electronics

MDPI

Article

Study of Network Anomaly Detection for In-Vehicle Ethernet Using Fuzzy Clustering

Siwen Liu, Yue Jia, Kaihang Zhang, Yujing Wu, Yihu Xu and Yinan Xu *

College of Engineering, Yanbian University, Yanji 133002, China; 2024050061@ybu.edu.cn (S.L.); 2024050064@ybu.edu.cn (Y.J.); 2689332993@ybu.edu.cn (K.Z.); yjwu@ybu.edu.cn (Y.W.); xuyh@ybu.edu.cn (Y.X.)
* Correspondence: ynxu@ybu.edu.cn

Abstract

Along with the swift evolution of autonomous driving and internet technologies, In-Vehicle Ethernet has evolved into the core backbone network underpinning the new generation of in-vehicle networks (IVNs). Since In-Vehicle Ethernet is susceptible to a host of cybersecurity threats—such as data pilferage, data falsification, and malicious unauthorized access—it is imperative to enhance its defense capabilities. This research focuses on anomaly identification for In-Vehicle Ethernet communication networks, with a specific focus on the intrinsic data features of the AVTP protocol and potential cyber-attack vectors targeting the network. This work develops a novel network anomaly detection approach rooted in the Fuzzy clustering algorithm. This effectively enhances the cybersecurity performance of In-Vehicle Ethernet. Experimental results demonstrate that the Fuzzy clustering algorithm proposed in this study achieves 97.4% accuracy in detecting anomalous data, outperforming the traditional K-Means and OPTICS clustering algorithms by 6.4% and 14.5% respectively in anomaly detection rate. This further elevates the cybersecurity performance of In-Vehicle Ethernet and forges a robust foundation for the stable operation and iterative advancement of intelligent connected vehicles (ICVs).

Keywords: In-Vehicle Ethernet; network security; anomaly detection; AVTP protocol; fuzzy clustering algorithm

1. Introduction

As vehicles grow increasingly intelligent and connected, the connectivity between on-vehicle electronic control units (ECUs) and external communication networks is intensifying. While this integration brings substantial convenience to drivers, it also exposes IVNs to new cybersecurity risks. By exploiting vulnerabilities in these networks, hackers can steal sensitive vehicle data, thereby endangering the safety of drivers and passengers as well as public security [1]. IVNs are growing increasingly vulnerable to malicious intrusions, with attack vectors evolving toward greater complexity and diversity—ranging from simple protocol-level vulnerabilities to sophisticated network infiltration tactics. Therefore, the effective detection of network attacks is critical to safeguarding the safety and reliability of intelligent connected vehicles [2].

Traditional in-vehicle bus networks such as LIN, CAN, and FlexRay connect control devices such as the engine, window motors, and brake system controllers, while In-Vehicle Ethernet connects multimedia devices such as high-definition cameras, radar, and dash-cams [3,4]. When connecting external devices via In-Vehicle Ethernet, the vehicle faces security threats such as malicious code implantation, data transmission interception, and

Check for updates

Received: 12 January 2026
Revised: 6 February 2026
Accepted: 9 February 2026
Published: 10 February 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) license.

data tampering. These threats can lead to intrusion into the vehicle control system and leakage of sensitive information [5]. Therefore, building anomaly detection systems (ADSs) capable of accurately identifying attack behaviors and continuously monitoring network traffic is an important research direction in the field of vehicle network security [6].

The network anomaly detection system primarily detects malicious network attacks and abnormal network data. Traditional network anomaly detection technologies rely on methods such as port access control and attack signature detection, making it difficult to accurately identify new types of network attacks, such as malware attacks, attacks that tamper with attack code, attacks that change attack paths, and attacks that alter the frequency of data transmission. Network anomaly detection methods based on vehicular networks mainly include real-time anomaly detection methods such as deep learning and statistics [7]. S. Rajapaksha et al. classified anomaly detection techniques for IVNs into three main classes: traditional machine learning methodologies, deep learning paradigms, and hybrid learning frameworks. Further improvements are needed in areas such as increasing dataset size, detecting low-frequency attacks, and reducing detection latency [8]. J. Zhang et al. proposed a network anomaly detection method based on multi-objective optimization to address the contradictions among different network anomaly detection metrics. This method demonstrates good accuracy, low false positive rate, and short anomaly detection time in Denial of Service (DoS) attack detection [9]. M. Han et al. proposed a network anomaly detection system based on three overlapping wavelet algorithms. The system detected AVTP frame injection attacks, PTP synchronization attacks, CAM table overflow attacks, DoS attacks, and replay attacks with an accuracy rate of 99.65% [10]. Currently, the network anomaly detection technology for vehicle networks mainly adopts network anomaly detection methods based on supervised learning detection modes, which are difficult to meet the real-time and security requirements of ICV. Therefore, network anomaly detection methods based on unsupervised learning detection models are key technologies for ensuring the network security of IVNs. Nevertheless, existing unsupervised paradigms still suffer from prominent limitations: failure to dynamically update the distribution of normal data, inadequate extraction of key features from high-dimensional data, and difficulties in achieving real-time detection. This study focuses on network anomaly detection for the AVTP protocol in In-Vehicle Ethernet. Leveraging an unsupervised learning paradigm, we propose a fuzzy clustering-based anomaly detection approach that achieves end-to-end training via a unified framework fusing reconstruction error, fuzzy clustering, and entropy regularization. Furthermore, boundary optimization is integrated to refine the differentiation of ambiguous boundary data, ultimately boosting the accuracy of network anomaly detection while cutting down computational latency.

The remainder of this paper is organized as follows: Section 2 presents the AVTP protocol data frame format and associated network security analysis for In-Vehicle Ethernet; Section 3 details the technical underpinnings of the fuzzy clustering algorithm; Section 4 carries out experimental validation and performance assessment using an In-Vehicle Ethernet anomaly detection dataset; and Section 5 summarizes the key findings and contributions of the study.

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

With the rapid development of ICV, the installation of entertainment systems, Advanced Driver Assistance Systems (ADASs), and autonomous driving systems has significantly improved the driving experience, but it has also brought about cybersecurity issues such as security vulnerabilities and hacker attacks. Therefore, traditional IVNs are no longer completely independent and secure network systems [11]. The AVTP protocol for In-Vehicle Ethernet, which adapts to various Audio and Video (AV) formats, is widely

used in autonomous driving and in-vehicle entertainment scenarios. ICV enhance the environmental perception capabilities of autonomous driving and in-vehicle entertainment systems by integrating multimodal sensors, enabling core functions such as vehicle environment prediction, path planning, and collision warning [12]. The open architecture and inherent complexity of the AVTP protocol render this protocol susceptible to security vulnerabilities such as data loss, transmission delays, and network intrusions in the course of data transmission processes. To ensure the network security performance of automobiles and prevent data tampering and packet loss, it is essential to guarantee the reliability of critical data frames, thereby enhancing the security of the vehicle bus network [13,14].

2.1. Introduction to the AVTP Protocol for In-Vehicle Ethernet

The AVTP protocol for In-Vehicle Ethernet is a core AV data transmission protocol defined by the IEEE 1722 standard [15]. It requires Ethernet to operate in 100 Mbit/s full-duplex mode to meet the high-bandwidth, low-jitter, and low-latency AV communication requirements of autonomous driving systems and in-vehicle entertainment systems [14]. The AVTP protocol is based on Time-Sensitive Networking (TSN) traffic scheduling and time synchronization mechanisms to ensure the stability of AV data transmission [16]. The AVTP protocol uses timestamps, priority classification, and precise frame synchronization to achieve deterministic transmission and synchronous playback of multi-channel media streams.

An AVTP protocol data frame comprises a header, a destination MAC address, a source MAC address, an 802.1Q header, an Ethernet-type field, an AVTP packet, and a cyclic redundancy check/frame check sequence (CRC/FCS). Correspondingly, an AVTP protocol packet comprises a header, stream IDs, the presentation time, format and packet information, and the AV data to be transmitted. The header specifies the type and sequence number of the AV data, which serves to verify whether any data packets have been lost. Meanwhile, the data Stream Identifier (Stream ID) denotes the name of the corresponding data stream. In addition, the presentation time refers to the duration required for the receiver to deliver the received AV data to the application layer for subsequent presentation. The AVTP protocol supports a variety of uncompressed AV data formats within its transmitted AV data payload, including 61883-6, 3550/AAF, and AVTP H.264. These frame formats are distinguished by distinct header architectures and corresponding applicable technical specifications. Specifically, the AVTP 61883-6 frame format is engineered specifically for audio signal transmission within in-vehicle entertainment systems and infotainment platforms; AVTP 3550/AAF is designed for high-resolution audio processing; and AVTP H.264 is dedicated to AV synchronization processing. Figure 1 presents the format of the AVTP protocol data frame applied to In-Vehicle Ethernet.

Data transmission using the AVTP protocol requires information exchange between the sending end (Transmit) and the receiving end (Receive). Efficient communication is achieved when the sending and receiving ends are located in the identical Virtual Local Area Network (VLAN) segment. At the sending end, AV data are collected and generated before being encapsulated into AVTP data packets. Each packet incorporates core elements including AV data, presentation time, stream ID, and additional control information. During transmission, the sending end sets priorities and schedules data packets according to Time-Sensitive Networking (TSN) protocols such as IEEE 802.1Qbv, thereby reducing transmission delay. At the receiving end, according to the format of the AVTP protocol data packets, the header information is parsed to obtain control information such as presentation time and data Stream ID, and AV payload data is extracted to achieve synchronous playback of multi-channel media streams.

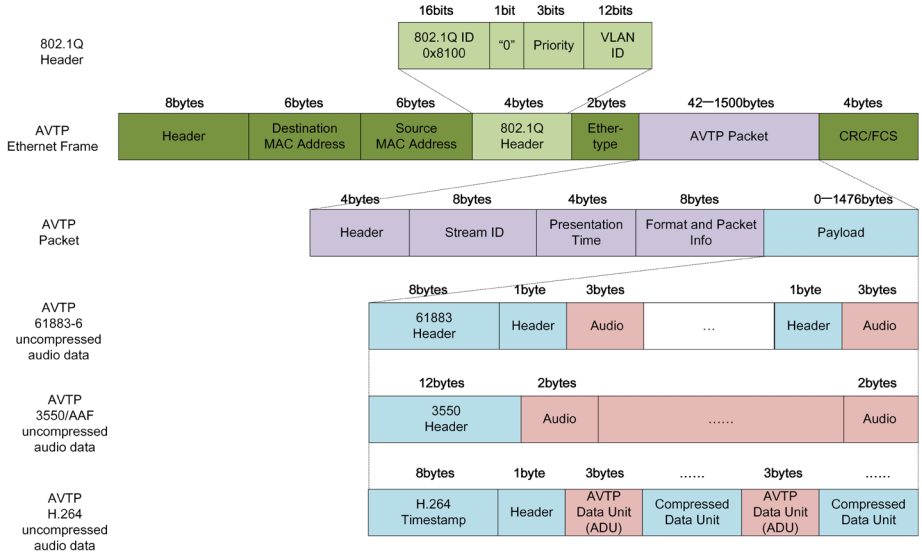


Figure 1. AVTP protocol data frame format.

2.2. Network Security Analysis of In-Vehicle Ethernet

Boasting high bandwidth and ultra-low latency, In-Vehicle Ethernet has emerged as the backbone network for data transmission in ADAS and autonomous driving systems, providing core network support for automotive intelligence. However, In-Vehicle Ethernet faces new types of network attacks, such as malware attacks, code tampering, changes in attack paths, and alterations in data transmission frequencies, which seriously threaten vehicle safety and user privacy [17,18]. Among them, malware attacks tamper with the instructions of in-vehicle ECUs by implanting disguised control programs, directly interfering with critical vehicle functions such as steering and braking. Code tampering bypasses the signature verification mechanism of in-vehicle software, replaces code snippets to inject malicious data, and forces ECUs to execute erroneous operations. Attack path modification leverages the multi-node communication topology of in-vehicle networks to circumvent security isolation zones and launch attacks on the core control units of the in-vehicle network. Altering data transmission frequency forges the message sending timing in automotive Ethernet, and the injection of abnormally high or low-frequency signals disrupts the communication synchronization between in-vehicle sensors and ECUs.

During data transmission using the AVTP protocol, both the sending and receiving ends become targets for hacker attacks. B. Dong et al. described the hacker's attack process. Hackers exploit security vulnerabilities to attack the sending or receiving end of data transmission, interfering with normal data transmission and processing, compromising data integrity during communication, and affecting the normal operation of vehicle systems [19]. When a vehicle connects to external networks such as WiFi, Bluetooth, or a mobile phone, hackers can exploit the network interface to steal the vehicle's location information, AV data, and user privacy information, enabling them to carry out cyberattacks such as remote vehicle control, sensitive information theft, and control command tampering [20].

Traditional research on network security for vehicular Ethernet mainly employs firewall-based network access control, cryptographic data protection, and anomaly detection techniques [21]. Firewall-based network access control restricts data traffic between network nodes and prevents malicious traffic intrusion by setting filtering rules in the vehicle network. E. Allen et al. proposed a network security method for In-Vehicle Ethernet networks based on a distributed TCAM firewall. Each area gateway uses TCAM's preset security policy rules for senders, data types, and receivers to match data in each area in real time, preventing unauthorized and forged data transmission and ensuring vehicle network

security [22]. F. Klement et al. proposed a protocol-independent firewall (Man-in-the-OBD). Deploy a firewall between the third-party device and the car's OBD-II interface, and use methods such as filtering, modifying, or delaying messages on the OBD-II interface to block unauthorized malicious traffic [23]. Cryptographic technology encrypts transmitted data using encryption algorithms and verifies data integrity and sender identity using methods such as Message Authentication Code (MAC) and digital signatures. It accurately detects unauthorized data tampering and enhances network resistance to attacks. J. Chen et al. proposed a In-Vehicle Ethernet network security solution based on key management technology. By randomly generating keys and distributing them using elliptic curve and Schnorr signature algorithms, and encrypting authentication data using AES-128 and HMAC algorithms, the efficiency of In-Vehicle Ethernet information processing is improved, ensuring the confidentiality, integrity, and authenticity of communication [24]. Y. Zhu et al. proposed a secure communication scheme for In-Vehicle Ethernet leveraging the post-quantum cryptographic algorithm NTRUEncrypt. At a 128-bit security level, this post-quantum solution achieves session key negotiation speeds 66.06 times quicker than the Elliptic Curve Diffie-Hellman (ECDH) protocol and 1530.98 times faster than the Rivest-Shamir-Adleman (RSA) protocol, thereby delivering a highly efficient and robust session key exchange mechanism tailored for In-Vehicle Ethernet environments [25]. Compared to static defense technologies such as cryptography and firewalls, network anomaly detection technology can dynamically analyze the behavioral characteristics of data traffic and proactively identify different network attack patterns. It can promptly detect potential threats and provide more flexible and proactive security protection for vehicle networks. Li et al. proposed the FOAP scheme to tackle the closed-set limitations and coarse-grained drawbacks inherent in existing application fingerprinting methods. This scheme enables Android application identification and method-level user behavior inference over encrypted traffic by filtering irrelevant traffic via structural similarity, automatically annotating network flows, and constructing spatiotemporal context models—while also exhibiting compatibility with the AProxy traffic obfuscation defense mechanism. Experimental results have validated the accuracy and practicality of the proposed identification approach [26]. Notably, Ni et al. proposed a novel side-channel attack method based on radio frequency (RF) energy harvesting, and designed the AppListener automated framework for this purpose. By capturing RF energy signals emitted by Wi-Fi modules, the attack achieves high-precision identification of mobile devices, associated applications, and fine-grained user activities via a three-layer classification algorithm—with targeted defensive countermeasures also put forward by the research team [27]. Y. Wang et al. proposed an anomaly detection method based on a weighted histogram algorithm, utilizing the characteristics of the AVTP (In-Vehicle Ethernet) protocol. This method demonstrates good accuracy, low false alarm rate, and short anomaly detection time in replay attack detection, thus improving the operational stability and anti-attack capability of In-Vehicle Ethernet [28]. R. Xu et al. developed a network anomaly detection method grounded in self-supervised learning and graph neural networks. This method detected network attacks such as denial-of-service attacks, brute-force attacks, and reconnaissance attacks with an F1 score of over 94%, and categorized the network attacks accordingly [29]. T. Hoang et al. devised an anomaly detection method rooted in semi-supervised learning for Convolutional Adversarial Autoencoders (CAAEs). This method combines convolutional autoencoders with generative adversarial networks to detect denial-of-service attacks, fuzzing attacks, and spoofing attacks in vehicular networks using a small amount of labeled data [30].

To ensure the security of data transmission in In-Vehicle Ethernet, prevent data from being stolen or tampered with during transmission, and ensure the confidentiality and integrity of data frames, an in-depth analysis of network attack types and security protection

systems should be conducted to design efficient network anomaly detection methods and security protection strategies, thereby comprehensively improving the network security of In-Vehicle Ethernet.

3. Network Anomaly Detection Method Based on Fuzzy Clustering Algorithm

To address the network security performance of In-Vehicle Ethernet, a network anomaly detection method based on Fuzzy clustering is proposed. The network anomaly detection method for In-Vehicle Ethernet is presented in Figure 2. This method consists of three parts: data preprocessing, data autoencoder compression and dimensionality reduction, and network anomaly detection. By enhancing the ability to detect abnormal AV, an efficient and reliable solution is provided for secure communication via In-Vehicle Ethernet.

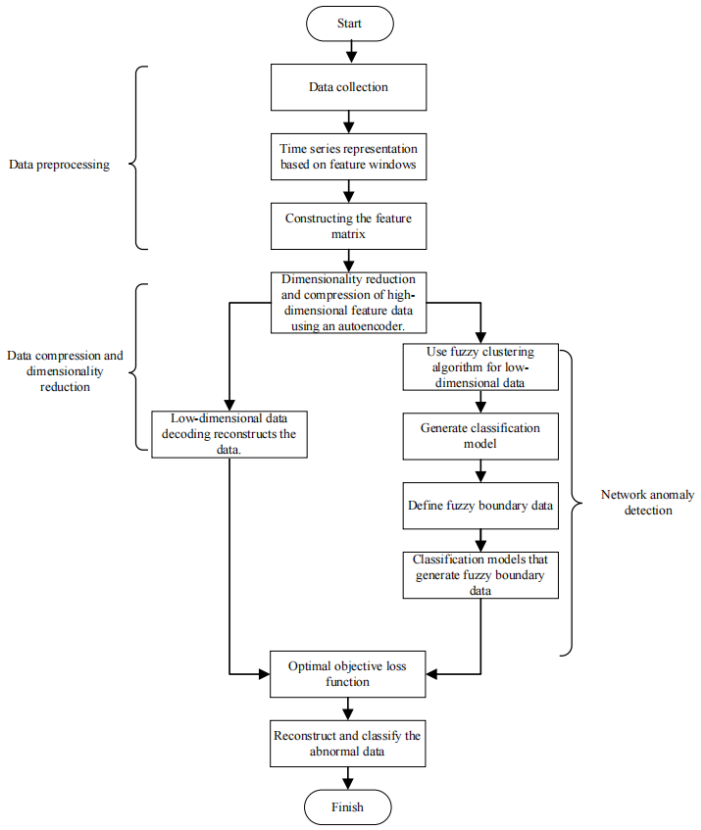


Figure 2. Flowchart of Network Anomaly Detection.

3.1. Data Preprocessing

An AVTP protocol data frame has a fixed length of 438 bytes, where each byte holds an integer value ranging from 0 to 255. The first 58 bytes contain critical control information such as the MAC address, the 802.1Q header, the Ethernet-type field, AVTP protocol header information, the flow identifier, and the payload field. Meanwhile, Jeonga et al. visualized the payloads of 100 consecutive stream AVTPDUs and identified a consistent pattern in the first 58 bytes, which enables effective differentiation between benign and injected data [31]. This study performs network anomaly detection on the first 58 key byte segments of each data frame, effectively monitoring network traffic and detecting abnormal data while reducing the complexity of data processing and satisfying the real-time performance requirements of In-Vehicle Ethernet networks.

Figure 3 shows a feature-based sliding window mechanism. The sliding window mechanism can dynamically update data to naturally connect the data before and after it while maintaining the continuity of the context for a short period of time. When new data

arrives, the window automatically updates its content and performs real-time network anomaly detection.

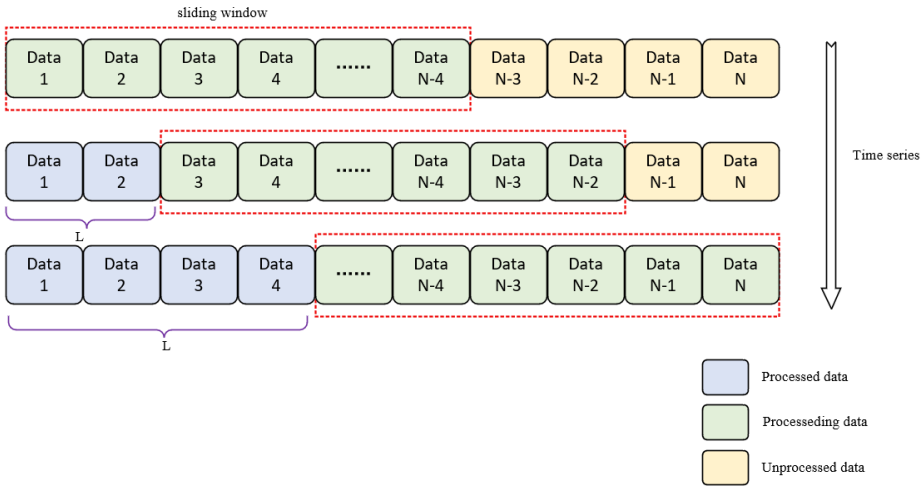


Figure 3. Feature-based sliding window mechanism.

N represents the total count of data frames. For each data frame in the set $(Frame\ 1,\ Frame\ 2,\ \cdots,\ Frame\ N)$, we extract its first 58 bytes and use these byte sequences to form the feature matrix Y, as shown in Formula (1). Specifically, the row dimension of Y is defined by the sliding window size ω, and the column dimension corresponds to the feature values of the first 58 bytes from each individual data frame.

$$Y = \begin{bmatrix} Y_{1,1} & \cdots & Y_{1,58} \\ \vdots & \ddots & \vdots \\ Y_{\omega,1} & \cdots & Y_{\omega,58} \end{bmatrix}_{\omega \times 58} \tag{1}$$

$Y_{i,j}$ denotes the data frame located in the i-th row and j-th column. The feature vector associated with the i-th data frame, denoted as Y_i, is given in Equation (2).

$$Y = [Y_{i,1}, Y_{i,2}, Y_{i,3}, \cdots\cdots, Y_{i,58}] \tag{2}$$

3.2. Dimensionality Reduction in Autoencoder Data

Autoencoders can be used to achieve core functions such as data feature extraction, data dimensionality reduction, and data reconstruction for specific tasks. Figure 4 is a flowchart of the autoencoder. An autoencoder maps a high-dimensional input to a low-dimensional latent space, and a decoder reconstructs the original data from the latent space, capturing the core features of the data by minimizing the reconstruction error. Autoencoders can automatically extract key features from data, effectively reduce the dimensionality of high-dimensional data, and efficiently reduce the complexity of data processing.

The network data of In-Vehicle Ethernet is characterized by high dimensionality and dynamic changes. Traditional network anomaly detection methods struggle to effectively extract key features and accurately capture dynamic data changes when processing high-dimensional data, failing to meet the real-time and security requirements of vehicular communication networks. Autoencoders learn normal data feature patterns autonomously through unsupervised learning, without relying on labeled attack samples. Meanwhile, self-encoders have relatively low computational complexity and a small number of parameters, making them suitable for deployment in the limited resource environment of automotive Electronic Control Units (ECUs). This study adopts a symmetric fully con-

nected autoencoder architecture, where both the encoder and the decoder consist of three network layers. To balance the accuracy of high-dimensional feature extraction with the limited computing resources of in-vehicle ECUs, this design not only ensures the effective compression of key information from AVTP frames and the accurate restoration of core data features but also avoids computational latency caused by an excessive number of layers, thus well adapting to the real-time requirements of in-vehicle scenarios.

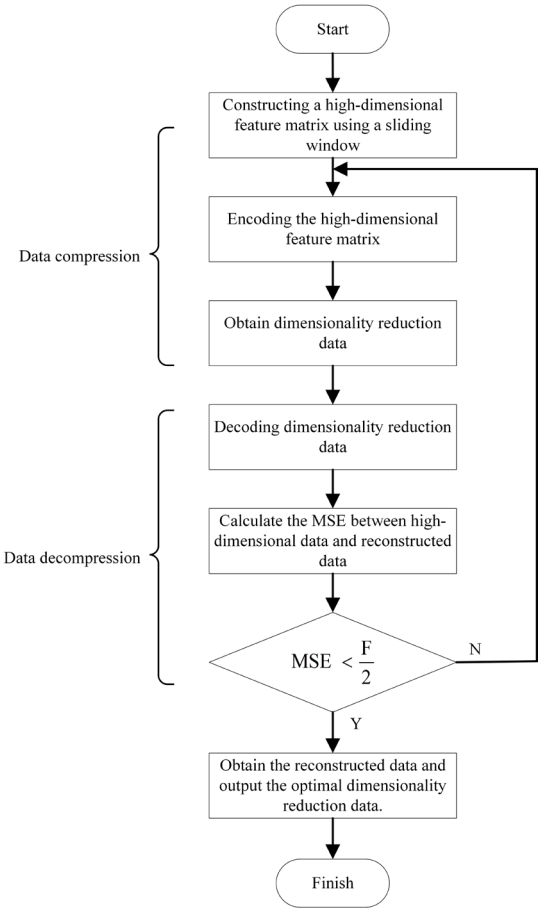


Figure 4. Flowchart of the autoencoder.

For the 58-byte feature matrix, the data is scaled to a uniform range to avoid the scale differences in different features affecting model training. Formula (3) uses an autoencoder to map the input high-dimensional data Y to low-dimensional data X, where ℓ is the activation function; W_enc is the weight matrix in the encoding process, which performs a weighted transformation on the input high-dimensional data Y; and b_enc is the result after bias adjustment and weighting in the encoding process.

$$X = \ell_{enc}(W_enc \cdot Y + b_enc) \tag{3}$$

Formula (4) is an activation function that can preserve protocol features related to attacks and effectively alleviate the problem of gradient vanishing, where Y' denotes the data generated by the linear transformation. In Formula (3), Y' corresponds to the data derived from the linear transformation $(W_enc \cdot Y + b_enc)$. In Formula (5), Y' refers to the data yielded by the linear transformation $(W_dec \cdot X + b_dec)$.

$$\ell_{enc} = \ell_{dec} = max(0, Y') \tag{4}$$

Formula (5) uses a decoder to restore and map low-dimensional data X to high-dimensional data Z, where ℓ is the activation function; W_dec is the weight matrix of the decoding process; which is used to perform weighted calculations on the input low-dimensional data X; and b_dec is the result of the weighted adjustment of the bias peak after decoding. The optimization goal of the decoder is to minimize reconstruction error and ensure that the compressed data retains the core information of the original data.

$$Z = \ell_{dec}(W_dec \cdot X + b_dec) \tag{5}$$

The reconstruction error in an autoencoder is a key indicator of a model's ability to restore low-dimensional data to the original data. Equation (6) quantifies the discrepancy between the input data and the reconstructed data generated by the decoder. In this formulation, N stands for the number of samples, and the mean square error of the difference between the true value Y_i and the predicted value Z_i of all samples is calculated, which quantifies the degree to which the autoencoder preserves the original features.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(Y_i - Z_i)^2 \tag{6}$$

Formula (7) is used to determine the degree of data reconstruction. Among them, F is a reconstruction similarity threshold between 0 and 1. In this paper, a reconstruction similarity threshold of $F = 0.5$ is chosen to ensure that the data retains key features while preventing an increase in the complexity of model training. When $MSE \geq 0.25$, it is necessary to retrain the autoencoder model. When $MSE < 0.25$, it is used for anomaly detection in subsequent fuzzy clustering.

$$MSE < \frac{F}{2} \tag{7}$$

Low-dimensional data is obtained by feature extraction and data compression of high-dimensional vehicle Ethernet data through autoencoders. Redundant information in the data is accurately removed; key features that reflect the core characteristics of the data are obtained, and the performance degradation of the model caused by excessive dimensionality is avoided. Low-dimensional data represents the distributional differences between normal and abnormal data, providing efficient feature input for subsequent detection algorithms such as Fuzzy clustering, thus effectively ensuring and improving detection accuracy.

3.3. Network Anomaly Detection

Figure 5 is a flowchart of network anomaly detection using the Fuzzy clustering analysis method. In low-dimensional data, cluster centers are calculated by minimizing the objective loss function, and the data are classified into normal and abnormal data based on the membership degree of the data calculated from the cluster centers. Among them, cluster centers represent the core features of normal and abnormal data, and membership degree indicates the probability value of each data point belonging to a certain cluster.

In traditional autoencoders, since the loss of the autoencoder depends on the reconstruction loss of the original data, it is difficult to distinguish normal data from abnormal data in low-dimensional data. In this study, fuzzy clustering into an autoencoder framework to enable their joint learning. In contrast to other unsupervised anomaly detection methods—such as hard clustering algorithms, isolation forests, and standalone autoencoders—fuzzy clustering is better suited to addressing the high noise and dynamic topological features inherent in vehicular data, by virtue of its membership degree assignment and boundary optimization mechanisms. Moreover, the joint learning of fuzzy

clustering and the autoencoder further caters to the real-time performance requirements of vehicular communication scenarios, which not only effectively reduces the false alarm rate but also enhances the detection accuracy of anomaly identification in vehicular networks.

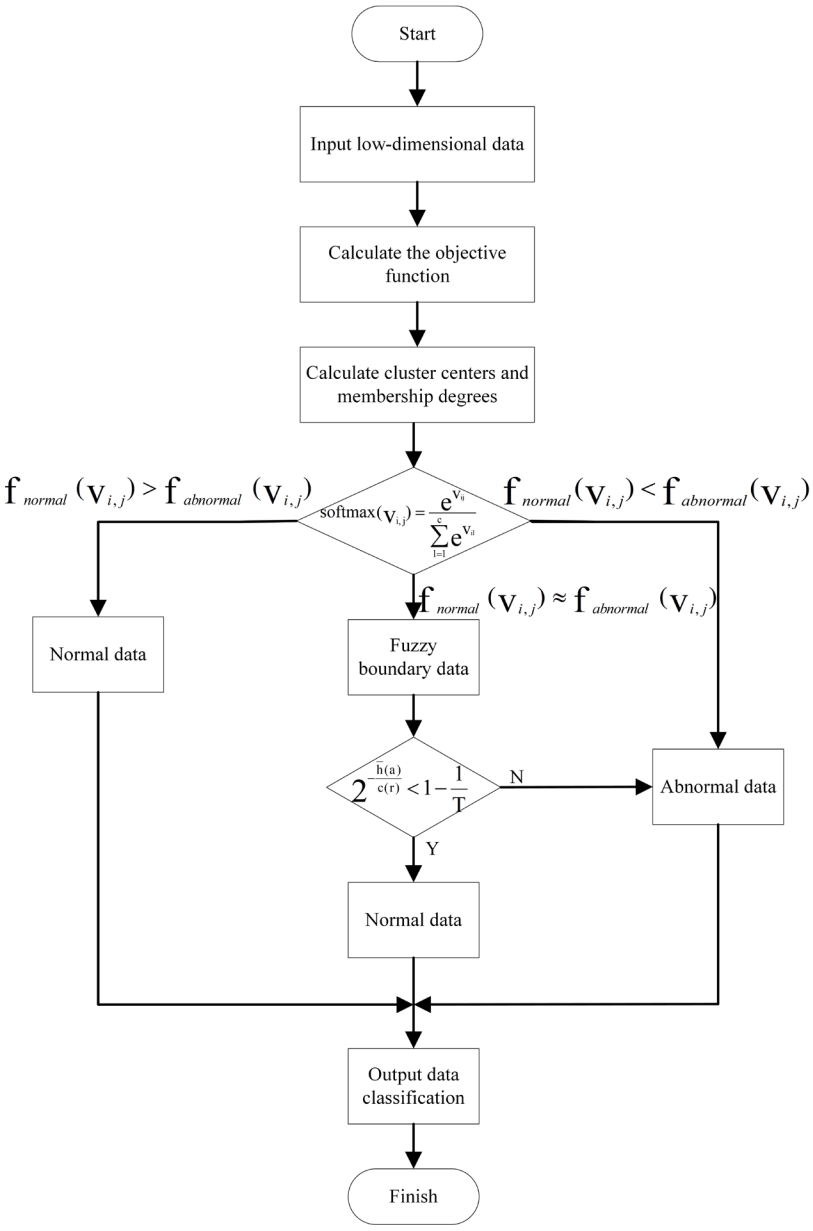


Figure 5. Flowchart for network anomaly detection.

Formula (8) represents the target loss function for Fuzzy clustering. By integrating three losses—reconstruction error, Fuzzy clustering, and entropy regularization—the model's data reconstruction accuracy, Fuzzy clustering effect, and stability of the clustering process are measured to reconstruct and classify low-dimensional data.

$$\Delta\mathcal{L} = \lambda_1 \sum_{i=1}^{N} (Y_i - Z_i)^2 + \lambda_2 \sum_{i=1}^{N} \sum_{j=1}^{c} f(V_{i,j}) (X_i - C_j)^2 + \lambda_3 \sum_{i=1}^{N} \sum_{j=1}^{C} f(V_{i,j}) lnf(V_{i,j}) \quad (8)$$

where $\lambda_1 \sum_{i=1}^{N} (Y_i - Z_i)^2$ is the reconstruction error loss term, which measures the model's data reconstruction accuracy and quantifies the discrepancy between the original data Y_i and the reconstructed data Z_i. $\lambda_2 \sum_{i=1}^{N} \sum_{j=1}^{c} (X_i - C_j)^2$ is the Fuzzy clustering loss term, which reflects the effect of Fuzzy clustering. $\lambda_3 \sum_{i=1}^{N} \sum_{j=1}^{C} f(V_{i,j}) lnf(V_{i,j})$ is the entropy

regularization loss term. $f(V_{i,j})$ is the membership degree of data i belonging to cluster j. N represents the number of samples of low-dimensional data x_i, C denotes the count of cluster centers, while $V_{i,j}$ denotes the membership degree corresponding to different cluster centers.

Formula (9) is a constraint condition for Fuzzy clustering, ensuring that the membership degree is within the range of [0, 1], representing the probability value of the data belonging to different categories, and the sum of the probability values is equal to 1.

$$\sum_{j=1}^{c} f(V_{i,j}) = 1 \tag{9}$$

Formula (10) is the judgment condition for calculating normal and abnormal data. Among them, the membership value of normal data $f_{normal}(V_{i,j})$ is much greater than the membership value of abnormal data $f_{abnormal}(V_{i,j})$, and the data is judged to be normal data; on the contrary, it is abnormal data. If the membership values of normal data $f_{normal}(V_{i,j})$ and abnormal data $f_{abnormal}(V_{i,j})$ are similar, the data is determined to be fuzzy boundary data.

$$\sum_{j=1}^{c} f_{normal}(V_{i,j}) + \sum_{j=1}^{c} f_{abnormal}(V_{i,j}) = 1 \tag{10}$$

The location of cluster center C is calculated using Formula (11). Cluster centers are determined by employing the objective function, constraints, and Lagrange algorithm in Fuzzy clustering. The position of C is dynamically adjusted using an objective function, ensuring that C is always located in a high-density region of the data distribution. The position of C is used to distinguish between normal data and abnormal data.

$$C_j = \frac{X_i \sum_{i=1}^{N} V_{i,j}}{\sum_{i=1}^{N} V_{i,j}} \tag{11}$$

The membership degree of the cluster centers is calculated using Formula (12). By adjusting the clustering of low-dimensional data by using the membership value $V_{i,j}$ between low-dimensional data point i and the cluster center, the membership value can accurately reflect the probability of each data point belonging to different clusters, thereby achieving the purpose of network anomaly detection.

$$softmax(V_{i,j}) = \frac{e^{v_{ij}}}{\sum_{l=1}^{c} e^{v_{il}}} \tag{12}$$

Fuzzy boundary data consists of data points with similar membership degrees, so further data classification is needed for fuzzy boundary data. Formulas (13) to (15) employ the classification method using multiple binary trees to classify fuzzy boundary data. The average path length of the fuzzy boundary data point r in different tree t structures is calculated using Formula (13) to preliminarily classify abnormal data, where r represents the sample data of the fuzzy boundary and t represents the number of trees.

$$\overline{h}(a) = \frac{1}{t} H_{r \times t} \cdot 1_{t \times 1} \tag{13}$$

The expected path length $c(r)$ is calculated using Formula (14). By comparing the average path length $\overline{h}(a)$ of each data point with the expected path length $c(r)$, data is described as anomalous if the actual average path length deviates significantly from the expected value. Among them, $H(r-1)$ is the harmonic number of normal data dispersion

characteristics, and $\frac{2(r-1)}{r}$ is the correction term, which adjusts the expected value of path length $c(r)$ according to the data point r.

$$c(r) = 2H(r-1) - \frac{2(r-1)}{r} = 2[ln(r-1) + 0.577] - \frac{2(r-1)}{r} \quad (14)$$

Based on the relative relationship between the average path length $\overline{h}(a)$ and the expected path length $c(r)$, the anomaly score $s(a,r)$ is calculated as shown in Formula (15).

$$s(a,r) = 2^{-\frac{\overline{h}(a)}{c(r)}} \quad (15)$$

The abnormal score $s(a,r)$ is judged by using a random number T, as shown in Formula (16). When the value of T is larger, the closer the abnormal score $s(a,r)$ is to 1, indicating abnormal data.

$$T\epsilon[2, +\infty] \quad (16)$$

By calculating the average path length $\overline{h}(a)$, the expected path length $c(r)$ and the abnormal score $s(a,r)$, the fuzzy boundary data can be further classified.

4. Experimental Simulation and Performance Result Analysis

CANoe serves as a specialized simulation platform for the design and performance assessment of in-vehicle bus systems, with native support for a full range of in-vehicle communication protocols such as LIN, CAN, FlexRay and In-Vehicle Ethernet. For the present study, we built an experimental testbed for In-Vehicle Ethernet by combining the CANoe.Ethernet tool with a hardware development board and further validated the overall performance of the proposed anomaly detection algorithm on this platform.

In-Vehicle Ethernet's network topology is depicted in Figure 6. Among them, data is sent from the transmit node, abnormal data is simulated and injected in the Ethernet IG node, the Gateway node realizes data transmission between different buses, data is received in the receive node, multimedia data is played in the Player node, and abnormal data is detected in the Detected node. All of the above communication nodes are linked together via Ethernet Bus (Eth 1).

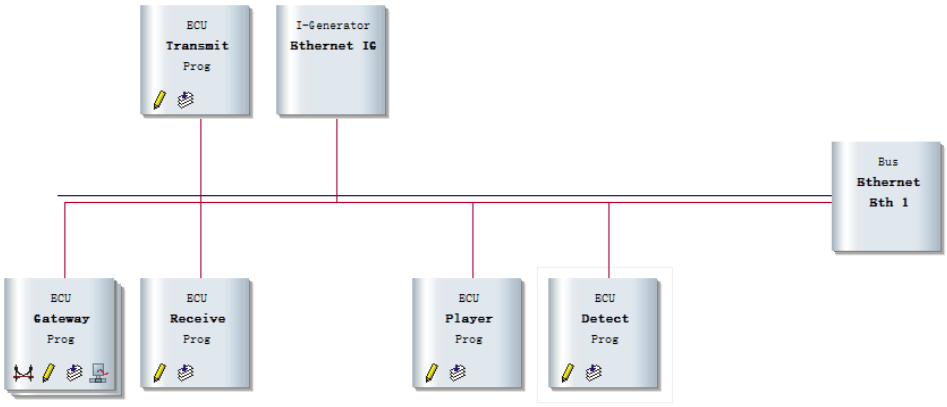


Figure 6. Network topology architecture of In-Vehicle Ethernet.

The hardware experimental environment of In-Vehicle Ethernet is shown in Figure 7. Among them, three MPC5646C hardware development boards for In-Vehicle Ethernet implement the Transmit node, Ethernet IG node, and Receive node. The CANoe.Ethernet system on the PC has a Gateway node, a Player node, and a Detected node. The VN5610 hardware features an Ethernet communication interface.

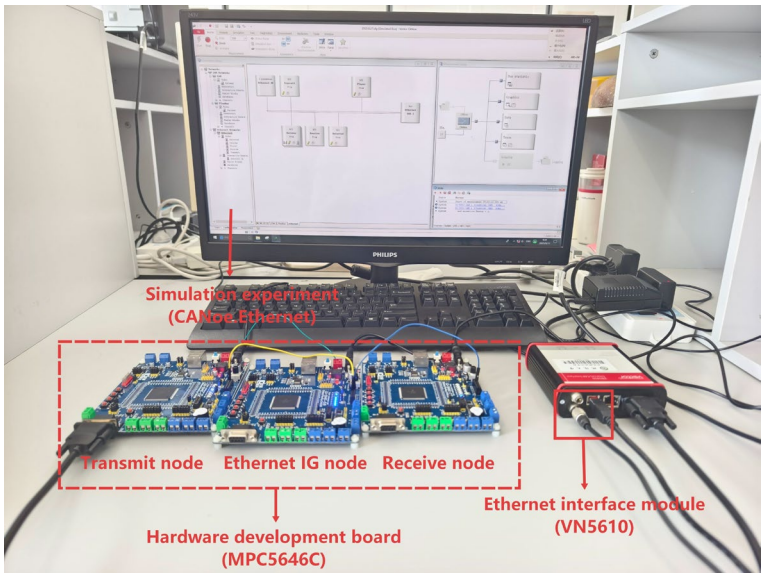


Figure 7. Actual experimental environment.

To ensure the authenticity and reproducibility of the experimental results, this study focuses on an AVTP protocol dataset for In-Vehicle Ethernet, which is stored in the PCAP file format and can be visualized and analyzed using dedicated tools such as Wireshark. Within In-Vehicle Ethernet environments, adversaries may launch replay attacks by injecting AVTPDUs (Audio Video Transport Protocol Data Units) that carry audio and video payloads. Specifically, pre-transmitted AVTPDUs are re-injected into the in-vehicle Ethernet link over a short timeframe, leading the video playback terminal to output duplicate video frames erroneously. This disruption impairs the normal operation of the in-vehicle audio and video system and degrades user experience.

This study performs experimental validation on the AVTP protocol data of In-Vehicle Ethernet via a randomly selected dataset (RSD), as shown in Table 1. The RSD in Table 1 lists packet counts for normal and attack samples. As a dedicated dataset for In-Vehicle Ethernet anomaly detection, the RSD is partitioned into training and test subsets at an 8:2 split, with a balanced distribution of benign and malicious samples in each subset. Specifically, the training set is used for model training and parameter optimization, while the test set serves to assess the model's anomaly detection performance on unseen data. The attack data consists of 36 consecutive AVTPDUs extracted from a public dataset. These 36 consecutive AVTPDUs can reconstruct the complete video replay attack behavior. Repeated injection of these sequential AVTPDUs into automotive Ethernet generates the anomalous packets in the public RSD, which simulates attack scenarios against the AVTP protocol and provides realistic and valid attack samples for the training and validation of the anomaly detection model.

Table 1. Dataset of the AVTP Protocol for In-Vehicle Ethernet.

Database	Normal Dataset (Packets)	Abnormal Dataset (Packets)	Size (KB)
RSD	771,043	199,656	433,729
Attack	0	36	16

Obtain the optimal target loss function during network anomaly detection and data compression dimensionality reduction. Among them, the loss function is composed of three parameters: reconstruct loss weights λ_1, fuzzy loss weights λ_2, and entropy regularization weights λ_3, as shown in Table 2. To determine the optimal hyperparameter configuration,

we tune the hyperparameters (λ_1, λ_2, and λ_3) via the backpropagation of the loss function, which serves to validate both the effectiveness and stability of the optimized hyperparameter combination. Within a predefined hyperparameter search range, we perform repeated validation experiments on distinct combinations of λ_1, λ_2, λ_3; the optimal combination ultimately selected achieves a dual balance between minimizing the target loss function and optimizing training efficiency. Balancing the target loss function ensures the model simultaneously focuses on the two core tasks of feature extraction and clustering-based anomaly detection, thereby avoiding performance bias induced by training dominated by a single loss term and guaranteeing the collaborative optimization of these two interrelated tasks. Optimizing training efficiency, in turn, effectively curbs redundant computational operations of the model, reduces invalid overhead in high-dimensional data reconstruction and clustering iterative processes, accelerates the convergence speed of the loss function, and shortens the overall inference latency. This not only enables the model to meet the stringent real-time performance requirements of automotive Ethernet networks but also clarifies the model's sensitivity to the adjustment of hyperparameters (λ_1, λ_2, and λ_3).

Table 2. Optimal Parameter Combination Table.

Parameter Name	Parameter Optimal Value
Reconstruct loss weights λ_1	0.01
Fuzzy loss weights λ_2	0.1
Entropy regularization weights λ_3	0.001

The three parameters take values in the ranges of (0–0.5], (0–1], and (0–0.01] respectively. Reconstructing weight values represents the ability to restore the original data. To prevent the reconstruction weight value from being too large, the upper limit of the reconstruction weight value is set to 0.5. Fuzzy clustering algorithms can classify data efficiently. The magnitude of the fuzzy loss weight value directly affects the density of the cluster. Based on the (0–1] range of the fuzzy loss weight λ_2, network abnormal data are classified. The discrimination and stability of clustering are adjusted according to the entropy regularization weight value (0–0.01]. If the weight value exceeds 0.01, the range of the fuzzy boundary is expanded. Specifically, focusing exclusively on the reconstruction loss weight λ_1 enhances the model's ability to recover the original data features, yet impairs its capacity to distinguish between normal and anomalous samples, resulting in degraded anomaly detection performance. Prioritizing only the fuzzy loss weight λ_2 improves the accuracy of cluster separation, but leads the model to overemphasize cluster distributions, making it prone to misclassifying normal data with similar feature patterns as anomalies. Concentrating solely on the entropy regularization loss weight λ_3 stabilizes the clustering process; however, an excessively large value expands the range of data with ambiguous decision boundaries, prolonging classification latency for such samples and failing to meet real-time operational requirements.

Figure 8 shows the effect of using Fuzzy clustering algorithm to classify low-dimensional data. In the test set data, normal and abnormal data were classified in an 8:2 ratio. In this diagram, green represents normal data, red represents abnormal data, and blue represents data with fuzzy boundaries.

Figure 9 shows the effect of low-dimensional data classification based on the K-Means clustering algorithm. K-Means clustering selects two data points as initial cluster centers, calculates the distance from each data point to the center, and classifies the data based on the distance. Among them, green represents normal data, and red represents outlier data. In Figure 9, some abnormal data were classified into the normal cluster because they were closer to the normal cluster center, resulting lead to unsatisfactory data classification results.

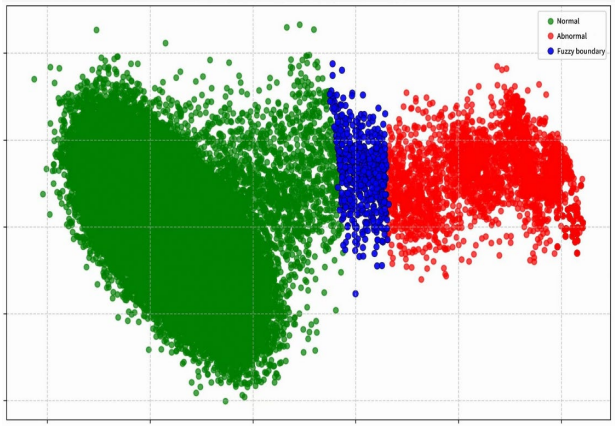


Figure 8. Effect of low-dimensional data classification based on Fuzzy clustering algorithm.

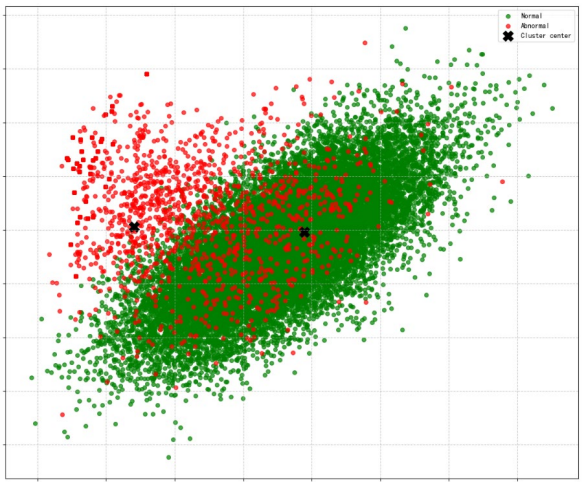


Figure 9. Effect of low-dimensional data classification based on K-Means clustering algorithm.

Figure 10 shows the effect of low-dimensional data classification based on the Optics density clustering algorithm. Optics density clustering discovers clusters based on data density. It constructs an ordered list of samples by calculating the core distance and reachability distance of the data, reflecting the density distribution structure of the data, thereby classifying the data. In Figure 10, the density difference between the green normal data and the red abnormal data is not significant, resulting lead to unsatisfactory data classification results.

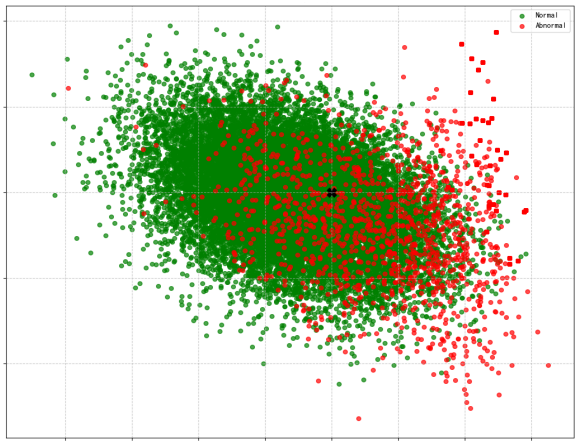


Figure 10. Effect of low-dimensional data classification based on Optics clustering algorithm.

This study experimentally verified the effectiveness and reliability of the fuzzy clustering algorithm and evaluated its performance based on four metrics: TP, TN, FP, and FN. Among them, TP (True Positive) denotes the count of normal data frames correctly classified as normal. TN (True Negative) refers to the count of abnormal data frames accurately categorized as abnormal. FP (False Positive) represents the count of abnormal data frames erroneously judged as normal. FN (False Negative) denotes the count of normal data frames mistakenly classified as abnormal. Based on the four performance metrics (TP, TN, FP, and FN), we calculate four key indicators for network anomaly detection: accuracy, precision, recall, and the F1-score (the harmonic mean of precision and recall).

Accuracy denotes the ratio of correctly identified data samples (true positives (TPs) and true negatives (TNs)) within the entire dataset. It is used to measure the reliability of network anomaly detection results, as shown in Formula (17).

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{17}$$

Precision refers to the percentage of correctly predicted data within the set of categorized normal data samples, as expressed by the following Formula (18).

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{18}$$

Recall denotes the proportion of all normal data points correctly categorized, as given by Formula (19). Elevated recall corresponds to a reduced likelihood that a normal data frame is misclassified as an abnormal one.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{19}$$

The F1-score corresponds to the harmonic mean of precision and recall values, as given by Equation (20). It is suitable for situations where the ratio of normal data to abnormal data is unbalanced.

$$\text{F1_score} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{20}$$

Centered on the fuzzy clustering algorithm proposed in this study, an exhaustive performance comparison and analysis were performed against four classic and mainstream comparative algorithms: the traditional K-Means clustering algorithm, the OPTICS density clustering algorithm, the LOF algorithm, and the Isolation Forest algorithm. Figure 11 illustrates the network anomaly detection outcomes of all five algorithms, and Table 3 tabulates their respective network anomaly detection performance metrics.

As shown in Table 3, all four evaluation metrics (Accuracy, Precision, Recall, and F1_score) derived from the Fuzzy clustering algorithm exceed 95%; specifically, the precision rate reaches 99.7%. The four evaluation indicators based on the K-means clustering algorithm are between 87% and 92%, and the four evaluation indicators based on the Optics density clustering algorithm are between 80% and 85%. Experimental results show that the Fuzzy clustering algorithm proposed in this study improves the network anomaly detection rate by 6.4% and 14.5% respectively compared with the traditional K-means clustering algorithm and Optics density clustering algorithm and reduces the computation time by 49.1% and 62.7% respectively. In comparison with the state-of-the-art LOF and Isolation Forest algorithms, the proposed method achieves 11.2% and 16.4% improvements in network anomaly detection accuracy, with respective reductions of 80.2% and 41.5% in computational time. Fuzzy clustering algorithms have significantly improved the ef-

ficiency of network anomaly detection, providing a solid theoretical foundation for the cybersecurity of automotive Ethernet technologies.

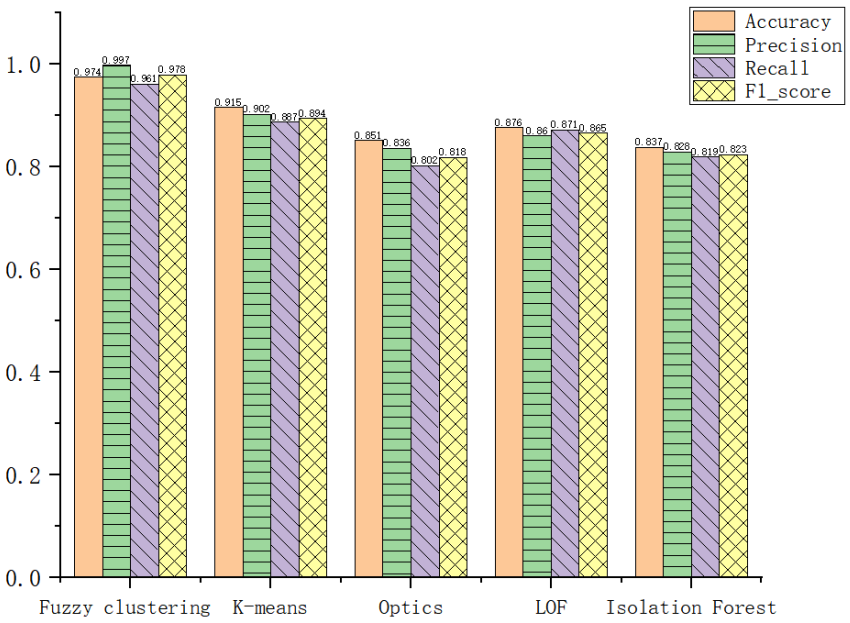


Figure 11. Network anomaly detection results.

Table 3. Network Anomaly Detection Performance Table.

Algorithm	Accuracy	Precision	Recall	F1_Score	Times
Fuzzy clustering Algorithm	0.974	0.997	0.961	0.978	113.53 s
K-means clustering Algorithm	0.915	0.902	0.887	0.894	223.25 s
Optics clustering Algorithm	0.851	0.836	0.802	0.818	304.30 s
LOF Algorithm	0.876	0.860	0.871	0.865	574.12 s
Isolation Forest Algorithm	0.837	0.828	0.819	0.823	194.22 s

5. Conclusions

The present study addresses the problem of network anomaly detection in In-Vehicle Ethernet and advances a novel detection method leveraging Fuzzy clustering algorithms. In complex vehicular network environments, Fuzzy clustering algorithms have significant advantages in network anomaly detection performance and stability, accurately identifying abnormal network attack behaviors with a detection accuracy of 97.4%. Furthermore, the Fuzzy clustering algorithm exhibits a reduced detection latency for network anomaly detection compared with other algorithms, meeting the real-time performance demands of In-Vehicle Ethernet communication networks. Future research needs to combine network anomaly detection technology with proactive prevention and fault-tolerant control technologies to construct a comprehensive protection system, thereby comprehensively improving the security and reliability of In-Vehicle Ethernet.

Author Contributions: Conceptualization, S.L.; methodology, S.L.; software, S.L.; validation, S.L.; formal analysis, S.L.; data curation, S.L.; writing—original draft preparation, S.L.; writing—review and editing, S.L.; investigation, Y.J.; resources, Y.J.; visualization, K.Z.; supervision, Y.W., 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 work was supported by the Jilin Province Natural Science Foundation, Grant Nos. YDZJ202501ZYTS641 and YDZJ202301ZYTS409, and by the National Natural Science Foundation of China, Grant Nos. 62161049 and 62201492.

Data Availability Statement: Data supporting this study's key findings is not publicly available for confidentiality reasons, and data access requests may be directed to the corresponding author.

Conflicts of Interest: The authors declare no competing interests to disclose.

Abbreviations

The abbreviations employed in this manuscript are as follows:

ICV	Intelligent Connected Vehicle
IVNs	In-Vehicle Networks
ECUs	Electronic Control Units
ADAS	Advanced Driver Assistance Systems
AVTP	IEEE 1722 Audio Video Transport Protocol
ADSs	Anomaly Detection Systems
MAC	Message Authentication Code
AV	Audio and Video
ID	Identifier
DoS	Denial of Service
LIN	Local Interconnect Network
CAN	Controller Area Network
RSD	Randomly Selected Dataset
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative

References

1. Almutairi, A.; Al Asmari, A.F.; Alanazi, F. Deep learning based predictive models for real time accident prevention in autonomous vehicle networks. *Sci. Rep.* **2025**, *15*, 20844. [CrossRef]
2. Dash, N.; Chakravarty, S.; Rath, A.K. An optimized LSTM-based deep learning model for anomaly network intrusion detection. *Sci. Rep.* **2025**, *15*, 1554. [CrossRef]
3. Min, J.; Park, Y. Fault recovery of 10BASE-T1S automotive ethernet with bus/ring hybrid topology. *J. Commun. Netw.* **2025**, *27*, 1–9. [CrossRef]
4. Toghuj, W.; Turab, N. Automotive Ethernet architecture and security: Challenges and technologies. *Int. J. Electr. Comput. Eng.* **2023**, *13*, 5209–5221. [CrossRef]
5. Douss, A.B.C.; 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]
6. Quadar, N.; Chehri, A.; Debaque, B. Intrusion detection systems in automotive ethernet networks: Challenges, opportunities and future research trends. *IEEE Internet Things Mag.* **2024**, *7*, 62–68. [CrossRef]
7. Lampe, B.; Meng, W. Intrusion detection in the automotive domain: A comprehensive review. *IEEE Commun. Surv. Tutor.* **2023**, *25*, 2356–2426. [CrossRef]
8. Rajapaksha, S.; Kalutarage, H.; Al-Kadri, M.O. Ai-based intrusion detection systems for in-vehicle networks: A survey. *ACM Comput. Surv.* **2023**, *55*, 1–40. [CrossRef]
9. Zhang, J.; Gong, B.; Waqas, M. Many-objective optimization based intrusion detection for in-vehicle network security. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 15051–15065. [CrossRef]
10. 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]
11. Lo Bello, L.; Patti, G.; Leonardi, L. A perspective on ethernet in automotive communications—Current status and future trends. *Appl. Sci.* **2023**, *13*, 1278. [CrossRef]
12. Khan, M.A. Intelligent environment enabling autonomous driving. *IEEE Access* **2021**, *9*, 32997–33017. [CrossRef]
13. da Luz, L.F.M.; de Araujo-Filho, P.F.; Campelo, D.R. Multi-stage deep learning-based intrusion detection system for automotive Ethernet networks. *Ad Hoc Netw.* **2024**, *162*, 103548. [CrossRef]

14. Muslam, M.M.A. Enhancing security in vehicle-to-vehicle communication: A comprehensive review of protocols and techniques. *Vehicles* **2024**, *6*, 450–467. [CrossRef]
15. Liu, W.; Hua, M.; Deng, Z.; Meng, Z.; Huang, Y.; Hu, C.; Song, S.; Gao, L.; Liu, C.; Shuai, B.; et al. A systematic survey of control techniques and applications in connected and automated vehicles. *IEEE Internet Things* **2023**, *10*, 21892–21916. [CrossRef]
16. Deng, L.; Xie, G.; Liu, H. A survey of real-time ethernet modeling and design methodologies: From AVB to TSN. *ACM Comput. Surv. (CSUR)* **2022**, *55*, 1–36. [CrossRef]
17. 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]
18. Almehdhar, M.; Albaseer, A.; Khan, M.A. Deep learning in the fast lane: A survey on advanced intrusion detection systems for intelligent vehicle networks. *IEEE Open J. Veh. Technol.* **2024**, *5*, 869–906. [CrossRef]
19. Dong, B.; Zhao, X.; Fan, Z. Research on lightweight encryption methods for in-vehicle ethernet in intelligent automobiles. In Proceedings of the 2024 6th International Conference on Internet of Things, Automation and Artificial Intelligence (IoTAAI), Guangzhou, China, 26–28 July 2024; pp. 521–526.
20. Rumez, M.; Grimm, D.; Kriesten, R. An overview of automotive service-oriented architectures and implications for security countermeasures. *IEEE Access* **2020**, *8*, 221852–221870. [CrossRef]
21. Jeong, S.; Kim, H.K.; Han, M.L. Aero: Automotive ethernet real-time observer for anomaly detection in in-vehicle networks. *IEEE Trans. Ind. Inform.* **2023**, *20*, 4651–4662. [CrossRef]
22. Allen, E.; Bowden, Z.; Ransbottom, J.S. The advantages of distributed TCAM firewalls in automotive real-Time switched zonal networks. In Proceedings of the ISOC Symposium on Vehicle Security and Privacy (VehicleSec), San Diego, CA, USA, 26 February 2024; pp. 1–10.
23. Klement, F.; Pöhls, H.C.; Katzenbeisser, S. Man-in-the-OBD: A modular, protocol agnostic firewall for automotive dongles to enhance privacy and security. In *International Workshop on Attacks and Defenses for Internet-of-Things*; Springer: Cham, Switzerland, 2022; pp. 143–164.
24. Chen, J.; Zuo, Q.; Jin, W. Study of network security based on key management system for in-vehicle ethernet. *Electronics* **2024**, *13*, 2524. [CrossRef]
25. 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]
26. Li, J.; Zhou, H.; Wu, S.; Luo, X.; Wang, T.; Zhan, X.; Ma, X. FOAP: Fine-Grained Open-World android app fingerprinting. In Proceedings of the 31st USENIX Security Symposium (USENIX Security 22), Boston, MA, USA, 10–12 August 2022; pp. 1579–1596.
27. Ni, T.; Lan, G.; Wang, J.; Zhao, Q.; Xu, W. Eavesdropping mobile app activity via Radio-Frequency energy harvesting. In Proceedings of the 32nd USENIX Security Symposium (USENIX Security 23), Anaheim, CA, USA, 9–11 August 2023; pp. 3511–3528.
28. Wang, Y.; Wu, Y.; Xu, Y. Research on network intrusion detection based on weighted histogram algorithm for in-vehicle ethernet. *Sensors* **2025**, *25*, 3541. [CrossRef]
29. Xu, R.; Wu, G.; Wang, W. Applying self-supervised learning to network intrusion detection for network flows with graph neural network. *Comput. Netw.* **2024**, *248*, 110495. [CrossRef]
30. Hoang, T.N.; Kim, D. Detecting in-vehicle intrusion via semi-supervised learning-based convolutional adversarial autoencoders. *Veh. Commun.* **2022**, *38*, 100520. [CrossRef]
31. 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]

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.