optimization latency significantly. Simulation results verify that 10 rounds achieve the optimal balance between detection performance and real-time efficiency.

In terms of computational complexity, the overall time complexity of this optimization strategy is $O(G \times K \times C)$, where G is the actual iteration number, K is the population size, and C is the constant cost of single loss evaluation. The early stopping rule limits

the upper bound of G , effectively reducing total computational overhead and end-to-end detection latency.

4. Experimental Simulation and Result Analysis

CANoe (Vector Informatik GmbH, Stuttgart, Germany) serves as a dedicated simulation platform for the design of in-vehicle bus network topology and the analysis of network performance. It is compatible with multiple in-vehicle communication protocols, such as LIN, CAN, FlexRay, and In-Vehicle Ethernet. In this paper, the CANoe-Ethernet tool is employed to build an experimental simulation environment for In-Vehicle Ethernet, and the performance of the proposed algorithm is verified through the integration of software and hardware.

The network topology architecture of the In-Vehicle Ethernet is depicted in Figure 5. In detail, data is transmitted from the Talker node, while the Ethernet IG node is used to simulate and inject abnormal data. The Gateway node facilitates data transmission between different buses, the Listener node undertakes the task of receiving data, the Player node is in charge of playing multimedia data, and the Detected node is responsible for detecting abnormal data. All the above-mentioned communication nodes are connected via the Ethernet Bus (Eth 1).

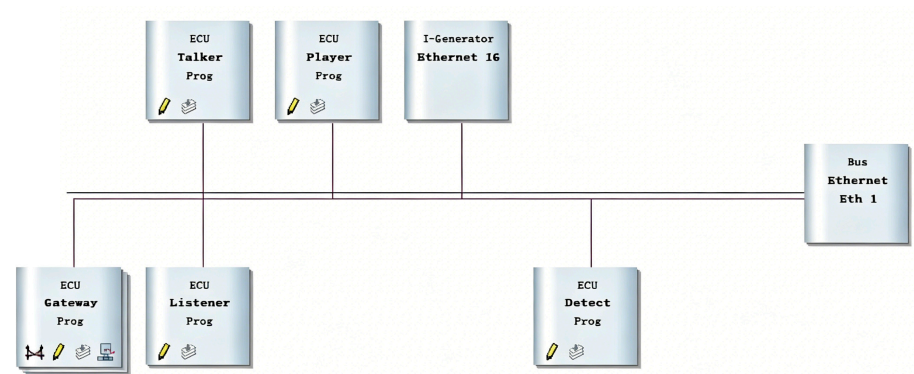


Figure 5. Network topology architecture of In-Vehicle Ethernet.

Figure 6 shows the schematic diagram of the hardware experimental device, which clearly displays the structure of the In-Vehicle Ethernet test platform. A PC host running the CANoe. The Ethernet system is connected to the VN5610 communication interface through a USB port. The transmitting node (Talker Node), listening node (Listener Node), and Ethernet Interactive Generation node (Ethernet IG Node) are all linked to the VN5610 communication interface, together forming a complete hardware architecture for In-Vehicle Ethernet communication. With CANoe. As the core simulation platform, the VN5610 device not only provides hardware authorization and real-time bus interaction support but also acts as a communication interface for In-Vehicle Ethernet. This study has built an In-Vehicle Ethernet simulation test environment, which consists of sending nodes, listening nodes, abnormal traffic generation nodes, and detection nodes. Figure 7 shows the actual hardware connections and software topology, allowing for the realistic reproduction of the entire process of message transmission, anomaly injection, and intrusion detection of In-Vehicle Ethernet in real vehicle operating scenarios.

In this study, we selected a subset from the In-Vehicle Ethernet AVTP protocol dataset for experimental verification, which is denoted as the Randomly Collected Dataset (RCD). Detailed specifications of the dataset are provided in Table 1. Specifically, the RCD dataset consists of normal AVTP traffic and replay attack traffic. It is split into a training set and a test set at a ratio of 7:3 to maintain uniform data distribution. The replay attack

samples were constructed in accordance with the native frame encapsulation mechanism of the AVTP protocol: 36 consecutive AVTPDUs, which form a complete audio-video transmission frame, were extracted from real in-vehicle audio and video streams. In real-world attack scenarios, adversaries typically implement AVTP frame replay attacks by repeatedly injecting this 36-AVTPDU sequence into the In-Vehicle Ethernet, which is a typical and realistic security threat to in-vehicle AVTP streams.

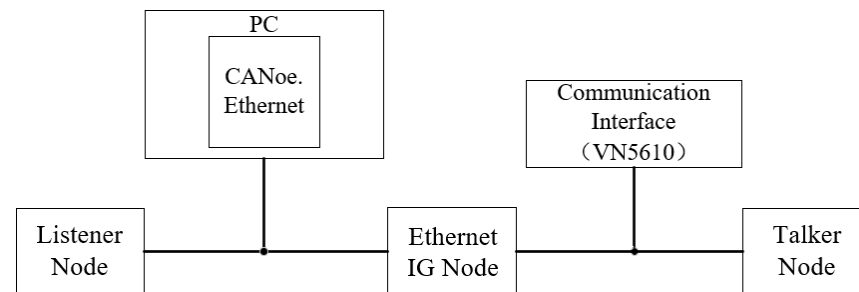


Figure 6. Network topology diagram of In-Vehicle Ethernet.

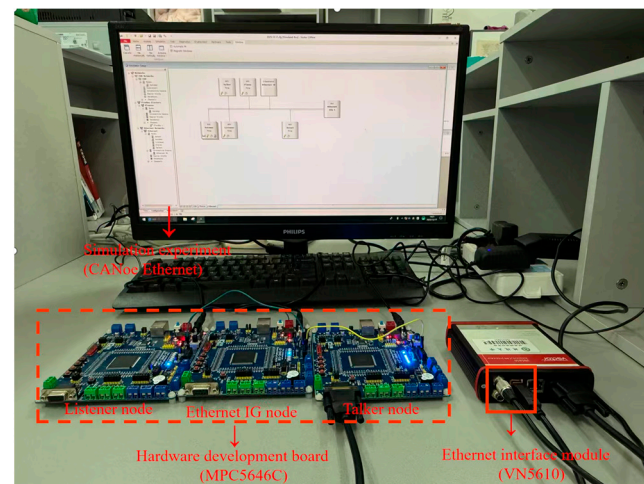


Figure 7. Actual experimental environment.

Table 1. In-Vehicle Ethernet AVTP protocol dataset.

Database	Normal (Packets)	Abnormal (Packets)	Size (KB)
RCD	771,043	199,656	433,729
Label	0	36	16

To assess the performance of the proposed network intrusion detection model, this study employs core statistical indicators derived from the confusion matrix for quantitative evaluation. The confusion matrix corresponding to the proposed algorithm is shown in Figure 8, which is obtained by statistically analyzing the correlation between the model's predicted results and actual labels. This matrix yields four basic metrics: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). Specifically, TP refers to the number of attack samples accurately identified by the model; TN represents the count of normal samples correctly classified; FP stands for normal samples mistakenly identified as attacks; FN denotes attack samples incorrectly categorized as normal. This study adopts TP, TN, FP and FN to compute four core intrusion detection indicators, including accuracy, precision, recall and F1_score. The F1_score refers to the harmonic mean of precision and recall.

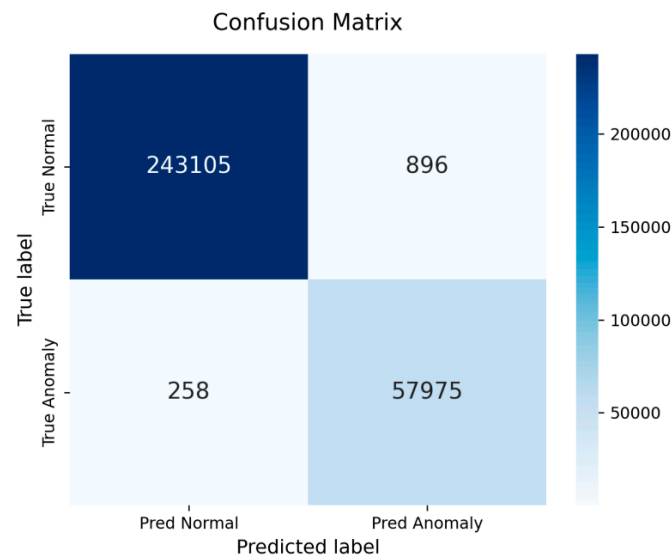


Figure 8. Confusion matrix of the Feature Masking algorithm.

Accuracy is defined as the proportion of correctly classified samples to the total number of samples, which reflects the overall classification performance of the model in handling both normal and abnormal AVTP data frames, and its specific calculation method is given in Equation (23). When tested on the test set of the RCD dataset, the feature mask algorithm proposed in this study achieved an accuracy of 99.6%.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \quad (23)$$

Precision is the proportion of actual attack samples among all samples predicted as attacks, with its specific definition provided in Equation (24). A higher precision value indicates a lower false alarm probability of the in-vehicle system, which is crucial for avoiding unnecessary interference with vehicle driving. In the experiment, the feature mask algorithm proposed in this study reached a precision of 98.5%.

$$Precision = \frac{TP}{TP + FP} \times 100\% \quad (24)$$

Recall reflects the model's capability to identify actual attack samples, specifically referring to the proportion of all real attack samples that are correctly detected, and its definition is detailed in Equation (25). As a core security indicator for in-vehicle intrusion detection, recall is closely related to driving safety: a lower recall means more attacks fail to be detected, which in turn raises potential risks to driving safety. In the experimental tests, the feature mask algorithm proposed in this study achieved a recall of 99.5%.

$$Recall = \frac{TP}{TP + FN} \times 100\% \quad (25)$$

The F1_score, also known as the F1 harmonic mean, is the harmonic mean of precision and recall, with its specific calculation formula shown in Equation (26). This indicator integrates the advantages of both precision and recall, taking into account both false positive and false negative cases simultaneously. It is particularly suitable for objectively evaluating the overall detection performance of the model in scenarios with an imbalanced class distribution. In the experiment, the feature mask algorithm proposed in this study achieved an F1_score of 99.0%.

$$F1_score = 2 \times \frac{precision \times recall}{precision + recall} \times 100\% \quad (26)$$

To guarantee fair comparison, all baselines and the proposed method share identical 16-dimensional input features, a 7:3 train-test split and Z-score normalization. For baseline configurations, Naive Bayesian and Decision Tree use default scikit-learn parameters without extra tuning; XGBoost adopts 100 estimators, max depth 6 and learning rate 0.1 as the standard setting; the CNN baseline consists of two convolution-pooling layers and two fully connected layers, with ReLU activation and Adam optimizer. All baselines use widely accepted standard configurations without targeted fine-tuning, while the optimal hyperparameters of our method are derived from the adaptive differential evolution module in Section 3.3.

This study conducts a comparative analysis between the proposed feature masking method and four mainstream detection baselines, including Naive Bayesian, Decision Tree, XGBoost and CNN. The overall detection performance of all algorithms is visualized in Figure 9, while Table 2 provides a detailed comparison of their detection performance metrics.

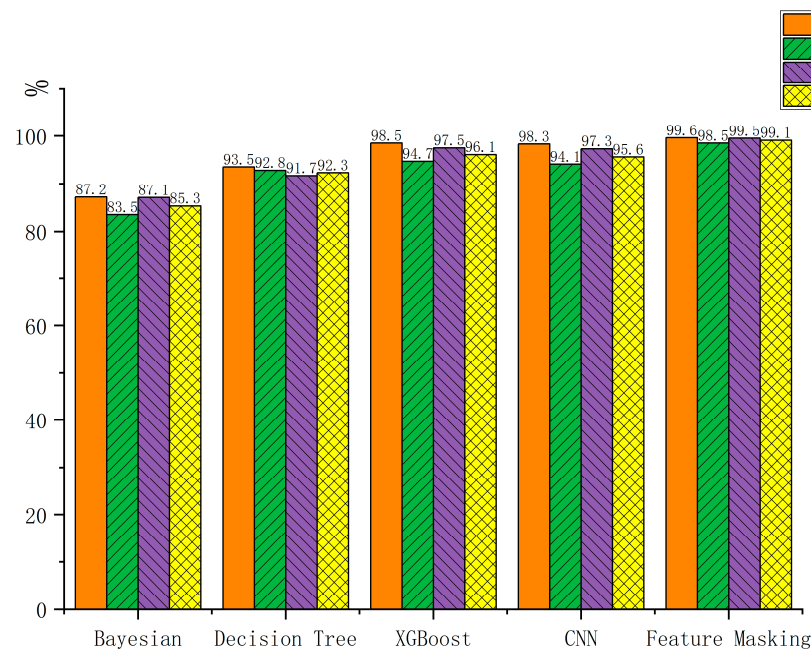


Figure 9. Comparison chart of network intrusion detection results.

Table 2. Comparison of network intrusion detection performance.

Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1_Score (%)	Time (μs/Sample)
Bayesian	87.2	83.5	87.1	85.3	454
Decision Tree	93.5	92.8	91.7	92.3	556
XGBoost	98.5	94.7	97.5	96.1	618
CNN	98.3	94.0	97.3	95.6	2625
Feature Masking	99.6	98.5	99.5	99.0	483

Table 3 summarizes the optimal hyperparameters derived from adaptive parameter optimization. Specifically, the max tree depth parameter D denotes the maximum allowable depth of each base tree within the model, which serves to regulate the complexity of individual trees and mitigate the risk of model overfitting. The search intervals for each hyperparameter are configured as follows: learning rate $\eta \in [0.01, 0.3]$, max tree depth $D \in [3, 20]$, and number of training rounds $T_m \in [50, 200]$. A simulated search space is

constructed based on these three value ranges to perform combinatorial parameter search, and the resulting optimal parameter configuration is presented in Table 3.

Table 3. Optimal hyperparameter values.

Parameter Name	Optimal Parameter Value
Learning rate η	0.02
Max tree depth D	6
rounds T_m	120

Experimental results show that the four evaluation metrics (Accuracy, Precision, Recall, F1_score) of the lightweight Bayesian algorithm range from 83.5% to 87.2%, while those of the Decision Tree algorithm lie between 91.7% and 93.5%. For more sophisticated baseline models, XGBoost and CNN deliver overall detection metrics between 94.0% and 98.5%, outperforming traditional lightweight methods. In sharp contrast, all four evaluation metrics of the proposed feature mask algorithm exceed 98.5%, demonstrating outstanding comprehensive intrusion detection performance.

To quantitatively measure the performance improvement of the proposed algorithm, this study uses the relative improvement rate of recall for calculation, with the specific formula given in Formula (27). Here, $\text{Recall}_{\text{proposed}}$ refers to the intrusion detection recall rate of the feature mask algorithm proposed in this study, and $\text{Recall}_{\text{baseline}}$ represents the recall rate of the baseline algorithm used for comparison.

$$\Delta\text{Recall} = \frac{\text{Recall}_{\text{proposed}} - \text{Recall}_{\text{baseline}}}{\text{Recall}_{\text{baseline}}} \times 100\% \quad (27)$$

To quantify the optimization extent of the algorithm in real-time detection performance, the relative reduction rate of computation time is adopted for calculation, as shown in Formula (28). Here, $\text{Time}_{\text{proposed}}$ denotes the detection computation time of the feature masking algorithm proposed in this paper, and $\text{Time}_{\text{baseline}}$ represents the detection computation time of the baseline comparison algorithm.

$$\Delta\text{Time} = \frac{\text{Time}_{\text{baseline}} - \text{Time}_{\text{proposed}}}{\text{Time}_{\text{baseline}}} \times 100\% \quad (28)$$

Experimental results confirm that, compared with the traditional Bayesian and Decision Tree algorithms, the proposed feature mask algorithm improves intrusion detection accuracy by 12.4 percentage points and 6.1 percentage points, respectively. Although the Bayesian algorithm has slightly lower inference latency, its detection capability is far from sufficient to meet the security requirements of in-vehicle Ethernet. Compared with the state-of-the-art XGBoost and CNN models, the proposed algorithm not only achieves higher detection accuracy and recall but also reduces inference latency by 21.8% and 81.6%, respectively. The proposed algorithm achieves a favorable balance between detection performance and operational efficiency, thus providing an efficient intrusion detection scheme for safeguarding in-vehicle Ethernet security.

5. Conclusions

This study focuses on solving the network intrusion detection problem in In-Vehicle Ethernet and innovatively puts forward an intrusion detection method based on the feature mask algorithm. In the complex operating environment of in-vehicle networks, the feature mask algorithm demonstrates remarkable advantages in detection accuracy, response speed, and system stability, which allows it to accurately identify abnormal network behaviors

with a detection accuracy reaching 99.5%. Looking ahead, this study will further focus on the network security challenges of In-Vehicle Ethernet, and construct a comprehensive integrated protection system that combines intrusion detection and intelligent response by integrating fault-tolerant control mechanisms, so as to improve the overall security and reliability of in-vehicle networks.

Author Contributions: Conceptualization, Y.J.; methodology, Y.J.; software, Y.J.; validation, Y.J.; formal analysis, Y.X. (Yihu Xu); investigation, X.L.; resources, Y.X. (Yinan Xu); data curation, Y.W.; writing—original draft preparation, Y.J.; writing—review and editing, Y.J.; visualization, M.L.; supervision, Y.W. and Y.X. (Yihu Xu); project administration, Y.X. (Yinan Xu); funding acquisition, Y.X. (Yinan Xu). All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Jilin Province Natural Science Foundation, Grant No. YDZJ202501ZYTS641, and was also supported by the National Natural Science Foundation of China, Grant Nos. 62161049 and 61763047.

Data Availability Statement: The data that support the findings of this study are not publicly available due to privacy concerns. Requests for access to the data should be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

- De Vincenzi, M.; Costantino, G.; Matteucci, I. A systematic review on security attacks and countermeasures in automotive ethernet. *ACM Comput. Surv.* **2024**, *56*, 1–38. [\[CrossRef\]](#)
- Lampe, B.; Meng, W. A survey of deep learning-based intrusion detection in automotive applications. *Expert Syst. Appl.* **2023**, *221*, 119771. [\[CrossRef\]](#)
- Le, H.; Alsmadi, I. Intrusion detection for internet of vehicles CAN bus communications using machine learning: An empirical study on the CICIoV2024 Dataset. *Future Internet* **2026**, *18*, 42. [\[CrossRef\]](#)
- Zou, L.; Luo, X.; Zhang, Y. HC-DTTSVM: A network intrusion detection method based on decision tree twin support vector machine and hierarchical clustering. *IEEE Access* **2023**, *11*, 21404–21416. [\[CrossRef\]](#)
- Hellemans, W.; Le Jeune, L.; Rabbani, M.M. Toward a real-time intrusion detection system for modern in-vehicle networks. *IEEE Trans. Intell. Transp. Syst.* **2025**, *26*, 18665–18679. [\[CrossRef\]](#)
- Liu, X.; Yu, H. A federated intrusion detection system for edge environments using multi-index hashing and attention-based KNN. *Symmetry* **2025**, *17*, 1580. [\[CrossRef\]](#)
- Rathore, R.S.; Hewage, C.; Kaiwartya, O. In-vehicle communication cyber security: Challenges and solutions. *Sensors* **2022**, *22*, 6679. [\[CrossRef\]](#) [\[PubMed\]](#)
- Huang, K.; Xian, R.; Xian, M. A comprehensive intrusion detection method for the internet of vehicles based on federated learning architecture. *Comput. Secur.* **2024**, *147*, 104067. [\[CrossRef\]](#)
- Cataldo, C. Ethernet Network in the Automotive Field: Standards, Possible Approaches to Protocol Validation and Simulations. Master's Thesis, Politecnico di Torino, Torino, Italy, 2021; pp. 1–92.
- Douss, A.B.; Abassi, R.; Sauveron, D. State-of-the-art survey of in-vehicle protocols and automotive Ethernet security and vulnerabilities. *Math. Biosci. Eng.* **2023**, *20*, 17057–17095. [\[CrossRef\]](#) [\[PubMed\]](#)
- Muslam, M.M.A. Enhancing security in vehicle-to-vehicle communication: A comprehensive review of protocols and techniques. *Vehicles* **2024**, *6*, 450–467. [\[CrossRef\]](#)
- Deng, L.; Xie, G.; Liu, H. A survey of real-time ethernet modeling and design methodologies: From AVB to TSN. *ACM Comput. Surv.* **2022**, *55*, 1–36. [\[CrossRef\]](#)
- IEEE Std 802.3-2022; IEEE Standard for Ethernet. Institute of Electrical and Electronics Engineers: New York, NY, USA, 2022.
- IEEE Std 802.1Qav-2009; IEEE Standard for Local and Metropolitan Area Networks—Virtual Bridged Local Area Networks Amendment 12: Forwarding and Queuing Enhancements for Time-Sensitive Streams. Institute of Electrical and Electronics Engineers: New York, NY, USA, 2009.
- IEEE Std 802.1AS-2020; IEEE Standard for Local and Metropolitan Area Networks—Timing and Synchronization for Time-Sensitive Applications. Institute of Electrical and Electronics Engineers: New York, NY, USA, 2020.
- Althunayyan, M.; Javed, A.; Rana, O. A survey of learning-based intrusion detection systems for in-vehicle networks. *Comput. Netw.* **2026**, *277*, 112031. [\[CrossRef\]](#)

17. Rumez, M.; Grimm, D.; Kriesten, R. An overview of automotive service-oriented architectures and implications for security countermeasures. *Electronics* **2020**, *8*, 221852–221870. [\[CrossRef\]](#)
18. Luo, F.; Hou, S. Security Mechanisms Design of Automotive Gateway Firewall. In *Proceedings of the WCX SAE World Congress Experience, Detroit, MI, USA, 9 April 2019*; SAE International: Warrendale, PA, USA, 2019; pp. 1–8.
19. Zhu, Y.; Liu, Y.; Wu, M. Research on secure communication on In-Vehicle Ethernet based on post-quantum algorithm NTRUEncrypt. *Electronics* **2022**, *11*, 856. [\[CrossRef\]](#) [\[PubMed\]](#)
20. Chen, J.; Zuo, Q.; Jin, W. Study of network security based on key management system for In-Vehicle Ethernet. *Electronics* **2024**, *13*, 2524. [\[CrossRef\]](#)
21. Kim, T.G.; Park, H.; You, I.; Kwak, B.I. XGBoost-Based anomaly detection framework for SOME/IP in in-vehicle networks. *Systems* **2026**, *14*, 196. [\[CrossRef\]](#)
22. Iorio, M.; Reineri, M.; Risso, F.; Sisto, R.; Valenza, F. Securing SOME/IP for in-vehicle service protection. *IEEE Trans. Veh. Technol.* **2020**, *69*, 13450–13466. [\[CrossRef\]](#)
23. Jeong, S.; Jeon, B.; Chung, B. Convolutional neural network-based intrusion detection system for AVTP streams in automotive ethernet-based networks. *Veh. Commun.* **2021**, *29*, 100338. [\[CrossRef\]](#)
24. Han, M.L.; Kwak, B.I.; Kim, H.K. TOW-IDS: Intrusion detection system based on three overlapped wavelets for automotive ethernet. *IEEE Trans. Inf. Forensics Secur.* **2022**, *18*, 411–422. [\[CrossRef\]](#)
25. da Luz, L.F.M.; de Araujo-Filho, P.F.; Campelo, D.R. Enhancing cybersecurity of automotive ethernet networks with deep learning-based intrusion detection systems. In *Proceedings of the Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC), Natal, Brazil, 19–23 May 2025*; SBC OpenLib (SOL): Alegre, Brazil, 2025; pp. 222–231.
26. Bamber, S.S.; Katkuri, A.V.R.; Sharma, S. A hybrid CNN-LSTM approach for intelligent cyber intrusion detection system. *Comput. Secur.* **2025**, *148*, 104146. [\[CrossRef\]](#)
27. Farrukh, Y.A.; Wali, S.; Khan, I. AIS-NIDS: An intelligent and self-sustaining network intrusion detection system. *Comput. Secur.* **2024**, *144*, 103982. [\[CrossRef\]](#)
28. Dakic, P.; Zivkovic, M.; Jovanovic, L.; Bacanin, N.; Antonijevic, M.; Kaljevic, J.; Simic, V. Intrusion detection using metaheuristic optimization within IoT/IIoT systems and software of autonomous vehicles. *Sci. Rep.* **2024**, *14*, 22884. [\[CrossRef\]](#) [\[PubMed\]](#)
29. Wang, X.; Geng, Y.; Luo, X. Detection of dummy data injection attacks by using particle swarm optimization-attention temporal graph convolutional network model in power system. *Eng. Appl. Artif. Intell.* **2026**, *171*, 114259. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.