Low-latency Dimensional Expansion and Anomaly Detection empowered Secure IoT Network

Wenhao Shao, Yanyan Wei, Praboda Rajapaksha, Dun Li, Zhigang Luo, Noel Crespi

***Abstract*—The Internet of Things (IoT) consists of a myriad of smart devices and offers tremendous innovation opportunities in industry, homes, and businesses to enhance the productivity and the quality of life . However, ecosystem of infrastructures and the services associated with IoT devices have introduced a new set of vulnerabilities and threats, resulting in abnormal values of information collected by sensors, jeopardizing system security. To secure sensor networks, it must be possible to detect such anomalies or sequences of patterns in IoT devices that significantly deviate from normal behavior. To perform this task, this paper proposes a real-time streaming anomaly detection method based on a Bloom filter combined with hashing. This method expands the data dimensions through a hashing algorithm, and then adopts competitive learning (Winner-Take-All) to build a multi-layer Bloom Filter anomaly detection model. The feasibility of the proposed algorithm is verified theoretically using two datasets, KDD (to detect anomalies at the TCP/IP network level) and Credit (to detect anomalies during credit card transactions). The simulation results show that the proposed in this paper can effectively identify anomalies in the simulation data streams, with almost 95% accuracy for both datasets.**

***Index Terms*—Internet of Things, System Security, Sensor Devices, Anomaly Detection, Bloom Filter**

I. Introduction

THE security of IoT (Internet of Things) systems has been the focus of many researchers [1], [2]. The proliferation of IoT devices and the ever-expanding scale of the data generated through these devices pose serious challenges to the security of existing IoT systems. One main reason for this risk is the sensor equipment; much of it is installed in a complex natural environment and has been in use for several years, sometimes decades. Given these conditions, this equipment may generate abnormal information which could then be transmitted to the network. Since the data collected by IoT sensors are high-dimensional and large-scale [3], it is very challenging to detect the abnormal information they can produce. Therefore, an accurate data flow detection method is needed to ensure that the information collected by the Wireless Multimedia Sensor Network (WMSN) [4]–[6] is secure by detecting sensor faults in real-time, thereby maintaining the safety and stable operation of the network system. In general, such a large-scale smart device network requires the use of the stream computing framework [7], [8] to process real-time data streams.

The abnormal information caused by sensor anomalies usually contains different types of data, such as mosaic images, missing frames in video, and incorrect data detection [9]. If abnormal information is not discovered in time, it will cause the information transmitted by the network to be invalid, jeopardizing the entire IoT security system. Therefore, anomaly detection algorithms for the IoT platforms will play a huge role in maintaining system security and ensuring the safety of both property and life [10].

Conventional anomaly detection algorithms are based on classification methods, including k-nearest neighbors and clustering, statistical-based approaches, information theory, and spectrum-based methods [11]. Even though each of these methods has its advantages and disadvantages, classification-based anomaly detection algorithms have shown high detection efficiency, and thus they have been very widely used in IoT security systems.

In 2019, Deng F et al. [12] proposed an abnormal traffic detection framework based on a Bloom Filter (BF). Thanks to its rapid processing capability, their framework can accurately detect abnormal traffic from the sensor with low time complexity. Their approach performed better than the conventional anomaly detection algorithms and can be utilized for real-time applications. However, single-layer bloom filters and direct attenuation of dimensions are not capable of enhancing the performance of anomaly detection. On the other hand, some methods perform well in anomaly detection, but they do not apply to real-time scenarios [13]. Apart from these limitations, the existing models have three main drawbacks: i) They cannot respond quick enough to perform online anomaly detection of large-scale data streams from various sensors; ii) They make it very difficult for a solidified detection model to give correct results on unseen data types as the scale of the data stream continues to expand and new types of data continue to emerge; and iii) In the high throughput and large-scale data streams, abnormal sensor information is overwhelmed by the large amount of collected information, and so the abnormal sensor information is difficult to detect.

To address the above-mentioned issues, this paper proposes a novel semi-supervised anomaly detection model, FJLT-BF (Fast J-L Transform Bloom Filter) for IoT secure systems. The FJLT-BF model can be updated in real-time, which ensures that our proposed model remains sensitive to newly arrived anomalies. The proposed algorithm combines Bloom Filter and dimensional expansion theory, which allows it to respond to the real-time detection of large-scale data streams promptly and to distinguish between normal and abnormal data more accurately. The proposed anomaly detection model is constantly being updated to ensure detection accuracy. The proposed algorithm is divided into four steps: 1) pre-process

Manuscript received xx xx, 20xx; revised xx xx, xxxx.

the dataset through FJLT-FLSH [14], a locality-sensitive hashing algorithm composed of a fast transformation matrix; 2) use labeled data to train and generate a Bloom detection model; 3) pre-prepare a test set and calculate the threshold; and 4) a detection model utilizes the identified threshold and applies it to the detection system. To analyze the anomalies in video streams that are part of the IoT systems, we used two datasets belonging to different domains to identify their classification performances. The first dataset is called the KDD and is used to detect anomalies at the TCP/IP network level, the second dataset is named Credit and contains streaming records to detect anomalies during credit card transactions. Based on our analyses, the model proposed in this paper significantly outperforms the current mainstream algorithms, achieving 95% accuracy for both datasets. This work offers the following main contributions:

- Proposes a rapid stealth anomaly detection model that can meet the high throughput requirements of IoT security systems;
- Develops an updatable algorithm model that can effectively prolong the decay period of algorithm performance and thereby ensure the persistence of a system;
- Proves (theoretically) that the proposed algorithm can effectively detect invisible anomalies in different data streams.

The remainder of this paper is organized as follows. We review related work in Section 2 and then describe the proposed model and algorithm in Section 3. The theoretical analysis is presented in detail in Section 4, followed by a description of the experiments and results in Section 5.

II. Related work

IoT System security has long been an important research direction for its special significance. Anomaly detection algorithms as a means of system security protection naturally attract a large number of researchers in recent years. Following explains the state-of-the-art on anomaly detection systems used in IoT systems and how the anomaly detection enhanced through integrating Bloom filters.

A. Anomaly detection in IoT

The threat of the IoT system mainly comes from two aspects. One is the wear and tear of sensor equipment, which leads to the collection of wrong information or information with large errors. The second is man-made malicious intrusions. Anomalies due to sensor attrition are more common than man-made malicious intrusions. Due to the large-scale, high-throughput characteristics of IoT data, it is difficult to detect abnormal information in the large-scale data stream. Hence, anomaly detection can be considered the most important aspect of the IoT systems security. Abnormal causes are divided into the following three categories: node anomalies, network anomalies, and data anomalies [15], [16]. Node anomaly refers to the abnormality of sensor network nodes that occur at the sensor layer in Figure 1 due to hardware or software failures. Network anomalies generally occur at the network layer indicating that data is distorted due to external interference during the wireless transmission process. Abnormal data generally refers to man-made malicious attacks, which may occur in any layer. And the external manifestations of all abnormalities are data fluctuations. Therefore, to detect abnormal information in the data stream, it is necessary to continuously check the data objects in the continuous data stream, analyze them one by one, and make judgment.

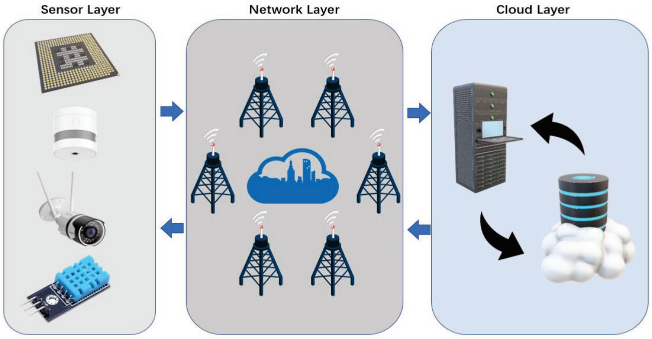


Fig. 1. The IoT Framework: From left to right, perception layer, network layer, application layer. The perception layer is responsible for information collection, the network layer is responsible for transmitting information; the application layer is responsible for processing and analyzing data

In 2017, Satoru et al. proposed a method for extracting precise failures and identifying their causes from network syslog data [17]. In 2019, Francesco Cauteruccio et al [18]. proposed a new method for automatic anomaly detection in heterogeneous sensor networks based on the coupling of edge and cloud data analyses. The approach focuses on detecting unexpected sensor data generated by the sensor system itself or by the environment under review; however, this method ignores the limitations of the computing power of edge devices [19]. In 2020, Sun et al. [20] proposed a two-stage network intrusion prevention system. In the first stage of their intrusion detection approach, support vector machines (SVMs) are used as detection algorithms to discover suspicious behaviors inside smart meters. In the second stage, the Temporal Fault Propagation Graph (TFPG) technique is used to generate attack routes to identify attack events. This study also inspired the work of our study, that is, using an SVM to separate abnormal information. In 2021, Cui et al. [21] introduced a blockchain-based decentralized asynchronous federated learning framework for anomaly detection in IoT systems to ensure data integrity, prevent sensor node failures, and improve the system operating efficiency. In 2021, Niu et al. [22] proposed a novel residual generator designed to detect indirect faults in sensor networks and a new cooperative decision-making strategy to ensure the stability of the detection results. However, this solution can only detect data inconsistencies caused by intermittent failures, and is less sensitive to abnormal sensor wear. In 2021, Sun et al. [23] proposed HVDC attack and defense control based on the FDIAs detection method. First, a squeeze excitation-based double convolutional neural network (SE-DCNN) is proposed to achieve fast identification of attack frequency types based on time- and frequency-domain signals. In 2022, Xiong et al. proposed a new algorithm, 2DP-FL, that separates anomalies by adding noise when training the local model, distributing the global model, and maintaining the safe operation of the system [24]. These solutions have played an important role in maintaining system security; however, there are still some limitations in the face of large-scale data streams and time performance. There is still a lack of new abnormal judgment technologies that can respond quickly and update in real-time.

B. Bloom Filter for Anomaly Detection

Bloom Filter is one of the most successful data structures that can respond quickly and low latency. Several existing methods have demonstrated the fast processing capability of the Bloom filter in large-scale high-throughput data streams. For example, in 2018, Groza et al. [25] proposed an intrusion detection mechanism that utilizes Bloom filters to test frame periodicity based on message identifiers and partial data fields. In 2019, Deng et al. [26] proposed a framework for abnormal traffic detection based on a Bloom Filter. Their framework can accurately retrieve information from large-scale real-time data with a low time complexity. In 2022, Teng et al. [27] proposed an improved hashing algorithm combined with a Bloom filter, which effectively improved the performance of hashing and enhanced the ability of the technology to monitor and detect large-scale high-speed network traffic in current IoT systems. However, the Bloom Filter sacrifices some data features while realizing fast data processing, which is an inevitable result of the hash function.

In 2018, Dasgupta et al. [28] proposed a novel anomaly detection algorithm that simulated the biological recognition process of odors. Their model recognizes objects in terms of their similarity to previous objects. Drosophila odor recognition process is divided into two steps:1) The first step assigns a hashing rule to each odor (the hashes between different odors may cross, but not repeat). The odor receptor neurons (ORNs) receive odor information and then transmit the signals emitted by the receptor neurons to Kenyon cells (KCs) via projection neurons (PNs) (per the hashing rules in the KC layer) and build a hash model of normal data flow. 2) The second step identifies the cells by determining the hash model and the novelty of each cell. This method offers a new perspective for anomaly detection models in secure IoT systems. There are a few drawbacks to this approach: the influx of a large amount of data from the sensor will inevitably affect the robustness of the model, and over time, the construction rules of the initial IoT anomaly detection model will become outdated. Any anomaly detection algorithm applied to IoT systems should have the ability to handle large-scale datasets and be capable of continuously updating the model over time.

Therefore, this study proposes a novel algorithm that uses a Bloom Filter as the basis of its detection model. Owing to the rapid response characteristics of the Bloom Filter, this detection model can respond to real-time data streams because our algorithm applies the dimension-expansion theory proposed in [14]. This approach helps to distinguish between normal and abnormal data, which can help protect IoT system security. The integration steps of Dimensional Expansion Theory into our proposed approach are explained in detail in Section III-B. The proposed algorithm also has thresholds and conditions for model updating; therefore, the detection model is updated continuously over time, ensuring the best performance.

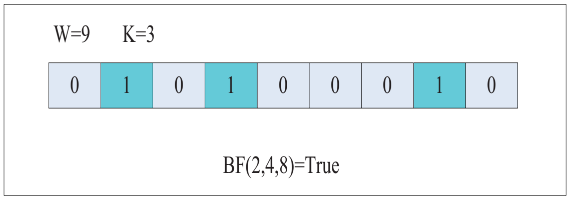


Fig. 2. Traditional Bloom Filter Model: W represents the length of the bloom, and K represents the number of bits the data occupies in the model

III. System Model

This section describes the framework of the IoT security system. The algorithm proposed in this work is mainly based on the Bloom filter and expansion theory, which can effectively combine the advantages of the fast processing of a Bloom filter and the advantages of dimension expansion, enabling the detection model to quickly process large-scale data streams

A. Bloom Filter Model

Bloom Filter (BF) was proposed by Bloom in 1970 [29] as a probabilistic data structure based on hashing. This data structure reduces the space occupied by the hash code by allowing a few errors in the hash code, which has become a common processing method for large-scale data [30]. In recent research, BF has been used as a solution for fast data retrieval and processing in IoT security systems.

1) Basic Bloom Filter: A BF consists of a vector array of n Boolean values, initially all set to 0 (false), and an element can be added to the bloom filter but not deleted. When an element ’x’ has to be added to the set, the element is hashed with ’k’ hash functions and ’n’ array positions are obtained, and the values in those indexes are changed to 1.

Currently, the function of a traditional BF is to determine whether a given element v is present in a set S. Suppose that a set of binary coding models with length w and its elements are initiated with a value of 0. The idea here is to map the elements in the collection to the model one by one through k hash functions and set the bits in the bit vector at the index of those hashes to 1. When k positions mapped by the query point in the model are 1. This proves that the query point exists in the set, that is, $v \in S$ and thus, output = True; otherwise, output = false, indicating that the query point is not in the set S.

Figure 2 shows a traditional Bloom Filter of length $w = 9$ with $k = 3$ hash functions. In this filter, the second, fourth, and eighth positions are activated. Therefore, when evaluating the element-wise value of $(2, 4, 8)$ after mapping, the model returns ’True’ as its output, which indicates that the model performs element-wise operations.

2) False positive value of a Bloom Filter: When an element refers to the same index as that of another element, the new element changes its value to 1, but the previous element has a value of 1. Hence, if an element is not present in the set, its existence returns a value of one, which is called a False Positive.

Figure 3 shows the process of false positives in the BFs. Suppose that there are two elements in a set $S(x_1, x_2)$. The

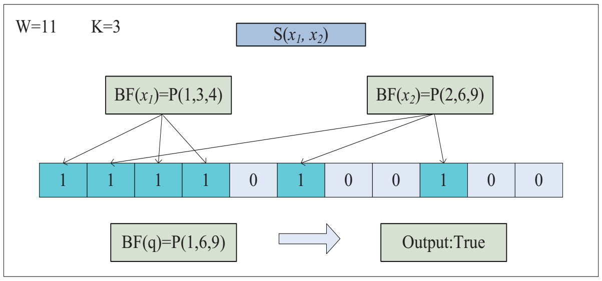


Fig. 3. False positive of Bloom Filter:False positives due to overlapping positions after data projection

mapped positions of x_1 are 1, 3, 4 while that of of x_2 are 2, 6, 9. A query point q is mapped to the model at positions 1, 6, and 9. When identifying set q that are mapped to positions 1,6,9, the output from the BF returns True, but positions 1 and 9 are already mapped with some elements in x_1 and x_2 sets. This behavior of the BF indicates that it is a False Positive state. The previous related literature [31] has given the calculation method of false positive probability P_{fp} as mentioned in Equation 1, where m is the length of BF, which is equal to W in Figure 3; n is the number of elements in the model, k is the number of hash functions, and the probability of mapping each element to a fixed position is $\frac{1}{m}$. When all elements after the k^{th} positions need to be activated, the total number of mappings is $n \cdot k$ times.

$$P_{fp} = (1 - (1 - \frac{1}{m})^{kn})^k. \tag{1}$$

$(1 - \frac{1}{m})^{kn}$ indicates the probability that a position in the model is 0 after all elements are mapped and $1 - (1 - \frac{1}{m})^{kn}$ when the probability is set to 1. Therefore, the probability that all positions of the element are set to one is indicated by $(1 - (1 - \frac{1}{m})^{kn})^k$.

B. The basis of dimensional expansion theory

Johnson and Linden-strauss [32] proposed the J-L theorem in 1984, which is to explore the law of similarity loss in the process of data dimension transformation It contains two important points: First, it proves that the minimum value of the dimensional reduction of high-dimensional data sets and the data of the original number of dimensions of the set are irrelevant. The second point proves the relationship between the loss of the relative distance (mainly the Euclidean distance) after dimensional reduction and reduced dimension k. A locally sensitive hashing (LSH) algorithm [33]–[36] inspired by the J-L theorem was later developed.

Equation 2 describes the relationship between the related parameters in the high-dimensional data-dimensional reduction process.

$$k \geq 4(\frac{\varepsilon^2}{2} - \frac{\varepsilon^3}{3})^{-1} \ln(n) \tag{2}$$

where k represents the dimension of the dataset after the dimensional reduction, n is the size of the dataset, and ε is the relative distance loss during the dimensional reduction process. The relationship between ε and relative distance is expressed by the following equation:

$$(1-\varepsilon)\|u-v\|^2 \leq \|f(u)-f(v)\| \leq (1+\varepsilon)\|u-v\|^2 \tag{3}$$

In Equation 3, u and v represent two points in a dataset of size n,$\| \cdot \|^2$ represents the Euclidean distance between the two data points, and $f()$ represents the description of the data after dimensional reduction. According to the value of k, the relative distance between two data points in the processed dataset varies between $[1-\varepsilon, 1+\varepsilon]$.

The J-L theorem provides theoretical support for the selection of the parameters of the data dimensional reduction process, and it corrects the previous error of researchers always linking the dimensions of the data after dimensional reduction to the initial dimensions. Subsequently, many scholars have proposed a series of related expansion theories [37], [38] based on the J-L theorem, which has supplemented the theoretical gaps in the dimensional reduction process of high-dimensional datasets, and has had a significant impact on the production of more scientific dimensional reduction schemes.

1) Dimension Expansion in Support Vector Machine and Kernel Method: Kernel methods (KMs) are pattern recognition algorithms that can be used to determine dependencies in a dataset [39]. More versatile kernel methods include support vector machines and Gaussian processes. The core idea of KMs is to Transform the original data into a suitable high-dimensional feature space through a certain nonlinear mapping (generally referred to as matrix projection), and then use a general linear classifier in this new space for further analysis. One advantage of KMs is that as it is a linear segmentation method, there is no overfitting.

The support vector machine (SVM) technique is a type of kernel method that is mainly used for classification [40]. SVMs divide the dataset linearly according to the corresponding category or type, usually into two different categories based on the dataset characteristics through the hyperplane. In the basic SVM approach, a dataset contains different types of data in the current vector space, and then a line (two-dimensional data) or a surface (three-dimensional data) from the vector space is used to divide the dataset into different categories. For example, as shown in Figure 4, the number of dimensions can be set to two and the dataset is divided into two categories.

Figure 4 describes the operation flow of SVM segmentation when the dataset is linearly separable, but the actual dataset is usually linearly inseparable (difficulty in finding a good hyperplane that can divide the data points). This situation is usually due to the subhigh cohesiveness [54] of the data, that is, the data of different categories intersect in the vector space; thus, it is impossible to divide the dataset into two categories. When dealing with this type of dataset, the usual practice is to project the dataset into a higher-dimensional space so that it can be successfully classified.

Figure 5 presents a visualization of this situation. Figure 5(a) depicts the data points that are not linearly separable in two-dimensional space, and Figure 5(b) shows the indivisible data points mapped in three-dimensional space (as it is difficult to obtain suitable higher-dimensional specific dimensions, only the ideal situation is discussed). From this figure, it can be concluded that we can easily find a suitable

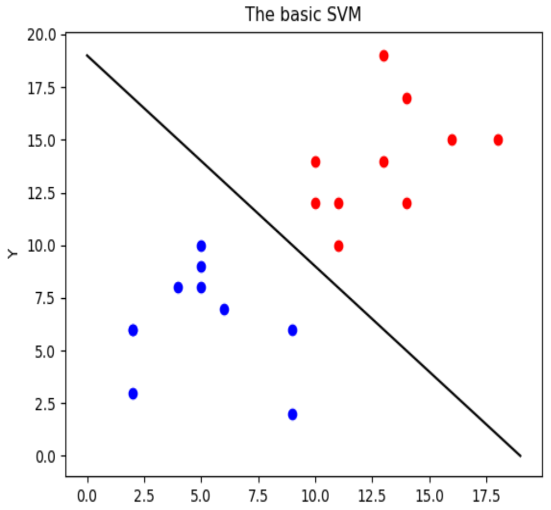
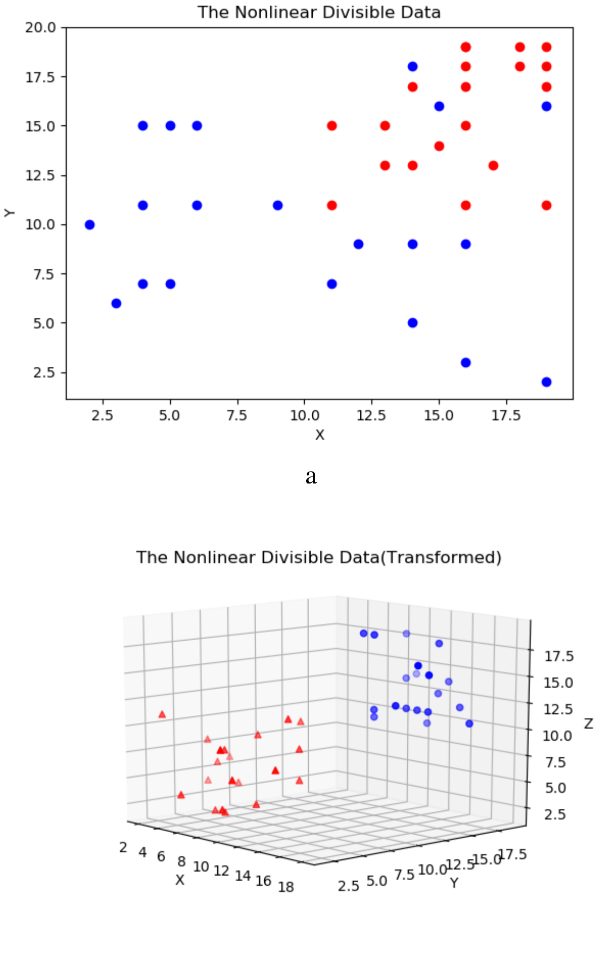


Fig. 4. Basic SVM segmentation: Linear segmentation in 2D space



a

b

Fig. 5. a is Linear indivisible data points are displayed in two-dimensional space; b is Three-dimensional space after SVM processing

hyperplane in three-dimensional space to correctly divide the dataset according to its category. Hence, when data points are projected into a higher-dimensional space, the difference between different types of data points will increase, which enhances the identification of abnormal data and inspires the work of this paper

C. Real-time anomaly detection for IoT

This work builds a BF-based real-time data stream anomaly detection model FJLT-BL that effectively combines BF and dimensional expansion theory and can be successfully applied to anomaly detection for large-scale data streams of the IoT. This section introduces the proposed model from the following two aspects: model construction process and real-time data stream detection.

1) The framework of real-time data stream using FJLT-BF: The real-time detection process of the data stream is illustrated in Figure 6. When the prior data from the collection of the wire multimedia sensor are processed, three important components are obtained: the detection model, growth threshold, and abnormality determination threshold. When the real-time data stream collected by the sensor is input to the model, the model simply preprocesses the data stream with reference to the prior data processing method and then adjusts the dimension of the data object to Bloom through the FJLT-FLSH algorithm [14], [42], [43], filtering the width of the model and retaining the position of the most significant feature of the data through a competitive learning strategy [44], [45]. Finally, the stored location information in the model was compared with the location of the significant feature value of the data object to calculate the outliers of the data object. The size of the abnormal values was compared with the abnormal threshold. If the abnormal value was greater than the threshold, the data were judged to be abnormal. Otherwise, the data were judged to be normal and the abnormal values were compared with the growth threshold. If it is less than the growth threshold, then the location information of the salient features of this piece of data is input to the model, the corresponding location is activated, and the model is updated. To make the model more stable, the activation threshold of a certain position within a time window is usually calculated during the experiment such that when the number of activations or collisions reaches the activation threshold, the model is updated to prevent the robustness from being reduced by a rapid model update rate. Simultaneously, each activated position in the model will automatically fail if it is not repeatedly activated for a long time.

2) FJLT-BF Model construction: The construction of the model was divided into five steps:

Step 1: Preprocesses the data set, which is mainly divided into two processing processes. The first is numerical processing, i.e., the normalization and standardization of the data set. Normalization and standardization can make the features between different dimensions have a certain degree of comparison in terms of value, which can greatly improve the accuracy of the classifier, and can also make the optimization process of the optimal solution smoother and easier to correct so that it more quickly converges to the optimal solution.

This article has been accepted for publication in IEEE Transactions on Network and Service Management. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2023.3246798

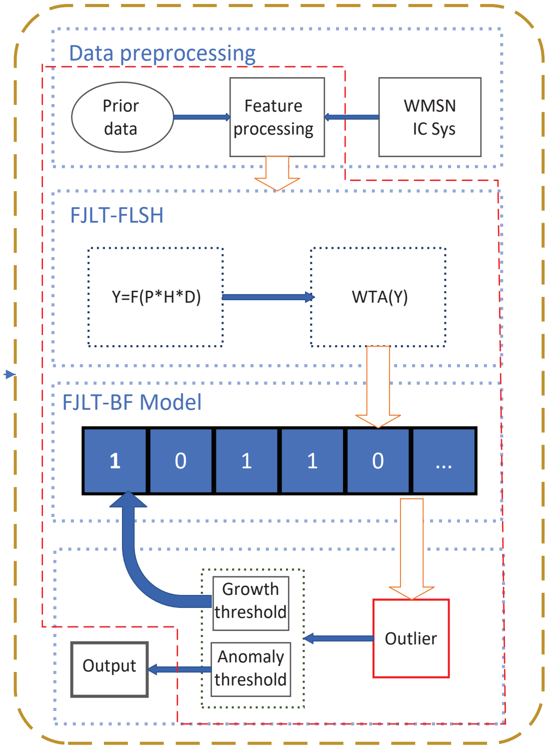


Fig. 6. FJLT-BF Detection Model Framework: This framework describes the dual process from data training to anomaly detection in secure IoT systems

Step 2: Processes the data by FJLT-FLSH algorithm. Dimensional transformation is done first, so that the prior data is transformed to the appropriate dimension through the FJLT matrix, and different types of data are separated. Next, the competitive learning strategy is applied to linearly reduce the dimensionality of the converted data, that is, to ensure that the more obvious feature values in the data set are retained. These are used to build a new index structure.

Step 3: Records the significant dimensional positions of the data processed by the FJLT-FLSH algorithm. The index position is reserved for the entire a priori data set in the BF, and the corresponding position is set to 1. This action builds the anomaly detection model.

Step 4: Train the threshold. The existing normal data and abnormal data are classified into two groups, and the abnormal average value, abnormal variance value, and maximum abnormal value and minimum abnormal value of the two data sets are calculated. According to each group of four values, two appropriate thresholds are calculated (generally set as the average value plus or minus t times the standard deviation) and the specific values can be adjusted according to different data sets.

Step 5: Verify the threshold and the model. After preprocessing a set of data with known labels, it is input to the model to calculate the outliers. The model's output is classified according to the outliers of the data and the two existing thresholds, which are com- pared to the original labels. If the accuracy rate reaches the appropriate range, the effectiveness of the model is proved.

D. Fast J-L Transform Bloom Filter (FJLT-BF): the anomaly detection algorithm proposed in this study

The algorithm is mainly divided into four steps: first, the data are preprocessed, and in the process, each data object is standardized. This step makes the model trained by the algorithm closer to the real data; the second step is to imitate the process of identifying odor information by the olfactory sense of fruit flies, use the improved FJLT-FLSH algorithm to expand the original dimension of the data to M, and pass the neural network of the fruit flies. Competitive learning retains the most significant $p\%$ characteristic of each data object. The purpose of this step was to simplify the separation of different types of data and effectively reduce the false positives of the BF. The number of misjudgments in the BF; the third step is to initialize the BF of length M, and map the position of each data object's reserved feature to the BF, which is activated according to the BF, the number of units and the calculation Equation 1 for false positives, calculate the theoretical false positives of the model at this time. If the false positives at this time are higher than 5%, then theoretically, the detection accuracy of the model is less than 95%. The value of M must be initialized. In a specific application process, it is necessary to set different false positive peaks according to different application scenarios and train a suitable model. The fourth step is to output a set of normal data and abnormal data prepared in advance to the model; calculate the average, variance, and standard deviation of the abnormal value of each data; verify the model twice according to the three parameters; and judge the validity of the model, that is, whether the two sets of data are clearly distinguished. If the model effect is poor, adjust the parameter M and the parameter p until a suitable model is trained. Then according to the abnormal value of the two sets of data, the corresponding growth threshold and abnormal threshold are calculated; theoretically, the p value is unchanged, the larger the M value, the lower the false positive of the model, and the better the detection effect; the M value is unchanged, the greater the p value is, The activation unit in the model is more likely to collide, false positives increase, and the more similar data is retained, the more uncertain the impact on the detection effect of the model (different data sets will produce different effects, after experimental testing, under normal circumstances, the data similarity retention gain is higher than the false positive).

The pseudocode of the algorithm is as follows.

Because this algorithm uses the FJLT-FLSH algorithm in [14] when processing real-time data streams, the time complexity of the algorithm in this chapter for the real-time detection of data streams in actual operation is:

$$E[|P|] = O(\varepsilon^{p-4} \log^{p+1} n) \tag{4}$$

IV. Analysis of Changes in Data Objects in the Process of Dimensional Expansion

Due to the distortion of space, the data projection process generally causes a loss of similarity between data, except for certain matrices (such as Fourier transform and Hadamard transform) [46] [47]. Data dimensional reduction will improve the efficiency of data processing, and having higher

Algorithm 1 FJLT-FLSH

```
Require:
  Input:Normal Data and Test Data
  Process by algorithm 1 Step 1,2,3
  $Model_{F-BF} = \Sigma X^N$
  Output: $Model_{F-BF}$, Threshold Outliers $\alpha$; Threshold Extends $\beta$
Ensure:
  Input: $data(X)$;
  Step 1: $X^P = Pre(X)$, Data Preprocess;
  Step 2: $X^E = FJLT(X^P, M)$; Extend Dimension to $M$
  Step 3: $X^N = WTA(X^E, K)$; Reserve k Features by competition learning.
  Step 4: $Ols = 1 - Intersect(Model_{F-BF}, X^n)$. Calculate outliers
  Step 5: Judge: Outliers
  If $Ols > \alpha$: Output: The data(X) is Anomaly Data
  Else: Output: The data(X) is normal Data
  If $Ols < \beta$: $Model_{F-BF} = Model_{F-BF}.Add(X^N)$. Update Model.
  Step 6: $Model_{F-BF} = Model_{F-BF} * f$. Update Model.
```

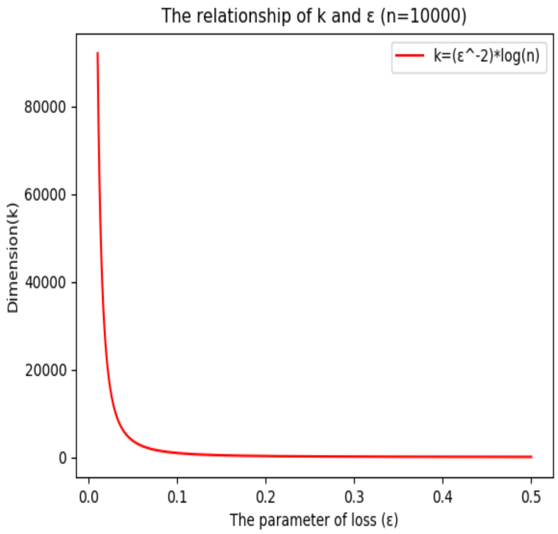


Fig. 7. The relationship of k and ε under the size of 10000

dimensions helps to effectively distinguish between normal and abnormal data, such as constructing an SVM classifier through a kernel function to segment the data.

A. Dimensional expansion theory

Traditional data dimensional reduction is designed to directly reduce the feature dimensions of high-dimensional datasets (usually through principal component analysis, support vector product, etc.) to reduce query time and space occupation. However, even if dimensional reduction is performed by principal component analysis (PCA) and other methods, the accuracy might be reduced because the process of dimensional reduction must conform to the J-L theorem. According to the J-L theorem, there is a relationship in Equation 2 between the reduced dimension k of the dataset and loss parameter ε. That is, the more the dimensionality of a dataset is reduced, the more similarity within the dataset is lost [42].

Figure 7 depicts the relationship between the loss parameter ε during the projection process and dimension k after projection. Based on Figure 7, the value of dimension k after projection must be sufficiently large to reduce the similarity loss during the projection process.

The FJLT-FLSH algorithm is a locality-sensitive hashing algorithm combined with a fast matrix transformation [14], which has the following relationship:

$$k = \varepsilon^{-2} \log(n) \tag{5}$$

There are two theorems for extension of the J-L theorem: it shows that losses caused during data transformation are unavoidable and that there is a bottleneck in the preservation of the degree of similarity between the transformed and original data.

Theorem 1: (Advantage of Dimensional extension) Let the size of the original data be n and dimension be l.After the dimensional reduction of the data, the dimension is k, the distance loss is ε_1, and the corresponding maximum regression distance $RS_1 = 1 - \varepsilon_1$. The function represents the highest value that the query precision of the algorithm can reach. After dimension expansion of the data, the dimension is $M = mk$, the loss parameter is ε_2, and the corresponding maximum regression distance $RS_2 = 1 - \varepsilon_2$,in which $k < l < M$. Then, the difference of the maximum regression distance is defined as follows: $\Delta RS = RS_1 - RS_2$

Theorem 2: (The Maximum Regression Distance Difference Theorem) sets the original dataset to S_0. After dimension reduction, the dataset was S_1, and after dimension extension, the dataset was S_2. Compared to the dimension reduction dataset, the dimension extension dataset can obtain a higher Maximum Regression Distance.

$$\begin{aligned} RS &= RS_2 - RS_1 \\ &= (1-\varepsilon_2) - (1-\varepsilon_1) \\ &= \frac{\sqrt{\log n}}{\sqrt{k}} - \frac{\sqrt{\log n}}{\sqrt{mk}} \\ &= \frac{\sqrt{m(m-1)\log n}}{m\sqrt{k}} \end{aligned} \tag{6}$$

In Equation 6, when the values of n and k are fixed, the M value increases as the m value increases, since the value of k does not change, RS_1 does not change. The difference in the maximum regression distance ΔRS is larger, and the maximum regression distance RS_2 of the extended-dimension dataset is superior to the maximum regression distance obtained by the reduced-dimension dataset.

When the values of n and m are fixed, as the k value increases, ΔRS gradually decreases; that is, RS_1 and RS_2 gradually approach each other. Assuming that k breaks through limit $(k < l)$ and becomes infinitely close to positive infinity, $\lim\limits_{k\to+\infty} = \Delta RS = 0$. And, $\lim\limits_{k\to+\infty} \varepsilon_1 = \lim\limits_{k\to+\infty} \varepsilon_2 = 0$ can be introduced, and the loss parameter at this time is infinitely close to 0 and the maximum regression distance $\lim\limits_{k\to+\infty} RS_1 = \lim\limits_{k\to+\infty} RS_2 = 0$, in which the peak of the query precision is infinitely close to 1, and $\lim_{k\to+\infty} \Delta RS = 0$. When k is infinitely close to and greater than $\log n$, then the loss parameter $\lim_{k\to\log n} \varepsilon = 1$, $\lim_{k\to\log n} RS_1 = \lim_{k\to\log n} RS_2 = 0$, $\lim_{k\to\log n} \Delta RS = 0$

The theorem above indicates that a dataset with extended dimensions can obtain a higher maximum regression distance than a dataset with reduced dimensions. Therefore, the peak of query accuracy that can be achieved by extending the dimensions is much higher than the peak of query precision obtained by reducing the dimensions. Table 1 shows a comparison between the maximum regression distance, the converted dimension k and the loss parameter ε, which is calculated using Equations 5 and 6.

TABLE I
EXAMPLE OF THE RELATIONSHIP BETWEEN THE MAXIMUM REGRESSION SIMILARITY AND k AND ε VALUES UNDER THE DIFFERENT DATA SET SIZES

n=10000			n=1000000		
k	ε	RS	k	ε	RS
10	0.9487	0.0513	10	Nan	Nan
100	0.3033	0.6967	100	0.3715	0.6285
1000	0.0949	0.9051	1000	0.1175	0.8825
10000	0.0303	0.9697	10000	0.0371	0.9629

In the FJLT-FLSH algorithm, the main reason for enlarging the data dimension is that the FJLT matrix used in the third chapter obeys the J-L theorem; that is, the larger the value of dimension k after projection, the smaller the similarity loss caused by the projection. Therefore, the dimensionality of the data is enlarged, the similarity between the data objects is better preserved, and a higher maximum regression similarity is obtained. Simultaneously, because of the existence of the Fourier transform and kernel function [46], the time consumed by the dimensional expansion process is acceptable. The algorithm in this study amplifies the dimension of data features, can more efficiently retain the similarity between data objects, obtains higher query accuracy, and can better simulate the recognition process of biological sensory nerves. Finally, we can conclude that we can better distinguish the differences between different data only by amplifying the received signal.

Anomaly detection is the process of segmenting data. This study found that in the process of dimensional expansion, similar and dissimilar data differ in their degree of similarity loss, as explained in Section IV-B. To explore the advantages of dimension expansion on anomaly detection tasks, this study analyzes two directions: from the problem of data similarity loss in the dimension expansion and from the analysis of the false positive impact of BF.

B. Data similarity loss analysis

Under limited conditions (when the value corresponding to a data point retains a limited number of decimals), the probability of any two points colliding in a two-dimensional plane is much greater than the probability of any two points colliding in a three-dimensional space. Similarly, normal and abnormal data will increase the dissimilarity between the two types of data after dimensional expansion. The normal data have a similar data structure, and thus, the similarity between them only occurs with a smaller distortion during the dimensional expansion. Therefore, it is easier to separate abnormal data from normal data by using dimensional expansion.

We can verify the relationship between similarity loss and dilation dimension by simulating data points.

For example, assume that there are three data points in a three-dimensional space, namely A, B, and C. Consider the coordinates of $A = (x, 2x, 3x)$, $B = (2x, 4x, 5x)$, and $C = (x, 7x, 4x)$. First, A is considered as the target point. The distance from B to A is $3x$ and the distance from C to A is $\sqrt{26}x \approx 5.09x$. Therefore, we can assume that B is the nearest neighbor of A relative to C. Based on Euclidean space, the similarity between points A and B is higher than that between points A and C. We can randomly select the following matrix M to expand the dimensionality of the original data points:

$$M = \begin{pmatrix} 2y & y & y & y \\ 4y & 3y & y & y \\ 5y & 6y & 2y & 1y \end{pmatrix} \tag{7}$$

$\hat{A}$, $\hat{B}$ and $\hat{C}$ are the resultant data points when data points A, B, and C are projected by the projection matrix Min into three-dimensional space. The resultant matrix can be represented as:
$\hat{A} = (25xy, 25xy, 9xy, 6xy)$,
$\hat{B} = (45xy, 44xy, 16xy, 11xy)$,
$\hat{C} = (50xy, 46xy, 16xy, 12xy)$.

The Euclidean distance between points A and B is $D_{AB} = \sqrt{835}xy \approx 28.89xy$ and the distance between points A and C is $D_{AC} = \sqrt{1135}xy \approx 34xy$. This indicates that the distance between point A and point B increases by $\triangle D_{AB} = 25.89xy$, and the distance between point C and point A increases by $\triangle D_{AC} = 28.91xy$ and thus, $\triangle D_{AC} > \triangle D_{AB}$.

This study further analyzes this phenomenon and finds that when matrix M is the sum of the squares of the coefficient roots of any rank greater than 1, the conclusion remains robust, and because the number of dimensions in the process of dimension expansion is large, the conclusion is general. We can conclude from these facts that dimensional expansion can more easily separate dissimilar data. This further proves that, in the process of dimensional expansion, the degree of similarity loss between similar and dissimilar data is different. This is consistent with the previous assumption in this study, that is, when the input signal is amplified, the data structure is distorted, and it becomes easier to distinguish signals of different data.

C. The relationship between BF false positives and dimensional extension

Equation 8 shows how the false positive calculation method of Bloom Filter is represented theoretically, where m is the length of BF, that is, the expanded dimension of the data set, k is the number of elements required for a piece of data in BF, and n is the size of the data set [31].

$$P_{fp} = (1 - (1 - \frac{1}{m})^{kn})^k \approx (1 - e^{\frac{kn}{m}})^k \tag{8}$$

This article has been accepted for publication in IEEE Transactions on Network and Service Management. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2023.3246798

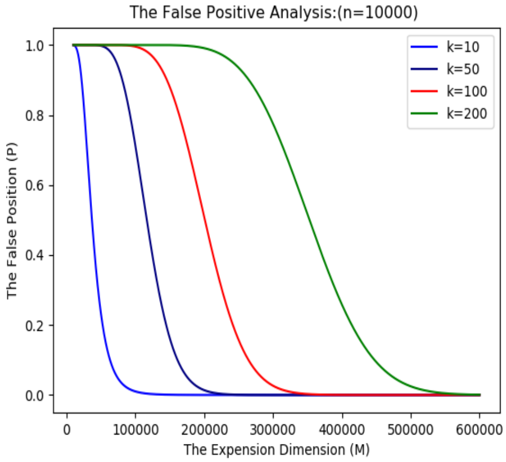


Fig. 8. False positive analysis under different k values

Equation 8 indicates that with the continuous expansion of dataset dimensions, the false positives of BF is gradually reduced, so the probability of misjudgment is gradually reduced, resulting in abnormal data becoming more significant, which is not conducive to hiding abnormal data. It is useful for distinguishing abnormal data from normal data.

Figure 8 shows the theoretical curve of the BF false positives under different k values, which are expanded to different dimensions. This clearly shows that when the dataset is expanded to higher dimensions, there are fewer false positives. Figure 8 also shows the difference in the rate of false-positive decline under different k values. The smaller the value of k, the faster the rate of false-positive decline, and vice versa. However, the smaller the value of k, the greater the loss of data similarity. The larger the value of k, the lesser the loss of similarity. Therefore, when selecting the value of k, it is important to weigh the decreased rate of false positives and loss of similarity to select the appropriate parameters. We used the following approach to select the appropriate parameters. In the first stage of the proposed algorithm, the dataset is expanded to m dimensions and then reduced to k dimensions. In the first stage of the algorithm proposed by the paper, the data set is expanded to m dimensions, and then reduced to k dimensions. Therefore, m in the BF represents k in JL. In the second stage, the dataset was quickly reduced to k dimensions through competitive learning. Therefore, k in the J-L theorem is the k value in the BF. The false positives and conversion of the J-L equation during the two stages are given below: **The first stage**: expand the data set with size n and original dimension d to m dimensions:

$$\triangle_1 = P_{fp} + \varepsilon \approx (1 - e^{\frac{kn}{m}})^k + \sqrt{\frac{\log n}{m}} \tag{9}$$

At this time, the independent variable is m, which is the expanded dimension of the data set.

The second stage: build a model through competitive learning, that is, keep k-dimensional data after dimensionality expansion

$$\triangle_2 = P_{fp} + \varepsilon \approx (1 - e^{\frac{kn}{m}})^k + \sqrt{\frac{\log n}{k}} \tag{10}$$

The independent variable is k, which is the number of bits used to represent the data point in the BF. By setting $x = m; y = k$, the binary function is obtained as follows:

$$\triangle = \triangle_1 + \triangle_2 = 2(1 - e^{\frac{-yn}{x}})^y + \sqrt{\log n}\sqrt{\frac{x+y}{xy}} \tag{11}$$

When $\triangle$ takes the minimum value, the selected parameter is theoretically optimal.

Through the analysis of the two parts of the detection model construction process, surface dimension expansion plays a positive role in the abnormal detection process. The first positive effect is to increase the distance between abnormal and normal data, which makes it easier to strip the abnormal data hidden in the normal data. The second positive effect is to reduce false positives in the BF detection model, that is, to reduce the hidden space of abnormal data and improve detection accuracy. Finally, Equation 11 can be used to improve parameter selection.

V. Experimental Results

A. Datasets

It is difficult to collect a dataset of fault information for IoT devices and it is common practice to verify the validity of the proposed model through simulation experiments. In this study, the fault information extracted from sensory data stream was simulated using the KDD [48], [49] and Credit [50] datasets. This article uses two data sets to evaluate the algorithm proposed in this article. The first dataset is the KDD dataset, which is one of the few public domains that utilizes TCP/IP level information and is embedded with domain-specific heuristics to detect intrusions at the network level. At present, this dataset has become a benchmark dataset in the field of anomaly detection. This dataset has 41 features, but the scale of the dataset is large and suitable for us to explore Anomaly detection and analysis of large-scale real-time data streams. The second dataset is a credit card fraud dataset, which sets a certain label for each entry, is suitable for training the model, and can be used to evaluate the detection accuracy of the proposed algorithm. This dataset contains 5000 pieces of data, and we can use this dataset as a data stream to evaluate the our model.

B. Experiments and Analyses

This study mainly evaluates the algorithm from two aspects, as mentioned below.
i) The total accuracy of the anomaly detection was Ar_1, which is the accuracy of the detection model for detecting all data objects. It represents the total number of correctly classified items-Tt divided by the total number of classified items-Tn.
$Ar_1 = Tt/Tn$.
ii) The detection accuracy rate of abnormal data Ar_2 where the total number of detected abnormal items Ta divided by the total abnormal items in the dataset Tan.
$Ar_2 = Ta/Tan$

The model detection accuracy rate is the most representative indicator to evaluate the algorithm, which guarantees the effectiveness of the algorithm and therefore, we can use this metric to evaluate our anomaly detection algorithms.

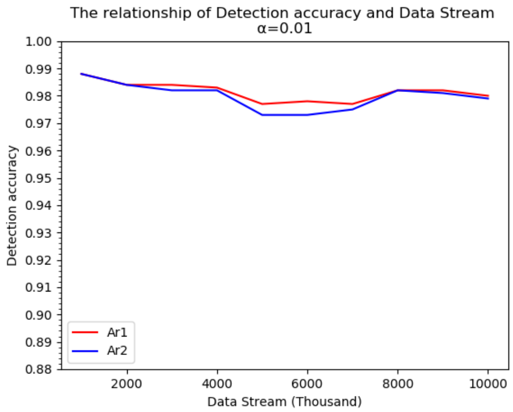


Fig. 9. Changes in detection accuracy when threshold is α

In order to verify the feasibility of the FJLT-BF proposed in this article, we designed the following six sets of experiments.

*1) **Experiment 1 :*** To prove the algorithm proposed in this paper, it is first necessary to verify the detection accuracy when the model remains unchanged for hyper-parameters.

To train the model, randomly select M_n pieces of normal data, expand it to the M dimension, and the expanded dimension M is related to M_n, and then pass the expanded data through competitive learning to retain k features that best represent the data. Then, sets the remaining k features of each piece of data in a one-dimensional space of length M, in which, the position of the projection is set to 1, and the remaining positions are set to 0 to generate the model.

The second step is to train the threshold for multiple itineraries, each of which takes M_n pieces of normal data and M_n pieces of abnormal data. After the same dimensional expansion to M dimensions, and retaining k features through competitive learning, calculate the intersection of each data and mode, calculate its novelty, and select a value based on its average and variance, which can distinguish most abnormal data (it needs to be able to distinguish at least 95% of abnormal data). Set this value as the abnormal threshold α, and then select a value according to the training set of normal data, which contains at most 5% of the normal data as the extended threshold β.

The third step is to test the accuracy of the model and calculate the novelty of the test set after the same projection. The abnormal threshold for this group of experiments was $\alpha = 0.01$. Due to the uninterrupted data flow, this study sets 1000 data points from KDD dataset as a time window, $M_n = 5000, M = 20000$, repeated 10 times, and then average value was taken.

Based on Figure 9, the detection accuracy of the model proposed in this study was in the range from 97% to 98.9%, which indicates that our algorithm proposed in this study has high detection accuracy. Although the detection accuracy of our model remains at a high rate as the time window continues, it decreases over time and therefore, it is required to update the model continuously.

*2) **Experiment 2**:* On the basis of Experiment 1, Experiment 2 adds the model update threshold β, but does not set the activation parameter π (i.e, $\pi = 1$). In this set of experiments, we extracted three subsets from the KDD dataset for testing purposes. First, the abnormal threshold for the first batch of experiments is set to be $\alpha = 0.01$. Due to the uninterrupted data flow, we set 1000 data points to be in one time Window (i.e. $M_n = 5000, M = 20000$). These experiments are repeated 10 times, and the average accuracy is taken to evaluate the performance. The dataset used in this group of experiments is a subset of the KDD dataset, which is different from the subset extracted in experiment one.

Figure 10 shows the experimental results for this Experiment 2. We can conclude that the detection accuracy of the model was significantly improved when the model update with threshold value. However, due to setting the model activation threshold to 1, the number of units in the model is easily activated, and thus, the detection accuracy of the model shows a significant increase in the early stage. However, When it rises to the peak value, the probability of misclassificaiton increases because of the false positive of the Bloom filter increases. Hence, the accuracy of the detection model shows a downward trend. Through this set of experiments, we realized that there is an upper limit to the endurance of the model, where if the activation parameter activation setting is set to 1, the $M - bit$ filter we set will be rapidly occupied, resulting in a rapid increase in the detection accuracy of the proposed model and then a rapid decline. If the number of activation units in the model increases rapidly, the detection ability of our model reaches the upper limit at the earliest convergence. At this point, the model is homogenized by a new type of data, which makes it difficult to distinguish different types of data.

*3) **Experiment 3**:* In the third set of experiments, we added the relevant model update process to set the updated threshold β and the activation threshold π. First, we construct the model and then, the number of initially activated units, such as the initial model length IML, the abnormal threshold α and the expansion threshold β in the model, are recorded through two training sets. Then data flow was detected followed by the previous step. With the continuous influx of data flow, when the abnormal value of a data object is less than the expansion threshold β, a dimensional feature that does not match the object and the model is added to the activation array. When the number of occurrences of an element of is greater than or equal to the activation parameter π, the position of the element in the model Mode is activated and set to 1. At this time, the value of IML is also increased by 1. In this set of experiments, in order to highlight the update of the model, we took 2000 data as a time window, and count the number of model activation units to show the model update.

Figure 11 shows the change curve of the detection accuracy Ar_1, Ar_2 with in terms of time windows. We can observe that there is a value above the accuracy for each time window. This value represents the number of active units in the model and can be interpreted as the model length ML. Following conclusions were drawn from this Figure 11. i) When we

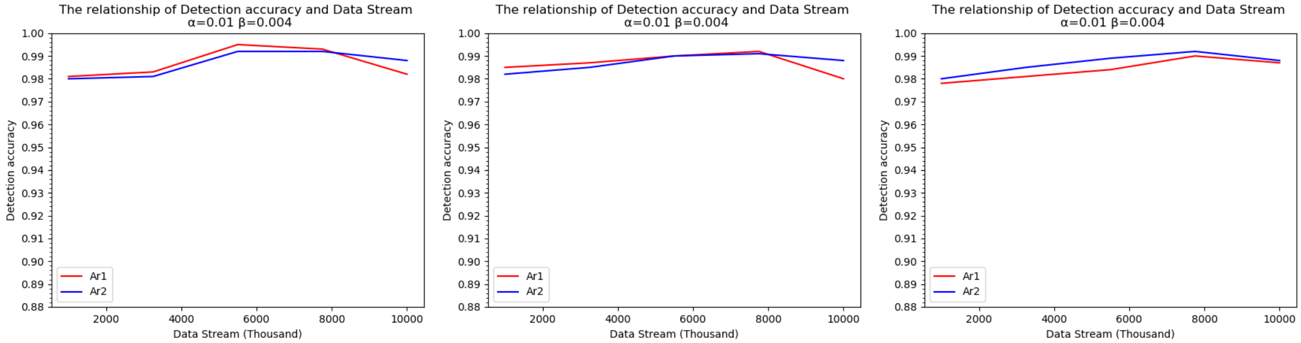


Fig. 10. Detection accuracy after adding threshold β

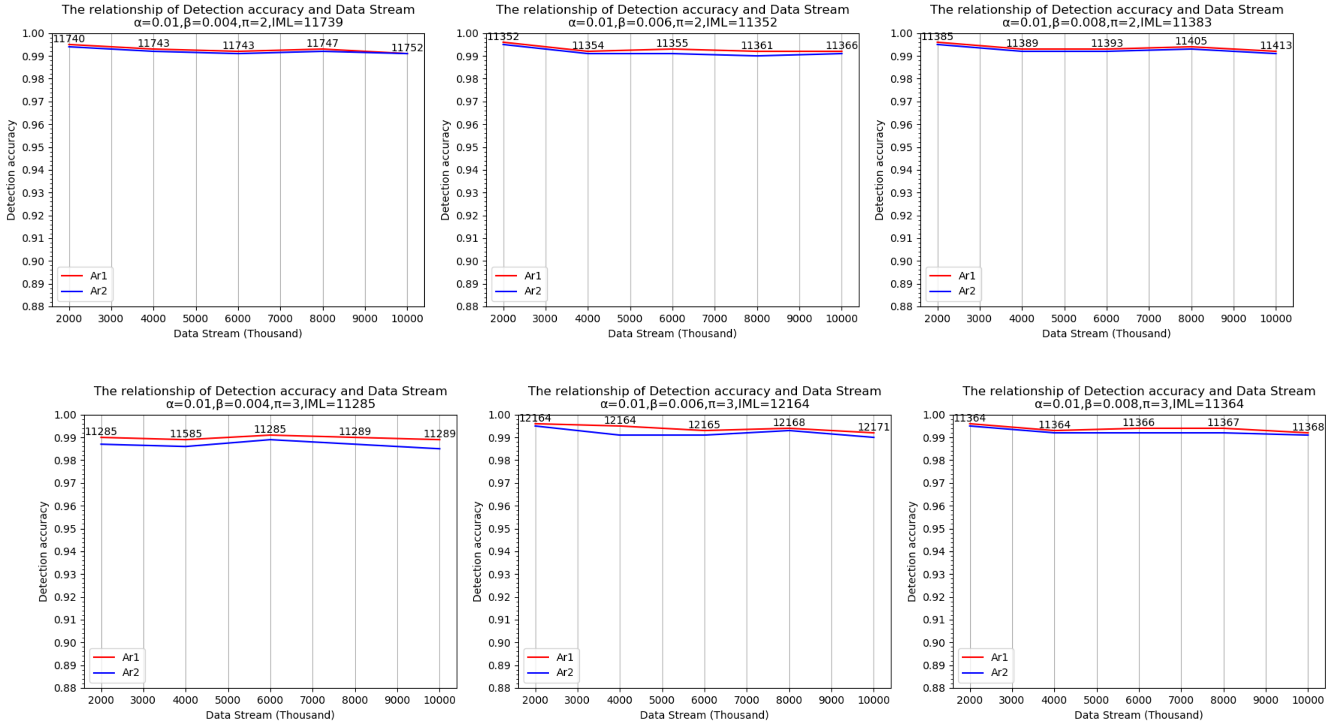
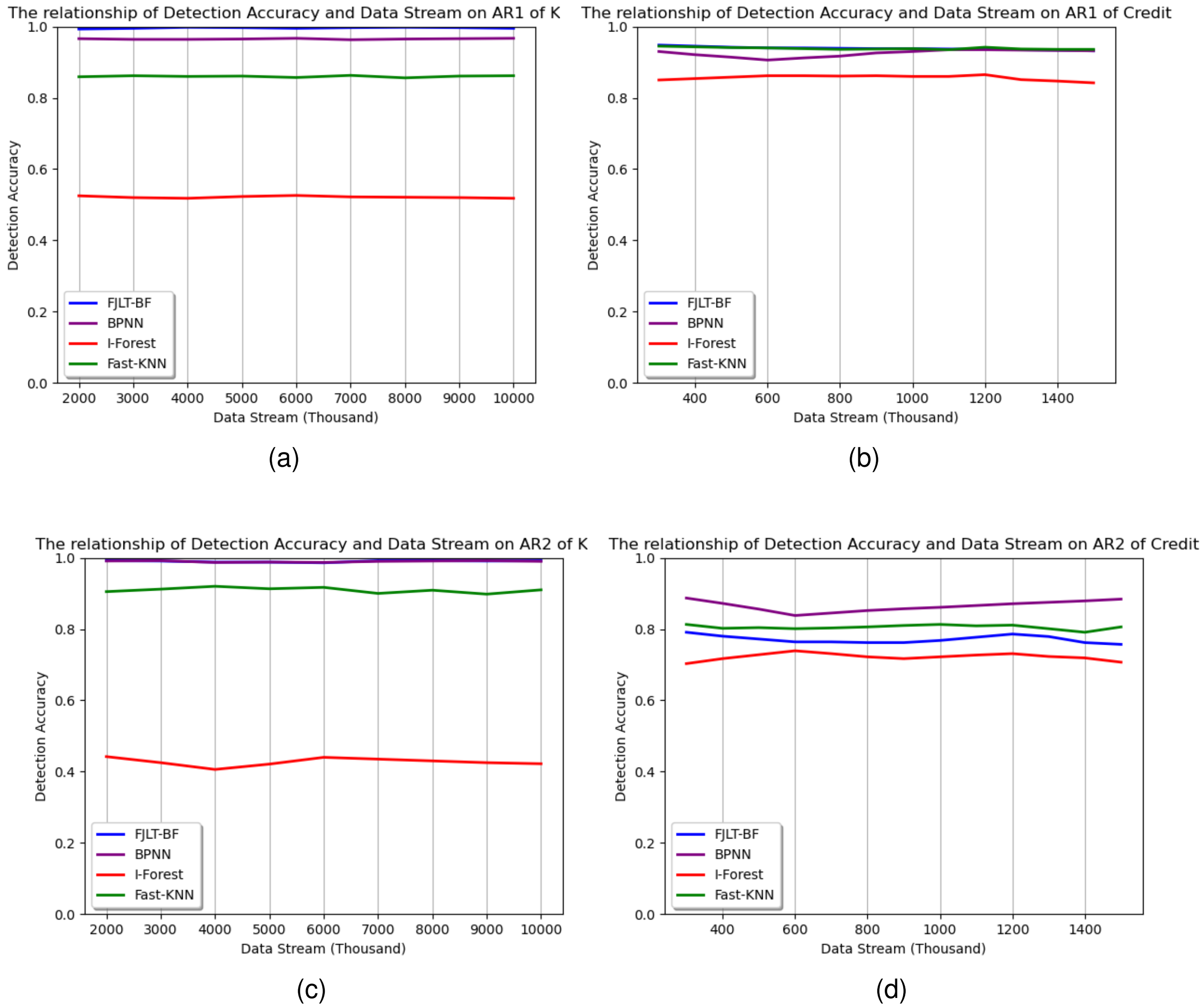


Fig. 11. Anomaly detection results under different thresholds and activation parameters π

add the update function to the model, the detection both accuracy Ar_1 and Ar_2 are both higher than the accuracy under a fixed model length. ii) After comparing Experiments 2 and 1, it can be concluded that when the activation function is set for the model, the detection accuracy of the model remains unchanged with the continuous influx of data flow. The detection accuracy in Experiment 1 has decreased with the continuous influx of data streams, which proves that the model proposed in this study has achieved significant performances in the abnormal detection of data streams. iii) Figure 11 clearly shows that the number of model activation units, ML, automatically increased with the continuous influx of data flow. By setting different activation parameters π, the growth rate of ML can achieve different values. The larger the value of π, the slower the growth of the ML and the smaller the value of π, the slower the growth of the ML. If π is set to a too large value, the model is likely to experience many time windows, and the model is still not updated, which causes the detection accuracy of the model decrease. Additionally, the β value is another important parameter for updating the model. Setting different expansion thresholds β results in different model growth rates. In other words, the larger the β value, the faster the model growth, and vice versa. The model grew slower if the β value is large. When the dimension of abnormal data added to the model, then the abnormal data appears in future can recognized as normal data, resulting in a significant decrease in the accuracy of detecting abnormal data in the subsequent data stream; thus, the value of β is generally selected to be a smaller value.



(a)

(b)

(c)

(d)

Fig. 12. Comparison of the accuracy of four anomaly detection technologies

*4) **Experiment 4**:* To compare the accuracy of the proposed algorithm with several baseline models, we calculated two accuracy evaluation metrics (Ar_1 and Ar_2) using two datasets. We used three baseline models in our evaluations: IForest [51] - Isolated Forests proposed by Zhou Zhihua [52], the BPNN neural network algorithm [53] and the Fast-KNN [54]. In Experiment 4, these baseline models are compared with the FJLT-BF algorithm proposed in this paper. We first divided the pre-processed dataset to analyze time-series data and then entered the trained model to calculate the outliers in the time windows to determine the abnormality. To evaluate FJLT-BF algorithm, we executed other baselines models in the same operating.

The experimental results are shown in Figure 12, where subimages (a) and (b) are the comparison of the detection accuracy of the four algorithms on the two datasets under the Ar_1 indicator. In terms of Ar_1, the model proposed in this paper shows better performances compared to other models. Among them, the IForest algorithm showed less accuracy on the KDD dataset. The accuracy of the algorithm proposed in this study and the BPNN can reach more than 0.95%. On the credit card dataset, the detection accuracy of the IForest has reached more than 0.85%, which indicates that different data types will affect the efficiency of the anomaly detection model.

Subfigures (c) and (d) show the anomaly detection accuracy of the four algorithms in the two datasets under the Ar_2 metric. Based on the Ar_2 metric our F-BF algorithm proposed in this study performs well on the KDD dataset and performed better than BPNN and IForest. However, on the credit card fraud dataset, the BPNN performed better than that of the other algorithms used in this experiment. However, one drawback of the BPNN is that it is costly, in terms of training time and space, to train a NN for anomaly detection. The Fast-KNN algorithm needs to calculate the distance between a large number of high-dimensional data points during the calculation process, which consume lots of time to converge and also requires additional space. In summary, the novel anomaly detection algorithm FJLT-BF proposed in this study has higher efficiency than the baseline anomaly detection models.

TABLE II
THE COMPARISON OF TIME PERFORMANCE

Times(s)	1	2	3	4	5	6	7	8	9	10
IForest	3.1	3.2	3.1	3.2	3.3	3.3	2.9	3.1	3.2	3.2
Fast-KNN	50.2	52.1	49.5	50.5	49.2	49.6	48	52.6	51	53
BPNN	296.5	295.7	291.2	290.7	300.2	287.9	297.8	281.5	299.5	293.6
F-BF	12.6	12.8	12.6	12.7	12.8	12.4	12.9	12.8	12.6	12.7

5) ***Experiment 5:*** In this set of experiments, we compared the time efficiency of four anomaly detection algorithms (BPNN, Fast-KNN, IForest, and F-BF) under the same execution platforms and configuration. The total time consumed during the execution of the algorithm consists of two parts: model building and model checking. To explore the time loss of the four algorithms in actual operation, this study separately obtains the timestamp when the program starts and terminates. The difference between the two timestamps is calculated to obtain the execution time of the algorithm. To ensure the authenticity of the results, this study conducts 10 tests on the four sets of algorithms, each acquisition builds a model, and calculate the total time to test 1500 sample data points. The experimental results are shown in Table 2.

It can be concluded that the total time spent by the IForest was the smallest, followed by the F-BF algorithm proposed in this study. The BPNN requires the most time because of the need to iteratively train the model. In our experiments, we found that NN training consume more time, and in the detection process, the neural network algorithm for detecting 1500 data takes approximately 0.6 seconds. The execution time of the Fast-KNN is also high because it needs to calculate the distance between multiple data. In short, the algorithm proposed in this study simplifies the model training process and improves the efficiency of model training. In addition, the optimization of the model in terms of execution time in this study has low latency when processing data. It is known from Experiment 4 that the detection accuracy of the algorithm in this study was much higher than that of the isolated forest algorithm.

6) ***Experiment 6:*** In our analysis, we designed a set of experiments to explore the impact of dimensional expansion on the detection rate Ar_1 and Ar_2 using four different datasets. The first dataset is 2000 normal data extracted to train the model, and the second is to use another 1000 normal data and 450 abnormal data (there are only 495 abnormal data in credit card fraud) as a training set. Then, a detection model was constructed using the method explained in Experiment 1, and the threshold α was calculated using two training sets. Finally, another 1500 pieces of data were selected for testing.

Experiment 6 has two important parameters: expanded dimension M and number of features reserved for competitive learning k. The selection and distribution of the value k is based on the J-L theorem introduced in Section IV-C, Figure 8, and Equation 11 and the k setting range is near the inflection point. This experiment is divided into three groups as follows and the experimental results are listed in Table III.

1) First, set the model length $M = 30000$; that is, all datasets are expanded to 30,000 dimensions, and then k ($k = 1000, 1500, 2000, 2500, 3000$) features are retained. Finally, the detection accuracy Ar_1 and Ar_2 are calculated through the test set;
2) Set the total length of the model to $M = 50000, k = 1000, 2000, 3000, 4000, 5000$. Calculate the detection accuracy Ar_1, Ar_2;
3) Set the fixed length $k = 3000$, select different model total lengths $M = 10000, 20000, 30000, 50000, 60000, 90000$. In other words, the fixed number of retained features is calculated, and the detection accuracy Ar_1, Ar_2 under different model lengths.

TABLE III
THE EXPERIMENTAL RESULT OF EXPERIMENT 6

	$M = 30000$				
k	1000	1500	2000	2500	3000
α	0.001	0.00067	0.0005	0.0004	0.00033
Ar_1	0.914	0.919	0.922	0.926	0.927
Ar_2	0.677	0.698	0.718	0.744	0.744

A. The influence of α value on the test result under M=30000

	$M = 50000$				
k	1000	2000	3000	4000	5000
α	0.001	0.0005	0.00033	0.00025	0.0002
Ar_1	0.918	0.923	0.93	0.933	0.933
Ar_2	0.695	0.732	0.759	0.77	0.771

B. The influence of α value on the test result under M=50000

	$M = 50000, \alpha = 0.00033$					
k	10000	20000	30000	50000	60000	90000
Ar_1	0.921	0.924	0.927	0.93	0.931	0.937
Ar_2	0.706	0.722	0.744	0.759	0.762	0.791

C. Optimal Exploration with Fixed M and α Values

Table 3 shows the relationship between the detection accuracy, extended dimension, and number of retained features. From Table $3.A$ and Table $3.B$, we can conclude that, as the number of retained features increases, the detection accuracy of the model also gradually increases. However, Table $3.A$ shows that when $k = 2500$ and $k = 3000$, the detection accuracy is essentially the same. Therefore, there is a bottleneck in the increase in k; that is, when k reaches a certain value, the detection accuracy does not increase, which is in line with the limit under the JL theorem. Table $3.C$ shows the relationship between the detection accuracy and the total length M of the model when the k value is fixed. From Table $3.C$, it can be seen that as M increases, the two detection accuracy Ar_1, Ar_2 are both It is increasing, which shows that the extended dimension can separate the abnormal data hidden in the normal data, thereby improving the detection accuracy of the model. This further proves the argument put forward in the third section of this article that dimensional expansion makes it easier to distinguish between normal and abnormal data.

VI. Conclusion and Future Works

This paper proposes a new real-time anomaly detection model, the FJLT-BF, for streaming data (device logs, content data, sensor environment information). This model can discover the fault(s) or wear of a sensor network in an IoT device in real-time, and thus maintain the stability of the IoT system. The FJLT-BF model differs from existing work, mainly due to our novel algorithm that can effectively combine the advantages of the fast processing of a Bloom filter and the advantages of dimension expansion, enabling the detection model to quickly process large-scale data streams . The FJLT-BF model solves the problem of identifying data anomalies in data streams due to the wear and tear of sensors or other IoT devices. In addition, the FJLT-BF model can identify malicious intrusion data hidden in large-scale data flow. Our proposed model provides a novel perspective for the protection of IoT security systems, and it effectively integrates the theory of dimension expansion, which ensures that our proposed algorithm can more effectively detect abnormal data. We train our Bloom filter model and the required two thresholds on the training data. Then, during the detection process, these two thresholds are utilized to identify anomalies and expand the model to guarantee the model proposed in this paper. The algorithm can effectively maintain the security of the Internet of Things system and protect the security of the system from man-made or accidental damage. In the future, our work will focus on combining additional evaluation metrics and IoT datasets to explore additional metrics to evaluate our anomaly detection algorithm. Our research group also hopes to use the proposed model proposed to better and more effectively combine wireless multimedia sensors, so that it can be applied to IoT systems based on wireless multimedia sensors as well.

References

[1] Alzubi J A. Blockchain-based Lamport Merkle digital signature: authentication tool in IoT healthcare[J]. Computer Communications, 2021, 170: 200-208.

[2] Alzubi O A, Alzubi J A, Shankar K, et al. Blockchain and artificial intelligence enabled privacy-preserving medical data transmission in Internet of Things[J]. Transactions on Emerging Telecommunications Technologies, 2021, 32(12): e4360.

[3] Demirbaga U, Aujla G S. MapChain: A Blockchain-based Verifiable Healthcare Service Management in IoT-based Big Data Ecosystem[J]. IEEE Transactions on Network and Service Management, 2022.

[4] Benmansour F L, Labraoui N. A Comprehensive Review on Swarm Intelligence-Based Routing Protocols in Wireless Multimedia Sensor Networks[J]. International Journal of Wireless Information Networks, 2021, 28(2): 175-198.

[5] Bai F, Liu X Y, Zhang Y L, et al. Research on game model of wireless sensor network intrusion detection[C]//Proceedings of the 2019 International Conference on Embedded Wireless Systems and Networks. 2019: 373-378.

[6] Alzubi J A. Bipolar fully recurrent deep structured neural learning based attack detection for securing industrial sensor networks[J]. Transactions on Emerging Telecommunications Technologies, 2021, 32(7): e4069.

[7] Karim M R, Cochez M, Beyan O D, et al. Mining maximal frequent patterns in transactional databases and dynamic data streams: A spark-based approach[J]. Information Sciences, 2018, 432: 278-300.

[8] D'Alconzo A, Drago I, Morichetta A, et al. A survey on big data for network traffic monitoring and analysis[J]. IEEE Transactions on Network and Service Management, 2019, 16(3): 800-813.

[9] Garg S, Kaur K, Batra S, et al. A multi-stage anomaly detection scheme for augmenting the security in IoT-enabled applications[J]. Future Generation Computer Systems, 2020, 104: 105-118.

[10] Habeeb R A A, Nasaruddin F, Gani A, et al. Real-time big data processing for anomaly detection: A Survey[J]. International Journal of Information Management, 2019, 45: 289-307.

[11] Singh G, Khare N. A survey of intrusion detection from the perspective of intrusion datasets and machine learning techniques[J]. International Journal of Computers and Applications, 2021: 1-11.

[12] Deng F, Song Y, Hu A, et al. Abnormal traffic detection of IoT terminals based on Bloom filter[C]//Proceedings of the ACM Turing Celebration Conference-China. 2019: 1-7.

[13] Hasheminejad S M H, Salimi Z. FDiBC: a novel fraud detection method in bank club based on sliding time and scores window[J]. Journal of AI and Data Mining, 2018, 6(1): 219-231.

[14] Shao W, Xiao R, Huang J, et al. FJLT-FLSH: More Efficient Fly Locality-Sensitive Hashing Algorithm via FJLT for WMSN IoT Search[J]. IEEE Internet of Things Journal, 2019, 6(4): 7122-7136.

[15] Chandola V, Banerjee A, Kumar V. Outlier detection: A survey[J]. ACM Computing Surveys, 2007, 14: 15.

[16] Zarpelão B B, Miani R S, Kawakani C T, et al. A survey of intrusion detection in Internet of Things[J]. Journal of Network and Computer Applications, 2017, 84: 25-37.

[17] Kobayashi S, Otomo K, Fukuda K, et al. Mining causality of network events in log data[J]. IEEE Transactions on Network and Service Management, 2017, 15(1): 53-67.

[18] Cauteruccio, Francesco, et al. "Short-long term anomaly detection in wireless sensor networks based on machine learning and multi-parameterized edit distance." Information Fusion 52 (2019): 13-30.

[19] Qu G, Wu H, Li R, et al. DMRO: A deep meta reinforcement learning-based task offloading framework for edge-cloud computing[J]. IEEE Transactions on Network and Service Management, 2021, 18(3): 3448-3459.

[20] Sun C C, Cardenas D J S, Hahn A, et al. Intrusion detection for cybersecurity of smart meters[J]. IEEE Transactions on Smart Grid, 2020, 12(1): 612-622.

[21] Cui, Lei, et al. "Security and privacy-enhanced federated learning for anomaly detection in iot infrastructures." IEEE Transactions on Industrial Informatics 18.5 (2021): 3492-3500.

[22] Niu, Yichun, et al. "Distributed intermittent fault detection for linear stochastic systems over sensor network." IEEE Transactions on Cybernetics (2021).

[23] Sun K, Qiu W, Yao W, et al. Frequency injection based HVDC attack-defense control via squeeze-excitation double CNN[J]. IEEE Transactions on Power Systems, 2021, 36(6): 5305-5316.

[24] Xiong Z, Cai Z, Takabi D, et al. Privacy threat and defense for federated learning with non-iid data in AIoT[J]. IEEE Transactions on Industrial Informatics, 2021, 18(2): 1310-1321.

[25] Groza, Bogdan, and Pal-Stefan Murvay. "Efficient intrusion detection with bloom filtering in controller area networks."IEEE Transactions on Information Forensics and Security 14.4 (2018): 1037-1051.

[26] Deng, Fengjie, et al. "Abnormal traffic detection of IoT terminals based on Bloom filter."Proceedings of the ACM Turing Celebration Conference-China. 2019.

[27] Zhan, Teng, and Shiping Chen. "An improved hash algorithm for monitoring network traffic in the internet of things." Cluster Computing (2022): 1-16.

[28] Dasgupta, Sanjoy, Charles F. Stevens, and Saket Navlakha. "A neural algorithm for a fundamental computing problem." Science 358.6364 (2017): 793-796.

[29] Bloom, Burton H . Space/time trade-offs in hash coding with allowable errors[J]. Communications of the ACM, 1970, 13(7):422-426.

[30] Sinha K, Ram P. Fruit-fly Inspired Neighborhood Encoding for Classification[C]//Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. 2021: 1470-1480.

[31] Byun H, Lim H. Learned FBF: Learning-Based Functional Bloom Filter for Key-Value Storage[J]. IEEE Transactions on Computers, 2021.

[32] Johnson W B, Lindenstrauss J. Extensions of Lipschitz mappings into a Hilbert space[J]. Contemporary mathematics, 1984, 26(189-206): 1.

[33] P. Indyk and R. Motwani, "Approximate nearest neighbors: Towards removing the curse of dimensionality," in Proc. 30th Annu. ACM Symp.Theory Comput., May 1998, pp. 604–613.

[34] X. Gu, G. Dong, X. Zhang, L. Lan and Z. Luo, ”Towards Making Unsupervised Graph Hashing Robust,” 2020 IEEE International Conference on Multimedia and Expo (ICME), 2020, pp. 1-6, doi: 10.1109/ICME46284.2020.9102845.

[35] Li D, Zhang W, Shen S, et al. SES-LSH: shuffle-efficient locality sensitive hashing for distributed similarity search[C]//2017 IEEE International Conference on Web Services (ICWS). IEEE, 2017: 822-827.

[36] Xu K, Qiao Y. Randomized sampling-based fly local sensitive hashing[C]//2018 25th IEEE International Conference on Image Processing (ICIP). IEEE, 2018: 1293-1297.

[37] Frankl P , Maehara H . The Johnson-Lindenstrauss lemma and the sphericity of some graphs[J]. Journal of Combinatorial Theory, Series B, 1988, 44(3):355-362.

[38] Dasgupta S, Gupta A. An elementary proof of a theorem of Johnson and Lindenstrauss[J]. Random Structures & Algorithms, 2003, 22(1):60-65.

[39] Liu, Fanghui, et al. ”Random features for kernel approximation: A survey on algorithms, theory, and beyond.” IEEE Transactions on Pattern Analysis and Machine Intelligence 44.10 (2021): 7128-7148.

[40] Alzubi O A, Alzubi J A, Al-Zoubi A M, et al. An efficient malware detection approach with feature weighting based on Harris Hawks optimization[J]. Cluster Computing, 2022, 25(4): 2369-2387.

[41] Nie F, Zhu W, Li X. Decision Tree SVM: An extension of linear SVM for non-linear classification[J]. Neurocomputing, 2020, 401: 153-159.

[42] Ailon N, Chazelle B. Approximate nearest neighbors and the fast Johnson-Lindenstrauss transform[C]// Thirty-Eighth ACM Symposium on Theory of Computing. ACM, 2006:557-563.

[43] Singhal A, Buckley C, Mitra M. Pivoted document length normalization[C]//ACM SIGIR Forum. New York, NY, USA: ACM, 2017, 51(2): 176-184.

[44] Thilakaratne M, Falkner K, Atapattu T. A Systematic Review on Literature-based Discovery: General Overview, Methodology, & Statistical Analysis[J]. ACM Computing Surveys (CSUR), 2019, 52(6): 1-34.

[45] Tian, Ye, et al. ”Efficient large-scale multiobjective optimization based on a competitive swarm optimizer.” IEEE Transactions on Cybernetics 50.8 (2019): 3696-3708.

[46] Kita, Derek M., et al. ”High-performance and scalable on-chip digital Fourier transform spectroscopy.” Nature communications 9.1 (2018): 1-7.

[47] Sosa-Gómez, Guillermo, Omar Rojas, and Octavio Páez-Osuna. ”Using Hadamard transform for cryptanalysis of pseudo-random generators in stream ciphers.” EAI Endorsed Transactions on Energy Web 7.27 (2020): e1-e1.

[48] Sabhnani M, Serpen G. Application of Machine Learning Algorithms to KDD Intrusion Detection Dataset within Misuse Detection Context[C]//MLMTA. 2003: 209-215.

[49] Elkan C. Results of the KDD’99 classifier learning[J]. Acm Sigkdd Explorations Newsletter, 2000, 1(2): 63-64.

[50] Dal Pozzolo A, Caelen O, Johnson R A, et al. Calibrating probability with undersampling for unbalanced classification[C]//2015 IEEE Symposium Series on Computational Intelligence. IEEE, 2015: 159-166.

[51] Liu F T, Ting K M, Zhou Z H. Isolation-based anomaly detection[J]. ACM Transactions on Knowledge Discovery from Data (TKDD), 2012, 6(1): 1-39.

[52] Liu F T, Ting K M, Zhou Z H. Isolation forest[C]//2008 eighth ieee international conference on data mining. IEEE, 2008: 413-422.

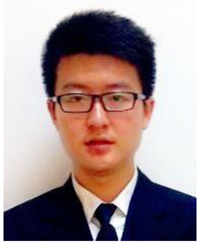
[53] Lalis J T, Gerardo B D, Byun Y. An adaptive stopping criterion for backpropagation learning in feedforward neural network[J]. International Journal of Multimedia and Ubiquitous Engineering, 2014, 9(8): 149-156.

[54] Chen, Yewang, et al. ”Fast density peak clustering for large scale data based on kNN.” Knowledge-Based Systems 187 (2020): 104824.

Wenhao Shao Doctoral student at the College of Computer, National University of Defense Technology, currently co-training at the Institut Polytechnique de Paris, Telecom-Sudparis. Master’s degree in 2020. His current research directions are machine learning, data mining, anomaly detection, application scenario mechanism in multi-modal anomaly detection in the Internet of Things and video anomaly detection in traffic scenarios.

Yanyan Wei Graduated from Henan University with a master’s degree in statistics in 2020. she is currently a lecturer at Zhengzhou University of Finance and Economics, teaching statistics. Proficient in R language, SPASS, research direction is information fraud in financial data.

Praboda Rajapaksha is a Research Fellow at the Institut Polytechnique de Paris, France. She holds a Ph.D. degree in Computer Science from the Institut Polytechnique de Paris, France (2021). She has more than eleven years’ experience as a Senior Lecturer and Assistant Professor in higher education sectors in different countries. Her primary research interests include NLP, ML and Deep Learning. She has collaborated in several European (ITEA, BPI, HE) and French national projects (FUI).



Dun Li received the B.S. degree in Human Resource Management from the Huaqiao University, Quanzhou, China, in 2013, and the M.S. degree in Finance from the Macau University of Science and Technology, Macau, China, in 2015. He is currently doing his Ph.D. degree in Information Management and Information Systems at Shanghai Maritime University. His main research interests include smart finance, big data, machine learning, IoT, and blockchain.

Zhigang luo Professor of National University of Defense Technology,college of computer, executive director of Hunan Artificial Intelligence Association, research direction is bioinformatics, parallel computing, artificial intelligence. Published more than 100 papers.

Noel Crespi (Senior Member, IEEE) joined Institut Mines-Telecom, Institut Polytechnique de Paris in 2002 and is currently Professor and MSc Programme Director, leading the Data Intelligence and Communication Engineering laboratory (DICE). He has played a key role in standardisation as a delegate in a number of committees and as the editor for CAMEL, the Intelligent Network standard for mobile networks. He was appointed as the coordinator for France Telecom’s standardisation activities for Core Network and then for all GSM/UMTS standards at the ETSI plenary committee. He is also an affiliate professor/researcher at KAIST (South Korea), Concordia University (Canada), and University of Goettingen (Germany). He is the scientific director the French-Korean laboratory ILLUMINE. His current research interests are in AI, Data Analytics, Internet of Things and Softwarisation.