International Journal of Machine Learning and Cybernetics (2021) 12:3129–3144
https://doi.org/10.1007/s13042-020-01253-w

ORIGINAL ARTICLE

A multiple-kernel clustering based intrusion detection scheme for 5G and IoT networks

Ning Hu[1] · Zhihong Tian[1] · Hui Lu[1] · Xiaojiang Du[2] · Mohsen Guizani[3]

Received: 29 August 2020 / Accepted: 7 December 2020 / Published online: 14 January 2021

Abstract

The 5G network provides higher bandwidth and lower latency for edge IoT devices to access the core business network. But at the same time, it also expands the attack surface of the core network, which makes the enterprise network face greater security threats. To protect the security of core business, the network infrastructure must be able to recognize not only the known abnormal traffic, but also new emerging threats. Intrusion Detection Systems (IDSs) are widely used to protect the core network against external intrusions. Most of the existing research works design anomaly detection models for a specific set of traffic attributes. In fact, it is difficult for us to find the specific correspondence between traffic attributes and attack behaviors. Worse, some traffic attributes will be missing in the IoT environment, which further increases the difficulty of anomaly analysis. In traditional solutions, the missing attributes are usually filled with zero or mean values. Sometimes, the attributes are directly discarded. Both of these methods may result in lower detection accuracy. To solve this problem, we propose an intrusion detection method based on multiple-kernel clustering (MKC) algorithms. Be different from zero value filling and mean value filling, the proposed method completes the absent traffic property through similarity calculation. Experimental results show that this method can effectively improve the clustering accuracy of incomplete sampled data, at the same time it can reduce the sensitivity of the anomaly detection model to the selection of traffic feature, and has a better tolerance for poor-quality traffic sampled data.

Keywords Network intrusion detection · Anomaly detection · Multiple kernel clustering · Machine Learning

1 Introduction

The 5G network provides higher bandwidth and lower latency for edge IoT devices to access the core network, which increases the efficiency of collaboration between edge side application and cloud-side service [61]. But at the same time, it also expands the attack surface of the core network, which makes the enterprise network face greater security threats. Due to limited computing capability, it is difficult for IoT devices to deploy heavy-weight security protection mechanisms [27, 47, 65] Security incidents that occurred in recent years have shown that by controlling IoT devices, it is possible to launch attacks on enterprise core networks or Internet infrastructure. For example, on Friday October 21, cybercriminals launched DDoS attacks on the DNS system in the US-East region via 30,000 maliciously manipulated Wi-Fi cameras, which is known as Dyn cyberattack incident [20]. The attack caused major Internet platforms and services to be unavailable to large swathes of users in Europe and North America. Similar incident includes the Ukrainian

✉ Hui Lu
luhui@gzhu.edu.cn

Ning Hu
huning@gzhu.edu.cn

Zhihong Tian
tianzhihong@gzhu.edu.cn

Xiaojiang Du
dxj@ieee.org

Mohsen Guizani
mguizani@ieee.org

[1] Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou 510006, China

[2] Department of Computer and Information Sciences, Temple University, Philadelphia, USA

[3] Computer Science and Engineering Department, Qatar University, Doha, Qatar

Power Center incident, hackers invaded Ukraine's power center through IoT devices [36].

As a network security protection technology, Intrusion Detection Systems (IDSs) are widely used to protect the core network against external intrusions. It was first proposed in 1980, the goal is to determine whether there is any malicious intrusion through behavior characteristics monitoring and analysis [6]. The early intrusion detection technology was focus on host security. With the rapid development of network technology, the research focus of intrusion detection technology turned to determine the intrusion behavior by analyzing network traffic [30]. Basically, intrusion detection includes misuse detection and anomaly detection [1, 5]. In the anomaly detection model, when the user's behavior pattern deviates from the normal standard by more than the threshold, it is regarded as abnormal behavior. In the case of the abuse detection model, when the user behavior pattern matches the existing malicious behavior pattern, it will be regarded as misuse behavior. Therefore, the key to improving the accuracy of intrusion detection lies in the recognition of network traffic patterns. Unfortunately, with the rapid development of operating systems, application software, and network technologies, both normal user behavior and attack behaviors are constantly changing. Especially the endless system vulnerabilities that lead to an ever-evolving variety of network attack methods which make the update speed of the signature database of malicious behaviors difficult to meet the detection requirements.

Over the last three decades, numerous machine learning algorithms have been widely utilized in network intrusion detection to make up this deficiency of manual analysis, such as support vector machines (SVM) (Vladimir [60], artificial neural networks (ANN) [29] and decision trees [44]. These research works show that machine learning methods can indeed improve the analysis efficiency of abnormal traffic, and can find some abnormal behaviors that cannot be identified by manual analysis [38, 52, 58, 64, 67–69]. However, judging from recent research results, there are still several challenges that need further exploration. First of all, how to perceive intrusion behavior without a known traffic signature database? As far as we know, most intrusion detection methods based on supervised learning and semi-supervised learning require prior data for training. The detection accuracy of these methods for unknown intrusions is generally low. Secondly, most traffic sampling data contains different attributes, for example, DARPA KDD CUP99 and NSL-KDD. Some research efforts try to improve the accuracy of anomaly detection by optimizing feature selection [24]. Is there another method that can reduce the sensitivity of detection accuracy to sample feature selection? Finally, when some packet attributes of the network traffic sampling data are missing, how to perform anomaly analysis based on these incomplete data?

1.1 Motivation

5G network introduces new application scenarios such as enhanced mobile bandwidth (eMBB), massive machine-type communications (mMTC), and ultra-reliable low-latency communications (uRLLC), which make it become a mobile communication infrastructure for a new generation of IoT information systems [19]. While the 5G network provides high bandwidth and low latency services, it also brings more severe security challenges to the core network [41, 42, 46, 57, 63], which endues the intrusion detection of IoT systems with different characteristics from the traditional network systems.

First, in the scenario of massive machine-type communications (mMTC), massive IoT devices will generate a large amount of network packets, causing huge analysis pressure for the intrusion detection system. According to statistics, by 2019, the data center traffic is three times that of 2014, with an average annual growth rate of 30%. In fact, the traffic base was very large, with 2.1ZB in 2014. As far as we know, there is no machine learning algorithm that can analyze such huge network traffic packets at line speed. Under normal circumstances, intrusion detection systems can only select a subset of network traffic for anomaly analysis. Therefore, how to select sampled data and the packet properties of network traffic have become important technical issues.

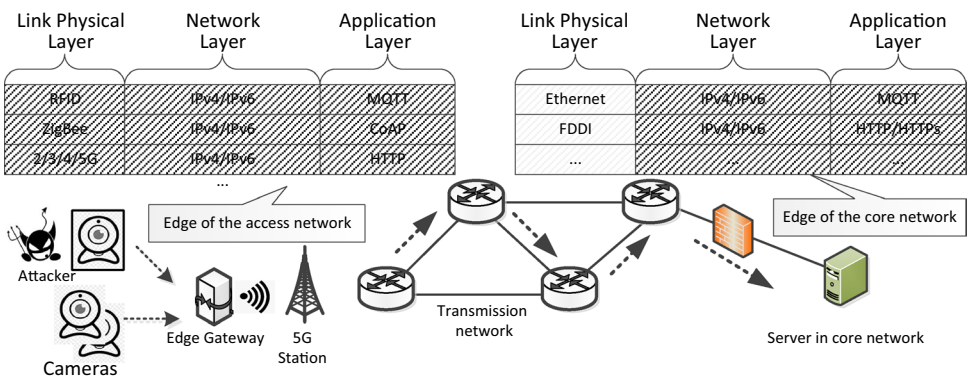

Fig. 1 Protocol stack of 5G IoT network

Secondly, a variety of IoT network protocols and emerging computing models make traffic analysis more complex. Figure 1 shows a typical 5G IoT application scenario. In this scenario, when data traffic begins from the IoT terminal through the edge gateway and transmission network to the edge of the core network, the data link layer protocol and application layer protocol will change. Most of existing research works on intrusion detection mainly focus on the analysis of the characteristics of the transport layer protocol. In fact, malicious attackers can implement network intrusion behaviors based on data link layer protocols and application layer protocols. On the other hand, the diversity of protocols results in very rich types and values of protocol fields, which brings the difficulty of feature selection and higher computational complexity to the analysis algorithm. In fact, many IoT intrusion detections not only need to analyze the network layer protocol but also need to analyze the link layer and physical layer protocols, which further increases the complexity of sample sampled data and the packet property selection [10].

Finally, since the access points of IoT terminals are scattered and numerous, the sample data provided by different access points may lack some protocol properties value. For example, the Aegean Wi-Fi Intrusion Dataset (AWID) is a comprehensive 802.11 network dataset, which was derived from real Wi-Fi traffic traces in 2015. Figure 2 shows the results of our analysis on the AWID data. It can be found that among the 575,643 samples, 329,821 of them have missing attributes, reaching more than 40%. If there is no corresponding data filling mechanism, cluster analysis will be difficult to conduct.

In short, the particularity of 5G/IoT networks makes intrusion detection and analysis more difficult and complicated, especially in terms of traffic properties selection and in case of properties absent. Traditional misuse detection-based IDS use supervised learning or semi- supervised learning method to recognized the malicious behavior. Most of these methods rely on the selection of protocol properties and massive prior data. Research results show that misuse detection-based IDS is very successful on detecting known intrusions, but is poor at unknown abnormal behaviors and 0-day attacks.

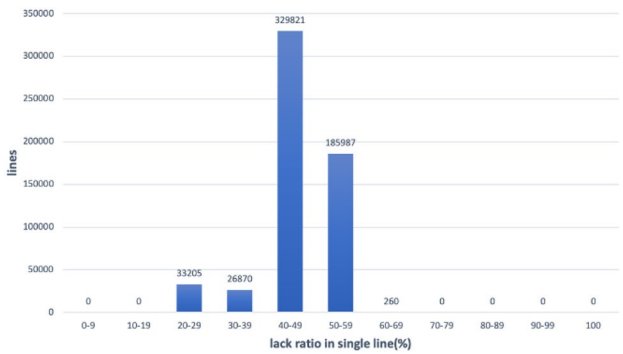

Fig. 2 Incomplete sampling data of AWID

Although the composition of the IoT protocol stack is relatively complex, the behavior of IoT application is not complicated due to the resource constraints of the IoT devices. At the same time, the distribution of IoT devices is scattered, making it difficult to organize complex collusion attacks. Therefore, there are obvious differences between the abnormal and normal behaviors. It is possible to divide these behaviors by using the clustering method, and then further determine which one is abnormal. Based on such regard and assumption, in this study, we choose a clustering method based on unsupervised learning for anomaly detection. Moreover, we try to use multi-view learning methods to reduce the influence of a single attribute on the detection results. In order to further enhance the practicality of the algorithm, the clustering analysis algorithm we proposed considers sample data with missing attributes for the first time.

1.2 Contribution

Based on the above considerations, we try to propose an anomaly detection method based on unsupervised learning in this paper. The major contributions of the proposed work are summarized as follows:

- Aiming at the difficulty of selecting traffic attributes in anomaly detection, we propose an analysis method based on multiple kernel clustering (MKC) algorithm. To reduces the sensitivity of anomaly detection accuracy to single feature selection, our method constructs multiple base kernels via different feature properties and combines these kernels to improve clustering performance.
- We further consider the pre-processing of sampled data with incomplete attributes. As we know, the existing multiple kernel clustering methods cannot address the situation when some feature properties of the traffic are absent. Most of the traditional solutions adopt the methods of mean value filling or zero value filling and even discard these absent properties, which may result in a lower detection rate. Our method supplements the incomplete base kernel with approximate values which are calculated based on sample data.
- Since it can only handle continuous numerical data, the existing multiple kernel clustering methods are mainly used for image recognition. This paper proposes a method to deal with characters and non-continuous data, such as enumerated type data, IP address and so on, which expands the application field of multiple kernel

clustering method. We also evaluate the performance of the design model on multiple benchmark data sets.

1.3 Organization

The remainder of this paper is organized as follows: Sect. 2 discusses related studies and analyses their limitations. The proposed anomaly detection based on multiple kernel K-means clustering algorithm is described in Sect. 3. Section 4 presents the results of experiments and evaluations. Section 5 presents the conclusions and future work.

2 Related works

The 5th generation of mobile communication technology (5G for short), as an extension of the 4G (LTE-A, WiMAX-A) system, provides three types of services, including enhanced mobile broadband (eMBB), massive machine-type communications (mMTC), and ultra-reliable and low latency communications (URLLC) (ITU 2017) These services ensure the Internet of Things applications such as machine-to-machine (M2M), Vehicles-to-Everything(V2X), device-to-everything (D2E), and provide the same user experience as that of the wire network. Unfortunately, in the context of 5G era, IoT networks are facing much greater security risks. The attack surface of 5G IoT not only involves terminal devices but also includes communication channel and application software. The expansion of the attack area of IoT makes the intrusion detection of IoT network become a research hotspot. There have been many literatures summarizing the types and research directions of IoT intrusion [3, 14, 54, 35]. According to these literatures, the research on IoT intrusion detection mainly includes intrusion detection method, intrusion detection system deployment, security threat model and verification method.

With the rapid development of machine learning technology, research on intrusion detection based on machine learning has attracted widespread attention in recent years. In general, IoT intrusion detection methods mainly include misuse detection, anomaly detection, and hybrid detection. Because the research content of this article is mainly aimed at the network intrusion detection problem of IoT, the terminal intrusion detection problem is not discussed here. The key idea of intrusion detection technology is to determine whether there is an intrusion incident happened by analyzing the hidden features in the sampled data, which include system log, network traffic, and so on. Because machine learning has inherent advantages in analyzing data, a large number of intrusion detection technologies based on machine learning algorithms have been proposed, which mainly include: unsupervised machine learning, supervised machine learning, and deep learning [22].

2.1 Supervised learning based intrusion detection

Before deep learning technology was proposed, intrusion detection technology based on supervised learning was the main research direction.

1. K-nearest Neighbor

K-nearest neighbor (k-NN) is a sample classification technique that does not require parameters. Data classification is achieved by calculating the Euclidean distance of the input sample (Soucy and Mineau). The k-NN classifier is widely used in the field of intrusion detection. For example, Liang et al. [39] use the Minimum Dependence Maximum Significance (MDMS) algorithm to select 6 features from the KDD1999 data set and use KNN to predict network traffic. The proposed method can better identify probe attacks and denial of service attack. The accuracy of the k-NN classifier is mainly affected by the value of k [17].

2. Support Vector Machines.

Compared with other algorithms, the method of support vector machine (SVM) can solve the problem of small samples and has better generalization ability. SVM is very suitable for classifying data sets that contain large features. SVM is simple to implement and easy to expand and can perform anomaly detection in real-time. Therefore, a large number of SVM-based intrusion detection methods have been proposed. For example,Ahmim et al. [2] use Z-score to normalize KDD1999 data, and uses compressed sampling method for feature compression, combined with SVM to classify the compression results. The proposed method has a low false positive rate (FPR) and can effectively detect denial of service attacks, probe attacks and other attacks. Chen et al. [13] use logarithms of the marginal density ratios (LMDRT) as a feature conversion technique to construct an IDS based on SVM.

3. Decision Trees.

The Decision tree (DT) has low computational complexity, and the constructed rules are easy to understand. Therefore, it is also widely used in the field of intrusion detection. For example, Senthilnayaki et al. [51] proposed a smart grid advanced metering infrastructure IDS based on a CART decision tree, and the experimental accuracy rate on the CICIDS2017 data set was 99.66%. According

to the method proposed, the highest accuracy rate on the CICIDS2017 data set is 96.665%, and the lowest false alarm rate (FAR) is 1.145%. The accuracy rate is higher than that of Naive Bayes (74.528%) and other algorithms. Effectively identify normal traffic and abnormal traffic.

4. Naive Bayes Networks.

Naive Bayes Networks (NB) is a probabilistic graph model that predicts the occurrence probability of events based on prior observations of similar events [16]. Naive Bayes Networks is mainly used to classify normal and abnormal behavior based on previous observations in a supervised learning model. The logic of the NB classifier is simple and easy to implement. It only needs a few samples to train and can obtain satisfactory results [56]. For example, Nuo [45] proposes a classification method based on the Naive Bayes model, tested on the KDD1999 data set, can effectively detect Trojan horse attacks, fake message attacks, denial of service attacks and remote user unauthorized access attacks, detection The detection rate (DR) reaches 87–97%.

5. Ensemble Classifiers.

To improve the performance of a single classifier, the ensemble classifier is proposed. The main idea is to combine multiple weak learning algorithms and then generate majority voting results for classification [32]. Bosman et al. [11] show that the EL algorithm produces more accurate results than each member classifier, but at the same time, due to the parallel use of multiple classifiers, the accuracy of EL leads to the cost of increasing time complexity [9].

2.2 Unsupervised learning based intrusion detection

Intrusion detection technology based on unsupervised learning performs intrusion detection on sample data without reference classifiers. The k-means algorithm has the advantages of strong interpretability and fast convergence speed. When k-means is used in combination with other classification algorithms, it can effectively improve the detection rate. For example, Shah et al. [53] uses an improved k-means algorithm to construct a high-quality training data set and uses a combination of SVM and an extreme learning machine (ELM) algorithm to construct an IDS, which can effectively identify denial of service attacks. The traditional k-means algorithm is sensitive to the initial value of the cluster center, and the accuracy is easily affected by noisy data and incomplete data [4]. Since the sample data can be represented using different units/scales, most existing distance-function or density-function based AD algorithms are sensitive to how data is expressed. To avoid the problem, literature [7] proposed an unsupervised stochastic forest-based Anomaly Detection algorithm, which is called usfAD. Noisy data has a greater negative impact on the accuracy of the clustering algorithm. Most of the existing clustering algorithms adopt a noise-free assumption. Iam-On [31] uses the multi-kernel k-means clustering method to analyze the noisy data. The experimental result shows that the approach is robust to the low level of noise. Guo et al. [28] study unsupervised anomaly detection in IoT systems and develops a GRU-based Gaussian Mixture VAE scheme, called GGM-VAE. According to the experiment results of simulation, the proposed scheme gets a 47.88% improvement in F1 scores on average.

The intrusion detection method based on the unsupervised algorithm can effectively deal with the large-scale traffic data problem that is increasing year by year in the network, reduce the computational overhead, and improve the detection accuracy. Therefore, with the increase of massive amounts of data in the network, unsupervised machine learning algorithms will be more widely used, but they are sensitive to noise and outliers, which are also problems faced by unsupervised machine learning algorithms in the field of intrusion detection.

2.3 Deep learning based intrusion detection

Deep learning technology can use a hierarchical structure to perform unsupervised feature learning and pattern classification of data, integrate feature extractors and classifiers into a framework, without the need to extract features manually. Deep learning can effectively process large-scale network traffic data and has higher efficiency and detection rate than traditional machine learning methods, but the training process is more complicated and the model interpretability is weak. Intrusion detection technologies based on deep learning mainly include deep auto encoders (AEs) [66], restricted boltzmann machine (RBM) [26], deep belief network (DBN) [23], recurrent neural network (RNN) [34], etc.

With the tremendous enrichment of machine learning theories, techniques such as reinforcement learning [12] and extreme learning [21] have also been applied to network intrusion detection.

3 A multiple-Kernel clustering-based anomaly detection scheme

3.1 Preliminary

3.1.1 Kernel K-means clustering (KKM)

Clustering is a type of unsupervised machine learning method, which can generalize the observed values into certain classes according to their features. By analyzing

enormous research results of intrusion detection, we found that the network behaviors with different purposes often caused network traffic with different characteristics. Based on this assumption, the network traffic caused by abnormal behaviors could be distinguished from normal network traffic. Here, we hope to classify network traffic through clustering methods.

K-means is a distance-based clustering algorithm, which is widely used due to its simplicity and ease of implementation. But the K-means algorithm does not perform well when processing linearly inseparable data. For example, to distinguish abnormal traffic from normal traffics, we treat each IP packet as a feature vector with multiple attributes, such as IP address, port, protocol type, and so on. Due to the mutual influence of these attributes, it is difficult to achieve linear segmentation in low-dimensional space. Therefore, it is difficult to obtain satisfactory clustering results directly using the K-means algorithm.

To make up for the shortcomings of the K-means algorithm, the kernel K-means algorithm is proposed. It assumes as follows: a set of point which cannot be linearly divided in a low-dimensional space is more likely to become linearly separable when it has been mapped into a high-dimensional space.

The key idea of the kernel clustering method is to map the data points of the input set into a high-dimensional feature space through a non-linear mapping and perform clustering in a new feature space. Because the nonlinear mapping increases the probability that the data points are linearly separable, a more accurate clustering result could be achieved.

The mapping function $\boldsymbol{\Phi}$ is defined as follows:

$$\boldsymbol{\Phi} : \boldsymbol{x} \mapsto \boldsymbol{\Phi}(\boldsymbol{x}) \in \boldsymbol{F}, \boldsymbol{x} \in \boldsymbol{X} \tag{1}$$

where $\boldsymbol{X}$ is the original input data set, and $\boldsymbol{F}$ is the high-dimensional feature space.

For example, if we want to map feature $\boldsymbol{x}$ into a three-dimensional space, the mapping function $\boldsymbol{\Phi}$ can be represented as follows.

$$\boldsymbol{\Phi}(\boldsymbol{x}) = \left(\boldsymbol{x}, \boldsymbol{x}^2, \boldsymbol{x}^3\right)^{\boldsymbol{T}} \tag{2}$$

For all $\boldsymbol{x}, \boldsymbol{z} \in \boldsymbol{X}$, the original data inner product is $\boldsymbol{x}, \boldsymbol{z}$, and the feature inner product in feature space is $\boldsymbol{\Phi}(\boldsymbol{x}), \Phi(\boldsymbol{z})$. Since the computational complexity of calculating $\boldsymbol{\Phi}(\boldsymbol{x}), \boldsymbol{\Phi}(\boldsymbol{z})$ in high-dimensional feature space is high, to improve the efficiency, we define kernel function $\boldsymbol{K}$ as follows. For all $\boldsymbol{x}, \boldsymbol{z} \in \boldsymbol{X}$ satisfies

$$\boldsymbol{K}(\boldsymbol{x}, \boldsymbol{z}) = \boldsymbol{\Phi}(\boldsymbol{x})^{\boldsymbol{T}} \boldsymbol{\Phi}(\boldsymbol{z}) \tag{3}$$

Obviously, the computational complexity of calculating the kernel function in the low-dimensional space is lower than the complexity of directly calculating the vector inner product in the high-dimensional space.

Given a kernel function $\boldsymbol{K} : \boldsymbol{R}^{\boldsymbol{N}} \times \boldsymbol{R}^{\boldsymbol{N}} \mapsto \boldsymbol{R}$ and a data set $\left\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(\boldsymbol{M})}\right\}$, where $\boldsymbol{x}^{(\boldsymbol{i})} \in \boldsymbol{R}^{\boldsymbol{N}}$ and $\boldsymbol{i} = 1, \ldots, \boldsymbol{M}$. For each $\boldsymbol{x}^{(\boldsymbol{i})}$ and $\boldsymbol{x}^{(\boldsymbol{j})}$, we calculate $\boldsymbol{K}_{\boldsymbol{ij}} = \boldsymbol{K}\left(\boldsymbol{x}^{(\boldsymbol{i})}, \boldsymbol{x}^{(\boldsymbol{j})}\right)$ and get a kernel matrix $\boldsymbol{MK}_{\boldsymbol{M \times M}}$, According to Mercer's theorem [43], the function $\boldsymbol{K}$ is an effective kernel function, if and only if the kernel matrix $\boldsymbol{MK}_{\boldsymbol{M \times M}}$ is a positive semidefinite matrix. According to this conclusion, for a given data set, we do not need to find a mapping function $\boldsymbol{\Phi}$, but only need to construct the kernel matrix with the training set, and determine whether it is a positive semi-definite matrix.

As one of the most important machine learning techniques, the kernel method provides a powerful and unified learning framework. It allows researchers to focus on algorithm design without considering the attributes of the data itself, such as strings, vectors, text, and graphs. The key

Table 1 Kernel K-Means Clustering Algorithm

Algorithm 1 Kernel K-Means Clustering

Input: Original input data set $\boldsymbol{X} = \{x^{(1)}, \ldots, x^{(M)}\}$, number of Clusters $\boldsymbol{N}$, kernel function $\boldsymbol{K}$.
Output: $\boldsymbol{N}$ disjoint clusters $\{\boldsymbol{C_1}, \boldsymbol{C_2}, \ldots, \boldsymbol{C_N}\}$

1. Initialize the $\boldsymbol{N}$ disjoint clusters: $\{\boldsymbol{C_1}, \boldsymbol{C_2}, \ldots, \boldsymbol{C_N}\}$;
2. For each data $x^{(i)}$ and every cluster $\boldsymbol{C_k}$
3. Calculate $\left\|\boldsymbol{\Phi}\left(\boldsymbol{x}^{(\boldsymbol{i})}\right) - \overline{\boldsymbol{m_k}}\right\|^2$ with Eq. (4) ;
4. Find the minimal $\left\|\boldsymbol{\Phi}\left(\boldsymbol{x}^{(\boldsymbol{i})}\right) - \overline{\boldsymbol{m_k}}\right\|^2$;
5. Add $x^{(i)}$ into cluster $\boldsymbol{C_k}$;
6. While $\{\boldsymbol{C_1}, \boldsymbol{C_2}, \ldots, \boldsymbol{C_N}\}$ is not converged go to step 2 ;
7. Return $\{\boldsymbol{C_1}, \boldsymbol{C_2}, \ldots, \boldsymbol{C_N}\}$;

idea of kernel k-means [50] clustering algorithm is mapping data from input space to a higher dimensional feature space through kernel function, such as Polynomial kernel function, Gaussian kernel function and Sigmoid kernel function, and then use k-means algorithm in the feature space.

The Kernel K-Means clustering model training algorithm is described in Table 1.

For kernel K-means clustering, the problem-solving model is described as follows:

Given data set $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(M)}\}$, our objective is to partition this dataset into $\boldsymbol{N}$ disjoint clusters $\{\boldsymbol{C}_1, \boldsymbol{C}_2, ..., \boldsymbol{C}_N\}$ by using kernel K-means method. First, we use the function $\boldsymbol{\Phi}(x)$ mapping the original $\boldsymbol{x}^{(i)}$ into a reproducing Hilbert space $\boldsymbol{H}$ [18].

Let $\overline{\boldsymbol{m}_k}$ be the mean of the $\boldsymbol{k}$-th cluster. The optimization objective of kernel K-means clustering is to minimize the sum of square of the within-cluster distance. By adopting a decision function $\boldsymbol{I}(\boldsymbol{x}^{(i)} \in \boldsymbol{C}_n) \rightarrow \{0, 1\}$, where $\boldsymbol{I}(\boldsymbol{x}^{(i)} \in \boldsymbol{C}_n) = 1$ if $\boldsymbol{x}^{(i)} \in \boldsymbol{C}_n$ is true, nor $\boldsymbol{I}(\boldsymbol{x}^{(i)} \in \boldsymbol{C}_n) = 0$, it can be represented as Eq. (4),

$$\min\left(\sum_{i=1}^{M}\sum_{k=1}^{\mathrm{N}} \boldsymbol{I}(\boldsymbol{x}^{(i)} \in \boldsymbol{C}_k)\|\boldsymbol{\Phi}(\boldsymbol{x}^{(i)}) - \overline{\boldsymbol{m}_k}\|^2\right) \tag{4}$$

where $\overline{\boldsymbol{m}_k} = \frac{\sum_{i=1}^{M} \boldsymbol{I}(\boldsymbol{x}^{(i)} \in \boldsymbol{C}_k)\boldsymbol{\Phi}(\boldsymbol{x}^{(i)})}{\sum_{i=1}^{M} \boldsymbol{I}(\boldsymbol{x}^{(i)} \in \boldsymbol{C}_k)}$.

In Eq. (4), $\|\boldsymbol{\Phi}(\boldsymbol{x}^{(i)}) - \overline{\boldsymbol{m}_k}\|^2$ can be calculate as follows:

$$\boldsymbol{K}(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(i)}) - \frac{2\sum_{j=1}^{M} \boldsymbol{I}(\boldsymbol{x}^{(j)} \in \boldsymbol{C}_k)\boldsymbol{K}(\boldsymbol{x}^{i}, \boldsymbol{x}^{\mathrm{i}})}{\sum_{j=1}^{M} \boldsymbol{I}(\boldsymbol{x}^{(j)} \in \boldsymbol{C}_k)} + \frac{\sum_{j=1}^{M}\sum_{l=1}^{M} \boldsymbol{I}(\boldsymbol{x}^{(j)} \in \boldsymbol{C}_k)\boldsymbol{I}(\boldsymbol{x}^{(l)} \in \boldsymbol{C}_k)\boldsymbol{K}(\boldsymbol{x}^{j}, \boldsymbol{x}^{l})}{\sum_{j=1}^{M}\sum_{l=1}^{M} \boldsymbol{I}(\boldsymbol{x}^{(j)} \in \boldsymbol{C}_k)\boldsymbol{I}(\boldsymbol{x}^{(l)} \in \boldsymbol{C}_k)} \tag{5}$$

3.1.2 Multiple-kernel K-means clustering (MKKM)

The traditional kernel method is a single-kernel method based on a single feature space and cannot effectively process the huge size and heterogeneous information. The limitations of the kernel method are more significant, and the construction and selection of the kernel function is still an open problem. To solve the above problems, multiple kernel learning was proposed [25].

In the case of multi-kernel, let $\boldsymbol{X} = \{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(M)}\}$ be a data set, each $\boldsymbol{x}^{(i)}$ has $\boldsymbol{m}$ properties and $\boldsymbol{\Phi}_k(\cdot) : \boldsymbol{x} \in \boldsymbol{X} \mapsto \boldsymbol{H}_k$ is the $\boldsymbol{k}$-th feature mapping function which maps X into a reproducing kernel Hilbert space $\boldsymbol{H}_k (1 \leq \boldsymbol{k} \leq \boldsymbol{m})$. So, for each $\boldsymbol{x}^{(i)}$, it can be represented as $\boldsymbol{\Phi}_\beta(\boldsymbol{x}) = \left[\beta_1\boldsymbol{\Phi}_1(\boldsymbol{x})^T, \ldots, \boldsymbol{\beta}_m\boldsymbol{\Phi}_m(\boldsymbol{x})^T\right]^T : \boldsymbol{x} \in \boldsymbol{X} \mapsto \boldsymbol{H}_k$, and $\boldsymbol{\beta} = \left[\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_m\right]^T$ is a coefficients matrix of m base kernels $\boldsymbol{k}_p(\cdot,\cdot)_{p=1}^{m}$. So, the multi-kernel function can be defined as follows:

$$\boldsymbol{K}_\beta(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{\Phi}_\beta(\boldsymbol{x}_i)^T\boldsymbol{\Phi}_\beta(\boldsymbol{x}_j) = \sum_{p=1}^{m} \boldsymbol{\beta}_p^2\boldsymbol{k}_p(\boldsymbol{x}_i, \boldsymbol{x}_j) \tag{6}$$

We can get Eq. (7) as follows:

$$\begin{aligned} &\min_{\boldsymbol{H},\boldsymbol{\beta}} \boldsymbol{Tr}\left(\boldsymbol{K}_\beta\left(\boldsymbol{I}_n - \boldsymbol{H}\boldsymbol{H}^T\right)\right) \\ &\boldsymbol{s.t.}\ \boldsymbol{H} \in \boldsymbol{R}^{n\times k}, \boldsymbol{H}^T\boldsymbol{H} = \boldsymbol{I}_k, \boldsymbol{\beta}^T 1_m = 1, \\ &\boldsymbol{\beta}_p \geq 0,\ \forall \boldsymbol{p} \end{aligned} \tag{7}$$

In Eq. (7), $\boldsymbol{I}_k$ is an identity matrix with size $\boldsymbol{k} \times \boldsymbol{k}$. By adjusting the $\boldsymbol{H}$ and β, we can get the optimization result of Eq. 7 in two ways: (1) Optimizing $\boldsymbol{H}$ while keeping β fixed. In this way, $\boldsymbol{H}$ can be obtained by solving a kernel k-means clustering optimization problem shown in Eq. (7); (2) optimizing β while keeping $\boldsymbol{H}$ fixed. In this way, β can be optimized via solving the following quadratic programming with linear constraints.

3.2 The proposed method

3.2.1 Basic idea

The effectiveness of the multi-kernel clustering algorithm is not only proven in theoretical research but also has been successful in many application fields, such as image analysis, pattern recognition, etc. For IoT network intrusion detection, in the absence of prior data and classification specification, we can firstly perform clustering analysis on the sampled data, and then confirm the clustering results. Based on this consideration, we try to apply the multi-kernel clustering method for network traffic anomaly detection.

Unfortunately, suffering from problems such as incomplete data and diverse data types, the multi-kernel clustering method is difficult to directly use for network abnormality analysis. In this paper, we try to propose an innovative method to solve the two problems as follows:

1. Incomplete data. As we know most existing multi-kernel clustering algorithms cannot perform cluster analysis when the kernel matrix is incomplete. Due to the lack of some attributes in the sampled data of the IoT network, it is difficult to construct a complete kernel matrix. For this reason, we try to adopt a method that integrates the information source filling and clustering tasks into the same optimization objective, to better combine the filling process and the clustering process and improve the performance of the algorithm.
2. Diverse data. In the image recognition application, the similarity of data can be obtained by calculating the Euclidean distance between pixels. But in the network traffic analysis, the situation is completely different. The diversity of IoT protocols causes the protocol attributes

of messages to differ not only in data types but also in their value ranges. This will cause some attributes to have too much weight in the clustering process, which will affect the final clustering results. We need to preprocess these data before clustering.

3.2.2 Multiple kernel K-means with incomplete kernels

The multiple kernel K-means clustering method is suitable for cluster analysis of data with multiple attributes. But it has a constraint, that is, it requires a complete kernel matrix. Due to the complexity and diversity of the IoT network environment, in actual situations, the sampling data that can be used for intrusion detection often has the problem of missing attributes, such as the example in Fig. 2. Most of the existing solutions use mean value filling or zero value filling to complete the kernel matrix. These approaches do not fully explore the potential correlations between the sample data and are not conducive to improving the detection accuracy. Inspired by the work of literature [40] and [62], this section proposes a method of constructing a kernel matrix based on incomplete data, filling in missing attribute values based on the similarity of sampled data, thereby improving the detection accuracy. The main principle of this method is described as follows:

For ease of explanation, we assume that there is a multi-view dataset $\boldsymbol{X} = \{\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_m\}$ which has m views, and each view $\boldsymbol{X}_p (1 \le p \le m)$ has two attributes, which can be denoted as follows:

$$\boldsymbol{X}_p = \left[\boldsymbol{X}_p^{(o)}, \boldsymbol{X}_p^{(u)}\right] =^{\mathrm{T}}$$

We assume that at least one sample in each view is observable. Suppose that the first attribute $\boldsymbol{X}_p^{(o)}$ is observable, and the second $\boldsymbol{X}_p^{(u)}$ is missing. Before cluster analysis, we must fill the absent information. The key idea is as follows:

To compute the kernel matrix between sample, we need to get a positive definite kernel function $\boldsymbol{\kappa}(\cdot,\cdot)$. We first calculate the k nearest neighbors of the observable attribute $\boldsymbol{X}_p^{(o)}$ and then calculate the kernel matrix between $\boldsymbol{X}_p^{(o)}$ and its $\boldsymbol{k}$ nearest neighbors. The kernel matrix is expressed as $\mathbf{K}_p^{(oo)}$. At the same time, the constraint that must be met is that the complete kernel matrix of the observable source should be equal to the known kernel matrix. So, the complete kernel matrices $\{\mathbf{K}_p\}_{p=1}^m$ should minimize the following formulation.

$$\begin{aligned} &\min_{\{\mathbf{K}_p\}_{p=1}^m, \mathbf{H}} \sum_{p=1}^{m} \mathrm{Tr}\left(\mathbf{K}_p\left(\mathbf{I}_{\boldsymbol{k}} - \mathbf{H}\mathbf{H}^{\mathrm{T}}\right)\right) \\ &s.t. \mathbf{K}_p\left(\boldsymbol{s}_p, \boldsymbol{s}_p\right) = \mathbf{K}_p^{(oo)}, \mathbf{K}_p \succcurlyeq 0, \forall p, \mathbf{H} \in \mathbb{R}^{n \times k}, \mathbf{H}^{\mathrm{T}}\mathbf{H} = \mathbf{I}_k, \end{aligned} \tag{8}$$

where $\boldsymbol{s}_p$ are the indices of missing instances of the p-th view.

The optimization goal w.r.t $\{\mathbf{K}_p\}_{p=1}^m$ in Eq. (8) is a programming problem with positive semi-definite constraints, and its computational efficiency is rather low when solving large-scale optimization problems. To overcome this defect, we propose to optimize the objective function:

$$\min_{\mathbf{K}_p} \mathrm{Tr}(\mathbf{K}_p\mathbf{U}), s.t. \mathbf{K}_p\left(\boldsymbol{s}_p, \boldsymbol{s}_p\right) = \mathbf{K}_p^{(oo)}, \mathbf{K}_p \succcurlyeq 0 \tag{9}$$

where $\mathbf{U} = \mathbf{I}_{\boldsymbol{k}} - \mathbf{H}\mathbf{H}^{\mathrm{T}}$. Consider that the decomposition of $\mathbf{K}_p = \mathbf{A}_p\mathbf{A}_p^{\mathrm{T}}$, $\mathbf{A}_p^{(o)}$ is the observable part and $\mathbf{A}_p^{(u)}$ is the unobservable part, i.e., $\mathbf{A}_p = \left[\mathbf{A}_p^{(o)}; \mathbf{A}_p^{(u)}\right]$. We can transform Eq. (9) into:

$$\min_{\mathbf{A}_p^{(u)}} \mathrm{Tr}\left([\mathbf{A}_p^{(o)}; \mathbf{A}_p^{(u)}]^{\mathrm{T}} \begin{bmatrix} \mathbf{U}^{(oo)} & \mathbf{U}^{(ou)} \\ \mathbf{U}^{(ou)\mathrm{T}} & \mathbf{U}^{(uu)} \end{bmatrix} \left[\mathbf{A}_p^{(o)}; \mathbf{A}_p^{(u)}\right] \right) \tag{10}$$

where $\begin{bmatrix} \mathbf{U}^{(oo)} & \mathbf{U}^{(ou)} \\ \mathbf{U}^{(ou)\mathrm{T}} & \mathbf{U}^{(uu)} \end{bmatrix}$ is a blocked form of $\mathbf{U}$.

By taking the derivative of Eq. (10) with respect to $\mathbf{A}_p^{(u)}$, we can obtain the close solution of $\mathbf{A}_p^{(u)}$ as follows:

Table 2 Filling Algorithm for Incomplete Kernel Matrix

Algorithm 2 Filling Algorithm for Incomplete Kernel Matrix

Input: Original input data set $\boldsymbol{X} = \{\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_m\}$, Number of Clusters k.

Output: *complete kernel matrix.*

1. Calculate the The kernel matrix of the observable samples in the p-th view;
2. While the objective function is not converged:
3. Keep $\mathbf{H}$ fixed, update $\{\mathbf{K}_p\}_{p=1}^m$;
4. Keep $\{\mathbf{K}_p\}_{p=1}^m$ fixed, update $\mathbf{H}$;
5. **Output**: complete kernel matrix

Table 3 Symbol description of algorithm 2

Symbol	Description
$\boldsymbol{X}$	A dataset with m views
$\boldsymbol{X}_p$	The p-th view of dataset
$\mathbf{K}_p$	Kernel matrix of the p-th view
$\mathbf{K}_p^{(oo)}$	The kernel matrix of the observable samples in the p-th view
$\mathbf{A}_p$	The decomposition of $\mathbf{K}_p$
$\mathbf{H}$	The clustering results
$\mathbf{I}_k$	An identity matrix with size $k \times k$

$$\mathbf{A}_p^{(u)} = \left(\mathbf{U}^{(uu)}\right)^{-1}\mathbf{U}^{(ou)\mathrm{T}}\mathbf{A}_p^{(o)} \tag{11}$$

Moreover, the optimal $\mathbf{H}$ can be obtained by taking the k eigenvectors corresponding the largest k eigenvalues of $\sum_{p=1}^{m}\mathbf{K}_p$.

The filling algorithm for Incomplete kernel matrix is described in Table 2 and the description of some symbols are listed in Table 3.

3.2.3 Algorithm computation complexity

The construction of the kernel matrix of the observable samples in the $\boldsymbol{p}$-th view is basically $\mathcal{O}(n^2)$, where n is the number of samples. When $\mathbf{H}$ is fixed, the updating of each kernel matrix will take $\mathcal{O}(n^3)$ time, due to the calculation of the inverse of $\mathbf{U}^{(uu)}$. So the updating of $\{\mathbf{K}_p\}_{p=1}^m$ takes $\mathcal{O}(mn^3)$ time in total. And the updating of $\mathbf{H}$ needs to conduct eigen decomposition on a $n \times n$ matrix, which costs $\mathcal{O}(\boldsymbol{n}^3)$ time. Assume that the iteration number is T. The computation complexity of our algorithm is $\mathcal{O}(\boldsymbol{n}^2 + \boldsymbol{T}(\boldsymbol{m}+1)\boldsymbol{n}^3)$.

3.2.4 Data pre-process

In machine learning tasks, the attributes of sample data are not always continuous values but maybe dispersed values, such as various attributes of IP packets. Usually, there are mainly two situations: (1) there is no significance between the values of discrete features, such as the value of protocol type; (2) the value of discrete feature has the meaning of magnitude such as message length. Since we need to calculate the similarity between different samples in the vector space, the distance in the vector space, to preserve the non-partial order characteristics of the sampled data, we use one-hot encoding for dispersed attributes without values significance. After the discrete features are one-hot encoded, the features of each dimension can be regarded as continuous features. We can normalize each dimension feature. For example, normalized to [− 1, 1] or normalized to a mean value of 0 and the variance of 1.

Table 4 The one-hot code for IP packet attribute

Attributes	Value	one-hot code
Protocol_type	{'tcp','udp', 'icmp'}	{ 001,010,100}
service	{'aol', 'auth', …}	$\{100..0\}_{68} \sim \{000..1\}_{68}$
flag	{'OTH', 'REJ',…}	$\{100..0\}_{11} \sim \{000..1\}_{11}$
land	{'0', '1'}	{ 01,10}
logged in	{'0', '1'}	{ 01,10}
is_host_login	{'0', '1'}	{ 01,10}
is_guest_login	{'0', '1'}	{ 01,10}

For example, Consider the IP protocol type: {"TCP", "UDP", "ICMP"}, if we directly use numbers to represent the value, it will destroy the distribution characteristics of the attribute, because the type of protocol characteristics is not set in numerical order. To solve the above problems, we use One-Hot Encoding. The method is to use N-bit status registers to encode N states. Each state has its own independent register bit, and at any time, among them only one bit is valid. For example: for {"TCP", "UDP", "ICMP"}, the one-hot encoding can be: {"001", "010", "100"}. Through this encoding method, we map the data to a sparse space, which solves the problem that the classifier cannot handle the attribute data.

Usually, the attributes of all IP packets include integer/real type and enumeration type. Since the clustering algorithm mainly analyzes digital data, the traffic data needs to be preprocessed first. The main process includes encoding, normalization, and dimensionality reduction.

Step1: Encoding.

Sampling data of network traffic usually consists of a group of IP packets. The protocol field of these IP packets usually includes a numeric type value and an enumerated type value. For the numeric type value, we keep its original value. For the enumerated type value, we use one-hot encoding and then use the Hamming distance to calculate the standard deviation during the normalization process. Take the NSL-KDD data set as an example, which contains seven enumerated attributes. The one-hot codes of these seven attributes are listed in Table 4.

Step2: Standardization and Normalization

Due to the large differences in the value of different protocol attributes, a unified standard cannot be used in the similarity calculation. This will make the attributes with a larger value get a higher weight in the clustering process, thereby affecting the final clustering accuracy. To solve this problem, we define a standardized function:$\boldsymbol{S}: \mathbf{x} \mapsto \boldsymbol{S}(\boldsymbol{x}) \in \boldsymbol{R}$. Suppose $\{\boldsymbol{x_i}\}_{i=1}^{\boldsymbol{m}} \in \boldsymbol{X}$ is sample data set with the size of $\boldsymbol{M}$ and each data has $\boldsymbol{N}$ attributes.

$$x'_{ij} = \frac{x_{ij} - \bar{x}}{\frac{1}{M}\left(\left|x_{1j} - \bar{x}\right| + \cdots + \left|x_{Mj} - \bar{x}\right|\right)} \tag{13}$$

$$S.t.\bar{x} = \frac{1}{M}\left(x_{1j} + \ldots + x_{Mj}\right)$$

For each attribute value x_{ij} of the sampled data, we use function S to calculate x'_{ij}. The function S is shown in Eq. (14). If the value of x_{ij} is a numeric type, $x_{ij} - \bar{x}$ means the algebraic difference between x_{ij} and $\bar{x}$. If the value of x_{ij} is an enumeration type, $x_{ij} - \bar{x}$ means the Hamming difference between x_{ij} and $\bar{x}$.

After the data has been standardized, it is further normalized and mapped to the [0, 1] interval. The calculation method is as follows:

$$x'_{ij} = \frac{x'_{ij} - min\left\{x'_{1j}, x'_{2j,\ldots,x'_{Mj}}\right\}}{max\left\{x'_{1j}, x'_{2j,\ldots,x'_{Mj}}\right\} - min\left\{x'_{1j}, x'_{2j,\ldots,x'_{Mj}}\right\}} \tag{14}$$

Step3: Dimensionality reduction.

Although the multiple kernel learning algorithm is not sensitive to the changes of individual attributes during the clustering process and has a good clustering effect, its computational complexity is higher, which is $O(N^3)$. Therefore, it is necessary to perform dimensionality reduction processing on the data before cluster analysis. From the original NSL-KDD, AWID and other test data, we found these data sets contain many data with the same characteristics. These data have no significant effect on the clustering results, but will increase the computational cost. Therefore, we use principal component analysis to reduce its dimensionality.

3.2.5 Anomaly detection based on multiple kernel K-means clustering

So far, after solving the problems of incomplete data and data diversity, we can perform anomaly detection on IoT network traffic based on the multi-kernel K-means clustering method. The procedure of the whole method is as follows: the first step is a feature selection of the sampled data. By using the technical advantages of multi-kernel learning, the selection of traffic characteristics can cover different protocol layers, such as: data link layer, network layer, transport layer, and application layer. The second step is data normalization. Finally, the multi-kernel K-means clustering method is used to cluster the data and the classification result of the traffic is obtained. The flow of the whole algorithm is shown in Fig. 3, and the pseudo code of the algorithm is shown in Table 5.

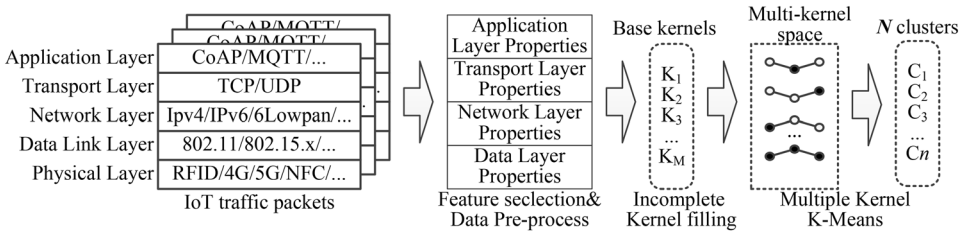

Fig. 3 Anomaly detection based on Multiple Kernel K-means Clustering

Table 5 Anomaly detection Algorithm based on MKKC

Algorithm 3 Anomaly detection based on MKKC

Input: Properties vector $\boldsymbol{X} = \{x^{(1)}, \ldots, x^{(M)}\}$, each $x^{(i)}$ has $\boldsymbol{L}$ properties; kernel matrix $\boldsymbol{K}$, number of Clusters $\boldsymbol{N}$

Output: $\boldsymbol{N}$ disjoint clusters $\{\boldsymbol{C_1}, \boldsymbol{C_2}, \ldots, \boldsymbol{C_N}\}$

1. **for** each $x^{(i)} \in \boldsymbol{X}$ **do**
2. **for** each property $\boldsymbol{p}$ of $x^{(i)}$ **do**
3. Encode $\boldsymbol{p}$ and Normalize $\boldsymbol{p}$ with Eq. (13) and (14);
4. construct kernel matrix $\boldsymbol{K}$ based vector $\boldsymbol{X}$
5. Filling the incomplete kernel matrix K with algorithm 2;
6. Perform Multiple Kernel K-Means with algorithm 1;
7. return $\{\boldsymbol{C_1}, \boldsymbol{C_2}, \ldots, \boldsymbol{C_N}\}$;

4 Experiments

4.1 datasets used

To verify the effectiveness of the proposed method, we select several data sets for testing, mainly including NSL-KDD (University of New Nrunswick), UNSW (Chinese Software Developer Network) and AWID (Awid dataset).

The NSL-KDD data set mainly performs data redundancy processing on KDD CUP99. Part of the duplicate data is removed from the training data and test data, but the format of the data itself is the same as the KDD CUP99 data set. Each piece of data in the NSL-KDD data set contains 41 attribute features, 32 continuous feature attributes, and 9 discrete feature attributes, and a class tag item is added to the last item of each attribute in the data, indicating that the data is normal or attack. There are four types of attack, which includes Dos, Probing, R2L, and U2R.

The UNSW_NB15 data set was collected by the Australian Centre for Cyber Security (ACCS) Cyber Range Laboratory in 2015. The laboratory uses the IXIA PerfectStorm tool to capture new and updated attack information from the CVE site, and uses the depdump tool to capture network traffic, ultimately obtaining 100 GB traffic generated a mixed data set of contemporary attacks caused by normal behavior and human behavior. Compared with the NSL-KDD data set, the biggest advantage of this data set is that it contains contemporary implicit attack methods, which more accurately reflects the real situation of contemporary network traffic. The UNSW_NB15 data set contains a total of more than 2.54 million pieces of data, which can be divided into 9 categories according to abnormal behaviors, namely Fuzzers, Analysis, Backdoors, Dos, Exploits, Generic, Reconnaissance, Shellcode, and Worms.

The Aegean Wi-Fi Intrusion Dataset (AWID) is a comprehensive 802.11 network dataset, which was derived from real Wi-Fi traffic traces in 2015. The AWID dataset is collected in the actual network environment via network equipment. Be different from the NSL-KDD data set and UNSW_NB15 data set, each record of AWID dataset has 155 attributes and contains link layer protocol information. In AWID dataset, there are three types of attack, which include Flooding, Impersonation, and Injection.

4.2 Experiment setup

The proposed method is experimentally evaluated on three widely used intrusion detection benchmark data sets listed in Sect. 4.1. We constructed two experiments. The first experiment evaluates the effectiveness of the filling algorithm for an incomplete kernel matrix. The second experiment evaluates the performance of anomaly detection based on MKKC. The experimental environment is built based on the Linux operating system running on a host with Intel CPU Core i7 3.6 GHz and 16G RAM. The development environment is Matlab2014a and simpleMKL toolbox.

4.3 Experiment 1: effectiveness of the filling algorithm for incomplete kernel matrix

For ease of description, we named Algorithm 2 proposed in Sect. 3 as MKKC-IC. We choose two other multi-kernel k-means methods for comparison. These two methods are MKKC-MF and MKKC-ZF, respectively, they use the mean value and zero value to fill in the missing attributes in the sample data. We use clustering accuracy (CA), normalized mutual information (NMI) and purity to evaluate the performance of the three multi-kernel k-means methods mentioned above.

Considering the high complexity of the algorithm, we randomly selected 500 sample data and 15 features in the benchmark data set for analysis. In fact, when detecting unknown abnormal behavior in a real network environment, a large amount of sampling data is often lacking. Therefore, our data selection strategy makes sense. We applying both a Polynomial kernel and a Gaussian kernel on the feature.

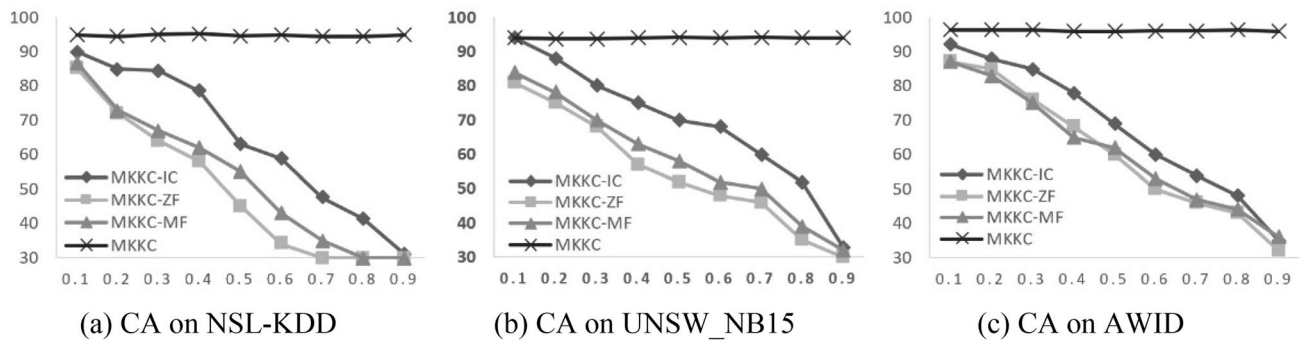

Fig. 4 Experimental result of Clustering Accuracy (CA)

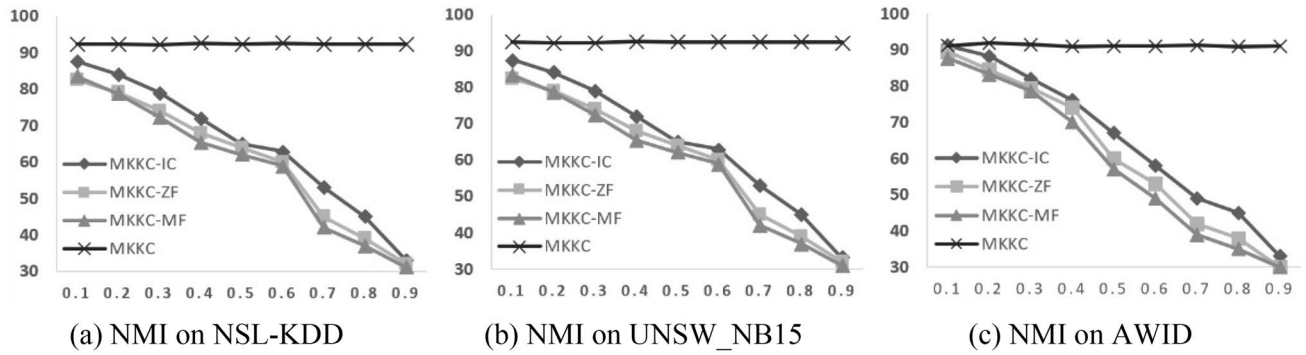

Fig. 5 Experimental result of Normalized mutual information

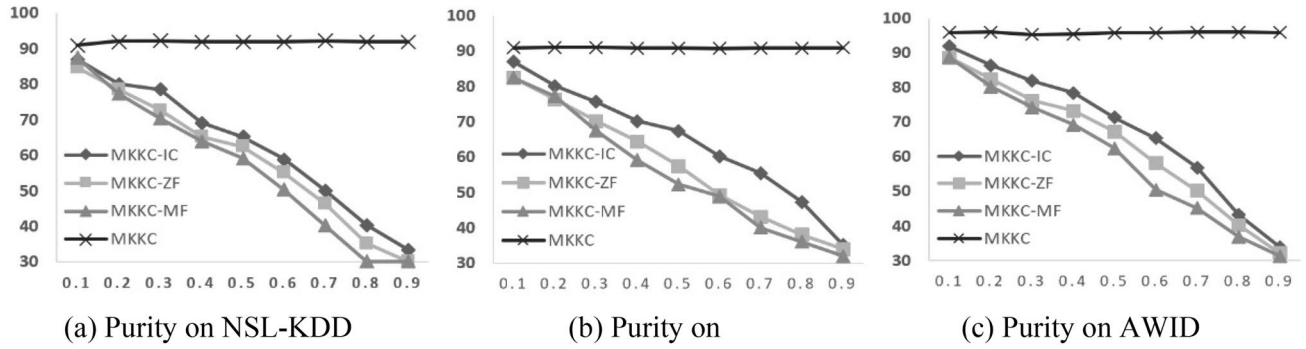

Fig. 6 Experimental result of Purity

In the experiment, we randomly select samples and adjust the proportion of missing data in the samples, while observing the changes in CA, NMI and purity of the above three methods.

The experimental results are shown from Fig. 4 to Fig. 6. The horizontal axis of the coordinate represents the missing ratio, and the vertical axis of the coordinate represents the clustering accuracy (CA), normalized mutual information (NMI) and purity respectively.

Figures 4, 5, 6 respectively show the performance of the four multi-kernel methods when processing incomplete sample data. In this experiment, the multiple kernel k-means (MKKM) is the reference target without the missing kernel matrix. Since there is no missing attribute, the performance of MKKM is the highest. With the increase in the missing ratio, we found that the performance closest to MKKM is MMK-IC method. This result confirms our assumption of data filling, that is, filling the missing kernel matrix based on similarity is helpful to increase the accuracy of clustering. Although the main application areas of multi-kernel

Table 6 ACC, NMI and Purity comparison (mean_std)

Datasets	MKKC	MKKC-ZF	MKKC-MF	MKKC-IC
Clustering accuracy (CA)				
NSL-KDD	94.93 ± 0.48	85.09 ± 0.37	86.93 ± 0.43	89.93 ± 0.48
UNSW_NB15	93.92 ± 0.16	80.95 ± 0.24	83.92 ± 0.17	87.92 ± 0.16
AWID	96.15 ± 0.24	87.10 ± 0.25	87.15 ± 0.21	92.15 ± 0.24
Normalized Mutual Information (NMI)				
NSL-KDD	94.40 ± 0.35	84.40 ± 0.35	85.35 ± 0.30	89.50 ± 0.42
UNSW_NB15	92.39 ± 0.08	82.39 ± 0.42	83.49 ± 0.18	87.55 ± 0.10
AWID	96.01 ± 0.18	89.40 ± 0.13	87.64 ± 0.24	91.10 ± 0.18
Purity				
NSL-KDD	91.92 ± 0.53	84.61 ± 0.43	87.49 ± 0.43	90.92 ± 0.53
UNSW_NB15	90.95 ± 0.14	82.44 ± 0.22	82.43 ± 0.32	86.95 ± 0.14
AWID	95.87 ± 0.25	88.62 ± 0.37	88.66 ± 0.25	91.87 ± 0.25

method are concentrated in image recognition, speech recognition, etc., and the test data of the algorithm is mostly image data. The results of this experiment show that the multi-kernel k-means clustering method can also explore the inherent characteristics of network traffic data and has a good clustering effect.

It can be seen from Table 6 lists the average performance of the four clustering algorithms when the sample missing rate is 10% and the highest performance is shown in bold. It can be seen from Table 6 that after using the MKKC-IC algorithm to fill the incomplete kernel matrix, the accuracy of the clustering results can be further improved. In this experiment, the MKKC-IC algorithm improves the performance of the traditional MKKC-ZF and MKKC-MF algorithms by 4%. In addition, Fig. 4 to Fig. 6 that when using the MKKC-IC algorithm to perform cluster analysis on the sampled data with missing attributes, the overall effect is better than the traditional MKKC-ZF and MKKC-MF algorithms.

4.4 Experiment of intrusion detection

To verify the effectiveness of the proposed intrusion detection method, this section uses true positive rate (TPR), false positive rate (FPR), precision, accuracy and F-score as evaluation metrics. Since the proposed method is based on multi-kernel clustering algorithm, three typical clustering algorithms are selected as the comparison objects, including density peaks (DP) algorithm [49], K-means algorithm and Gaussian Mixture Modelling (GMM) algorithm [48]. We first use the above algorithms to perform cluster analysis for NSL-KDD, UNSW_NB15, and AWID, and then perform statistics based on the clustering results. In the statistical process, only normal behaviors and abnormal behaviors are distinguished, and the clustering results are not accurately classified.

To test the algorithm's ability to identify abnormal traffic from small batches of data, 1000 records were selected from each data set for multiple tests. The number of abnormal traffic packets is increased from 100 to 500, each time increasing by 100. Finally calculate the average of the test results. Table 7 shows the result of performance test.

The experimental results in Table 7 show that the multi-kernel clustering method helps to obtain more stable and accurate anomaly detection results. In addition, the DP algorithm, K-means algorithm and GMM algorithm are susceptible to the influence of feature selection, which leads to big changes in TPR, accuracy and accuracy. In Table 7, the highest values of different experimental results are marked in bold.

5 Conclusion

The kernel method is a powerful tool to solve the linear inseparability of low-dimensional vector spaces. But a single kernel method is not good at solving the problem of high-dimensions vector clustering. The multi-kernel method is proved to be a more advanced and effective solution. However, in the actual application process, the lack of sampling data hinders the use of multi-kernel clustering algorithms. In this paper, we propose and discuss the issues of attribute missing in sample data for intrusion detection. we also propose an intrusion detection framework for 5G and IoT networks which is based on multiple

Table 7 Result of performance test

Algorithm	TPR (%)	FPR (%)	Precision (%)	Accuracy (%)	F-score (%)
NSL-KDD dataset					
MKKM-IC	**89.00**	**5.00**	**81.65**	**93.80**	**85.17**
Density peaks	75.00	8.75	68.18	88.00	71.43
K-means	68.00	11.75	59.13	84.20	63.26
GMMs	80.00	6.25	76.19	91.00	78.05
UNSW_NB15 dataset					
MKKM-IC	**85.00**	**6.25**	**77.27**	**92.00**	**80.95**
Density peaks	68.00	11.25	60.18	84.60	63.85
K-means	65.00	13.00	55.56	82.60	59.91
GMMs	79.00	8.00	71.17	89.40	74.88
AWID dataset					
MKKM-IC	**90.00**	**3.00**	**88.24**	**95.60**	**89.11**
Density peaks	80.00	7.50	72.73	90.00	76.19
K-means	70.00	6.75	72.16	88.60	71.07
GMMs	77.42	1.83	85.71	95.60	81.36

The highest values of different experimental results are marked in bold

kernel k-means with incomplete kernels. The experimental results show that our proposed method can indeed achieve high-accuracy clustering when the sampled data is incomplete. Unfortunately, this method still has some shortcomings in processing performance, and there are still some shortcomings when processing massive amounts of data. In the follow-up research work, we strive to improve and perfect the problem.

Existing multi-kernel clustering methods generally suffer from high computational complexity and cannot process large-scale sampled data in real time. The method proposed in this article is no exception. Therefore, in the follow-up research, we will try to further explore the topology characteristics of 5G IoT, and design a hierarchical clustering method to avoid the problem of massive data size.

Acknowledgements Acknowledgements and Reference heading should be left justified, bold, with the first letter capitalized but have no numbers. Text below continues as normal. Authors should thank those who contributed to the article but cannot be listed as an author.

Author contributions Methodology: NH; project administration: ZT; conceptualization: HL, XD and MG: all authors have read and agreed to the published version of the manuscript.

Funding Authors should describe sources of funding that have supported the work, including specific grant numbers, initials of authors who received the grant, and the URLs to sponsors' websites. If there is no funding support, please write "The author(s) received no specific funding for this study. This work was supported by National Natural Science Foundation of China (Grant no. 61976064).

Compliance with ethical standards

Conflicts of interest The authors declare that they have no conflicts of interest to report regarding the present study. The authors claim that none of the material in the paper has been published or is under consideration for publication elsewhere.

References

1. Agarwal R, Joshi MV (2001) PNrule: a new framework for learning classifier models in data mining (a case-study in network intrusion detection). Report No 10598:1–17. https://doi.org/10.1137/1.9781611972719.29
2. Ahmim A, Maglaras L, Ferrag MA, et al (2019) A novel hierarchical intrusion detection system based on decision tree and rules-based models. In: 2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS). IEEE, pp 228–233
3. Al-Garadi MA, Mohamed A, Al-Ali AK et al (2020) A survey of machine and deep learning methods for internet of things (IoT) security. IEEE Commun Surv Tutor 22:1646–1685. https://doi.org/10.1109/COMST.2020.2988293
4. Al-Yaseen WL, Othman ZA, Nazri MZA (2017) Multi-level hybrid support vector machine and extreme learning machine based on modified K-means for intrusion detection system. Expert Syst Appl 67:296–303. https://doi.org/10.1016/j.eswa.2016.09.041
5. Anderson JA (1995) An introduction to neural networks. MIT Press, Cambridge
6. Anderson JP (1980) Computer security threat monitoring and surveillance. James P. Anderson Co., Fort Washington
7. Aryal S, Santosh KC, Dazeley R (2020) usfAD: a robust anomaly detector based on unsupervised stochastic forest. Int J Mach Learn Cybernet. https://doi.org/10.1007/s13042-020-01225-0
8. Awid dataset wireless security datasets project (2020) http://icsdweb.aegean.gr/awid/features.html
9. Baba NM, Makhtar M, Fadzli SA, Awang MK (2015) Current issues in ensemble methods and its applications. J Theoret Appl Inf Technol 81:266–276
10. Benkhelifa E, Welsh T, Hamouda W (2018) A critical review of practices and challenges in intrusion detection systems for IoT: toward universal and resilient systems. IEEE Commun Surv Tutor 20:3496–3509. https://doi.org/10.1109/COMST.2018.2844742
11. Bosman HHWJ, Iacca G, Tejada A et al (2015) Ensembles of incremental learners to detect anomalies in ad hoc sensor networks. Ad Hoc Netw 35:14–36. https://doi.org/10.1016/j.adhoc.2015.07.013
12. Caminero G, Lopez-Martin M, Carro B (2019) Adversarial environment reinforcement learning algorithm for intrusion detection. Comput Netw 159:96–109. https://doi.org/10.1016/j.comnet.2019.05.013
13. Chen S, Peng M, Xiong H, Yu X (2016) SVM intrusion detection model based on compressed sampling. J Electr Comput Eng 2016:1–6. https://doi.org/10.1155/2016/3095971
14. Chettri L, Bera R (2020) A comprehensive survey on internet of things (IoT) toward 5G wireless systems. IEEE Internet Things J 7:16–32. https://doi.org/10.1109/JIOT.2019.2948888
15. Chinese Software Developer Network UNSW_NB15 (2020) https://download.csdn.net/download/asialeebird/10795133
16. D'Agostini G (1995) A multidimensional unfolding method based on Bayes' theorem. Nucl Instrum Methods Phys Res, Sect A 362:487–498. https://doi.org/10.1016/0168-9002(95)00274-X
17. Deng Z, Zhu X, Cheng D et al (2016) Efficient k NN classification algorithm for big data. Neurocomputing 195:143–148. https://doi.org/10.1016/j.neucom.2015.08.112
18. Dieudonné J (1969) Foundations of modern analysis. Academic Press, Cambridge
19. Du XJ, Wu D (2006) Adaptive cell relay routing protocol for mobile ad hoc networks. IEEE Trans Veh Technol 55:278–285. https://doi.org/10.1109/TVT.2005.861196
20. Dyn (2016) Incident Report for Oracle + Dyn. https://www.dynstatus.com/incidents/5r9mppc1kb77
21. Fossaceca JM, Mazzuchi TA, Sarkani S (2015) MARK-ELM: application of a novel multiple kernel learning framework for improving the robustness of network intrusion detection. Expert Syst Appl 42:4062–4080. https://doi.org/10.1016/j.eswa.2014.12.040
22. Fourati H, Maaloul R, Chaari L (2020) A survey of 5G network systems: challenges and machine learning approaches. Int J Mach Learn Cybernet. https://doi.org/10.1007/s13042-020-01178-4
23. Gao N, Gao L, Gao Q, Wang H (2014) An intrusion detection model based on deep belief networks. In: 2014 Second international conference on advanced cloud and big data, IEEE, pp 247–252
24. Garg S, Kaur K, Kumar N et al (2019) A hybrid deep learning-based model for anomaly detection in cloud datacenter networks. IEEE Trans Netw Serv Manage 16:924–935. https://doi.org/10.1109/TNSM.2019.2927886

25. Gönen M, Alpaydın E (2011) Multiple kernel learning algorithms. J Mach Learn Res 12:2211–2268
26. Gouveia A, Correia M (2017) A systematic approach for the application of restricted Boltzmann machines in network intrusion detection. In: Rojas I, Joya G, Catala A (eds) Advances in computational intelligence. Springer International Publishing, Cham, pp 432–446
27. Gu J, Sun B, Du X et al (2018) Consortium Blockchain-based malware detection in mobile devices. IEEE Access 6:12118–12128. https://doi.org/10.1109/ACCESS.2018.2805783
28. Guo Y, Ji T, Wang Q et al (2020) Unsupervised anomaly detection in IoT systems for smart cities. IEEE Trans Netw Sci Eng. https://doi.org/10.1109/TNSE.2020.3027543
29. Haykin S (1999) Neural networks: a comprehensive foundation, 2nd edn. Prentice Hall, Hoboken
30. Heberlein LT, Dias GV, Levitt KN, et al (1990) A network security monitor. In: Proceedings. 1990 IEEE Computer Society Symposium on Research in Security and Privacy, IEEE, pp 296–304
31. Iam-On N (2020) Clustering data with the presence of attribute noise: a study of noise completely at random and ensemble of multiple k-means clusterings. Int J Mach Learn Cybernet 11:491–509. https://doi.org/10.1007/s13042-019-00989-4
32. Illy P, Kaddoum G, Miranda Moreira C et al (2019) Securing Fog-to-Things environment using intrusion detection system based on ensemble learning. In: 2019 IEEE wireless communications and networking conference (WCNC), IEEE, pp 1–7
33. ITU (2017) Minimum requirements related to technical performance for IMT-2020 radio interface(s)
34. Kim J, Kim J, Thi Thu H Le, Kim H (2016) Long short term memory recurrent neural network classifier for intrusion detection. In: 2016 international conference on platform technology and service (PlatCon). IEEE, pp 1–5
35. Li R, Li X, Lin C, Collinson M, and Mao R (2019) A Stable Variational Autoencoder for Text Modeling. In: The 12th International Conference on Natural Language Generation (INLG). SIGGEN, pp 594–599
36. Li, R, Lin C, Collinson M, Li X, and Chen G (2019) A Dual-Attention Hierarchical Recurrent Neural Network for Dialogue Act Classification. In: The 23rd Conference on Computational Natural Language Learning (CoNLL), SIGNLL, pp 383–392
37. Li X, Lin C, Wang C, Li R, and Guerin F. Latent Space Factorisation and Manipulation via Matrix Subspace Projection (2020). In: The 37th International Conference on Machine Learning (ICML), PMLR, pp 5916–5926
38. Li M, Sun Y, Lu H et al (2019) Deep reinforcement learning for partially observable data poisoning attack in crowdsensing systems. IEEE Internet Things J 2019:1–1. https://doi.org/10.1109/jiot.2019.2962914
39. Liang J, Ma M, Sadiq M, Yeung K-H (2019) A filter model for intrusion detection system in Vehicle Ad Hoc Networks: a hidden Markov methodology. Knowl-Based Syst 163:611–623. https://doi.org/10.1016/j.knosys.2018.09.022
40. Liu X, Gao W, Zhu X et al (2019) Multiple Kernel k-means with incomplete Kernels. IEEE Trans Pattern Anal Mach Intell. https://doi.org/10.1109/TPAMI.2019.2892416
41. Meng S, Huang W, Yin X et al (2020) Security-aware dynamic scheduling for real-time optimization in cloud-based industrial applications. IEEE Trans Industr Inf. https://doi.org/10.1109/TII.2020.2995348
42. Ming Zhang, Xiaojiang Du, Nygard K (2005) Improving coverage performance in sensor networks by using mobile sensors. In: MILCOM 2005–2005 IEEE military communications conference, IEEE, pp 3335–3341
43. Minh HQ, Niyogi P, Yao Y (2006) Mercer's Theorem, feature maps, and smoothing, pp 154–168
44. Mitchell T (1997) Machine learning. McGraw Hill, Hoboken
45. Nuo Y (2018) A novel selection method of network intrusion optimal route detection based on naive Bayesian. Int J Appl Dec Sci 11:1. https://doi.org/10.1504/IJADS.2018.088631
46. Qi L, Hu C, Zhang X et al (2020) Privacy-aware data fusion and prediction with spatial-temporal context for smart city industrial environment. IEEE Trans Industr Inf. https://doi.org/10.1109/TII.2020.3012157
47. Qiu J, Tian Z, Du C et al (2020) A survey on access control in the age of internet of things. IEEE Internet Things J 7:4682–4696. https://doi.org/10.1109/JIOT.2020.2969326
48. Reynolds D (2009) Gaussian Mixture Models. In: Encyclopedia of Biometrics. Springer US, Boston, pp 659–66
49. Rodriguez A, Laio A (2014) Clustering by fast search and find of density peaks. Science 344:1492–1496. https://doi.org/10.1126/science.1242072
50. Schölkopf B, Smola A, Müller K-R (1998) Nonlinear component analysis as a kernel Eigenvalue problem. Neural Comput 10:1299–1319. https://doi.org/10.1162/089976698300017467
51. Senthilnayaki B, Venkatalakshmi K, Kannan A (2019) Intrusion detection system using fuzzy rough set feature selection and modified KNN classifier. Int Arab J Inf Technol 16:746–753
52. Shafiq M, Tian Z, Bashir AK et al (2020) CorrAUC: a malicious Bot-IoT traffic detection method in IoT network using machine learning techniques. IEEE Internet Things J. https://doi.org/10.1109/JIOT.2020.3002255
53. Shah R, Qian Y, Kumar D et al (2017) Network intrusion detection through discriminative feature selection by using sparse logistic regression. Future Internet 9:81. https://doi.org/10.3390/fi9040081
54. Singh T, Kumar N (2020) Machine learning models for intrusion detection in IoT environment: a comprehensive review. Comput Commun. https://doi.org/10.1016/j.comcom.2020.02.001
55. Soucy P, Mineau GW(2001) A simple KNN algorithm for text categorization. In: Proceedings 2001 IEEE International Conference on Data Mining. IEEE Comput. Soc, pp 647–648
56. Swarnkar M, Hubballi N (2016) OCPAD: one class Naive Bayes classifier for payload based anomaly detection. Expert Syst Appl 64:330–339. https://doi.org/10.1016/j.eswa.2016.07.036
57. Tian Z, Gao X, Su S, Qiu J (2020) Vcash: a novel reputation framework for identifying denial of traffic service in internet of connected vehicles. IEEE Internet Things J 7:3901–3909. https://doi.org/10.1109/JIOT.2019.2951620
58. Tian Z, Shi W, Wang Y et al (2019) Real-Time lateral movement detection based on evidence reasoning network for edge computing environment. IEEE Trans Industr Inf 15:4285–4294. https://doi.org/10.1109/TII.2019.2907754
59. University of New Nrunswick NSL-KDD (2020) http://nsl.cs.unb.ca/NSL-KDD/
60. Vapnik V (1998) Statistical learning theory. Wiley, Hoboken
61. Wang D, Chen D, Song B et al (2018) From IoT to 5G I-IoT: the next generation IoT-based intelligent algorithms and 5G technologies. IEEE Commun Mag 56:114–120. https://doi.org/10.1109/MCOM.2018.1701310
62. Wang S, Li M, Hu N et al (2019) K-means clustering with incomplete data. IEEE Access 7:69162–69171. https://doi.org/10.1109/ACCESS.2019.2910287
63. Wu X, Khosravi MR, Qi L et al (2020) Locally private frequency estimation of physical symptoms for infectious disease analysis in Internet of Medical Things. Comput Commun 162:139–151. https://doi.org/10.1016/j.comcom.2020.08.015
64. Xiao L, Wan X, Dai C et al (2018) Security in mobile edge caching with reinforcement learning. IEEE Wirel Commun 25:116–122. https://doi.org/10.1109/MWC.2018.1700291

65. Xue L, Yu Y, Li Y et al (2019) Efficient attribute-based encryption with attribute revocation for assured data deletion. Inf Sci 479:640–650. https://doi.org/10.1016/j.ins.2018.02.015
66. Yousefi-Azar M, Varadharajan V, Hamey L, Tupakula U (2017) Autoencoder-based feature learning for cyber security applications. In: 2017 International Joint Conference on Neural Networks (IJCNN). IEEE, pp 3854–3861
67. Zarpelão BB, Miani RS, Kawakani CT, de Alvarenga SC (2017) A survey of intrusion detection in Internet of Things. J Netw Comput Appl 84:25–37. https://doi.org/10.1016/j.jnca.2017.02.009
68. Zetter K (2016) Inside the Cunning, Unprecedented Hack of Ukraines Power Grid. https://www.wired.com/2016/03/inside-cunning-%0Aunprecedented-hack-ukraines-power-grid/%0A
69. SimpleMKL Toolbox (2008) http://asi.insa-rouen.fr/enseignants/~arakoto/code/mklindex.html