Peer-to-Peer Networking and Applications (2023) 16:1680–1695
https://doi.org/10.1007/s12083-023-01482-0

Anomaly traffic detection based on feature fluctuation for secure industrial internet of things

Jie Yin[1] · Chuntang Zhang[2] · Wenwei Xie[3] · Guangjun Liang[1] · Lanping Zhang[4] · Guan Gui[4]

Received: 25 November 2022 / Accepted: 30 March 2023 / Published online: 26 April 2023

Abstract
The detection of anomaly traffic in internet of things (IoT) is mainly based on the original binary data at the traffic packet level and the structured data at the session flow level. This kind of dataset has a single feature extraction method and relies on prior manual knowledge. It is easy to lose critical information during data processing, which reduces the validity and robustness of the dataset. In this paper, we first construct a new anomaly traffic dataset based on the traffic packet and session flow data in the Iot-23 dataset. Second, we propose a feature extraction method based on feature fluctuation. Our proposed method can effectively solve the disadvantage that the data collected in different scenarios have different characteristics, which leads to the feature containing less information. Compared with the traditional anomaly traffic detection model, experiments show that our proposed method based on feature fluctuation has stronger robustness, can improve the accuracy of anomaly traffic detection and the generalization ability of the traditional model, and is more conducive to the detection of anomalous traffic in IoT.

Keywords Anomaly traffic detection · Dataset construction · Feature fluctuations · Machine learning · Iot-23 dataset

1 Introduction

The internet of things (IoT), originated in the media field, is the third revolution of the information technology industry. Recently, the emergence of many advanced technologies, such as IoT, artificial intelligence (AI) and cloud computing have been considered as the most important trend for fast development of future intelligent society [1–4]. Since its establishment, it has been developing at an amazing speed, often changing its direction rapidly, and appearing in new and quite unexpected forms. Up to now, the application of IoT has covered all fields of daily life and industrial production, and is extending to a wider and deeper field at a rapid speed, such as edge computing [5], smart city [6], medical care [7] and smart cars [8]. The coverage of IoT devices is increasingly broad, from smart thermostats controlled by smart phones at home to advanced driverless vehicles equipped with hundreds of sensors. These devices rely on sensors to collect and transmit data to ensure their effective operation [9, 10].

With the breakthrough of fifth generation (5G) mobile communications and other key technologies, the development of IoT applications has made rapid progress. At the same time, affected by the novel coronavirus pneumonia situation, the demand for remote office has increased, followed by massive equipment access to the Internet. Hence, internet

✉ Jie Yin
yinjiejspi@163.com

Chuntang Zhang
b19020421@njupt.edu.cn

Wenwei Xie
jim.xie.cn@outlook.com

Guangjun Liang
lianggjun@126.com

Lanping Zhang
1021010526@njupt.edu.cn

Guan Gui
guiguan@njupt.edu.cn

[1] Computer Information and Cyber Security, Jiangsu Police Institute, Nanjing 210031, China

[2] Bell Honors School, Nanjing University of Posts and Telecommunications, Nanjing 210023, China

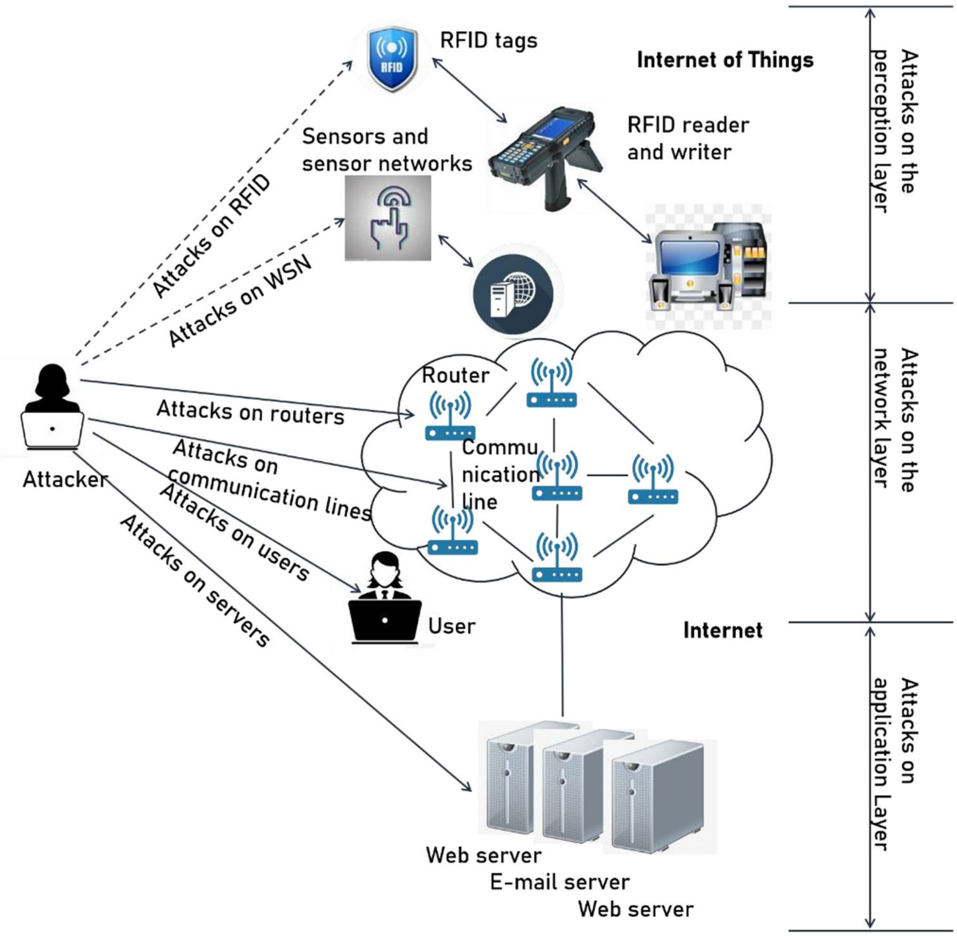
[3] Network Security, Trend Micro Incorporated, Nanjing 210012, China

[4] College of Telecommunications and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210003, China

of everything brings convenience, but also brings more attack risks for attackers. Attacks in IoT can be divided into network attacks and physical attacks [11]. Network attacks include passive attacks and active attacks. Nowadays, many security attacks do not directly involve the backbone network, but attack the entire network by attacking individual devices or distributed devices of the Internet of Things. Network attack security issues [12] mainly involve data security, privacy, replication, and threats to Radio Frequency Identification (RFID) systems. It can be seen from Fig. 1 that the types of attacks on IoT security include attacks on RFID, wireless sensor networks (WSN), routers and communication lines. Physical attacks will cause physical damage to IoT devices. In this case, no network is needed to attack devices.

Compared with Internet and mobile communication network security, secure IoT research is just studied at the early stage. Due to the wide interconnection and massive data of IoT, security research becomes more difficult. Therefore, it is particularly important to identify and classify the traffic data transmitted in the network, and the monitoring based on anomaly traffic, as one of the very important research means, has also been widely concerned. At present, most network and security monitoring tools use signature-based methods to detect anomaly, which requires high data structure and stability of features. In 2021, Zhao et al. proposed a method based on dynamic Auto-Encoder (AE) [13] to monitor anomaly traffic, and the monitoring accuracy of KDD99 dataset and UNSW-NB15 dataset is 93.1% and 98.5%. In the same year, this team proposed an anomaly flow detection method based on the Lightweight Neural Network (LNN) [14]. The monitoring accuracy on UNSW-NB15 dataset reached 98.94%, which further improved the accuracy of anomaly detection compared with the previous method. Hasan et al. [15] predicted attacks and anomalies on IoT network through careful comparison of multiple models such as decision tree (DT), random forest (RF), support vector machine (SVM) and artificial neural network (ANN) under NSL-KDD, Real Traffic and DS2OS datasets. Among them, the best accuracy of random forest records is 99.4%. Alrashdi et al. [16] proposed an intelligent anomaly detection IoT (AD IoT) system based on random forest machine learning algorithm to address IoT network security threats in smart cities. Using the UNSW-NB15 modern dataset to evaluate the accuracy of the model, the best result of the RF algorithm in the experiment is the highest classification accuracy of 99.34% and the lowest false positive rate.

Fig. 1 Six types of attacks over different layers in IoT



In general, on the open dataset, most of the experimental schemes have reached more than 90%, but these methods are rarely used in industrial practice. The general trend is to use the same simulation data set to train and test these models, and then report very high accuracy. However, in an ideal life scenario, the data used to train the model is very different from the data deployed to run the model in the production environment. At deployment time, this can adversely affect the performance of the model. This is because most public data sets uniformly cut, supplement and extract features for each traffic data collected, but the amount of information about the same feature in different data is different, which makes some traffic data lose its important features due to the unified feature extraction method. In order to solve the problem of different feature extraction methods between different types of data or even different datasets, and balance the differences between the network deploying the intrusion detection model and the network training the intrusion detection model, this paper proposes an IoT anomaly traffic detection method based on feature fluctuations. The experimental results show that the proposed method still maintains a good feature output environment when the features of the data change. At the same time, a variety of traditional anomaly detection models are used to evaluate the stability of output features, and good detection results are achieved. The main contributions of this paper are highlighted as follows:

- We propose a feature extraction method based on Wave Robust EMD distance to characterize anomaly traffic through the change of feature.
- We propose a calculation method of packet anomaly index and level based on sliding window, which describes the anomaly status of the packet.
- In the case of cross-dataset, the deployed anomaly detection model can still get stable output, using the features extracted by wave robust EMD, the accuracy of the trained neural network model reaches 97.07%.

2 Related work

It is a common method of anomaly detection to divide the types of detected network traffic according to the different features of network traffic. Its basic feature data, such as traffic size, packet length information, protocol information, and port traffic information, all express the running state of network traffic. The data features are integrated into a complete dataset after filtering, clipping and other operations. This kind of dataset is also commonly used by researchers in the encryption anomaly network traffic detection experiment. As shown in Table 8, researchers have made great breakthroughs in detection accuracy by using different models and technologies in network traffic classification and anomaly detection.

Bagui et al. [17] used three machine learning algorithms: logical regression, support vector machine and random forest to study the botnet traffic in the Internet of Things. The dataset used is from UCI's machine learning library [18], and the data is from IoT devices infected by gafgyt and mirai botnets. Their research shows that the detection rate of the three classifiers for botnets is as high as 99%, and the random forest performs best. Wang et al. [19] proposed the BalancePIC scheme to ensure the balance between IoT user privacy, data integrity of edge auxiliary IoT devices and computing costs. The data they use is based on the real data track collection of the perceptual network system, and is obtained by simulating the real network environment through various devices [20]. Simulation results show that the scheme can obtain more accurate data and better network traffic detection performance under the premise of protecting user privacy. Elmrabit et al. [21] evaluated 12 machine learning algorithms on three open datasets: CICIDS-2017, UNSW-NB15 and industrial control system (ICS) network attacks. From the evaluation indicators of accuracy, precision, recall, F1Score and Receiver Operating Characteristic (ROC) curves, the random forest algorithm performs best in the three data sets, with an accuracy rate of 99.9% in CICDS-2017. Liu et al. [22] used different machine learning schemes to analyze the problem of enhancing the security of IoT networks. The dataset used in the experiment is IoT network intrusion dataset created by researchers of Hacking and Countermeasure Research Lab [23]. The research results show that the KNN algorithm achieves 99% accuracy in the average running time of two minutes, and XGBoost achieves 97% accuracy in the running time of 10.8 seconds. From the above research results, it can be seen that the accuracy of machine learning algorithms in detecting abnormal network traffic on public data sets is more than 95%. However, in the actual deployment of the secure industrial Internet of Things, we rarely see the use of the above research scheme to detect and classify network traffic anomalies. One important reason is that the above research uses the same data set to model the IoT scenario. The training and testing models all use the same data set, thus obtaining high classification accuracy. However, in actual industrial deployment, the data of model operation is often different from the data used in training, which leads to a significant reduction in the performance of the model in actual application [24].

Public network traffic datasets generally process the collected data in an undifferentiated way, which will weaken the important features of the sample and strengthen the unimportant features of the sample. This leads to the poor performance of the anomaly flow detection model when the sample features differ greatly or are deployed across datasets. Lorbach et al. [25] studied the influence of different experimental data on the same machine learning model and used F1 score to evaluate the experimental effect. When the experiment used RatSI and Validation datasets for training and testing, the F1 score was 0.46. When the RatSI dataset is used for both training and testing, F1 scores 0.52. When the validation data set is used for training and testing, F1 scores 0.68. A simple cross dataset experiment shows that it is urgent to solve the problem of performance degradation when machine learning models are deployed on different data. Qin et al. [26] proposed using adaptive spatiotemporal transfer learning (ASTTL) to solve the problem of data differences between different data sets. ASTTL learns spatial features in transfer learning by adaptively evaluating the relative importance between marginal probability distribution and conditional probability distribution. Through a large number of experiments on four open datasets, the average classification accuracy of ASTTL is 66.3%, which is 9% - 14% higher than that of the method without migration learning. However, the method of transfer learning is not end-to-end learning. The two-stage structure increases the complexity and computation of the model. Sha et al. [27] proposed a new domain adaptive algorithm to solve the problem of cross dataset mismatch. The difference of statistical distribution is reduced by converting the characteristic representation of the data in the target dataset, and the model trained in the source dataset is used to the maximum extent to adapt to the target dataset. In binary learning, there is a big difference between the accuracy of training within data sets and that of training across data sets. For example, the accuracy of training and testing in the VQA dataset [28] is 65.7%, while the accuracy of the model decreases to 53.4% when using the Visual7W dataset [29]. It can be seen from the appeal study that when the data structure in the dataset is quite different or cross dataset evaluation is conducted, the common method is to use migration learning to achieve cross domain adaptation. This method is improved from the machine learning model. Therefore, we can think that the anomaly traffic detection method in the secure industrial Internet of Things should follow the following principles:

- Low complexity: The traffic data in the industrial Internet of Things changes rapidly and has a large amount of data, so relatively simple detection methods are needed to reduce the computing cost.
- Strong generalization ability: The attacks on IoT devices every day may be known or unknown, and the detection ability for unknown attacks is an important indicator to evaluate the anomaly detection model.
- Concise data: From the perspective of data itself, the original network traffic data contains a large number of features but few effective features. Therefore, selecting an effective feature selection method to process data samples can improve detection accuracy and reduce computational redundancy.

Based on the above principles, the method proposed in this paper optimizes the dataset itself and directly realizes end-to-end classification without complex classification models.

3 Problem description

It is important to use the right method to evaluate the performance of the anomaly traffic detection model. Detection models that use a single dataset for training and testing may not perform well in actual deployment because the data observed in real life is very different from the simulated data. Farah [30] performed very well with the IOTID20 dataset when training and testing in the same dataset, as shown in Table 1. This is because the data distribution of the training and test sets is similar. However, when deployed, the model must face different data, and the results can mislead the performance of the model. When one dataset is used to train the model and another dataset is used to test the model, the performance of the model degrades, as shown in Table 2. This is because the methods used for simulation on the two datasets are different. Performing cross-dataset assessments helps understand the problem that anomaly traffic detection models generated using the same simulated data are not well generalized when tested against different simulated attacks, that is, opponents may invade the network using new methods that cannot be reflected by the simulated data. The reason for the poor generalization ability of models can be analyzed from two perspectives: sample and features.

Table 1 AUC values for each category in the IoTID20 dataset

Model	Decision Trees				kNN ($k = 5$)			
Category	Dos	Normal	Scan	Weighted_AUC	Dos	Normal	Scan	Weighted_AUC
AUC	0.99	0.92	0.99	0.98	0.99	0.87	0.97	0.97
Model	Logistic Regression				Ensemble			
Category	Dos	Normal	Scan	Weighted_AUC	Dos	Normal	Scan	Weighted_AUC
AUC	0.91	0.76	0.92	0.90	0.99	0.94	0.98	0.98

Table 2 AUC classification across datasets, the training set is a Bot-IoT dataset, and the test set is an IoTID20 dataset

Model	Decision Trees				kNN ($k = 11$)			
Category	Dos	Normal	Scan	Weighted_AUC	Dos	Normal	Scan	Weighted_AUC
AUC	0.48	0.75	0.37	0.55	0.47	0.77	0.30	0.54
Model	Logistic Regression				Ensemble			
Category	Dos	Normal	Scan	Weighted_AUC	Dos	Normal	Scan	Weighted_AUC
AUC	0.87	0.61	0.88	0.78	0.49	0.66	0.33	0.52

3.1 Cause analysis of model generalization from the sample perspective

Public datasets are generally built to simulate platforms for collecting normal and anomaly network traffic data. Artificial simulation platforms collect limited types of data, and in actual industrial deployment, they may encounter a variety of attacks different from the type of data collected. Therefore, the great difference between the sample types that can be used for model training and the data in the reality environment is one of the important reasons that the performance of the model will be greatly reduced in practical application. Feature engineering for open datasets generally generates features by statistical means, quantitatively analyzing each collected feature, resulting in a limited number of features that may result in the loss of important information. This allows the model to identify only existing attacks in the training set and not unknown attacks.

As shown in Fig. 2, the worm virus is not part of the training set sample, so when it appears in the test set, the model cannot recognize the type of attack. At the same time, the uniform cleaning, processing and marking of the collected data greatly reduces the robustness of the model.

3.2 Reasons analysis of model generalization from the perspective of features

In machine learning, feature selection is very important. Good data and features often determine the upper limit of machine learning. Peker et al. [31] studied the classification accuracy of the number of features in the Naive Bayes model and found that the more features the data has, the better. When the key features in the study have been selected, the training accuracy will reach a high value, and then the overlapped features will have little effect on the accuracy, but will increase the computational load of the model and pay more cost.

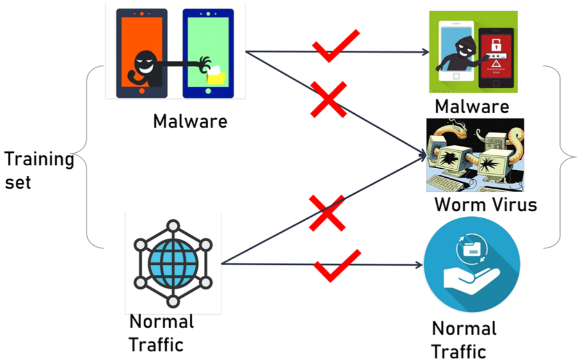


Fig. 2 Diagram of classification effect of models with different sample types of test set and training set

Figure 3 shows the distribution of the importance of each feature in the model trained on the IoT-23 dataset, and it can be seen from Fig. 3 that the importance of the different features in the model is not evenly distributed. When instable features account for the majority of the share, the model output depends on individual features and the output is unstable. In different attack scenarios, the statistical features are not common enough, which is quite different from the real scene.

4 Preliminary

Before explaining the method in this paper, a new hypothesis is introduced: since IoT traffic is usually generated automatically by devices with less human participation, the distribution of IoT traffic is stable over a period of time. Therefore, when an attack occurs, the stability of traffic is destroyed. Anomaly traffic can be detected through the change of traffic, that is, anomaly traffic can be detected according to the fluctuation of features, the detection process is shown in Fig. 4.

The data features used in this study are very important to the training of the model. Table 3 shows the 17 features used in training.

4.1 Describing fluctuations in features using statistics

Feature fluctuations refer to changes in the eigenvalues over time, assuming that the sequence of eigenvalues is $X = \left\{x_1, x_2, \ldots, x_n\right\}$ and that the eigenvalues at time t are x_t, then the feature fluctuation W_t time t can be expressed as

$$W_t = f(X_t) \quad (1)$$

where function $f(\cdot)$ represents the measure of the change of the eigenvalues. Hence, the mean and standard deviation of the features are defined respectively as [32]

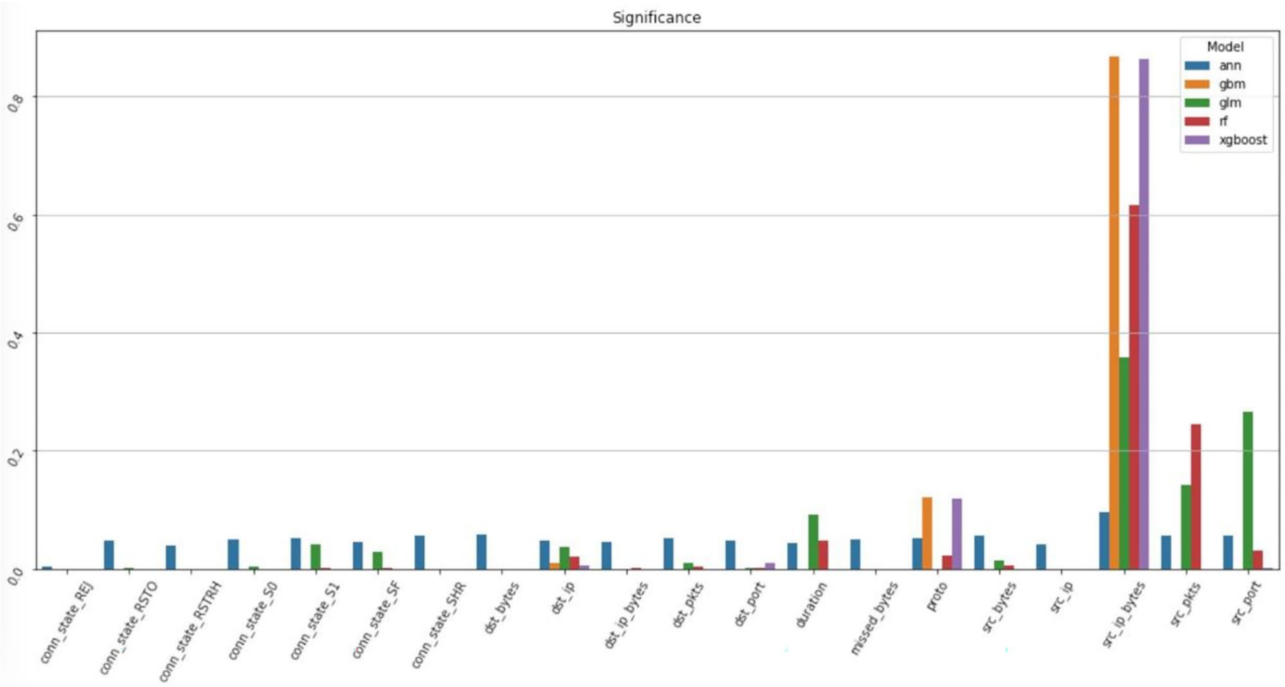


Fig. 3 The importance of multiple features in different models

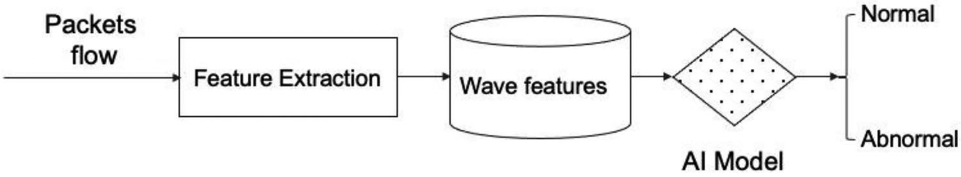


Fig. 4 Anomaly traffic detection process

Table 3 Features used for model training

Name	Description
duration	The duration of a TCP or UDP session
ip_tos	The type of service (ToS) field in IP header
ip_len	Total length field in IP header
ip_proto	Protocol field in IP header
ip_src	Source IP address
ip_dst	Destination IP address
udp_sport	Source UDP port
udp_dport	Destination UDP port
udp_len	Total size of UDP package
tcp_sport	Source TCP port
tcp_dport	Destination TCP port
tcp_seq	Sequence number field in TCP header
tcp_ack	Sequence number field in TCP header
tcp_reserved	Acknowledgement number field in TCP header
tcp_window	Window size field in TCP header
tcp_urgptr	Urgent point field in TCP header
tcp_flags	Options field in TCP header

$$f(X_t) = \frac{1}{n}\sum_{i-1}^{n} x_i, \tag{2}$$

$$f_d(X_t) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2}. \tag{3}$$

4.2 Describing fluctuations in features using a distance matrix

4.2.1 Distance matrix

The distance matrix [33] can describe the change in the eigenvalue at different times in the feature sequence as

$$D = \begin{bmatrix} d(x_1,x_1) & d(x_1,x_2) & \cdots & d(x_1,x_n) \\ d(x_2,x_1) & d(x_2,x_2) & \cdots & d(x_2,x_n) \\ \cdots & \cdots & \cdots & \cdots \\ d(x_n,x_1) & d(x_n,x_2) & \cdots & d(x_n,x_n) \end{bmatrix} \tag{4}$$

where element $d(x_i, x_j)$ in the matrix represents the distance of the eigenvalue at the moment i and j, and d represents the measure of distance, such as cosine distance, EMD distance, Wave Robust EMD distance, and so on. These functions are given as subsequent.

4.2.2 Cosine distance

It can be expressed as

$$d(x_i, x_j) = \frac{\mathbf{x_i} \cdot \mathbf{x_j}}{\left\|\mathbf{x_i}\right\|\left\|\mathbf{x_j}\right\|} \tag{5}$$

4.2.3 EMD

Earth Mover's Distance (EMD) [34] is a similarity measurement method proposed by IJCV Journal Article "The Earth Mover's Distance as a Metric for Image Retrieval" in 2000 [35], which can be used to measure feature changes and computed as the minimal value of a linear program. Rubner et al. [35] computed EMD using a linear program called transportation simplex and used it for content based image retrieval by comparing colour signatures. They obtained better performance than binwise measures. EMD can be used for histogram different measure. Ling and Okada [36] used the EMD for comparing different histogram descriptors and noted improved performance compared to ??2 and Euclidean distance. EMD can be used for pattern identification. Holmes et al. [37] use partial signature matching based on the EMD for identifying mammogram structures. They embed histograms into a learned Euclidean space to speed up computation.

Assuming that there are m features in common, the eigenvectors at x time are $(p_1, p_2, \cdots, p_m)$, and at y time are $(q_1, q_2, \cdots, q_m)$, each f_{ij} denotes the flow from p_i to q_j, d_{ij} denotes the distance from p_i to q_j. The EMD is defined as

$$\begin{aligned}
&EMD(x, y) = \frac{\sum_{i=1}^{m}\sum_{j=1}^{m} f_{ij} \times d_{ij}}{\sum_{i=1}^{m}\sum_{j=1}^{m} f_{ij}} \\
&EMDcost = \min \sum_{i=1}^{m}\sum_{j=1}^{m} d_{ij} \times f_{ij} \\
&\text{s.t.} \\
&f_{ij} \geq 0, i = 1, 2, \cdots, m; j = 1, 2, \cdots, m \\
&\sum_{j=1}^{m} f_{ij} \leq p_i,\ i = 1, 2, \cdots, m \\
&\sum_{i=1}^{m} f_{ij} \leq q_j,\ j = 1, 2, \cdots, m
\end{aligned} \tag{6}$$

And the cost function and the constraint are expressed as

$$\sum_{i=1}^{m}\sum_{j=1}^{m} f_{ij} = \min\left\{\sum_{i=1}^{m} p_i,\ \sum_{j=1}^{n} q_j\right\}. \tag{7}$$

4.2.4 Wave robust EMD distance

Traditional EMD distances treat two sets of eigenvalues indiscriminately, and the timing information of the features fails to be reflected in the EMD distance. Hence, this paper proposes a wave robust EMD (WREMD) method to improve EMD as

$$WREMD(x_i, x_j) = \frac{|i - j|}{n} \times EMD(x_i, x_j) \tag{8}$$

where x_i represents the eigenvalue of the moment i, x_j represents the eigenvalue of the j moment.

5 Experiment

Each IoT-23 scenario captures network traffic to get the original pcap file, then uses the Zeek tool to get a dataset of statistics (i.e., the ZeekFlows dataset), and tags each ZeekFlow according to the type of attack. When generating a ZeekFlows dataset, the original data packets need to be aggregated, which results in the ZeekFlows dataset not describing the change of traffic characteristics effectively. In this experiment, based on the IoT-23 dataset, a new dataset is obtained by tag construction and feature construction, and then the dataset is used for model training and evaluation. The detailed flow of the experiment is shown in Fig. 5.

5.1 IoT-23 dataset introduction

The IoT-23 dataset [38] consists of 23 different IoT network traffic scenarios, including captured samples (pcap files) in 20 malware scenarios and three network traffic for benign IoT devices. It was first released in January 2020 and captured from 2018 to 2019. These IoT network traffic was captured at Stratosphere Laboratory, FEL AIC Group, CTU University, Czech Republic. Malicious and benign scenarios operate in a controlled network environment with unrestricted Internet connectivity, just like any other real Internet of Things device.

Tables 4 and 5 show the data in the network security scenarios of the training set and the test set, respectively, including the name of the dataset, duration (in hours), number of packets, number of Zeek ID streams in the conn.log file (obtained by running the Zeek Network Analysis Framework on the original pcap file), size of the original pcap file, and possible names of the malicious software samples used to

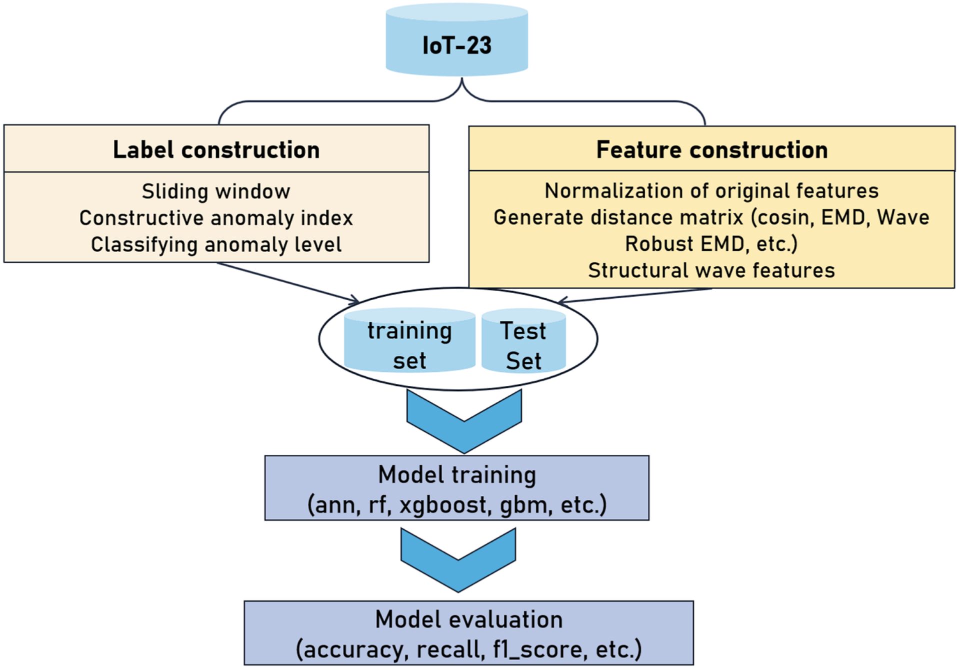

IoT-23
Label construction
Sliding window
Constructive anomaly index
Classifying anomaly level
Feature construction
Normalization of original features
Generate distance matrix (cosin, EMD, Wave Robust EMD, etc.)
Structural wave features
training set
Test Set
Model training
(ann, rf, xgboost, gbm, etc.)
Model evaluation
(accuracy, recall, f1_score, etc.)

Fig. 5 Dataset processing flowchart

infect the device. Malware capture can take a long time to execute. Because of the high volume of traffic generated by each infection, the pcap file is rotated every 24 hours. However, in some cases, the pcap file grows too fast and decides to stop capturing until 24 hours have elapsed. As a result, some captures differ in the number of hours.

The IoT-23 dataset includes the original pcap file and ZeekFlows data generated using the zeek tool. The ZeekFlows data contains malicious traffic tags, and the original pcap file does not contain malicious traffic tags. The IoT-23 data used in this paper is divided into multiple subsets with different sources, and the samples in the subset are not intersected, so it can be used for experimental research on samples from different sources between different data sets. As shown in Table 6, in this experiment, data from 12 different scenarios is selected for training and testing to simulate deployment across data set scenarios.

5.2 Labels construction

The pcap data of IoT-23 does not contain malicious traffic tags, and each packet is labeled by the following method.

Table 4 Summary of network security scenarios for training sets

Name of Dataset	Duration (hrs)	Packets	ZeekFlows	Pcap Size	Name
CTU-IoT-Malware-Capture-17-1	24	109000000	54659864	7.8GB	Kenjiro
CTU-IoT-Malware-Capture-20-1	24	50000	3210	3.9MB	Torii
CTU-IoT-Malware-Capture-21-1	24	50000	3287	3.9MB	Torii
CTU-IoT-Malware-Capture-33-1	24	54000000	54454592	3.9GB	Kenjiro
CTU-IoT-Malware-Capture-34-1	24	233000	23146	121MB	Mirai
CTU-IoT-Malware-Capture-36-1	24	13000000	13645107	992MB	Okiru

Table 5 A summary of the test set network security scenarios

Name of Dataset	Duration (hrs)	Packets	ZeekFlows	Pcap Size	Name
CTU-IoT-Malware-Capture-39-1	7	73000000	73568982	5.3GB	IRCBot
CTU-IoT-Malware-Capture-43-1	1	82000000	67321810	6GB	Mirai
CTU-IoT-Malware-Capture-48-1	24	13000000	3394347	1.2GB	Mirai
CTU-IoT-Malware-Capture-49-1	8	18000000	5410562	1.3GB	Mirai
CTU-IoT-Malware-Capture-52-1	24	64000000	19781379	4.6GB	Mirai
CTU-IoT-Malware-Capture-8-1	24	23000	10404	2.1MB	Hakai

Assuming that there are n packets in the pcap data, and that the sequence of packets is $X = \{x_1, x_2, x_3, ..., x_n\}$, the label of the packet is constructed in the following four steps.

5.2.1 The original tags are queried from the ZeekFlows data

Based on the data packet's time stamp, source IP address, source port, destination IP address, destination port and so on, the corresponding label is queried from ZeekFlows data, and the original label sequence $L = \{L_1, L_2, L_3, ..., L_n\}$ is obtained.

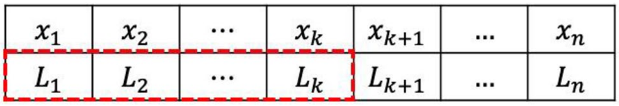
5.2.2 As shown in Fig. 6, the anomaly index is calculated using a sliding window of size *k*

The anomaly index S_1 corresponding to packet x_1 is calculated as shown in formula (9), where abnormal count (AC) represents the number of occurrences of the original tag of the exception, and total count (TC) represents the number of packets in the sliding window.

$$S_1 = \frac{AC}{TC}, S \in [0, 1] \tag{9}$$

As the sliding window slides over the packet sequence X, the anomaly index corresponding to each packet is calculated using a similar method, and the anomaly index sequence $S = \{S_1, S_2, S_3, ..., S_n\}$ is obtained. In this experiment, the sliding window size used is 5, that is, the parameter $k = 5$ is chosen.

5.2.3 The anomaly level is calculated based on the anomaly index



x_1	x_2	⋯	x_k	x_{k+1}	…	x_n
L_1	L_2	⋯	L_k	L_{k+1}	…	L_n

Fig. 6 Using the sliding window to calculate the anomaly index

The calculation method for the anomaly level of data package x_i is shown in formula (10), where D_i represents the anomaly level of data package x_i, m represents the number of anomaly levels, and *Floor* represents the integer part. In this experiment, the anomaly index is divided into five levels, that is, the parameter $m = 5$ is selected.

$$D_i = \begin{cases} Floor(m \times S_i), & S_i \in [0, 1) \\ m - 1, & S_i = 1 \end{cases} \tag{10}$$

Table 6 Security scenarios

Purpose	IoT-23 Dataset
Training	CTU-IoT-Malware-Capture-17-1
	CTU-IoT-Malware-Capture-20-1
	CTU-IoT-Malware-Capture-21-1
	CTU-IoT-Malware-Capture-33-1
	CTU-IoT-Malware-Capture-34-1
	CTU-IoT-Malware-Capture-36-1
Testing	CTU-IoT-Malware-Capture-39-1
	CTU-IoT-Malware-Capture-43-1
	CTU-IoT-Malware-Capture-48-1
	CTU-IoT-Malware-Capture-49-1
	CTU-IoT-Malware-Capture-52-1
	CTU-IoT-Malware-Capture-8-1

5.2.4 Determine the label of the packet

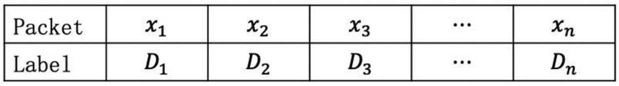
After rating the anomaly index of each packet, the anomaly rank sequence $D = \{D_1, D_2, D_3, \cdots, D_n\}$ is obtained, which is used as the label of the packet, as is shown in Fig. 7.

5.3 Feature construction

Assuming that there are n packets in the pcap data, and the sequence of packets is $X = \{x_1, x_2, x_3, \cdots, x_n\}$, the eigenvector is constructed in the following two steps.

5.3.1 Divide the packet into multiple groups

As shown in Fig. 8, m packets are divided into one group $m < n$, and $n - m + 1$ groups are obtained, that is, $G = \{G_1, G_2, G_3, \cdots, G_{(n-m+1)}\}$. In this experiment, the parameter $m = 32$ is taken.



Packet	x_1	x_2	x_3	⋯	x_n
Label	D_1	D_2	D_3	⋯	D_n

Fig. 7 Using the anomaly level as the label of data packet

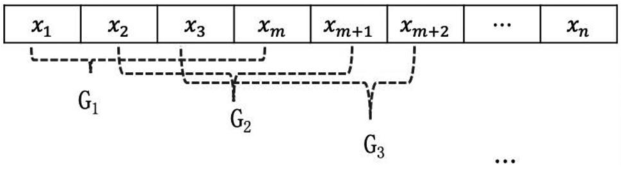


Fig. 8 Divide packets into groups

Table 7 Feature types and generation methods

The feature type	Build how
(1) Raw	Direct use of TCP/IP protocol header information for packets as raw features
(2) Mean	Use the average of the original features in each group as a new feature
(3) Std	Use the standard deviation of the original features in each group as a new feature
(4) Cosine	Use the cosine distance matrix of the original features in each group as a new feature
(5) EMD	Use the EMD distance matrix of the original features in each group as a new feature
(6) WREMD	Use the WREMD distance matrix of the original features in each group as a new feature

5.3.2 The protocol header information in the data package is used as the original feature

At the same time, the original feature is transformed to get a variety of new features. As shown in Table 7, these features are used for comparison experiments.

5.4 Experimental results

Select machine learning and neural network algorithm for model training, verify the malicious traffic monitoring performance of the proposed feature construction algorithm, and compare the performance of the algorithm from the accuracy index. Machine learning algorithms that participate in the evaluation are Artificial Neural Network (ANN), Random Forest (RF), eXtreme Gradient Boosting (XGBoost), Gradient Boosting Machine (GBM), and Generalized Linear Model (GLM). All the five machine learning algorithms are commonly used in anomaly traffic detection. They are very typical and effective as experimental schemes to evaluate this paper.

6 Performance evaluation

6.1 Experimental equipment

The server device used in this experiment is Intel Core i9-9820X CPU, with memory size of 128G and hard disk size of 1TB. The experimental programming language is running on Python, the machine learning framework is H2O, and the original dataset is IoT-23.

6.2 Evaluation indicators

The performance of the experiment is determined by *Accuracy*, *Recall*, *Precision*, F_1-*Score* four indicators to evaluate.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{11}$$

$$Recall = \frac{TP}{TP + FN} \tag{12}$$

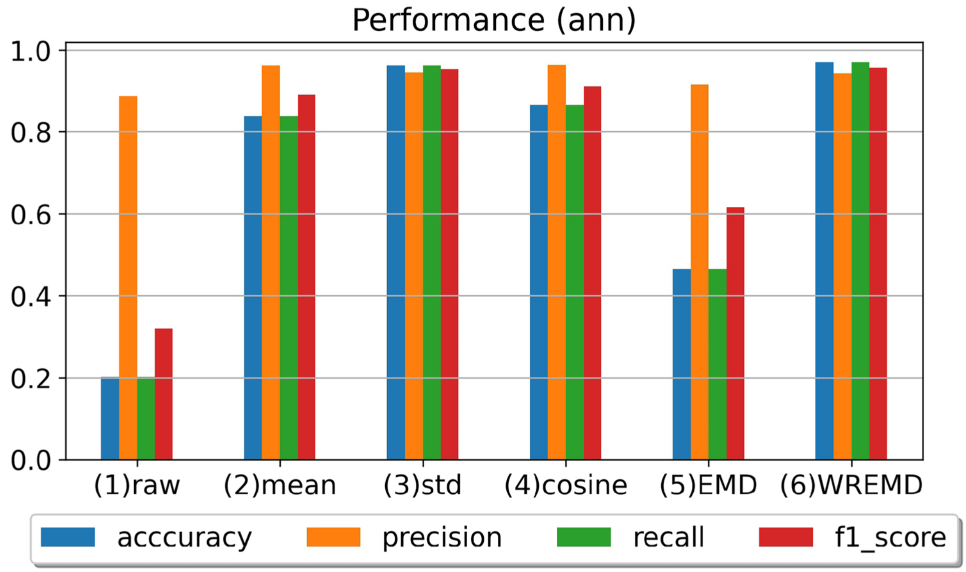


Fig. 9 Performance characterization of different feature extraction algorithms on ANN models

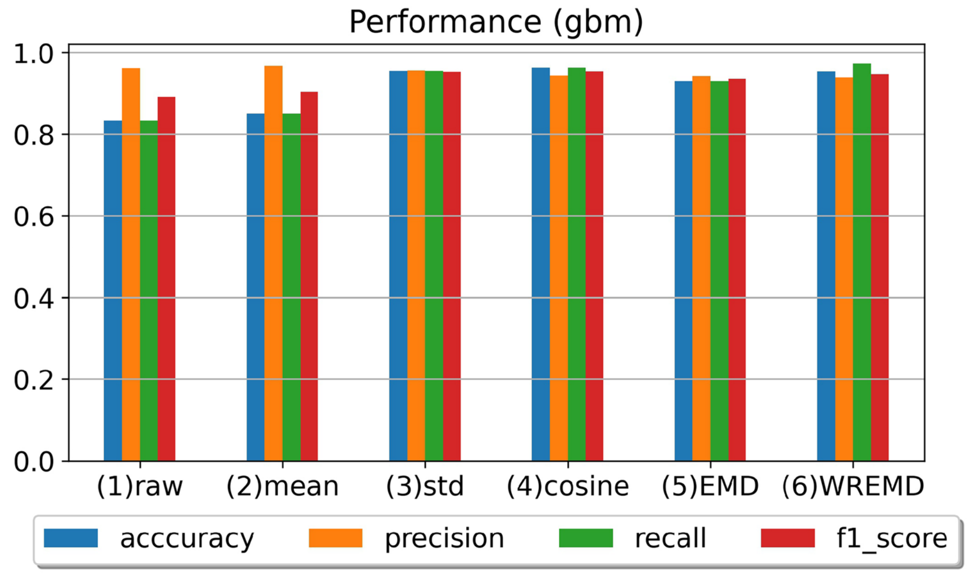


Fig. 10 Performance characterization of different feature extraction algorithms on GBM model

$$Precision = \frac{TP}{TP + FP} \tag{13}$$

$$F_1\text{-}Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{14}$$

where *TP* indicates that the positive sample is correctly identified as a positive sample; *TN* indicates that a negative sample is correctly identified as a negative sample; *FP* represents a false positive sample, i.e. a negative sample is misidentified as a positive sample; *FN* represents a false negative sample, i.e. a positive sample is misidentified as a negative sample.

6.3 Analysis of experimental results

To verify the validity of the dataset processing method proposed in this paper in the application of anomaly detection model, six experiments were carried out. The original features, mean, standard deviation, cosine distance matrix, EMD distance matrix and WREMD distance matrix were used as features to compare and study the different machine learning algorithm models. Figures 9, 10, 11, 12, 13 and 14 show the performance characterization of datasets built by different feature extraction algorithms on multiple anomaly detection models.

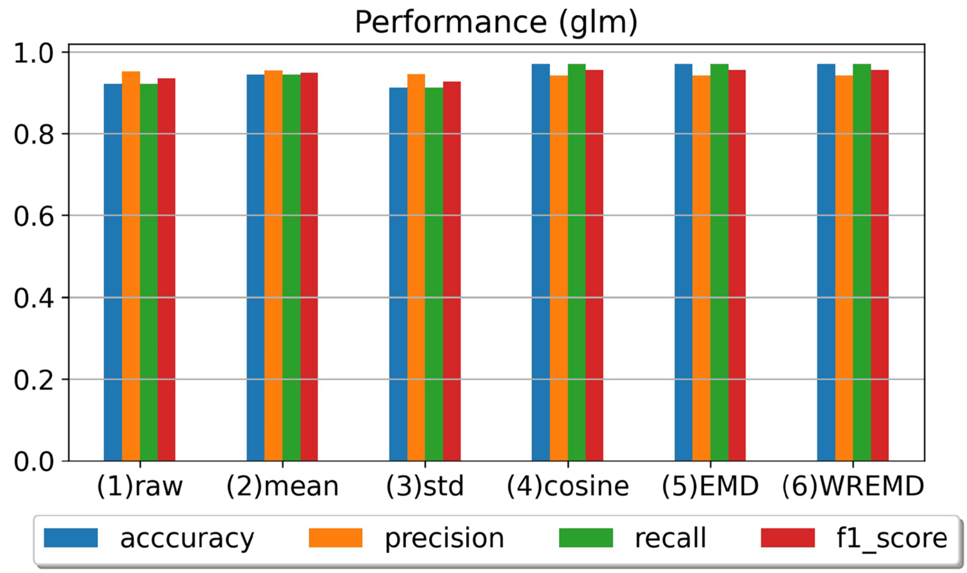


Fig. 11 Performance characterization of different feature extraction algorithms on GLM models

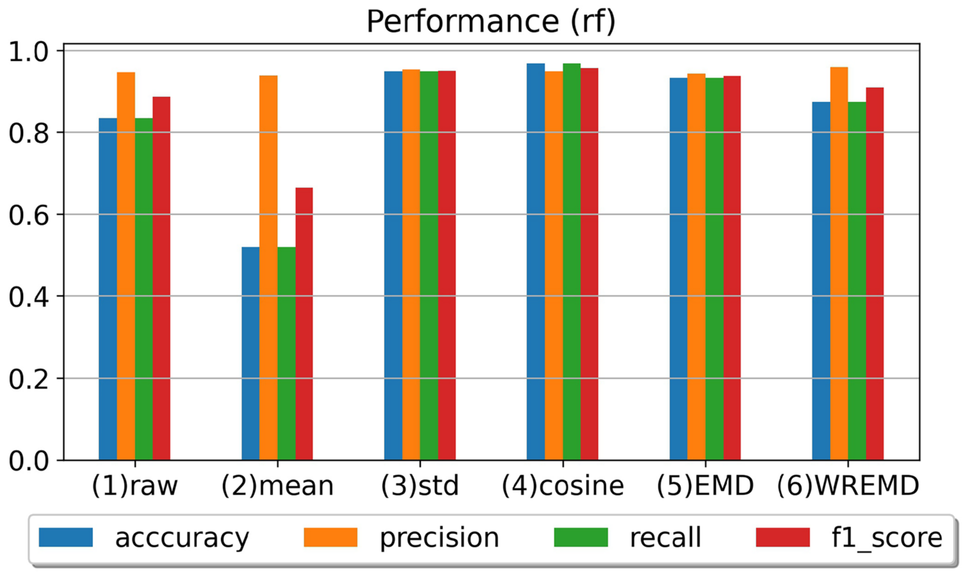


Fig. 12 Performance characterization of different feature extraction algorithms on RF models

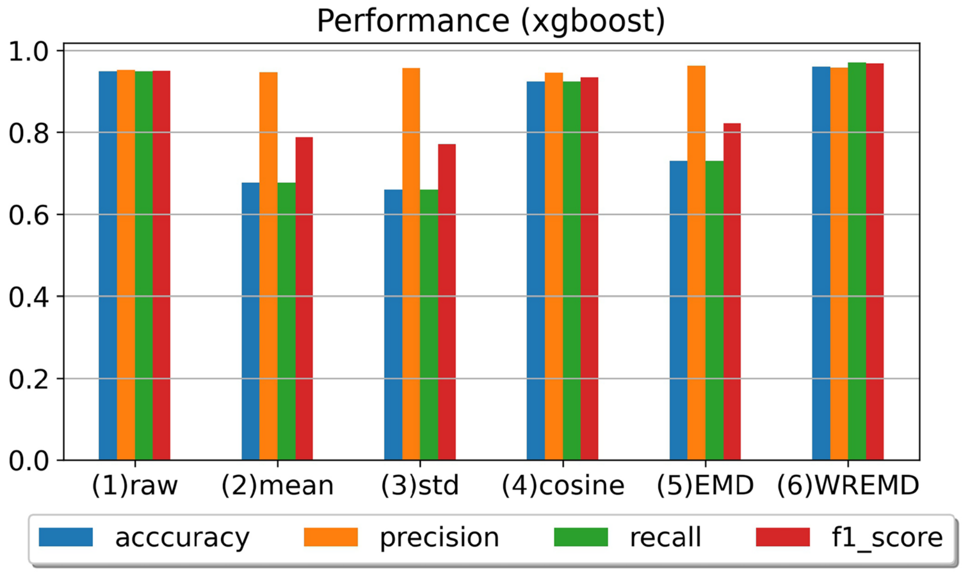


Fig. 13 Performance characterization of different feature extraction algorithms on XGBoost model

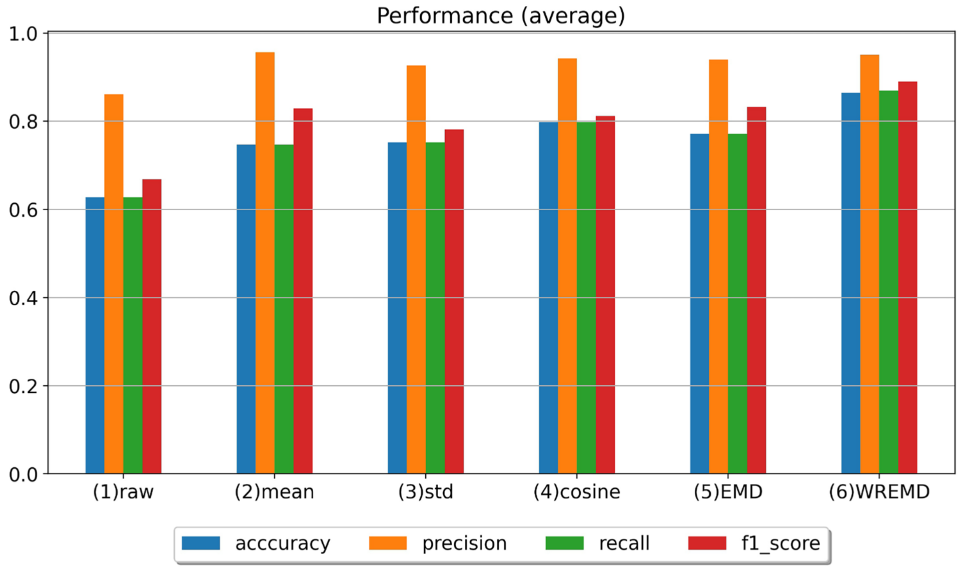


Fig. 14 Average performance characterization of different feature extraction algorithms on five machine learning models: ANN, RF, XGBoost, GBM and GLM

Table 8 Research on ML algorithm performance of anomaly detection and attack in IoT

Authors	Year	Objectives	Datasets	Results
Bagui et al. [17]	2021	Detect botnet network traffic in the Internet of Things	Machine learning library from UCI	Logistic regression, support vector machine and random forest classifier can detect botnet network traffic up to 99%, and random forest is the best
Wang et al. [19]	2020	Ensure the balance between the privacy of IoT users, the data integrity of edge auxiliary IoT devices and computing costs	Real data track collection based on perceptual network system	Error of 0.05 at the time of its worst case, and a high variance of weight ±5% scenario while employing the enhanced technique
Elmrabit et al. [21]	2020	Evaluating the classification performance of traditional machine learning algorithms	CICIDS-2017, UNSW-NB15 and Industrial Control System (ICS)	The random forest algorithm performs best in the three datasets, with accuracy of 99.9% in CICDS-2017
Liu et al.[22]	2020	Enhance the security of IoT network	Network intrusion dataset in the Internet of Things	KNN algorithm achieved 99% accuracy in the average running time of two minutes, and XGBoost achieved 97% accuracy in the running time of 10.8 seconds
Lorbach et al. [25]	2018	Study the effects of different experimental data on the same machine learning model.	RatSI and Validation	When the RatSI and Validation datasets were trained and tested respectively, the F1 score was 0.46. When the validation data set is used for training and testing, F1 scores 0.68.
Qin et al. [26]	2019	Study the data difference between different datasets	DSADS, HCI-HAR, USC-HAD and PAMAP	The average classification accuracy is 66.3%, which is 9% - 14% higher than that of the method without transfer learning
Sha et al. [27]	2018	Resolve cross dataset mismatches	Visual7W and VQA	The accuracy rate of training and testing in the VQA dataset was 65.7%, while the accuracy rate of the model decreased to 53.4% when assessed with the Visual7W dataset. The accuracy is improved to 58.5% after using domain adaptive method.

In the anomaly detection scenario, the fluctuation features extracted by WREMD performed significantly better in the ANN and GLM model. *Accuracy*, *Recall*, *Precision*, F_1-*Score* are higher than other feature extraction methods, and the accuracy rate is about 97%. The fluctuation features extracted by the Cosine algorithm perform better in RF, XGBoost and GBM models. From Fig. 14, it can be seen that, in general, the fluctuation features extracted by WREMD perform best in multiple models with an average accuracy of more than 85%.

7 Conclusion

Based on the construction of anomaly traffic detection dataset in IoT, in this paper, we propose a feature extraction method for feature fluctuation dataset, and make an experimental comparison on IoT-23 dataset. Experiments show that the use of fluctuation features can effectively alleviate model degradation when data fluctuations are large or across datasets. The anomaly index can well express the fluctuation degree of data package labels, which provides data support for feature extraction below. By comparing with four traditional feature extraction functions, it is verified that the fluctuation features extracted by WREMD algorithm are more stable and perform best in multiple anomaly detection models.

Author contributions Conceptualization, Jie Yin and Chuntang Zhang; methodology, Jie Yin; validation, Wenwei Xie, Guangjun Liang and Lanping Zhang; investigation, Jie Yin; writingoriginal draft preparation, Jie Yin and Lanping Zhang; writingreview and editing, Guan Gui; supervision, Guan Gui; project administration, Guan Gui; funding acquisition, Guan Gui All authors have read and agreed to the published version of the manuscript.

Funding This research was funded by the National Key Research and Development Program of China under grant number 2021ZD0113003.

Data availibility The data used to support the findings of this study are available from email request.

Declarations

Ethics approval There are no human subjects in this article and informed consent is not applicable.

Consent to publish All authors have read and agreed to the published version of the manuscript.

Conflict of interest The authors declare no conflict of interest.

References

1. Zhang K, Liang X, Lu R, Shen X (2014) Sybil Attacks and Their Defenses in the Internet of Things. IEEE Internet Things J 1(5):372–383
2. Nguyen DC, Ding M, Pathirana PN, Seneviratne A, Li J, Niyato D, Dobre OA, Poor HV (2022) 6G internet of things: A comprehensive survey. IEEE Internet Things J 9(1):359–383
3. Yang H, Liang S, Ni J, Li H, Shen X (2020) Secure and Efficient k NN Classification for Industrial Internet of Things. IEEE Internet Things J 7(11):10945–10954
4. Guo F, Yu FR, Zhang H, Li X, Ji H, Leung VCM (2021) Enabling massive IoT toward 6G: A comprehensive survey. IEEE Internet Things J 8(15):11891–11915
5. Caprolu M, Di Pietro R, Lombardi F, Raponi S (2019) Edge computing perspectives: Architectures, technologies, and open security issues. IEEE International Conference on Edge Computing (EDGE), pp 116–123
6. Lee R, Jang RY, Park M, Jeon GY, Kim JK, Lee SH (2020) Making IoT data ready for smart city applications. IEEE International Conference on Big Data and Smart Computing (BigComp), pp 605–608
7. Harsha Vardhini PA, Harsha MS, Sai PN, Srikanth P (2020) IoT based smart medicine assistive system for memory impairment patient. 12th International Conference on Computational Intelligence and Communication Networks (CICN), pp 182–186
8. Sanil N, Venkat PAN, Rakesh V, Mallapur R, Ahmed MR (2020) Deep learning techniques for obstacle detection and avoidance in driverless cars. International Conference on Artificial Intelligence and Signal Processing (AISP), pp 1–4
9. Zhang M, Leng W, Ding Y, Tang C (2018) Tolerating Sensitive-Leakage With Larger Plaintext-Space and Higher Leakage-Rate in Privacy-Aware Internet-of-Things. IEEE Access 6:33859–33870
10. Zhao M, Ding Y, Wu Q, Wang Y, Qin B, Fan K (2021) Privacy-Preserving Lightweight Data Monitoring in Internet of Things Environments. Wireless Pers Commun 116(3):1765–1783
11. Haji SH, Ameen SY (2021) Attack and anomaly detection in IoT networks using machine learning techniques: A review. Asian Journal of Research in Computer Science 9(2):30–46
12. Tsochev G (2020) Some security problems and aspects of the industrial internet of things. International Conference on Information Technologies (InfoTech), pp 1–5
13. Zhao R et al (2021) An efficient intrusion detection method based on dynamic autoencoder. IEEE Wireless Communications Letters 10(8):1707–1711
14. Zhao R et al (2022) A novel intrusion detection method based on lightweight neural network for internet of things. IEEE Internet Things J 9(12):9960–9972
15. Hasan M, Islam MM, Zarif M, Hashem M (2019) Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches. Internet of Things 7:2542–6605
16. Alrashdi I, Alqazzaz A, Aloufi E, Alharthi R, Zohdy M, Ming H (2019) AD-IoT: Anomaly detection of IoT cyberattacks in smart city using machine learning. IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC), pp 0305–0310
17. Bagui S, Wang X, Bagui S (2021) Machine learning based intrusion detection for IoT botnet. International Journal of Machine Learning and Computing 11(6):399–406
18. Meidan Y, Bohadana M, Mathov Y, Mirsky Y, Breitenbacher D, Shabtai A, Elovici Y (2018) N-BaIoT network-based detection of IoT botnet attacks using deep autoencoders. IEEE Pervasive Comput 17(3):12–22
19. Wang T, Bhuiyan MZA, Wang G, Qi L, Wu J, Hayajneh T (2020) Preserving balance between privacy and data integrity in edge-assisted internet of things. IEEE Internet Things J 7(4):2679–2689
20. Miao C, Jiang W, Su L, Li Y, Guo S, Qin Z, Xiao H, Gao J, Ren K (2015) Cloud-enabled privacy-preserving truth discovery in crowd sensing systems. 13th ACM Conference on Embedded Networked Sensor Systems, pp 183–196
21. Elmrabit N, Zhou F, Li F, Zhou H (2020) Evaluation of machine learning algorithms for anomaly detection. International

Conference on Cyber Security and Protection of Digital Services (Cyber Security), pp 1-8

22. Liu Z, Thapa N, Shaver A, Roy K, Yuan X, Khorsandroo S (2020) Anomaly detection on IoT network intrusion using machine learning, in International Conference on Artificial Intelligence. Computing and Data Communication Systems (icABCD), Big Data, pp 1–5
23. Kang H, Ahn D, Lee G, Yoo J, Park K, Kim H (2019) IoT network intrusion dataset. http://ocslab.hksecurity.net/Datasets/iot-network-intrusiondataset
24. Cao L, Liu Z, Huang TS (2010) Cross-dataset action detection. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp 1998–2005
25. Lorbach M, Kyriakou E, Poppe R, Dam E, Noldus L, Veltkamp R (2018) Learning to recognize rat social behavior: Novel dataset and crossdataset application. J Neurosci Methods 300:166–172
26. Qin X, Chen Y, Wang J, Yu C (2019) Cross-dataset activity recognition via adaptive spatial-temporal transfer learning. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 3(4)
27. Sha F, Hu H, Chao W (2018) Cross-dataset adaptation for visual question answering. IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 5716–5725
28. Antol S, Agrawal A, Lu J, Mitchell M, Batra D, Zitnick C, Parikh D (2015) Vqa: Visual question answering. Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp 2425–2433
29. Zhu Y, Groth O, Bernstein M, Li F (2016) Visual7w: Grounded question answering in images. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 4995–5004
30. Farah A (2020) Cross dataset evaluation for IoT network intrusion detection, University of Wisconsin-Milwaukee Dissertations. Available [online] https://dc.uwm.edu/etd/2491
31. Peker M, Arslan A, Sen B, Celebi FV, But A (2015) A novel hybrid method for determining the depth of anesthesia level: Combining relief feature selection and random forest algorithm (ReliefF+RF). International Symposium on Innovations in Intelligent Systems and Applications (INISTA), pp 1–8
32. Huang T, Sethu H, Kandasamy N (2016) A New Approach to Dimensionality Reduction for Anomaly Detection in Data Traffic. IEEE Trans Netw Serv Manag 13(3):651–665
33. Villamarin-Salomon R, Brustoloni JC (2008) Identifying Botnets Using Anomaly Detection Techniques Applied to DNS Traffic, in 5th IEEE Consumer Communications and Networking Conference. NV, USA, Las Vegas, pp 476–481
34. Tan Z, Jamdagni A, He X, Nanda P, Liu RP, Hu J (2015) Detection of Denial-of-Service Attacks Based on Computer Vision Techniques. IEEE Trans Comput 64(9)2519–2533
35. Rubner Y, Tomasi C, Guibas LJ (2000) The earth mover??s distance as a metric for image retrieval. Int J Comput Vis 40:99–121
36. Ling H, Okada K (2006) An efficient earth movers distance algorithm for robust histogram comparison. IEEE Transactions on PAMI 29(5):840–853
37. Holmes A, Rose C, Taylor C (2002) Transforming pixel signatures into an improved metric space. Image Vis Comput 20(9):701–707
38. Ling H, Okada K (2020) A labeled dataset with malicious and benign IoT network traffic, Prague, Czechia. Available [online] https://doi.org/10.5281/zenodo.4743746

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Jie Yin received his M.S. degree in Software Engineering from Nanjing University of Science and Technology, Nanjing, China in 2008. He is currently a Senior Engineering with the Department of Computer Information and Cyber Security, Jiangsu Police Institute, Nanjing, China. He is also with the Jiangsu Province Electronic Data Forensics and Analysis Engineering Research Center, Key Laboratory of Digital Forensics of Jiangsu Provincial Public Security Department, and State Key Lab. for Novel Software Technology, Nanjing University, Nanjing, China. He has published more than 10 international journal/conference papers. His recent research interests include machine learning, big data, and network security.

Chuntang Zhang iscurrently pursuing a Bachelor's degree in Communication Engineering at Bell Honor School, Nanjing University of Posts and Telecommunications, Nanjing, China. His research topic is focusing on machine learning for network security.

Wenwei Xie received a master's degree in MBA from the Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing, China in 2013. He is currently working toward network security in Trend Micro Incorporated, Nanjing, China. His research interests include IoT security, AI technology, and computer vision.

Guangjun Liang received a doctoral degree in Communication and Information Systems from Nanjing University of Posts and Telecommunications in 2018. He is currently serving as an associate professor at Jiangsu Police Institute. His research interests include resource allocation and optimization in wireless communication networks, and network security.

Lanping Zhang received her B.S. degree in Communication Engineering from Southwest Petroleum University, Chengdu, China in 2021. She is currently pursuing a master's degree in Communication Engineering at Nanjing University of Posts and Telecommunications, Nanjing, China. Her research interests include deep learning, cyber security, and its application in the Internet of Things.

Guan Gui received a Ph.D. degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2012. From 2009 to 2014, he joined Tohoku University as a research assistant as well as a postdoctoral research fellow, respectively. From 2014 to 2015, he was an Assistant Professor at the Akita Prefectural University, Akita, Japan. Since 2015, he has been a professor with the Nanjing University of Posts and Telecommunications, Nanjing, China. His recent research interests include intelligence sensing and recognition, intelligent signal processing, and physical layer security. Dr. Gui has published more than 100 IEEE Journal/Conference papers and won several best paper awards, e.g., ICC 2017, ICC 2014 and VTC 2014- Spring. He received the IEEE Communications Society Heinrich Hertz Award in 2021. Since 2022, he has been a Distinguished Lecturer of the IEEE Vehicular Technology Society. He is serving or served on the editorial boards of several journals, including IEEE Transactions on Vehicular Technology, IEICE Transactions on Communications, Physical Communication, Wireless Networks, IEEE Access, Journal of Circuits Systems and Computers, Security and Communication Networks, IEICE Communications Express, and KSII Transactions on Internet and Information Systems, Journal on Communications. In addition, he served as the IEEE VTS Ad Hoc Committee Member in AI Wireless, TPC Chair of PRAI 2022, TPC Chair of ICGIP 2022, Executive Chair of VTC 2021-Fall, Vice Chair of WCNC 2021, TPC Chair of PHM 2021, Symposium Chair of WCSP 2021, General Co-Chair of Mobimedia 2020, TPC Chair of WiMob 2020, Track Chairs of EuCNC 2021 and 2022, VTC 2020 Spring, Award Chair of PIMRC 2019, and TPC member of many IEEE international conferences, including GLOBECOM, ICC, WCNC, PIRMC, VTC, and SPAWC.