ISSN 1330-3651 (Print), ISSN 1848-6339 (Online)
https://doi.org/10.17559/TV-20210414030120
Preliminary communication

# Anomaly Detection Based on Multiple Streams Clustering for Train Real-Time Ethernet

Jing LIU*, Yunjuan PENG, Dalin ZHANG

**Abstract:** With the increasing traffic of train communication network (TCN), real-time Ethernet becomes the development trend. However, Train Control and Management System (TCMS) is inevitably faced with more security threats than before because of the openness of Ethernet communication protocol. It is necessary to introduce effective security mechanism into TCN. Therefore, we propose a train real-time Ethernet anomaly detection system (TREADS). TREADS introduces a multiple streams clustering algorithm to realize anomaly detection, which considers the correlation between the data dimensions and adopts the decay window to pay more attention to the recent data. In the experiment, the reliability of TREADS is tested based on the TRDP data set collected from the real network environment, and the models of anomaly detection algorithms are established for evaluation. Experimental results show that TREADS can provide a high reliability guarantee, besides, the algorithm can detect and analyze network anomalies more efficiently and accurately.

**Keywords:** anomaly detection; decay window; multiple streams; real-time Ethernet

## 1 INTRODUCTION

TCN technology combines Wired Train Bus (WTB) and Multifunction Vehicle Bus (MVB) is widely used in the traditional train communication network, which has the characteristics of high real-time and high reliability when the transmission message is less. However, with the increasing complexity of train network structure and demand of passengers for streaming files, the data transmission volume of the network is also growing rapidly. The bandwidth of fieldbus technology represented by WTB and MVB is only 1.5 Mbps, which cannot meet the demand. At present, real-time Ethernet is applied to TCN, which has the advantages of high transmission rate, low cost, good compatibility, flexible networking and so on, and is defined as a new generation of TCN [1]. With the deepening of the research on train Ethernet communication technology, TCN composed entirely of Ethernet will become the development trend. IP protocol in the network layer of Ethernet protocol stack is unreliable. When the network traffic is large, its conflict resolution will lead to the uncertainty of communication [2]. However, the transmission of key control information in TCMs must achieve high real-time and reliability. To make up for this problem, the Ethernet standard IEC61375-2-3 issued by International Electro Technical Commission (IEC) stipulates that TRDP should be applied to the real-time Ethernet of rail transit. TRDP is based on the IP network protocol and located between the application layer and transport layer of Ethernet protocol stack, which realizes the information exchange and allows all devices in the same network to communicate directly. Fig. 1 describes the communication process of TRDP. TRDP can use TCP or UDP as transport layer protocols for network data communication. Besides, it defines the communication mechanism of real-time periodic process data and real-time non-periodic message data. Through the transmission of process data and message data, the real-time and reliability of the train Ethernet in the large amount of data interaction is guaranteed.

Although the introduction of real-time Ethernet technology makes TCMS more compatible and facilitates the integration and interconnection between systems, the openness of its protocol makes TCMS more vulnerable to threats of information security [3, 4], which seriously affects the normal operation of trains. In 2003, the train signal system in Florida was shut down by the "SOBIG" virus, causing some trains to be delayed [5]; in 2008, the subway system of a city in Poland was attacked, the attacker used a TV remote control to change the track switch, resulting in the derailment of four carriages [6]; in 2012, a subway in Shenzhen, China, was brought to an emergency halt after its signal system was disrupted by passengers' WiFi signals.

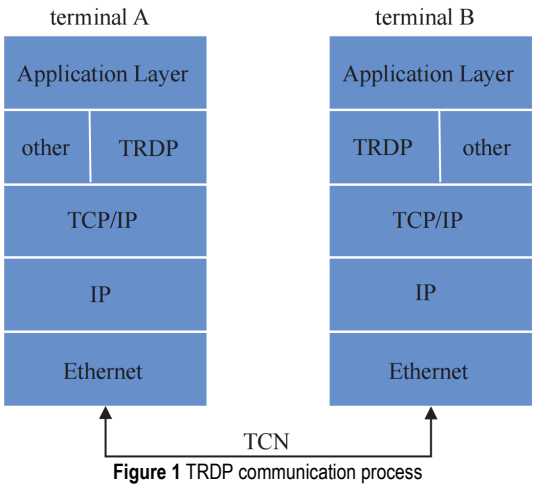


**Figure 1** TRDP communication process

All these incidents indicate that the security of TCMS needs more attention. In order to ensure the security of the TCMS and provide quick early warning of abnormal situations in the network, it is necessary to introduce effective security mechanism into TCN. As an efficient and active defense mechanism, anomaly detection system can monitor the abnormal situation in the network data stream in real time and make a report to guarantee the normal operation of the train. Therefore, it is of great practical significance to study TREADS.

At present, most of the commonly used train communication protocols use open standards, and TRDP is also open in the form of open source through "TCNOpen", which inevitably provides hackers with the opportunity to learn and invade, making the communication protocol have a certain vulnerability. Although the application research based on TRDP is increasing day by day, there is still a

large gap in the study of its network status monitoring and health assessment [7]. Therefore, this paper focuses on the anomaly detection method of TRDP.

TRDP transmits real-time periodic process data and real-time non-periodic message data, which are ordered sequences of continuous arrivals, that is, data stream. A large number of sensors is often distributed on the train, and the parameters collected by the sensors reflect the changes of physical parameters in the system. Many anomalies do not exist in isolation, but reflect in multiple parameters at the same time [8]. There are many methods for anomaly detection, including Isolation Forest (iForest) [9] based on the concept of isolation, One-Class Support Vector Machine (OCSVM) based on particle swarm optimization algorithm [10], anomaly detection model based on particle swarm optimization-support vector machines (PSO-SVM) and genetic algorithm-support vector machines (GA-SVM) optimization algorithms [4] and so on. Most of these methods are based on static data or single data stream, ignoring the persistence, quantity and correlation of data streams. Moreover, they failed to take into account that recent data and historical data should not receive equal attention. Based on the above considerations, this paper adopts multiple streams clustering algorithm to detect abnormal data transmitted in TRDP of real-time Ethernet. In addition, the training data of clustering algorithm does not need to have a classification label, and it can update the normal pattern through the continuous arrival of new data, which is self-adaptive.

Based on the above analysis, this paper constructs TREADS based on multiple stream clustering algorithm to achieve efficient and accurate network anomaly detection and analysis.

Compared with the existing work, the contribution of this paper can be summarized as follows:

- An adaptive multiple streams clustering algorithm considering the correlation of each data dimension and the importance of recent data is used to detect anomalies in real-time Ethernet.
- A complete real-time Ethernet anomaly detection system is constructed, which includes data acquisition layer, data transport layer, network performance monitoring module, data storage layer and data application layer. The system can collect and store TRDP data in a distributed way, and detect, analyze and display the data based on anomaly detection algorithm.
- iForest, Clustream, K-Means and DBSCAN models are built for comparison and evaluation with the multiple streams anomaly detection model adopted in this paper.

The following parts of this paper are organized as follows: Chapter 2 provides an overview of existing anomaly detection methods; Chapter 3 introduces the architecture of TREADS and the functions of each layer; Chapter 4 introduces the multiple stream clustering algorithm in detail; Chapter 5 describes the experiments carried out in this paper and discusses and analyzes the results; Chapter 6 summarizes the work of this paper.

## 2 RELATED WORK

Intrusion detection is an active and efficient security protection technology, which can identify and respond to abnormal situations in the network. Intrusion detection system refers to the system that monitors and discovers the abnormal use of network or computer, and takes corresponding protective measures. According to detection and analysis methods, it can be divided into abnormal detection system and misuse detection system [3]. The intrusion detection system discussed in this chapter mainly refers to the anomaly detection system. Anomaly detection system summarizes the rules of normal behavior to form prior knowledge, and then judges whether the data is abnormal by comparing whether the difference between the data and the normal behavior pattern exceeds the preset threshold.

Intrusion detection technology has been applied in industrial control scenarios for a long time. In 1986, Anderson established a feasible intrusion detection system for the first time and used innovative statistical algorithms for anomaly detection [11]. Intrusion detection technology has been maintaining a sustained and stable development. It has been widely used in different fields with constantly integrating intelligent detection technology. Due to the low efficiency of traditional time-oriented maintenance, a data-driven method combining principal component analysis and partial least squares method is proposed to identify and predict the state pattern of asynchronous generator, the method using principal component analysis to find first characterization of the most important factor in the equipment situation, then using partial least squares forecast system status and detect abnormal behavior [12]; in order to solve the security problem of field control system in industrial process automation, a multi model anomaly detection method was proposed in literature [13], and the corresponding intelligent detection algorithm was designed; in addition, to overcome the shortcomings of anomaly detection, the literature also designed a classifier based on Intelligent Markov model to distinguish attacks and faults; in order to detect attacks on critical infrastructure such as water treatment plants, a distributed attack detection method was proposed. This method detects attacks in real time by identifying anomalies in the physical process behavior of plants [14]; an integrated model of multiple classifiers was proposed to detect new types of distributed denial of service attacks, in which each classifier can target specific aspects or types of intrusion to provide a more powerful defense mechanism [15]. These methods have achieved good results in their respective fields. However, with the increase of attack intensity and the diversification of attack types, they are faced with problems such as the decrease of detection speed and the increase of false detection rate.

In recent years, the in-depth research and application of intelligent detection technology promote the development of intrusion detection, greatly improving the detection efficiency of the system. At present, many machine learning algorithms are used in intrusion detection, mainly divided into supervised learning and unsupervised learning. Literature [16] constructed an application based on machine learning to enhance domain knowledge, and used genetic algorithms and decision trees to automatically generate network connection classification rules; iForest was proposed by Zhou Zhihua from Nanjing University, it is an anomaly detection algorithm commonly used in the industry, which detects anomalies completely based on the concept of isolation [9]; Robust Random Cut Forest

improves iForest based on the characteristics of streaming data and it is applied to AWAS [17]; considering that the network communication of industrial control system and supervisory control and data acquisition system is a deterministic behavior, literature [18] designed a fuzzy c-means algorithm and fuzzy inference system clustering method, combining with quantitative events for a given network data condition (state) level of attack to complement more complex intrusion detection system to improve the efficiency and reduce the false positive rate; an intelligent intrusion detection system based on artificial neural network was constructed, which can identify shell code patterns in network traffic and improve the performance of feature-based detection method [19]; OCSVM is also a commonly used anomaly detection algorithm; in literature [10], the optimal support vector machine (SVM) was used to establish a normal communication behavior model, and the particle swarm optimization algorithm was designed to optimize the optimal SVM model parameters; literature [20] proposed an intelligent intrusion detection system based on deep neural network, and trained the detection model on KDDCup99 data set, and obtained high accuracy; an intrusion detection system based on anomaly detection and attack classification was constructed for the security protection of train real-time Ethernet communication. The system used PSO-SVM and GA-SVM algorithms to optimize the kernel function parameters of SVM, and established an iterative dichotomizer3 and classification and regression tree attack classification model [4]. Although these methods combined with machine learning have achieved relatively satisfactory results, they are still subject to certain limitations, such as convergence speed and parameter selection. In addition, they do not take into account the characteristics of data streams and the correlation between multiple data streams generated by multiple devices.

After analysis, we find that the above research methods mainly have one or more of the following problems:

(1) Modeling based on static data ignores the continuous and infinite characteristics of data stream.

(2) Historical and recent data are given same weight, but they should have different effects on the results.

(3) Anomalies are not isolation, the potential correlation between multiple data streams is not considered.

(4) Long training time, slow convergence speed, easy to fall into local optimum.

(5) In the actual application of anomaly detection, most of the data are unmarked, so supervised learning is not suitable for detecting new types of anomalies.

Based on the above analysis, we establish TREADS based on multiple streams clustering algorithm and evaluate its performance based on real-time collected data. The system can achieve efficient collection and simulation of TRDP data packets, and realize distributed storage of data. The anomaly detection algorithm is used to model high-dimensional data, and the decay technology is used to reduce the impact of historical data, so as to realize fast and accurate early warning in abnormal situations.

## 3 ANOMALY DETECTION SYSTEM FOR TRAIN REAL-TIME ETHERNET

As exhibited in Fig. 2, TREADS designed in this paper is mainly divided into five parts: data acquisition layer, data transport layer, network performance monitoring module, data storage layer and data application layer.

In the implementation process of the whole system, the data acquisition layer firstly obtains the length and sending cycles of TRDP packets under different services, and then selects the network card for data packet collection. The collected data packets and the corresponding basic information are transmitted to the network performance monitoring module through the data transport layer, and the data are processed by the Flink platform in the module. Finally, the processing results are stored on the Hadoop platform, and the anomaly detection algorithm is used to learn the data features. The model has the ability to adapt to the new data. The detection results can be obtained after entering the data into the model, which can effectively improve the processing speed. Finally, the system generates a report based on the exception data and displays it on a large screen. In this chapter, we will introduce the functions and workflow of each layer.

### 3.1 Data Acquisition Layer

Data acquisition layer is responsible for collecting and parsing TRDP packets. This layer collects data packets through tshark, a subproject of Wireshark that is an open source network data capture tool. Wireshark is a convenient, widely used, cross-platform and easily extensible network protocol analysis software, it uses WinPcap to directly exchange data packets with network card under Windows [21], and supports TRDP packet analysis. The system encapsulates tshark as a client program, and then deploys the acquisition client on several different network ports to capture packets in the real network.

### 3.2 Data Transport Layer

The data transport layer uses the Kafka message queue as the distributed real-time data transport channel and sends the network data collected by the upper layer to the Flink stream processing platform. Kafka has good throughput, built-in partition, replication and fault tolerance, which is a good solution for large-scale network data flow processing applications in this paper.

### 3.3 Network Performance Monitoring Module

Network performance monitoring module is one of the core modules of the system. This module uses the Flink real-time streaming data processing platform to calculate the network packets and obtain the out-of-order data and the number of packet loss in real time. The module receives the network data stream transported from the upper layer, and then uses Flink to summarize multiple performance statistics indicators of the data stream, and obtains the final statistical results to judge the out-of-order and packet loss situation of the data stream.

### 3.4 Data Storage Layer

Data storage layer is one of the core modules of the system. This layer stores the processed network data in the InfluxDB and MySQL database based on Hadoop platform to facilitate data access. InfluxDB is a kind of time-series database, which is often used in monitoring data statistics and can query and store time-series data with high performance. The Ethernet communication data in this paper is a kind of time-series data stream.

### 3.5 Data Application Layer

Data application layer includes historical data query, anomaly detection, large screen display and report generation module. Among them, anomaly detection is the core module, and multi-data stream clustering algorithm is used to realize network data detection here. We will describe this algorithm in detail in Chapter 4. The historical data query module first acquires the data stored in the data storage layer, and then detects the abnormal situation in the network through the anomaly detection module. Then, the report generation module generates a report according to the abnormal data, and the large screen display module presents the detection results.

## 4 MULTIPLE STREAMS CLUSTERING ALGORITHM

Ethernet communication data is a series of continuous and ordered sequences, that is, data stream. Data stream is characterized by infinite data volume, fast arrival speed, non-reproducible and changing with time, etc.

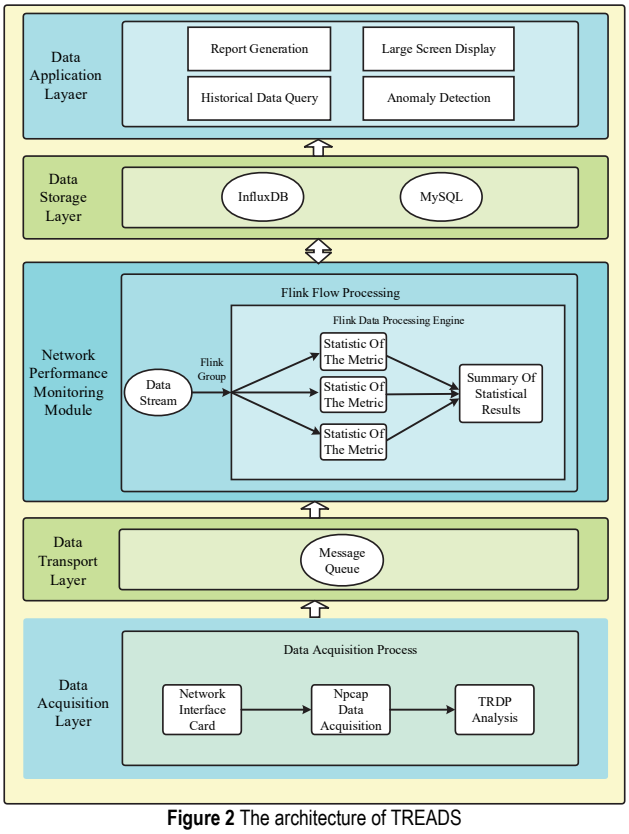


**Figure 2** The architecture of TREADS

It is impossible to store all the data. If the arrived data are not processed in time, the old data will be covered, resulting in the loss of valuable information [8]. Anomaly detection of Ethernet communication data is essentially the anomaly detection of streaming data. Most train workers pay more attention to the abnormal situation of data in the recent period of time, so it is necessary to take into account the time series range of data stream processing when building the model. In addition, Ethernet communication data stream continues to arrive over time, cannot be obtained at the time of detection, and abnormal patterns cannot be fully established from offline data. Data stream can be divided into single multiple data stream according to the level of dimensions. The key single data stream reflects the important state information of a certain equipment on the train, while the abnormality of multiple data streams is the result of the comprehensive action of the data of all dimensions, which reflects the correlation among the equipment. Considering the above features of Ethernet communication data, the anomaly detection algorithm in this paper focuses on three points: (1) it is self-adaptive to new data streams continuously arriving; (2) can establish the correlation between multi-dimensional data streams; (3) can give different weights to historical data and recent data, highlighting the importance of recent data.

D-Stream [22] is a clustering algorithm based on grid and density. It adopts an unsupervised learning method and can adaptively detect novel abnormal patterns with the arrival of new data streams. D-Stream follows the two-stage processing framework of Clustream and divides the process of clustering analysis into two parts: online and offline. The online part is responsible for mapping the received new data elements to the corresponding density grid in the multi-dimensional space, without calculating the distance or weight, which is more efficient than the algorithm without using grid. The offline part is responsible for calculating the density of the grid and clustering the grid based on the density. D-Stream uses density decay technology to capture the dynamic change of data stream. With the passage of time, the influence of historical data on clustering results continuously decays, which is more flexible than the algorithm that requires preset K value and time window size. In addition, it can identify clusters of any shape, and has good scalability for massive high-dimensional stream data. The algorithm speed will not slow down with the increase of data volume. It has great advantages in analyzing and processing a large amount of communication data in Ethernet. Therefore, we introduce D-Stream algorithm into TREADS to detect and analyze the abnormal situation in TRDP data stream.

### 4.1 Correlation of Multiple Data Streams

Considering a single data stream, sequences consist of a series of continuous and ordered binary < time stamp, value > are called time series data streams. Considering multiple data streams, sequences consist of a series of triple < streamID, time stamp, value > are called multiple time series data streams, where, streamID is the number of data streams from different data sources. Suppose that $X_1 = \{x_{11}, x_{12}, \ldots, x_{1d}\}$ and $X_2 = \{x_{21}, x_{22}, \ldots, x_{2d}\}$ are two data streams arriving at a certain time, $d$ represents the feature dimension of a single data stream. According to the feature of multiple time series data streams, the local correlation degree between data streams is:

$$\mathrm{Corr}\left(X_1, X_2\right) = \frac{L_{12}}{\sqrt{L_1 \times L_2}} \tag{1}$$

where:

$$L_1 = \sum_{i=1}^{d}\left(x_{1i} - \frac{1}{d}\sum_{i=1}^{d} x_{1i}\right)^2 \tag{2}$$

$$L_{12} = \sum_{i=1}^{d}\left(x_{1i} - \frac{1}{d}\sum_{i=1}^{d} x_{1i}\right)^2\left(x_{2i} - \frac{1}{d}\sum_{i=1}^{d} x_{2i}\right)^2 \tag{3}$$

### 4.2 The Overall Architecture of D-Stream

D-Stream algorithm implements an incremental data processing mode, that is, the data stream in the algorithm grows continuously over time [23]. The main idea of the clustering process is as follows: firstly, the density grid and grid group list are initialized, and then the data is read in and processed in a circular way. The data processing is divided into online and offline parts.

The online part is the process of updating. When a data stream arrives, for each time step, the online component constantly reads new data elements, maps the multidimensional data to the corresponding discrete density grid in the multidimensional space, and then updates the feature vectors of the density grid. The offline part is the processing of clustering. The offline component dynamically adjusts the cluster at each interval time step [24]. Fig. 3 shows the architecture of D-Stream algorithm. Due to the features of large and unlimited, it is impossible to store all data streams. Therefore, D-Stream divides the multidimensional data space into multiple density grids, and forms clusters on this basis.

The whole process of D-Stream algorithm is shown in Fig. 4. The algorithm firstly defines a set of discrete time steps, in which the steps are labeled by {0, 1, 2, ..., *n*, ...}. *t* represents the current time step. Suppose the current arrival multiple streams $X^t = \{x_1, x_2, x_3 \ldots, x_i\}$, each data record $X_i = \{x_{i1}, x_{i2}, \ldots, x_{id}\}$, *d* is the feature dimension of the data record. When t is 0, create an empty hash grid, read in a new $X_i$ at each time step, determine the mapping density grid *g*, if *g* does not exist in the grid, then insert it into grid and update the feature vector of grid *g*. When *t* first reaches the interval gap (integer parameter), the algorithm will call the function to initialize the grid and obtain an initial cluster.

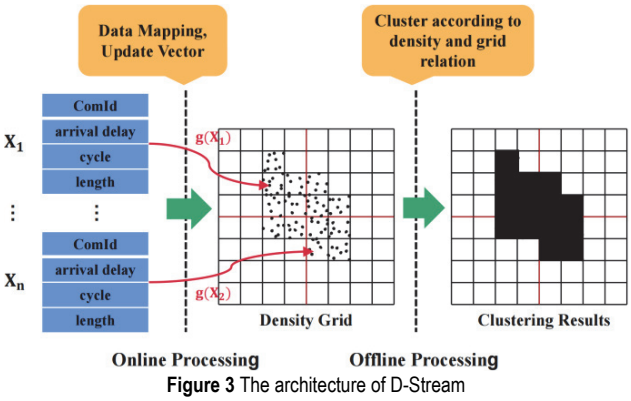


**Figure 3** The architecture of D-Stream

Thereafter, the grid will be detected at every time step of the gap to remove sporadic grids and adjust the cluster. Whether the grid is sporadic or not is determined by the interval of grid density, the interval is related to $D_m$ and $D_l$:

$$D_m = \frac{C_m}{N(1-\lambda)} \tag{4}$$

$$D_l = \frac{C_l}{N(1-\lambda)} \tag{5}$$

where, *N* represents the total number of grids in the data space, $C_m$ and $C_l$ are two constants, $C_m > 1$, $0 < C_l < 1$. In this interval, the grid is divided into three types: dense, sporadic and transitional.

**Algorithm 1** D-Stream

```
t = 0
initialize an empty hash table grid
while the data streams do not end do
    read X_i = (x_i1, x_i2, ..., x_id)
    determine g that contains X_i
    if g is not in grid then
        insert g into grid
    end if
    update the feature vector of g
    if t == gap then
        call the cluster initialization function
    end if
    if t % gap == 0 then
        detect and remove sporadic grids from grid
        call the cluster adjustment function
    end if
    t = t + 1
end while
```

**Figure 4** The overall process of D-Stream

### 4.3 Density Decay Technology

Because the data stream cannot be stored completely, the data stream processing method is particularly important in an efficient clustering algorithm. The commonly used data stream processing methods include sliding window, decaying window and tilted time frame. D-Stream uses a decay window technology. Decay window adopts the idea of increment, when data arrives, it does not consider its departure, but multiplies it by a weight, which is a time related function. The longer the data is retained in space, the smaller the weight will be. Therefore, on the whole, the impact of historical data on the current situation will be smaller and smaller [24].

The delay function used by D-Stream is as follows:

$$f(t) = \lambda^{t-t'}\left(0 < \lambda < 1\right) \tag{6}$$

where, *t* represents the time when the current data arrives in the grid, *t'* represents the arrival time of the last data in this grid, $\lambda$ represents decay factor. The larger the $\lambda$ is, the faster the data decay rate is, and the smaller the impact of historical data on the current clustering results is.

## 5 EXPERIMENTS

### 5.1 Train Real-Time Ethernet Data Description

TRDP [8] is the application layer protocol of railway network transmission protocol. According to the protocol type, the most basic data features of TRDP include service code ComId, packet arrival delay, sequence number, packet cycle, packet length and so on. Each type of service corresponds to its corresponding packet sending cycle, packet length and other information. In this paper, we first carry out feature matching on the sequence number, and classify the data packets by judging the continuity of the sequence number and whether it is lost, and then store the data into our Hadoop platform. We use the anomaly detection algorithm to mine the service data. In the process of data mining, because of the contingency of sequence number, it is not used as a learning feature. That is, we select ComId, delay, cycle and packet length as the learning features of the model. Tab. 1 shows the basic features of TRDP packets.

**Table 1** Basic features of TRDP network packet

| ID | Feature | Description |
|---|---|---|
| $X_1$ | ComId | the primary key in TRDP |
| $X_2$ | packet arrival delay | time difference between this packet and the last arrived packet |
| $X_3$ | sequence number | the sequence number of the packet under this ComId |
| $X_4$ | packet cycle | the cycle of sending packets under this ComId |
| $X_5$ | packet length | the length of the packet sent under this ComId |
| $Y$ | state | anomaly situation of the packet |

### 5.2 Data Acquisition and Packet Processing Method

In order to test the performance of TREADS in the actual operation scenario, we process the data packets captured in the real network and use them as the experimental data set. In the data collection stage, we use Wireshark to capture the data packets and form packet files. After that, we divide and process the files, extract multiple data packets under the service line, and generate background traffic to simulate the real network. In this paper, we use a Wireshark-based efficient segmentation method of big data files proposed by Liu [25] et al. By using Lua plug-in, we control each piece of data, which greatly speeds up the cutting process and further improves the traffic of network packets. We split the packets according to source IP and ComId, and put the packets with different IP and services into different files. The specific packet processing method is as follows:

(1) Use Wireshark to read packets from the packet files.
(2) Get Host and ComId of the current packet and check whether the corresponding ComId has been stored in the Host hash table. If it exits, go to step 3, otherwise go to step 4.
(3) Write the current packet into a generated file.
(4) Write the current ComId into the hash table and the current packet into a file with the current Host as the file name and ComId as the subfile name.
(5) Continue to read the next packet until all packets have been read.

### 5.3 System Performance Test

TREADS is mainly used for real-time monitoring of TCN, it has high requirements for the speed of network packet capture and processing. The performance requirements of TREADS mainly includes two aspects: one is to avoid the loss of data packets caused by the data acquisition layer and the data transport layer; the other is to ensure the real-time monitoring of abnormal TRDP data packets.

First, we perform a pressure test on the data acquisition layer. In order to prevent packet loss during transmission, the acquisition device starts Wireshark to capture packets at the same time. In this paper, 60 threads are started to send 1432 bytes of data packets with 1 ms as the sending cycle, so as to simulate the train equipment sending different service data packets. Test results are shown in Fig. 5.

Next, we read the Binlog logs stored in MySQL in Network Performance Monitoring Module with the degree of parallelism 1, test the processing speed of 26,018 data in the Flink processing engine at different degree of parallelism, and perform the network performance statistics with the degree of parallelism 1, 2, 4, and 6 respectively. Test results of the processing speed of this module are shown in Fig. 5.

Finally, we calculate the data processing speed of the whole system. In the experiment, the number of data analysis and storage processes is set to 4, the global parallelism of Network Performance Monitoring Module is set to 1. The packet sending device starts 60 threads to send packets with 30 ms as the cycle, and the end time of the last data processing of each module in the whole system is counted. The data processing speed of each module is shown in Fig. 6.

It can be seen from Fig. 5 that the number of packets sent and received by the system is always the same,which indicates that with the increase of packets, the data collector of the receiving device captures all the packets sent, and the probability of the system collecting a complete number of packets is100%. As can be seen from Fig. 6, the system can process the data in 8 seconds under the single degree of parallelism.

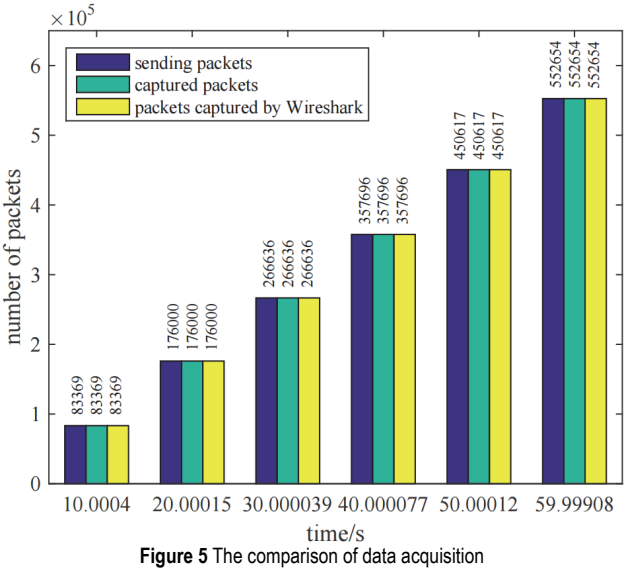


**Figure 5** The comparison of data acquisition

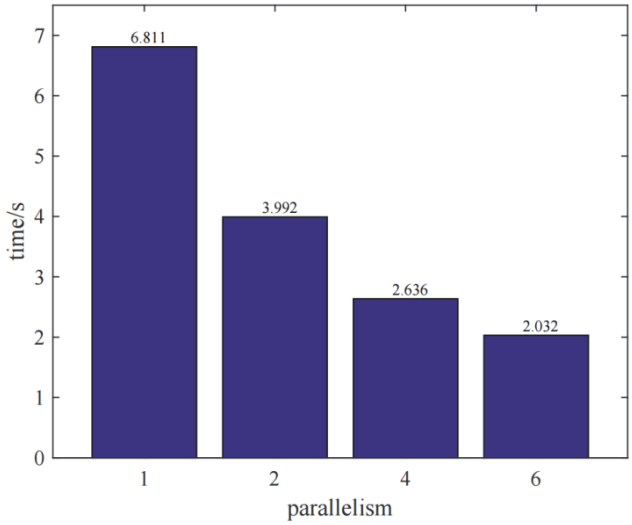


**Figure 6** The computational speed of different parallelism

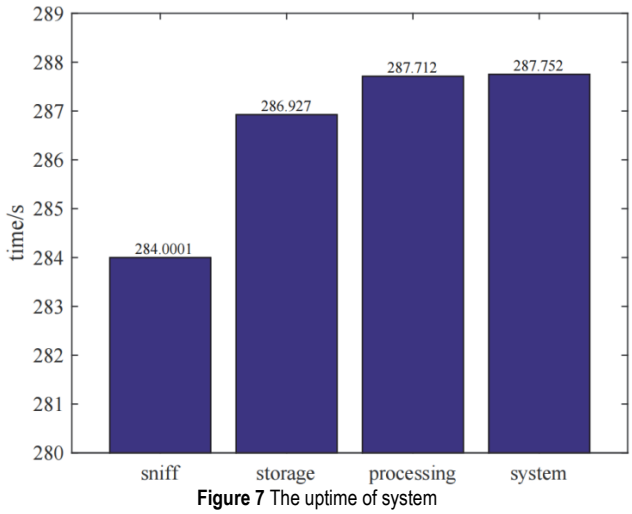


**Figure 7** The uptime of system

With the increase of the degree of parallelism, the processing speed continuously increases. In Fig. 7, the four modules run in parallel. The time for the whole system to receive all data is 287756 seconds, and the time difference from the end of data capture to the completion of processing is 3751 seconds, which is maintained at a low level.

After analyzing the above test results, we find that the system can achieve the expected results in packet capture and processing speed, the system can provide high reliability for real-time Ethernet anomaly detection of trains.

### 5.4 Algorithm Comparison

In order to evaluate the performance of D-Stream algorithm in real-time Ethernet anomaly detection of trains, we compare it with other four algorithms: iForest, CluStream, K-Means and DBSCAN. This experiment is implemented on WEKA 3.8.5 platform and MOA20.12 framework of Windows 10 system.

The experimental data set adopts the TRDP data set collected and processed in the real network environment. It contains 11770 pieces of data, among which there are 60 different kinds of services, each piece of data has 4 continuous attribute values and 1 flag character. When the value of the flag character is 0, it means that the data is normal, otherwise it is abnormal. It is worth noting that although the unsupervised algorithms are adopted in the experiment, we have to label each piece of data in the data set to compare the effect of the algorithms, that is, flag character mentioned above. In order to unify dimensional units, we normalize the data before the experiment, that is, the maximum value $x_{i\max}$ and minimum value $x_{i\min}$ of each attribute $x_i$ in the data are calculated, and then the data is compressed to the interval of (0, 1). The formula of data normalization processing is as follows [24]:

$$x_i = \frac{x_i - x_{i\min}}{x_{i\max} - x_{i\min}} \tag{7}$$

We use six metrics to evaluate the algorithms, they are Precision, Recall, Purity, SSQ, Homogeneity and Completeness. The following is a brief description of these evaluation indicators:

(1) Precision: The proportion of the samples that are actual positive in all the samples that are predicted to be positive.

(2) Recall: The proportion of the samples that are predicted to be positive in all the samples that are actually positive.

(3) Purity:

$$\text{Purity} = \frac{1}{n}\sum_{k=1}^{c}\max\left(s\right) \tag{8}$$

where, $s$ represents the number of the majority class in cluster $k$, $c$ represents the number of clusters. The larger the value, the better.

(4) SSQ: The sum of the squares of the distance between the data item and its cluster center reflects the cluster distribution. The smaller the value, the better. It is used to evaluate the clustering algorithm.

(5) Homogeneity: Each cluster only contains members of a single class, and the value range is (0.0, 1.0). The larger the value, the better. It is used to evaluate the clustering algorithm.

(6) Completeness: All members of a given class are assigned to the same cluster. The value range is (0.0, 1.0). The larger the value, the better. It is used to evaluate the clustering algorithm.

In the experiment, we set the decay factor of D-Stream as 0.92, and the parameters of iForest algorithm remain the default (subsample size: 256, tree height: 8, number of trees: 100). the epsilon of DBSCAN is 0.8, the k value of K-Means is 2, and the maximum number of microkernels of CluStream is 2. Tab. 2 shows the Precision and Recall scores of the five algorithms, and we show the highest scores in bold.

**Table 2** Scores of Precision and Recall for each algorithm

| algorithm | Precision | Recall |
|---|---|---|
| D-Stream | 1.000 | 0.95 |
| iForest | 0.998 | 0.629 |
| CluStream | 1.000 | 0.700 |
| K-Means | 1.000 | 0.620 |
| DBSCAN | 1.000 | 0.850 |

It can be seen from Tab. 2 that D-Stream algorithm performs best in Precision and Recall; the scores of two metrics of iForest are the lowest; CluStream, K-Means and DBSCAN have the highest scores on Precision, but they do

not perform well on Recall, especially K-Means has the lowest Recall score, which is only about 0.6.

We use Purity, SSQ, Homogeneity and Completeness to further evaluate the clustering algorithms. Tab. 3 shows the scores of these metrics of the four algorithms, among which the best scores are shown in bold.

**Table 3** Scores of other metrics for four clustering algorithms

| algorithm | Purity | SSQ | Homogeneity | Completeness |
|---|---|---|---|---|
| D-Stream | 1.000 | 382.97 | 0.99 | 0.52 |
| CluStream | 1.000 | 224.00 | 0.99 | 0.13 |
| K-Means | 1.000 | 243.64 | 0.99 | 0.13 |
| DBSCAN | 1.000 | 229.70 | 0.99 | 0.18 |

As can be seen from Tab. 3, D-Stream has three, CluStream has three, K-means has two indicators, DBSCAN has two highest scores. The SSQ score of D-Stream is higher than that of the other three algorithms, which indicates that the data in the cluster obtained by D-Stream in the clustering process is relatively scattered. We think that the reasons may be as follows: in the data stream environment, the density distribution of the data may change with time. The grid with higher initial density gradually decays because there is no data for a long time, and the density gradually decreases to sporadic. The state of the grid is changing, but the relevant parameters remain unchanged, which has a certain impact on the clustering effect. The scores of other three metrics of D-Stream are the highest, especially the Completeness, which indicates that the overall clustering effect of D-Stream is superior to the other three algorithms.

Based on the above analysis, D-Stream algorithm has more advantages in the detection of train real-time Ethernet anomalies than the other algorithms, and can make more accurate analysis.

## 6 CONCLUSION

In view of the security threats TCMS is facing, this paper studies train real-time Ethernet anomaly detection, and uses Hadoop platform and Flink real-time data processing platform to build an anomaly detection system. In this system, a method which can efficiently extract and segment TRDP packet files on demand is proposed to realize data acquisition simulation. The distributed stream processing mechanism is adopted to realize the network data storage by utilizing the big data platform. According to the features of TCN data stream, D-Streams, an efficient multiple stream clustering algorithm based on grid and density, is used to learn network data protocol feature library. Compared with other anomaly detection algorithms, this algorithm considers the correlation between high-dimensional data, reduces the influence of historical data on clustering results by using density decay technology, and can adapt to the data stream to improve the efficiency and accuracy of anomaly detection.

In the experimental stage, we first use Wireshark to collect and process TRDP packets in the real network environment, then we test the performance of the system, which proves that it can provide a high reliability guarantee for Ethernet anomaly detection. Finally, based on the collected data, we build the D-Streams and four benchmark models, and evaluate their detection results with Precision and other metrics. The excellent performance of D-Streams in multiple data stream anomaly detection environment is proved. In Summary, TREADS based on multiple data stream clustering algorithm can effectively improve the detection performance and provide further guarantee for the safety of the train operation.

## 7 REFERENCES


[1] Pei, Z. & Tan, X. (2014). A Performance Simulation Study of Train Communication Networks Based on Ethernet. *Journal of Southwest University of Science and Technology*, *000*(002), 66-71.

[2] Tingxu, L. (2019). *Research on Train Ethernet Network Communication Real Time Based on TRDP Protocol*. (Doctoral dissertation, Dalian Jiaotong University).

[3] Fei, L. *Design and Implementation of Train Communication Network Intrusion Detection System Based on Deep Packet Inspection*. (Doctoral dissertation, Huazhong University of Science and Technology).

[4] Duo, R., Nie, X., Yang, N., Yue, C., & Wang, Y. (2021). Anomaly Detection and Attack Classification for Train Real-Time Ethernet. *IEEE Access*, *9*, 22528-22541. https://doi.org/10.1109/ACCESS.2021.3055209

[5] Cherdantseva, Y., Burnap, P., Blyth, A., Eden, P., Jones, K., Soulsby, H., & Stoddart, K. (2016). A review of cyber security risk assessment methods for SCADA systems. *Computers & Security*, *56*, 1-27. https://doi.org/10.1016/j.cose.2015.09.009

[6] Shuai, Z. (2012). Security Status and Risk Analysis of Industrial Control System-One of the Security Risk Analysis of ics Industrial Control System. *Computer Security*, *000*(001), 15-19.

[7] Dong, Y., Wang, L., Li, H., & Li, Z. (2019). TRDP feature extraction and fault diagnosis based on SVM and SNMP. *Information Technology*, 43, *332*(07), 18-22.

[8] Pang, J. *Adaptive Anomoly Detection For Data Stream With Sequence-Based Slinding Windows Model*. (Doctoral dissertation, Harbin Institute of Technology).

[9] Liu, F. T., Ting, K. M., & Zhou, Z. H. (2012). Isolation-based anomaly detection. *Acm Transactions on Knowledge Discovery from Data*, *6*(1), 1-39. https://doi.org/10.1145/2133360.2133363

[10] Shang, W., Zeng, P., Wan, M., Li, L. & An, P. (2015). Intrusion detection algorithm based on OCSVM in industrial control system. *Security and Communication Networks*, *9*(10), 1040-1049. https://doi.org/10.1002/sec.1398

[11] Anderson, D., Frivold, T., & Vald, A. (1995). Next-generation Intrusion Detection Expert System (NIDES) A Summary.

[12] Marton, I., Sanchez, A., Carlos, S., & Martorell, S. (2013). Application of Data Driven Methods for Condition Monitoring Maintenance. *Chemical Engineering Transactions*, *33*, 301-306. https://doi.org/10.3303/CET1333051.

[13] Zhou, C., Huang, S., Xiong, N., Yang, S., Li, H., Qin, Y. & Li, X. (2015). Design and Analysis of Multimodel-Based Anomaly Intrusion Detection Systems in Industrial Process Automation. *IEEE Transactions on Systems*, Man, and Cybernetics: Systems, *45*(10), 1345-1360. https://doi.org/10.1109/TSMC.2015.2415763

[14] Adepu, S. & Mathur, A. (2018). Distributed attack detection in a water treatment plant: method and case study. *IEEE Transactions on Dependable & Secure Computing*, 1-1. https://doi.org/10.1109/TDSC.2018.2875008

[15] Das, S., Mahfouz, A. M., Venugopal, D., & Shiva, S. (2019). DDoS Intrusion Detection Through Machine Learning Ensemble. *2019 IEEE 19th International Conference on Software Quality, Reliability and Security Companion* (*QRS-C*), Sofia, Bulgaria, 2019, 471-477.

https://doi.org/10.1109/QRS-C.2019.00090
[16] Sinclair, C., Pierce, L., & Matzner, S. (2002). An application of machine learning to network intrusion detection. *Computer Security Applications Conference. IEEE.*
[17] Guha, S., Mishra, N., Roy, G. & Schrijvers, O. (2016). Robust Random Cut Forest Based Anomaly Detection on Streams. *Proceedings of The 33rd International Conference on Machine Learning, in Proceedings of Machine Learning Research* 48:2712-2721 Available from http://proceedings.mlr.press/v48/guha16.html.
[18] Jr, L. T. & Farnam, M. R. A Clustering Approach to Industrial Network Intrusion Detection.
[19] Shenfield, A., Day, D., & Ayesh, A. (2018). *Intelligent intrusion detection systems using artificial neural networks*. ICT Express, *4*(2), 95-99. https://doi.org/10.1016/j.icte.2018.04.003
[20] Jin, K., Shin, N., Jo, S. Y., & Sang, H. K. (2017). Method of intrusion detection using deep neural network. *IEEE International Conference on Big Data & Smart Computing*. IEEE. https://doi.org/10.1109/BIGCOMP.2017.7881684
[21] Hu, P. (2020). Research and Implementation of Ethernet Communication Protocol Parser of Train Control Center Based on Wireshark. *RAILWAY SIGNALLING & COMMUNICATION*(06), 48-51.
[22] Yixin, C. & Li, T. (2007). Density-based clustering for real-time stream data. *13th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '07)*. New York, NY, USA: *Association for Computing Machinery*, 133-142. https://doi.org/10.1145/1281192.1281210
[23] Yin, G. S., Yu, X., & Ning, H. (2009). Increment Clustering Algorithm Based On Grid. *Application Research of Computers* (06), 2038-2040.
[24] An, P. (2012). *Study of Cluster Algorithm of Data Stream*. (Doctoral dissertation, Harbin Institute of Technology).
[25] Liu, Z., Zhao, Y., Wan, S. (2020). An efficient segmentation method for large data packet files based on Wireshark.*China New Telecommunications*, *22*(19), 84-87.

**Contact information:**

**Jing LIU,** Engineer
(Corresponding author)
Beijing Mass Transit Railway Operation Corporation Limited,
Beijing, China
E-mail: 13811276617@139.com

**Yunjuan PENG**, MSc.
School of Software Engineering,
Beijing Jiaotong University,
Beijing, China
E-mail: mualuojuan@163.com

**Dalin ZHANG,** Professor
School of Software Engineering,
Beijing Jiaotong University,
Beijing, China
E-mail: dalin@bjtu.edu.cn