


RESEARCH ARTICLE

An improved X-means and isolation forest based methodology for network traffic anomaly detection

Yifan Feng¹ , Weihong Cai^{1*}, Haoyu Yue¹, Jianlong Xu¹, Yan Lin², Jiaxin Chen¹, Zijun Hu¹

1 College of Engineering, Shantou University, ShanTou, Guangdong, China, **2** Sangfor Technologies Incorporation, ShenZhen, Guangdong, China

* whcai@stu.edu.cn



Abstract

Anomaly detection in network traffic is becoming a challenging task due to the complexity of large-scale networks and the proliferation of various social network applications. In the actual industrial environment, only recently obtained unlabelled data can be used as the training set. The accuracy of the abnormal ratio in the training set as prior knowledge has a great influence on the performance of the commonly used unsupervised algorithms. In this study, an anomaly detection algorithm based on X-means and iForest is proposed, named X-iForest, which clusters the standard Euclidean distance between the abnormal points and the normal cluster centre to achieve secondary filtering by using X-means. We compared X-iForest with seven mainstream unsupervised algorithms in terms of the AUC and anomaly detection rates. A large number of experiments showed that X-iForest has notable advantages over other algorithms and can be well applied to anomaly detection of large-scale network traffic data.

OPEN ACCESS

Citation: Feng Y, Cai W, Yue H, Xu J, Lin Y, Chen J, et al. (2022) An improved X-means and isolation forest based methodology for network traffic anomaly detection. PLoS ONE 17(1): e0263423. <https://doi.org/10.1371/journal.pone.0263423>

Editor: Zhihong (Arry) Yao, Southwest Jiaotong University, CHINA

Received: October 27, 2021

Accepted: January 18, 2022

Published: January 31, 2022

Copyright: © 2022 Feng et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Data Availability Statement: All publicly available datasets (Shuttle, Mulcross, Satellite, BreastW) are available from the Outlier Detection DataSets (ODDS) ([url:http://odds.cs.stonybrook.edu/](http://odds.cs.stonybrook.edu/)). The simulation datasets are available from <https://github.com/Busa-yifan/Network-Anomaly-Detection-Simulation-Dataset>.

Funding: Grant numbers: 2016B010124012; Weihong Cai received award; Full name of funder: Science and Technology Planning Project of Guangdong Province; Grantees played a role in study design, data collection and analysis,

Introduction

In recent years, the network environment has become increasingly complex. Traffic data have exploded and mass infrastructure based on internet of things (IoT) technology and complex networks has had a significant impact on society and the economy [1–3]. Due to the increase in Internet services, network abnormalities including malicious attacks and poor quality of network services have become more diversified and substantially affecting the operation of web services and causing social and economic losses [4–7].

An abnormality is a pattern of data that does not conform to a clearly defined normal behaviour [8]. The cause of the abnormality may be equipment failure or malicious intrusion. To date, anomaly detection as a theme has already been applied in various surveys, review articles and books [9, 10], including the field of network traffic data anomaly detection [11]. It is very difficult to detect abnormal patterns in network traffic data because the types of services provided by the network and the user connection patterns are different, which means that the network traffic has different characteristics and that the pattern distribution is very irregular

publication decisions and manuscript preparation. Grant numbers: 2019B010116001; Weihong Cai received award; Full name of funder: Science and Technology Planning Project of Guangdong Province; Grantees played a role in study design, data collection and analysis, publication decisions and manuscript preparation.

Competing interests: The authors have declared that no competing interests exist.

[12]. In addition, threshold-based anomaly detection methods are more commonly used in the enterprise. There are other cutting-edge unsupervised anomaly detection algorithms, such as a local outlier factor [13] (LOF) and the histogram-based outlier score [14] (HBOS). However, there are currently few methods that can be stable and efficient in the application scenarios of network traffic detection. Therefore, an anomaly detection algorithm that can be applied to high-dimensional large-scale unlabelled data and maintain robustness and high accuracy in a complex and changing network environment is urgently needed.

Isolation forest (iForest) currently have many applications in industry. For example, in the field of semiconductor manufacturing, the high-dimensional and massive characteristics of optical emission spectroscopy (OES) data limit the achievable performance of anomaly detection systems. Puggini and McLoone [15] presented dimensionality-reducing variable selection and iForest-based anomaly detection to solve this problem. The iForest-based method has also been used in studies to detect abnormal situations in the etching process in semiconductor manufacturing and in smart grids, and the effectiveness of the algorithm has been proven through actual industrial data [16, 17]. This paper selects the iForest algorithm [18], which is efficient for massive data, as a basis. iForest can maintain relatively stable detection accuracy in massive data, and its advantages are a short training time and fast detection speed, making it suitable for anomaly detection in many scenarios with massive amounts of data.

However, iForest determines whether a sample is an outlier by calculating the anomaly score of the sample data. The anomaly ratio largely influences the calculation of the anomaly score, so iForest relies heavily on the setting of the anomaly ratio. In the actual industrial network environment, we can rely only on manual experience to set this ratio [19], which means that there is no guarantee that the iForest can achieve the best performance. Inaccurate parameter settings can also lead to low accuracy and high false detection rates in iForest. This paper proposes a solution to this problem, with the following key contributions:

- We propose a new X-iForest method based on iForest, using standard Euclidean distances and upper quartile method to quantify anomaly data.
- We combine X-means [20] with iForest to cluster the standard Euclidean distance values from the data to the cluster centre and effectively classify the data by distance.
- We use a multi-dimensional Gaussian distribution to simulate real network traffic data. Extensive experiments are conducted on 4 simulated datasets and 4 real-world datasets, and the results are compared with those of 6 other cutting-edge unsupervised algorithms.
- Our experimental results show that the proposed X-iForest has significant advantages in the area under the ROC curve (AUC) and anomaly detection rate (ADR) without the need for an accurate anomaly ratio as prior knowledge.

X-iForest maintains the efficiency of the iForest algorithm in high-dimensional big data training and detection processes while greatly improving the AUC and ADR. When presented with large unsupervised data with no accurate anomaly ratio and large changes in data distribution, the high performance, high accuracy and robustness demonstrated by X-iForest show that the algorithm is suitable for real and complex network traffic anomaly detection scenarios.

This paper is organized as follows: The second part reviews several network traffic anomaly detection methods commonly used in the industrial field. The third part explains the theoretical framework and architecture design of our method. The fourth part evaluates and compares the performance of our proposed method with other methods in the anomaly detection. The fifth part presents our summary and future prospects.

Related work

Following extensive investigations of actual industrial applications and recently published articles in the field of network health analysis and network traffic anomaly detection, the main methods can be classified as follows.

Threshold-based approaches

At present, the most widely used methods for network health analysis in the industry are based on active detection, such as ping detection and http detection. On the basis of active detection, a fixed parameter is set based on the experience of operation and maintenance personnel, such as determining a threshold according to a reasonable range of server delay times and then detecting abnormalities in network traffic data based on the threshold. This detection method does not incur excessive computational cost but cannot be flexibly applied to varying network traffic scenarios, especially for the accurate detection of outliers close to the normal distribution, which are difficult to identify accurately and which we define as swing points.

Statistical-based approaches

Anomaly detection algorithms based on statistics and probability models mainly make assumptions about the distribution of data and determine the “anomalies” defined under the assumptions, so extreme value analysis or hypothesis testing is often used. For example, they might assume a Gaussian distribution for the simplest one-dimensional data and then consider the data whose distance lies outside a specific range as anomalous points. After generalizing to high dimensions, one can assume that each dimension is independent and add the abnormality of each dimension. The most representative algorithm is HBOS. Paulauskas and Baskys [21] used the HBOS to detect anomalies in a computer network concluded that the HBOS with dynamic bins showed better results than other approaches in detecting rare events. The HBOS algorithm, which is based on the assumption of independence of each dimension of multi-dimensional data, divides n partitions in each dimension. The outlier corresponding to each interval depends on the density. The higher the density is, the lower the value. If the statistical assumptions are true, then this method is simple and very effective. However, this type of statistical method relies heavily on the distribution assumption of the data set, which also causes the robustness of this type of algorithm in the real data set to be relatively poor.

Distance-based approaches

The distance-based method is generally regarded as the basic method of outlier detection research. The distribution of abnormal points is different from that of normal points, so a series of algorithms are derived to identify abnormal points through distance-representing similarity. Wang et al. [22] proposed a distance-based proximity detection method using the K-nearest neighbour (KNN) algorithm to analyse and detect the abnormal flow data in a wireless sensor network (WSN). The KNN classifier is used for anomaly detection. For parameters k and n , the k-nearest neighbour distance is calculated for each point. The top n points are taken as anomalies by sorting the k-nearest neighbour distances in descending order. They implemented and tested the QualNet simulation platform and proved the effectiveness of the proposed KNN algorithm through simulation results. The principal component analysis (PCA)-based anomaly detection algorithm can also be regarded as a distance-based method. One method is to find k feature vectors and then calculate the weighted Euclidean distance from each sample to the hyperspace formed by these feature vectors and use it as the sample abnormality. Hoang and Nguyen [23] investigated the application scenarios of PCA algorithm

and proposed a new general formula for distance calculation and a PCA-based IoT detection method. They verified the feasibility of their proposed method through a variety of experiments. Distance-based approaches incur a very high computational cost for massive datasets, with loss of performance when applied to network traffic anomaly detection.

Density-based approaches

These approaches introduce the concept of LOF, in which each instance is assigned a score based on the neighbours' local density denoting a degree of outlierness. A potential outlier is identified by the relatively high LOF value. Based on this main idea, some extended models have been proposed. Gan and Zhou [24] combined the LOF with the Density-based spatial clustering of applications with noise (DBSCAN) algorithm to realize the adaptive dynamic adjustment of parameters to changing data. The method was used to optimise the accuracy of network traffic scenarios. The experimental results show that the method based on the improved LOF algorithm has high practical application value. The cluster-based local outliers factor (CBLOF) algorithm uses K-means to pre-classify the data before performing the LOF algorithm [25] and then uses the LOF algorithm separately for the data in each cluster. The problem with this type of method is the same as that of the distance-based method: a considerable amount of calculation when facing large, high-dimensional data. Moreover, density-based anomaly detection algorithms require relatively extensive prior knowledge and experience in the selection of parameters, which also limits the application of such algorithms to fast-changing network traffic anomaly detection.

Neural network-based approaches

In network traffic anomaly detection tasks, network traffic datasets are often massive and complex. In recent years, neural network-related algorithms have been proven to be well applied in complex scenarios and have very good performance [26]. Kim and Cho [27] proposed the C-LSTM method to extract more complex features by combining a coevolutionary neural network (CNN), long short-term memory (LSTM) and deep neural network (DNN) and verified in experiments that its performance is better than that of other state-of-the-art machine learning techniques. Wei and Wang [28] combined a convolutional neural network (CNN) and recurrent neural network (RNN), proposed a network anomaly detection method based on hierarchical spatiotemporal feature learning (HAST-NAD), and proved that the time series characteristics algorithm performs better than the spatial characteristics algorithm. Although deep learning has shown excellent performance in many tasks, it relies heavily on high-quality labelled data. In the actual industrial scenario of network traffic anomaly detection, it is often difficult to obtain a large amount of labelled data, which means that the use of neural network-based methods face the problem of cold starting. Moreover, network traffic data have different distributions on different servers, different applications, and different times; thus, generalization is a severe problem faced by neural-network-based methods.

Isolation forest-based approaches

Since the data in network health analysis and network traffic anomaly detection scenarios often involve large data volume, high-dimensional data and a small proportion of anomalies, iForest are very suitable for network traffic anomaly detection. Compared with other algorithms, iForest can maintain higher detection performance and accuracy with massive data. iForest is used as an ensemble learning method, which contains multiple isolation trees. Each tree can be regarded as a weak classifier. The training set used for each weak classifier is obtained by random sampling from the full dataset using the bootstrap method. The final

result is obtained by combining the results of all the isolation trees, which means that the iForest has a good generalization performance and can largely prevent overfitting. Hariri et al. proposed an extended isolated forest algorithm (EiForest) [29] that allows the branching hyperplanes to take on any slope as opposed to hyperplanes only parallel to the coordinate frame. EiForest addresses the impact of bias introduced in the standard isolated forest case on the anomalous scores for a given data point. The experimental results show that the EiForest possesses stronger robustness. Although the EiForest solves the problem caused by bias, it does not take into account the effect of the abnormal ratio on the detection results. Ding and Fei [30] used a sliding window frame and proposed an adaptive streaming data anomaly detection algorithm iForestASD based on iForest, which can be used to detect network traffic data generated in applications such as computer networks and sensor networks, and used experiments to prove that their proposed algorithm can effectively detect anomalies in the data stream. Pugini and McLoone proposed an anomaly detection method based on reduced dimensional variable selection and isolated forests [15]. The method is used to solve the problem that the high dimensionality and correlation of optical emission spectral data limit the performance of anomaly detection systems, and it is more interpretable compared to methods such as PCA. Wang et al. used the isolated forest algorithm for aero-engine fault detection [31], and the experimental results showed that the method has higher detection accuracy and shorter running time. Alsini et al. combined the local outlier factor (LOF) algorithm with isolated forests to solve the limitations of LOF in evaluating concrete mixtures [32], and experimental results demonstrated that the method was more effective in detecting anomalous sequences. Chen et al. combined Gaussian mixture model (GMM) with isolated forest for anomaly detection and identification of key behavioral attributes in continuous state monitoring (CM) data [33], and experimentally demonstrated that the method is more effective for high-dimensional data compared with other methods. Zhou et al. used the isolated forest algorithm to predict the final welding quality [34], and the experimental results showed that the isolated forest possessed better classification performance. Wang et al. proposed a general separation method based on linear prediction analysis and isolated forest [35], which separate multi-source partial discharge signals and distinguish various partial discharge signals. The experimental results confirmed that the method can effectively separate and distinguish various partial discharge signals. Ahmed et al. use isolated forests to detect covert data integrity assault (CDIA) utilizing non-labeled data in smart grid (SG) communication networks [17], and simulation results show that the method can handle non-labeled historical measurement datasets well and significantly improve attack detection accuracy. Xing et al. used isolated forest to identify the interest flooding attack [36], and the simulation results showed that the method has high attack detection accuracy and fast detection speed.

However, the unavailability of sufficient high-quality data and the requirement for a priori knowledge make these iForest-based methods still difficult to apply to real network traffic monitoring scenarios. We introduce X-Means and standard Euclidean distances to quantify the anomalies of the data for this problem, and address the problem of not having sufficient a priori knowledge and labeled data in realistic anomalous traffic monitoring.

Materials and methods

This section describes our approach and the algorithmic tools used. The proposed X-iForest is composed of iForest and X-means. After obtaining the normal clustering centres by iForest, the standard Euclidean distance from the data points to the normal clustering centres was used to quantify the degree of abnormality. The degree of abnormality was clustered using X-means

after filtering the extreme values by the upper quartile method. Finally the 2-means method was used as a classifier to classify the data into normal and abnormal classes.

X-means

X-means clustering is an improved algorithm based on K-means [20]. It is used to solve the main problem of K-means clustering, which requires prior knowledge about the number of clusters. In this method, the number of clusters k is estimated in an unsupervised manner based on the data set itself, using k_{\max} and k_{\min} respectively as upper and lower limits for the possible values of x . In the first step of X-means grouping, X-means performs a clustering where $x = k_{\min}$. In the next step, each cluster is regarded as an initial parent category, and a calculation is performed on each parent category which calculates the *BIC* scores before and after classification and compares them to decide whether to divide the parent class into two groups. The *BIC* score, which helps determine the best representation for sample data, is defined as:

$$BIC(\phi) = \hat{I}_{\phi}(D) - \frac{P_{\phi}}{2} \cdot \log R \quad (1)$$

where ϕ represents the model and $\hat{I}_{\phi}(D)$ is the log-likelihood of the data according to the ϕ model, and taken at the maximum likelihood point. P_{ϕ} is the number of parameters in model ϕ . For example, model ϕ_2 is better than model ϕ_1 if $BIC(\phi_2) > BIC(\phi_1)$.

In this way, clusters provide an accurate distribution of samples. As a result, the algorithm either replaces the parent generation with the centroid or maintains the centroid and keeps it. Then, the structure is continuously updated according to each choice until the estimated number of clusters reaches x_{\max} or has converged to the best structure.

Isolation forest algorithm

Algorithm 1 iTree(D, h, l)

Input: $D = (x_1, x_2, \dots, x_n)$ —dataset, h —the height of tree, l —height limit

Output: an iTree t

Initialize: $t = \emptyset$

```

1: if  $h \geq l$  or  $Size(D) \leq 1$  then
2:   return  $t$ 
3: else
4:   randomly select  $q_i$  a feature of  $D$ 
5:   randomly select a split point  $p \in (\min(q_i), \max(q_i))$ 
6:    $D_l \leftarrow \text{filter}(D, q_i < p)$ 
7:    $D_r \leftarrow \text{filter}(D, q_i \geq p)$ 
8:   repeat iTree( $D_l, h + 1, l$ ) and link the obtained tree as the left
   tree of  $t$ 
9:   repeat iTree( $D_r, h + 1, l$ ) and link the obtained tree as the
   right tree of  $t$ 
10: end if

```

The iForest was proposed by Liu et al. [18]. It has many applications in the industrial field and involves an ensemble of isolation trees, similar to random forests and decision trees. By definition, an iForest is composed of a certain number of isolation trees:

$$IF = \{t_1, \dots, t_T\} \quad (2)$$

where iTree is obtained according to the flow of Algorithm 1. The path length $h(x)$ of the sample x is the number of edges passed from the root node of the isolation tree to the leaf node.

The average number of steps required to isolate a sample x in a forest is then

$$h(x) = \frac{1}{T} \sum_{t \in IF} h_t(x) \quad (3)$$

Intuitively, clusters with high density need to be cut many times to be isolated, but those with low density can easily be isolated. Under this random segmentation strategy, abnormal points usually have shorter paths. Let n be the number of samples in the data set, $c(n)$ be the average path length of the tree, and $c(n)$ be defined as:

$$c(n) = 2H(n-1) - \frac{2(n-1)}{2n} \quad (4)$$

where $H(i)$ is the harmonic number estimated as:

$$H(i) \approx \ln(i) + 0.5772156649 \quad (5)$$

The normalized anomaly score $s(x, n)$ of sample is defined as:

$$s(x, n) = 2^{\frac{h(x)}{c(n)}} \quad (6)$$

If instances return s very close to 1, then they are definitely anomalies, if instances have s much smaller than 0.5, then they are quite safe to be regarded as normal instances, and if all the instances return $s \approx 0.5$, then the entire sample does not really have any distinct anomaly.

X-iForest: Improved isolation forest based on X-means

Although iForest are more suitable for massive unlabelled data than other algorithms to a certain extent, similar to other unsupervised algorithms, the performance of the algorithm is very dependent on the settings of the abnormal ratio. The actual network conditions are very complicated, causing the definition and data distribution to change very quickly, so it is necessary to always use the newly captured unlabelled data as the training set to update the model. In this case, we can rely only on manual experience to determine the value of the abnormal ratio, but an inaccurate abnormal ratio would destabilize the performance of the iForest, resulting in low accuracy and a high false detection rate. In this study, the X-means clustering algorithm is used to improve the iForest to improve the performance of algorithm anomaly detection when the real anomaly ratio is unknown, even exceeding the performance of the original iForest algorithm with an accurate abnormal ratio.

According to the actual situation in industry, we treat the dataset as a completely unlabelled dataset; that is, the training set has no labels, and the abnormal ratio is unknown. Our algorithm is described as follows:

Algorithm 2 X-iForest Algorithm

Input: $D = (x_1, x_2, \dots, x_n)$ —dataset, k_{max} —the upper limit of the number of clusters in X-means
Output: the index list of normal values $Nlist$, the index list of abnormal values $Alist$
Initialize: iForest, $Nlist = \emptyset$, $Alist = \emptyset$, distance list $Dlist = \emptyset$
 1: set the contamination parameter of iForest to 2 times the background knowledge of the relevant field
 2: get preliminary classification result $L = (l_1, l_2, \dots, l_n)$ of D with iForest
 3: **for** each label l_i in L **do**
 4: **if** $l_i = 1$ **then**
 5: $Nlist$ append x_i
 6: **else**
 7: $Alist$ append x_i

```

8:   end if
9: end for
10: calculate the cluster center  $C$  of  $Nlist$ 
11: for each sample  $s_i$  in  $Alist$  do
12:   calculate the standard Euclidean distance  $d_i$  between  $s_i$  and  $C$ 
13:    $Dlist$  append  $d_i$ 
14: end for
15:  $MAX \leftarrow \text{box plot } (Dlist)$ 
16:  $Dlist \leftarrow \text{filter } (Dlist, d_i < MAX)$ 
17: perform X-means clustering on  $Dlist$  and calculate the cluster
    center of each cluster to get the result  $X = \{xclu_1, xclu_2, \dots, xclu_k\}$ , where  $k \leq k_{max}$ 
18: perform K-means clustering with  $k = 2$  on  $X$  and get the result
     $K = (kclu_1, kclu_2)$ , where the center of  $kclu_1 < kclu_2$ 
19: for each sample  $s_i$  in  $kclu_1$  do
20:    $Nlist$  append  $s_i$ 
21:    $Alist$  remove  $s_i$ 
22: end for
23: return  $Nlist, Alist$ 

```

For other algorithms that require a priori knowledge, they need to learn the true abnormal ratios of many datasets to do a priori fitting. The closer the regression results are to the true ratios the more effective these algorithms will be, and once they encounter a little noise (which is extremely common and unavoidable in reality) the algorithms will be much less effective or even ineffective. Such algorithms, which rely on a priori knowledge, are not stable in reality. Our proposed algorithms can perform well as long as they have basic background knowledge of the scenario in a real-world application. For example, in network traffic anomaly detection, the usual anomaly rate in this domain is about 0.02 to 0.05, and we simply need to set the parameters higher than that to achieve excellent detection results. In contrast, we prefer to call our algorithm background knowledge-based rather than a priori knowledge-based.

First, iForest are used for preliminary detection, and the contamination parameter is set to 2 times the background knowledge of the relevant field. This is to detect as many outliers as possible. In actual situations, the cost of misclassifying an abnormal value as a normal value is higher than the misvaluation of a normal value as an outlier. The first task should be to detect as many outliers as possible. Since our estimated anomaly ratio is higher than the true ratio, there may be cases where the normal value is misclassified as an outlier after the initial detection. We need to minimize misclassification as much as possible while maximizing the abnormal detection rate. This part is reflected in lines 1 to 8 in Algorithm 2.

Second, we calculate the cluster centre of this part of the normal data in the preliminary detection and then calculate the standard Euclidean distance from the suspected abnormal value to the normal cluster centre. The standardized Euclidean distance is an improvement scheme intended to address the shortcomings of the simple Euclidean distance [37], mainly for data with large differences in data distribution in each dimension.

Let $\mathbf{a} = (x_{11}, x_{12}, \dots, x_{1n})$ and $\mathbf{b} = (x_{21}, x_{22}, \dots, x_{2n})$ be the observed data, where s_n is the standard deviation and where the standardized Euclidean distance between \mathbf{a} and \mathbf{b} is:

$$d = \sqrt{\sum_{k=1}^n \left(\frac{x_{1k} - x_{2k}}{s_k} \right)^2} \quad (7)$$

The abnormal ratio assumed in the preliminary calculation is higher than the true ratio, so the data evaluated to be normal after the initial detection are the data whose distribution in the data set most conform to the normal definition and do not include swing points. The closer the distance to the cluster centre is, the closer the data are to the normal data distribution, and

the farther the distance is, the higher the abnormality of the data. This step is shown in lines 9 to 13 in Algorithm 2.

Third, extreme values in the distance value are filtered out through box plot [38]. Let Q_1 be the upper quartile and Q_3 be the lower quartile. The *MAX* and *MIN* observations can be defined as:

$$MAX = Q_1 - 1.5 \times (Q_3 - Q_1) \quad (8)$$

$$MIN = Q_3 + 1.5 \times (Q_3 - Q_1) \quad (9)$$

Extreme values affect subsequent clustering by causing the overall characteristics of the data to shift. Therefore, in this step, we filter out extreme values whose distance value is greater than the *MAX* observation value, and classify these points directly as anomaly values. Since the Euclidean distance calculation result is not less than 0, we do not consider the case of values less than the *MIN* observation value. This part is reflected in line 14 and line 15 in Algorithm 2.

Fourth, the distance value calculated in the second step is used for X-means clustering with $k_{min} = 2$. This setting is used because this part of the data contains misdetected normal values and real abnormal values, so these data should be divided into at least 2 clusters to prevent the data distribution from being too uniform and causing X-means to collect the data into 1 cluster. Line 16 in Algorithm 2 are this part.

Finally, we calculate the standard Euclidean distance from the cluster centre of each cluster obtained by X-means clustering to the normal cluster centre. Then, K-means clustering with $k = 2$ is performed on these distance values, and all clusters are divided into normal and abnormal categories. This step is shown in lines 17 to 22 in Algorithm 2.

To show the process of X-iForest more intuitively, we determine the approximate range of normal and abnormal values based on the distribution of actual network traffic data and randomly generated a simulation data set within the distribution range to demonstrate the steps of the algorithm. We simulated normal data $X = (x_1, \dots, x_n) \in \mathbb{R}^{n \times 6}$ with $n = 500$ samples and the abnormal data $Y = (y_1, \dots, y_n) \in \mathbb{R}^{n \times 6}$ with $n = 30$ samples. For each feature, the elements in X and Y are independent random samples of normal distribution and abnormal distribution, respectively. The data set has a total of 530 samples, each of which has 6 dimensions. The results after X-means clustering are shown in Fig 1(a). To make the results more intuitive, we use PCA for dimensionality reduction [39]. The triangle represents the centre of a cluster, and the point with the same colour as the triangle represents the data grouped into that category. The blue dot represents the normal data detected in the first step. The marked numbers represent the distance from the centre of each cluster to the centre of the normal cluster. The abnormal data are divided into 4 clusters by X-means. The smaller the standard Euclidean distance from the centre of the normal cluster is, the closer the cluster is to the normal data distribution.

The result of K-means clustering is shown in Fig 1(b); the smaller type is shown in green, which represents that the points contained in the clusters of this type are close to the normal data distribution of the preliminary detection, so these points are considered to be normal values that were misevaluated as abnormal and reclassified into the normal class. The data contained in the other cluster are shown in red, representing classification as anomaly data. Fig 1(c) shows the final anomaly detection result.

Experiments

Evaluation metric

The metrics for evaluating outlier detection are the *ADR* and *AUC*. The *ADR* refers to the ratio of detected abnormal values to all abnormal values. In industrial applications, the cost of

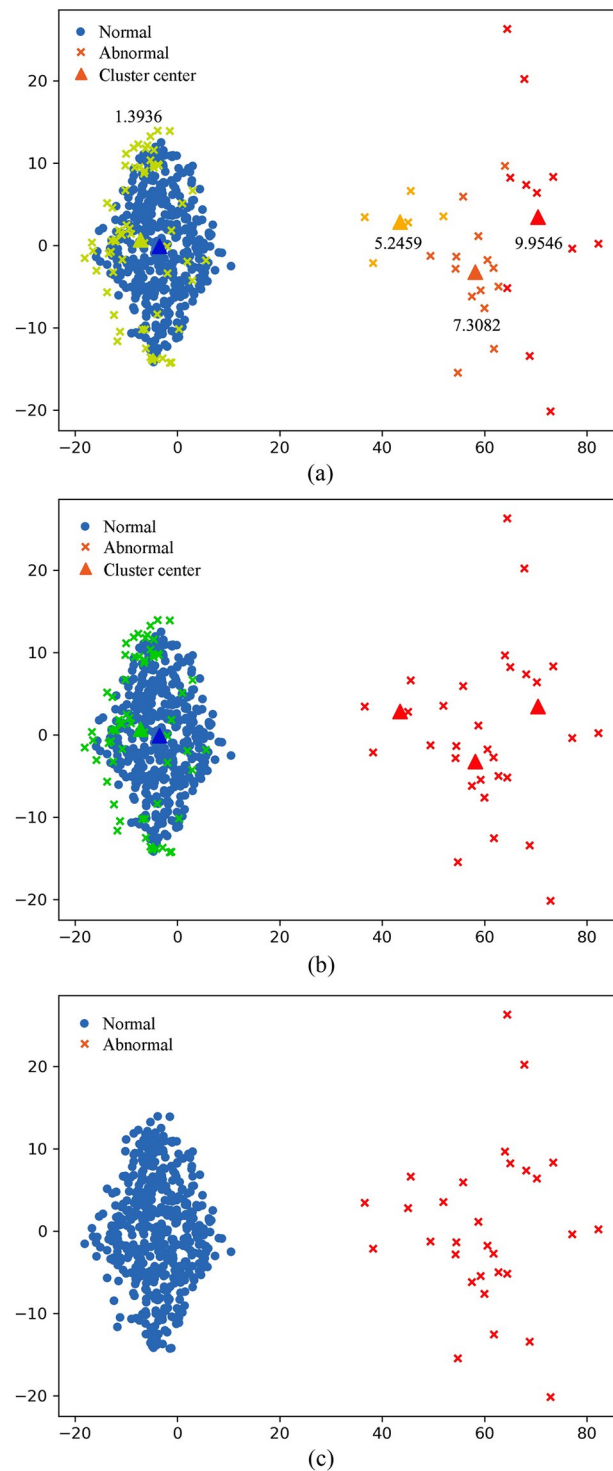


Fig 1. The result of X-means clustering on standard Euclidean distance from the abnormal cluster centers to the normal cluster center. (a): The result of K-Means clustering. (b): The final anomaly detection result of the generated data bu using X-iForest. (c): Demonstration of X-iForest on a test dataset.

<https://doi.org/10.1371/journal.pone.0263423.g001>

erroneous anomaly detection is much greater than the cost of false detection of normal values. The first task of an anomaly detection algorithm in practical applications should be to identify outliers as much as possible, so the *ADR* is an important evaluation indicator in industrial applications. *DN* is the number of abnormal data detected correctly, and *AN* is the number of abnormal data in the entire test set. The *ADR* is defined as:

$$ADR = \frac{DN}{AN} \quad (10)$$

The *AUC* is calculated from the recall (*REC*) and false positive rate (*FPR*). Let *TP* be the number of detected true positives, *FP* the number of detected false positives, *TN* the number of detected true negatives and *FN* the number of detected false negatives. The recall and false positive rate are defined as follows:

$$REC = \frac{TP}{TP + FN} \quad (11)$$

$$FPR = \frac{FP}{FP + TN} \quad (12)$$

Data

To better evaluate the performance of algorithms in different network environments, we used four simulation datasets, which are obtained by simulating the real network traffic data captured on the load balancer. We desensitize the existing real data, expand the real data set according to the distribution of real network traffic data. The simulated data are sampled from a mixture of two multivariate Gaussian distributions. The multivariate Gaussian distribution mixture model can be represented by the following equation [40]:

$$p(x; \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x - \mu)^T \Sigma^{-1}(x - \mu)\right). \quad (13)$$

where *n* is the dimension of the feature and the nondimensional vector μ_i is defined as the mean value of the *i*-th feature, $\mu = (\mu_1, \mu_2, \dots, \mu_n)$ is a *n*-dimensional vector. Here, Σ is the covariance of the feature, which is an *n*-dimensional matrix:

$$\Sigma = \begin{bmatrix} \sigma(x_1, x_1) & \cdots & \sigma(x_1, x_n) \\ \vdots & \ddots & \vdots \\ \sigma(x_n, x_1) & \cdots & \sigma(x_n, x_n) \end{bmatrix} \in \mathbb{R}^{n \times n} \quad (14)$$

The multivariate Gaussian distribution is chosen because it is the maximum entropy distribution for a given mean and variance [40]. Therefore, a minimum number of assumptions are imposed on the simulation data. We usually assume that there is a correlation between the various features of network traffic data and that a multivariate Gaussian distribution can automatically capture the correlation between the features, which makes this approach suitable for the simulation of network traffic data. We use two multivariate Gaussian distributions to simulate the normal data and abnormal data of real network traffic data and sample them to form simulation datasets. The configuration of the 4 simulation datasets is as follows:

Simulation 1. The first simulation dataset has a total of 6-dimensional features; the training set contains 1 million samples, and the abnormal ratio is approximately 2%. The dataset is

characterized by a large amount of data, and abnormal points are abnormally distributed in all dimensions. According to the above characteristics, the distribution of the abnormal data is quite different from that of the normal data, so the challenge of abnormality detection is less difficult.

Simulation 2. The abnormal ratio, number of samples, and dimension of the simulation2 set have the same settings as simulation1. However, each anomaly point presents anomalies in a few random features only. We assume that the features are independent of each other in the simulation2, so the covariance matrix Σ can be expressed as:

$$\Sigma = \begin{bmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_6^2 \end{bmatrix} \in \mathbb{R}^{6 \times n} \quad (15)$$

We randomly select 3 features from 6 features, calculate the variance and mean of these 3 features from normal data, and use the variance and mean of abnormal data for the remaining 3 features. After the combination, Σ and μ are obtained, and a multi-dimensional Gaussian distribution of abnormal data is constructed. Compared with the first simulation data set, there are more swing points. Swing points arise when the point is at the edge of the normal data distribution and abnormal data distribution, presenting a challenge in the anomaly detection task. Compared with simulation1, the data distribution in simulation2 is more challenging as a task of abnormal data detection.

Simulation 3. The third simulation dataset uses the configuration of simulation1 with the same proportion of outliers and number of dimensions. The number of samples in the training set in this data set is only 730 to test the performance of the outlier algorithm with a small sample data set.

Simulation 4. The fourth simulation dataset uses the configuration of simulation1 with the same proportion of outliers. The difference is that the dimension of the fourth data set is extended to as high as 50, and the data distribution of a few dimensions is quite different from that of other dimensions. In addition, to simulate the existence of the real data set, some extreme values are randomly added to the abnormal points of the data set. The challenge of this data set to the abnormal detection algorithm lies in the high dimensionality of the data set and the influence of extreme values.

Other datasets

To make the experimental data more convincing, we measure the performance of our proposed method for detecting outliers through a number of publicly available datasets. The Shuttle, Satellite, and BreastW datasets are selected from the UCI data set, and the Mulcross data set is selected from the ODDS data set [41]. Table 1 lists the statistical characteristics of each

Table 1. Statistical characteristics of the other experimental dataset.

| Dataset | Cardinal number | Number of attributes | Abnormal points |
|-----------|-----------------|----------------------|-----------------|
| Shuttle | 49097 | 9 | 3437 |
| Mulcross | 262144 | 4 | 26214 |
| Satellite | 6435 | 36 | 2036 |
| BreastW | 683 | 9 | 239 |

<https://doi.org/10.1371/journal.pone.0263423.t001>

data set. These datasets contain different cardinal number, number of attributes and anomaly proportions, which means that they can be used to test the robustness of the algorithms, which is key to their ability to perform well in complex and changing network environments.

Verification of algorithm validity

We used the *AUC* and *ADR* to jointly evaluate the performance of iForest under different contamination parameters. The satellite dataset from the ODDS dataset is used here, and the abnormal ratio of the training set is 0.32. Fig 2 plots the ROC curve for simultaneous assessment of *AUC* and *ADR* and the Avg curve for the sum average of *AUC* and *ADR*. Near the true abnormal ratio(0.32) of the data set, the point of the ROC curve is closest to the upper left corner, indicating that the point in this area is closest to the optimal critical value. Moreover, the Avg curve also reaches the highest value at *contamination* = 0.34, which is near the true anomaly ratio. This proves that an accurate abnormal ratio has an important impact on the performance of iForest.

To verify the effectiveness of the X-iForest algorithm, we tested EiForest, iForest, LOF, CBLOF, PCA, HBOS, and KNN on a total of 8 datasets mentioned earlier and compare them with X-iForest. For the 8 datasets, we use the same training set and test set for all algorithms, and adjust the abnormal ratio of other algorithms except X-iForest to the true abnormal ratio to optimize the performance of these algorithms and ensure the objectivity of the experiment.

Table 2 counts the *AUC* results of all algorithms. The bold font in the table indicates the best performance in the horizontal experiment. X-iForest performs the best in 6 out of 8 datasets. X-iForest maintains a high *AUC* in each data set, which also means that the algorithm has high performance and good robustness. Fig 3 shows a histogram of *AUC* results. Based on the average *AUC* value, iForest can maintain high detection accuracy and stability with different datasets. The improvement of X-iForest on the basis of the iForest enables an 8.1%

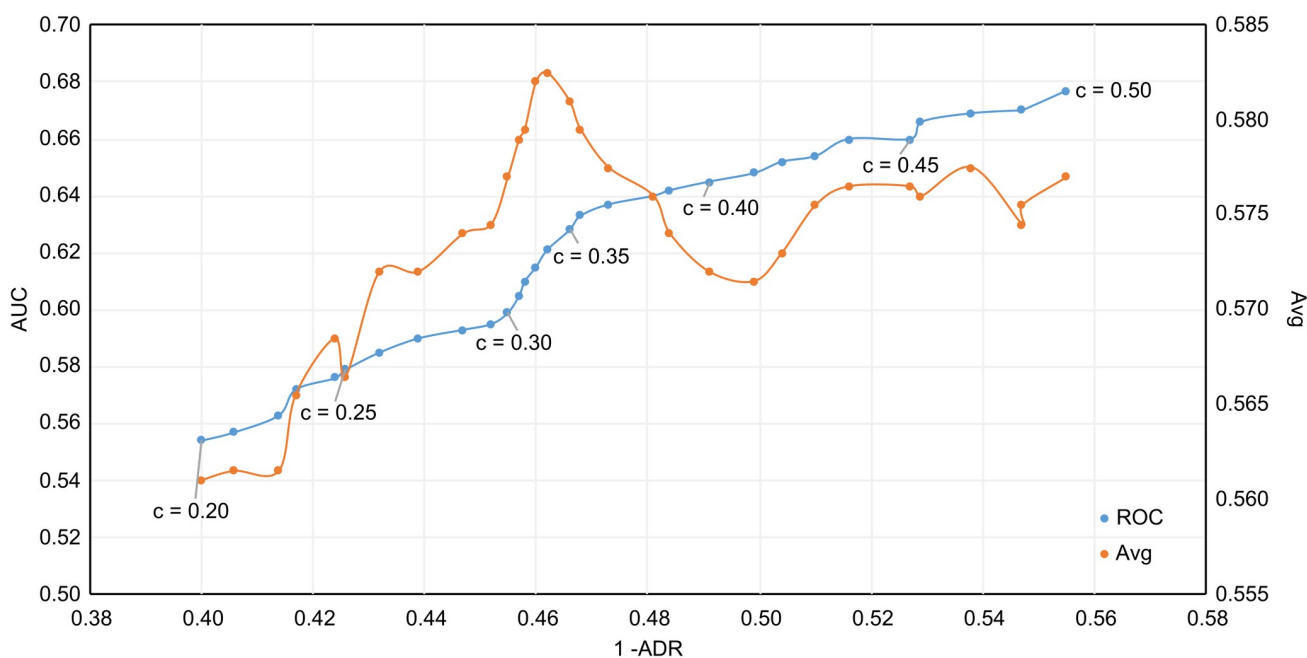


Fig 2. The performance of iForest under different contamination parameters in the dataset with a abnormal ratio of 0.32, *c* represents the contamination parameter.

<https://doi.org/10.1371/journal.pone.0263423.g002>

Table 2. AUC of X-iForest and other algorithms.

| Dataset | X-iForest | EiForest | iForest | LOF | PCA | HBOS | CBLOF | KNN |
|--------------|--------------|----------|---------|-------|-------|--------------|-------|--------------|
| Simulation 1 | 1.0 | 0.99 | 0.941 | 0.498 | 0.958 | 0.957 | 0.733 | 0.99 |
| Simulation 2 | 0.823 | 0.8 | 0.761 | 0.711 | 0.701 | 0.756 | 0.664 | 0.719 |
| Simulation 3 | 0.763 | 0.672 | 0.638 | 0.735 | 0.621 | 0.633 | 0.738 | 0.874 |
| Simulation 4 | 1.0 | 0.971 | 0.929 | 0.51 | 0.954 | 0.954 | 0.743 | 0.978 |
| Shuttle | 0.968 | 0.945 | 0.895 | 0.741 | 0.969 | 0.973 | 0.699 | 0.765 |
| Mulcross | 0.98 | 0.915 | 0.868 | 0.57 | 0.96 | 0.455 | 0.752 | 0.445 |
| Satellite | 0.723 | 0.679 | 0.638 | 0.585 | 0.641 | 0.646 | 0.681 | 0.646 |
| BreastW | 0.978 | 0.968 | 0.932 | 0.529 | 0.764 | 0.653 | 0.812 | 0.94 |

<https://doi.org/10.1371/journal.pone.0263423.t002>

improvement in the average AUC in the 8 experimental datasets over that of the iForest, which means that the former has better detection accuracy and better robustness.

The ADR is an important indicator for industrial applications, including network traffic anomaly detection applications. We compared the ADRs of X-iForest and other algorithms, and the results are shown in Table 3. X-iForest has the highest ADR in all 8 datasets, which shows that X-iForest can complete the anomaly detection task well. Fig 4 shows the histogram of ADR results. The average values indicate that the performance of X-iForest is very stable and far better than those of the other algorithms. The results from the AUC experiments show that the EiForest algorithm has a good performance and robustness. However, the ADR experimental results show that EiForest did not achieve high ADR values in the Simulation2 and Satellite Dataset. The average ADR of X-iForest is 19.5% higher than that of iForest and 10.1% of EiForest, which means that X-iForest can better complete the anomaly detection task.

Based on the results, X-iForest has the best performance in both simulation datasets and the real datasets, maintaining high AUC and ADR performance, which shows that X-iForest can complete anomaly detection tasks well while maintaining high accuracy. The AUC of X-

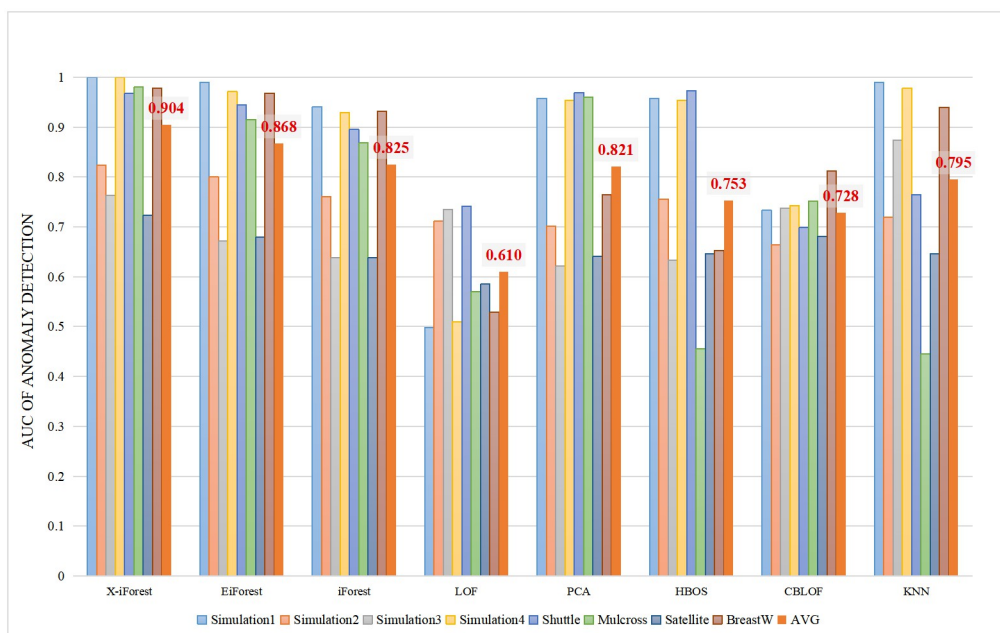


Fig 3. The AUC results of the proposed algorithm X-iForest and others algorithms.

<https://doi.org/10.1371/journal.pone.0263423.g003>

Table 3. ADR of X-iForest and other algorithms.

| Dataset | X-iForest | EiForest | iForest | LOF | PCA | HBOS | CBLOF | KNN |
|--------------|--------------|----------|---------|-------|-------|-------|-------|-------|
| Simulation 1 | 1.0 | 1.0 | 0.883 | 0.02 | 1.0 | 1.0 | 0.467 | 1.0 |
| Simulation 2 | 0.85 | 0.783 | 0.533 | 0.433 | 0.417 | 0.6 | 0.333 | 0.45 |
| Simulation 3 | 0.82 | 0.385 | 0.41 | 0.564 | 0.359 | 0.63 | 0.487 | 0.769 |
| Simulation 4 | 1.0 | 1.0 | 0.86 | 0.03 | 1.0 | 1.0 | 0.487 | 0.957 |
| Shuttle | 0.982 | 0.899 | 0.796 | 0.607 | 0.967 | 0.976 | 0.401 | 0.648 |
| Mulcross | 1.0 | 1.0 | 0.762 | 0.229 | 0.929 | 0.019 | 0.504 | 0.0 |
| Satellite | 0.6 | 0.38 | 0.514 | 0.432 | 0.291 | 0.304 | 0.409 | 0.53 |
| BreastW | 0.986 | 0.958 | 0.917 | 0.389 | 0.528 | 0.306 | 0.639 | 0.903 |

<https://doi.org/10.1371/journal.pone.0263423.t003>

iForest in Simulation3 with a small sample size is not the highest, and there is no large gap with other algorithms. We speculate that X-iForest may not be the best choice for the small sample data set. However, the ADR of Simulation 3 shows that X-iForest still accomplishes the anomaly detection task well. Although the Satellite dataset is characterised by high-dimensional and massive volume, the relatively concentrated distribution of the data and the close distribution of outliers and normal values result in many swing points in the data, which means that it is more difficult to do anomaly detection on this dataset. The X-iForest algorithm does not achieve high AUC(0.723) and ADR(0.6) on Satellite dataset, but it still performs much better than other algorithms. Compared to other algorithms, X-iForest shows robustness to different types of data, especially large-scale high-dimensional data, which also proves that X-iForest can be well applied to network health analysis and the field of abnormal network traffic detection.

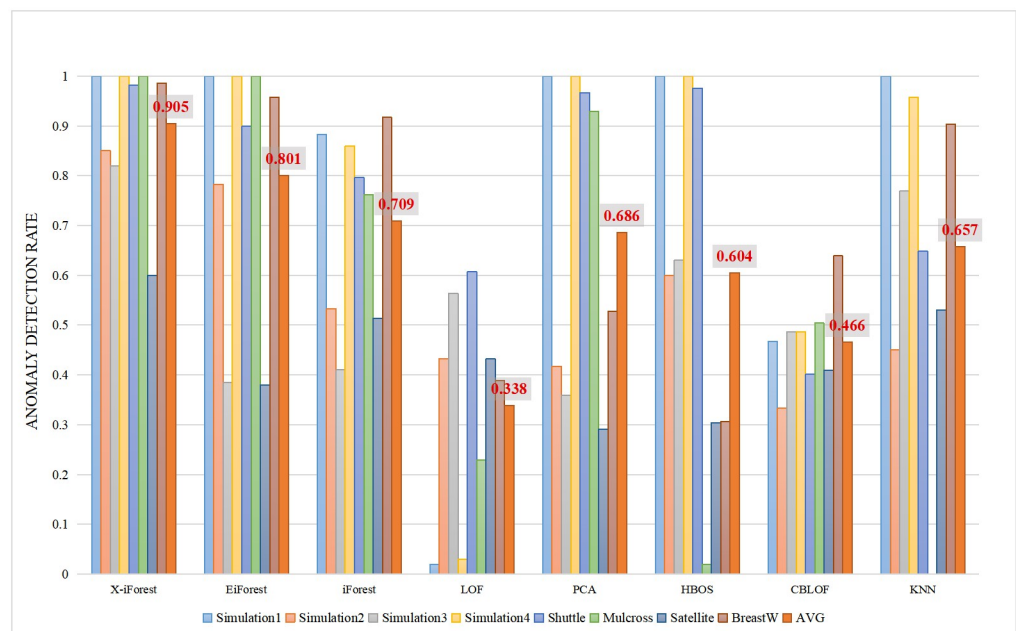


Fig 4. The ADR results of the proposed algorithm X-iForest and others algorithms.

<https://doi.org/10.1371/journal.pone.0263423.g004>

Conclusion

The complex network environment and the surge of traffic data make the detection of network traffic anomalies a considerable challenge facing enterprises today. Due to the varying network environment, the definition and distribution of normal and abnormal data have also varied, resulting in a lack of sufficient labelled network traffic data for enterprises. In addition to the explosive growth of traffic data, the current unsupervised anomaly detection algorithms commonly used in industrial applications cannot be well implemented in a real complex network environment. The main research purpose of this paper is to realize the application of unsupervised algorithms in actual dynamic network environments. It is hoped that even when the abnormal ratio of the training data is unknown, the abnormal traffic data in the network traffic can still be found quickly and accurately. In this paper, we combine iForest and X-means to propose the novel algorithm X-iForest. Experiments prove that X-iForest exhibits high accuracy and high performance with massive data in complex networks. The experimental results meet our expectations, and the performance of the X-iForest algorithm is better than that of the iForest algorithm with a precise abnormal ratio. X-iForest has an average AUC score of 8.1% higher than iForest and an average ADR score of 19.5% higher across the 8 datasets. Compared with other algorithms, X-iForest also shows excellent accuracy and robustness without the need to provide precise abnormal ratios. The average AUC score of X-iForest is 15.1% higher than the average score of other algorithms, and the average ADR score of X-iForest reaches 90.4% in the ADR score, while the average ADR score of other algorithms is only 57.6%. This represents a significant advantage of X-iForest and can be well applied to the task of network traffic anomaly detection.

In the next step, we plan to deploy the X-iForest algorithm on a load balancer based on the Flink framework, making full use of the distributed stream processing features of the Flink framework to generate multiple trees in an iForest by placing them on different nodes to speed up training and detection and further validate the performance of the algorithm in real-world network scenarios.

Acknowledgments

The authors are very grateful for the insightful comments and suggestions of the anonymous reviewers and the editor who have helped to significantly improve the quality of this article.

Author Contributions

Conceptualization: Yifan Feng, Weihong Cai, Haoyu Yue.

Data curation: Yifan Feng, Weihong Cai, Haoyu Yue, Jiaxin Chen, Zijun Hu.

Investigation: Yifan Feng, Weihong Cai, Haoyu Yue, Jianlong Xu, Jiaxin Chen, Zijun Hu.

Methodology: Yifan Feng, Weihong Cai, Haoyu Yue, Jianlong Xu.

Resources: Weihong Cai, Yan Lin.

Software: Yifan Feng, Weihong Cai, Haoyu Yue, Jiaxin Chen, Zijun Hu.

Writing – original draft: Yifan Feng, Weihong Cai, Haoyu Yue.

Writing – review & editing: Weihong Cai, Jianlong Xu.

References

1. Kim KH, Cho SB. Modular Bayesian networks with low-power wearable sensors for recognizing eating activities. *Sensors*. 2017; 17(12):2877. <https://doi.org/10.3390/s17122877> PMID: 29232937

2. Ronao CA, Cho SB. Recognizing human activities from smartphone sensors using hierarchical continuous hidden Markov models. *International Journal of Distributed Sensor Networks*. 2017; 13(1):1550147716683687. <https://doi.org/10.1177/1550147716683687>
3. Cai W, Du X, Xu J. A personalized QoS prediction method for web services via blockchain-based matrix factorization. *Sensors*. 2019; 19(12):2749. <https://doi.org/10.3390/s19122749> PMID: 31248105
4. Abdellatif AA, Ahmed E, Fong AT, Gani A, Imran M. SDN-based load balancing service for cloud servers. *IEEE Communications Magazine*. 2018; 56(8):106–111. <https://doi.org/10.1109/MCOM.2018.1701016>
5. Chaudhary R, Kumar N. LOADS: Load optimization and anomaly detection scheme for software-defined networks. *IEEE Transactions on Vehicular Technology*. 2019; 68(12):12329–12344. <https://doi.org/10.1109/TVT.2019.2948222>
6. Du X, Xu J, Cai W, Zhu C, Chen Y. Oprc: An online personalized reputation calculation model in service-oriented computing environments. *IEEE Access*. 2019; 7:87760–87768. <https://doi.org/10.1109/ACCESS.2019.2925778>
7. Ahmed M, Mahmood AN, Hu J. A survey of network anomaly detection techniques. *Journal of Network and Computer Applications*. 2016; 60:19–31. <https://doi.org/10.1016/j.jnca.2015.11.016>
8. Agrawal S, Agrawal J. Survey on anomaly detection using data mining techniques. *Procedia Computer Science*. 2015; 60:708–713. <https://doi.org/10.1016/j.procs.2015.08.220>
9. Lee W, Stolfo SJ, Mok KW. Adaptive intrusion detection: A data mining approach. *Artificial Intelligence Review*. 2000; 14(6):533–567. <https://doi.org/10.1023/A:1006624031083>
10. Lee W, Stolfo S. Data mining approaches for intrusion detection. 1998.
11. Michalak M, Wawrowski U, Sikora M, Kurianowicz R, Bialas A. Outlier Detection in Network Traffic Monitoring. In: *Proceedings of the 10th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2021)*, Vienna, Austria; 2021. p. 4–6.
12. Yu J, Liu F, Zhou W, Yu H. Hadoop-based network traffic anomaly detection in backbone. In: *2014 IEEE 3rd International Conference on Cloud Computing and Intelligence Systems*. IEEE; 2014. p. 140–145.
13. Breunig MM, Kriegel HP, Ng RT, Sander J. LOF: identifying density-based local outliers. In: *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*; 2000. p. 93–104.
14. Goldstein M, Dengel A. Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm. *KI-2012: Poster and Demo Track*. 2012; p. 59–63.
15. Puggini L, McLoone S. An enhanced variable selection and Isolation Forest based methodology for anomaly detection with OES data. *Engineering Applications of Artificial Intelligence*. 2018; 67:126–135. <https://doi.org/10.1016/j.engappai.2017.09.021>
16. Susto GA, Beghi A, McLoone S. Anomaly detection through on-line isolation forest: an application to plasma etching. In: *2017 28th Annual SEMI Advanced Semiconductor Manufacturing Conference (ASMC)*. IEEE; 2017. p. 89–94.
17. Ahmed S, Lee Y, Hyun SH, Koo I. Unsupervised machine learning-based detection of covert data integrity assault in smart grid networks utilizing isolation forest. *IEEE Transactions on Information Forensics and Security*. 2019; 14(10):2765–2777. <https://doi.org/10.1109/TIFS.2019.2902822>
18. Liu FT, Ting KM, Zhou ZH. Isolation forest. In: *2008 eighth IEEE international conference on data mining*. IEEE; 2008. p. 413–422.
19. Spiekermann D, Keller J. Unsupervised packet-based anomaly detection in virtual networks. *Computer Networks*. 2021; 192:108017. <https://doi.org/10.1016/j.comnet.2021.108017>
20. Pelleg D, Moore AW, et al. X-means: Extending k-means with efficient estimation of the number of clusters. In: *ICML*. vol. 1; 2000. p. 727–734.
21. Paulauskas N, Baskys A. Application of Histogram-Based Outlier Scores to Detect Computer Network Anomalies. *Electronics*. 2019; 8(11):1251. <https://doi.org/10.3390/electronics8111251>
22. Wang L, Li J, Bhatti UA, Liu Y. Anomaly detection in wireless sensor networks based on KNN. In: *International Conference on Artificial Intelligence and Security*. Springer; 2019. p. 632–643.
23. Hoang DH, Nguyen HD. A PCA-based method for IoT network traffic anomaly detection. In: *2018 20th International conference on advanced communication technology (ICACT)*. IEEE; 2018. p. 381–386.
24. Gan Z, Zhou X. Abnormal network traffic detection based on improved LOF algorithm. In: *2018 10th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*. vol. 1. IEEE; 2018. p. 142–145.
25. He Z, Xu X, Deng S. Discovering cluster-based local outliers. *Pattern Recognition Letters*. 2003; 24(9–10):1641–1650. [https://doi.org/10.1016/S0167-8655\(03\)00003-5](https://doi.org/10.1016/S0167-8655(03)00003-5)
26. Abbasi M, Shahraki A, Taherkordi A. Deep learning for network traffic monitoring and analysis (NTMA): A survey. *Computer Communications*. 2021. <https://doi.org/10.1016/j.comcom.2021.01.021>

27. Kim TY, Cho SB. Web traffic anomaly detection using C-LSTM neural networks. *Expert Systems with Applications*. 2018; 106:66–76. <https://doi.org/10.1016/j.eswa.2018.04.004>
28. Wei G, Wang Z. Adoption and realization of deep learning in network traffic anomaly detection device design. *Soft Computing*. 2021; 25(2):1147–1158. <https://doi.org/10.1007/s00500-020-05210-1>
29. Hariri S, Kind MC, Brunner RJ. Extended isolation forest. *IEEE Transactions on Knowledge and Data Engineering*. 2019.
30. Ding Z, Fei M. An anomaly detection approach based on isolation forest algorithm for streaming data using sliding window. *IFAC Proceedings Volumes*. 2013; 46(20):12–17. <https://doi.org/10.3182/20130902-3-CN-3020.00044>
31. Wang H, Jiang W, Deng X, Geng J. A new method for fault detection of aero-engine based on isolation forest. *Measurement*. 2021; 185:110064. <https://doi.org/10.1016/j.measurement.2021.110064>
32. Alsini R, Almakrab A, Ibrahim A, Ma X. Improving the outlier detection method in concrete mix design by combining the isolation forest and local outlier factor. *Construction and Building Materials*. 2021; 270:121396. <https://doi.org/10.1016/j.conbuildmat.2020.121396>
33. Chen H, Ma H, Chu X, Xue D. Anomaly detection and critical attributes identification for products with multiple operating conditions based on isolation forest. *Advanced Engineering Informatics*. 2020; 46:101139. <https://doi.org/10.1016/j.aei.2020.101139>
34. Zhou L, Zhang T, Zhang Z, Lei Z, Zhu S. A new online quality monitoring method of chain resistance upset butt welding based on Isolation Forest and Local Outlier Factor. *Journal of Manufacturing Processes*. 2021; 68:843–851. <https://doi.org/10.1016/j.jmapro.2021.06.005>
35. Wang YB, Chang DG, Qin SR, Fan YH, Mu HB, Zhang GJ. Separating multi-source partial discharge signals using linear prediction analysis and isolation forest algorithm. *IEEE Transactions on Instrumentation and Measurement*. 2019; 69(6):2734–2742. <https://doi.org/10.1109/TIM.2019.2926688>
36. Xing G, Chen J, Hou R, Zhou L, Dong M, Zeng D, et al. Isolation Forest-Based Mechanism to Defend against Interest Flooding Attacks in Named Data Networking. *IEEE Communications Magazine*. 2021; 59(3):98–103. <https://doi.org/10.1109/MCOM.001.2000368>
37. Chomboon K, Chujai P, Teerarassamee P, Kerdprasop K, Kerdprasop N. An empirical study of distance metrics for k-nearest neighbor algorithm. In: *Proceedings of the 3rd international conference on industrial application engineering*; 2015. p. 280–285.
38. Frigge M, Hoaglin DC, Iglewicz B. Some implementations of the boxplot. *The American Statistician*. 1989; 43(1):50–54. <https://doi.org/10.2307/2685173>
39. Abdi H, Williams LJ. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*. 2010; 2(4):433–459. <https://doi.org/10.1002/wics.101>
40. Bishop CM. *Pattern recognition and machine learning*. springer; 2006.
41. Xu D, Wang Y, Meng Y, Zhang Z. An improved data anomaly detection method based on isolation forest. In: *2017 10th International Symposium on Computational Intelligence and Design (ISCID)*. vol. 2. IEEE; 2017. p. 287–291.