A Cloud User Behavior Authentication Model Based on Multi-label Hyper-network

Ruoshui Liu[1], Xin Wang[2], Juan Du[1], Ping Xie[1]

[1]Information Engineering College, Henan University of Science and Technology, China
[2]School of Business and Management, Postdoctoral Research Station,
Shanghai International Studies University, China
liuruoshui@msn.com, wangxin@shisu.edu.cn, 447916581@qq.com, xieping_1984@bupt.edu.cn

Abstract

With the advent of the Big Data era, user information security is particularly important. How to build trust between users and the cloud is an important issue. In response to this problem, this paper proposes a cloud user behaviour authentication model based on multi-label hyper-network, which implements the fine-grained division of user behaviour and improves the accuracy of anomaly detection. This method trains the user's normal behaviour database into a hyper-network, adds the current user behaviour as an instance to the hyper-network for classification. If a label is successfully found in this classification, it is identified as a normal user. Otherwise, the model updates the weight of the hyper-network, replaces the super edge, and looks for the label again. If the label is found, it is identified as a risk user, otherwise it is identified as malicious user. The simulation results show that there is a significant improvement in the accuracy of classification. Applying the method in this paper to the detection of user behavior can effectively improve the detection rate of user behavior, realize fine-grained analysis of user behavior, and improve the processing ability of user behavior.

Keywords: User behavior, Authentication, Multi-label, Hyper-network

1 Introduction

In recent years, with the development of information technology and the integration of the world economy, mankind has entered an era of Big Data. The rapid growth of the number of mobile terminals gives rise to an irreversible trend of the widespread application of mobile cloud computing [1]. Transition of cloud computing from the desktop market to the mobile market becomes the main direction of development. However, there appears to have a variety of complex issues. Among them, the three aspects of "user-environment-service" are particularly outstanding. Mobile intelligent terminals access information from mobile cloud services via the mobile Internet or Internet of Things [2-7], so that various types of integrated services can be provided. The provision of green, reliable and timely terminal service is the core function of mobile cloud services. From the user's point of view, the establishment of a trust relationship between users and the cloud becomes the most paramount [8]. Fine classification of user behavior before the cloud service enters the substantive service provision process has become a important issue to be addressed urgently [9]. User behaviour classification problem is actually the subset of classification problem. Classification algorithm is also an important research direction in the fields of pattern recognition, data mining and anomaly detection, and it has attracted much attention. The characteristics of datasets are used to construct a classifier in the classification problem, and then this classifier is utilised to assign categories to unknown categories of objects. At present, the single-label classification problem has been studied in depth, and there are many mature related algorithms such as Naive Bayes (NB) [10], Support Vector Machine (SVM) [11], K-Nearest Neighbour (KNN) [12] and so on. The multi-label classification algorithm uses the dependency between behavioural labels to improve the performance of the classifier. For some less complex and smaller relational data, the algorithm enables better classification accuracy. However, it is difficult to learn the characteristics of nodes through the classification of related models.

With the increasing research on classification algorithms, there are more and more practical applications emerged in the fields of data mining, interest recommendation, anomaly detection etc. In this paper, the main contributions of this work are summarized as follows. We propose a cloud user behaviour authentication model based on multi-label hyper-network, which implements the fine-grained division of user behaviour and improves the accuracy of anomaly detection. The simulation results show that it can effectively improve the accuracy of classification.

*Corresponding Author: Xin Wang; E-mail: wangxin@shisu.edu.cn
DOI: 10.3966/160792642019122007006

2 Related Work

Classification is an important research topic in the fields of pattern recognition, data mining and so on. Traditional machine learning is mainly classified into two categories, i.e., binary-class and multiclass classifications. Each sample has only one category label, called single label learning. However, there are a large number of samples with multiple labels at the same time in the real world. For example, a news report, it can belong to not only the economic category but also the law. A picture may contain more semantic information, such as the sea, the beach and the city. A gene may have multiple functions, such as translation and transcription. There are a variety of effects arising from the function of protein to a cell, so there will be many functional classes. These samples with multiple labels are called multi-label data. Compared with samples in single label data having only one tag, and samples in multi label data may have multiple labels at the same time. It gives rise to the complexity and variability. The complexity and variability of multi label data make it more difficult to construct the multi label learning algorithm.

Human beings tend to naturally decompose complex problems into several simple problems in the divide and conquer fashion. This method, however, destroys the correlation between labels. There is a certain correlation between tags in multi label data. Therefore, making full use of the correlation among tags in multi-label training data to build prediction model [13-15] is always one of the hot topics in multi-label learning. As we all know, multi-label classification is a kind of supervised learning [16]. In the case of support vector machines (SVM), one-vs.-all (OVA) is the most common solution to the multi-label classification problem. However, due to the extremely uneven training set, its precision is very limited. A new framework of multi-label classification is proposed by Daengduan who uses undersampling technology to alleviate the problem of imbalance and improve the accuracy of the algorithm. Some scholars also put forward the multi-label classification algorithm [17] in which the Naive Bayes Classifier (NBC) is used as the basic model. The algorithm makes the edge of the classification more distinct, and the model is more expressive.

Classifier chain [18] is an effective method to solve the multi-label classification task. It can keep the computational complexity within the acceptable range and model the label relevance. However, the disadvantage is that the redundancy of potential labels and tags is ignored, and the accuracy of the model is not comparable to others. The mRMR-CC algorithm is proposed by Huang et al. [19] who devises a new classifier chain. It selects an additional compact attribute set for each of the basic classifiers. The algorithm considers not only the label correlation but also the label redundancy. The experimental results show that the model improves the attribute selection and prediction performance of the classifier chain method.

Nowadays, the multi-label classification in the social network environment has become an important field of data mining research. Given the label (or source) of some nodes, the task is to infer the labels of other nodes in the same network. The classification method using association relationships among related instances has outperformed the traditional classifiers. In a socialised network environment, the collective reasoning process will severely limit the efficiency of the classification if we just want to predict a specific node's label. A new concept of the core network is proposed by Zhang et al. [20] propose a new heuristic core network discovery algorithm (HCN), and they compare two classification algorithms, i.e., HCN-wvRN and HCN-SCRN. These two algorithms can deal with large-scale social networks in an effective way while maintaining classification accuracy. The proposed algorithm greatly improves the classification efficiency. The drawback is that in many multi-label classification tasks, the algorithm does not provide label correlation, giving rise to the difficulty of direct learning from data samples in medium scale training set.

With the development of multi-label classification algorithm, label embedding (LE) is an important family of multi-label classification algorithms that digest the label information jointly for better performance. Researchers Haung et al. proposed a LE algorithm considering the interest cost function [21], which improved the classification accuracy. Scholar Claudia proposed a heterogeneous information network in literature [22]. By mining the linkage structure of heterogeneous information networks, multiple types of relationships among different class labels and data samples can be extracted. Then we can use these relationships to effectively infer the correlations among different class labels in general, as well as the dependencies among the label sets of data examples inter-connected in the network. Empirical studies on real-world tasks demonstrate that the performance of multi-label classification can be effectively boosted using heterogeneous information networks. In some extreme cases, the scholar Lin has designed a two-stage framework for the classification of extreme data. The weight adjustment phase improves the diversity of the model. The members with low quality are deleted in the prediction stage, so that the prediction accuracy can be subsequently improved. Experiments show that this method can produce more accurate prediction results while reducing the size of the model successfully.

Multi-label classification has been widely used in all fields, such as image classification [23-24], medical diagnosis [25], mail classification [26], user interest

recommendation [27] and so on. This paper proposes a cloud user behaviour authentication model based on multi-label hyper-network. This model applies multi-label to anomaly detection of user behaviour and implements the fine-grained division of user behaviour simply and efficiently. Therefore, the complexity of algorithm can be reduced, and the accuracy of behaviour analysis can be improved. Firstly, the normal user behaviour database is used in the model as training set in a hyper-network. Then the current user behaviour is added as an unknown instance to the hyper-network for classification. The data will be identified as the normal user if the classification of a successful label is made. Instead, we update the weight of the hyper-network to replace the super edge. Then look for the lable again, if successful classification is identified as a risk user, if not successfully classified as a malicious user.

In Section 2, we review some of the methods related to multi-label classification. The rest of this paper is organized as follows. Section 3 introduces specific methods. The simulation experiment is showed in Section 4. Finally, Section 5 gives concluding remarks.

3 Multi-label Feature Selection and Construction

3.1 Feature Selection

Feature selection and feature construction are commonly methods to improve the recognition of feature set. The related feature vectors are selected from the original feature set to reduce the number of features, and they are combined to build new high-level features, so that they have better discernment ability. In this paper, we define to measure the importance of features. It is used to evaluate the discrimination ability of each feature vector, and it is calculated based on Equation (1).

$$Dis = \frac{1}{1+e^{-5(D_p - D_q)}}, \tag{1}$$

where D_p is the average distance between a feature vector and its nearest neighbour eigenvectors of the different class, and D_q is the average distance between a feature vector and its farthest eigenvector of the same class. Equation (1) is used to maximize the distance of feature between different classes but minimize the distance of feature within the same classes. Then it calculates the discernment ability of each feature vector and sorts them. We select feature vectors with good discriminative ability to construct feature. The behaviour instance S is used as the training set, and D_p and D_q are calculated as shown in Equations (2) and (3).

$$D_p = \frac{1}{|S|}\sum_{i=1}^{|S|} \max_{j,k} dis\left(V_{ji}, V_{ki}\right), \tag{2}$$

$$D_q = \frac{1}{|S|}\sum_{i=1}^{|S|} \min_{j,k} dis\left(V_{ji}, V_{ki}\right), \tag{3}$$

where $j \neq k$ and $class\left(V_j\right) = class\left(V_k\right)$.

In Equations (2) and (3), $dis\left(V_j, V_k\right)$ is used to represent any measure of the distance between two approximate eigenvectors V_j and V_k. Here, we introduce the *Czekanowski* distance to evaluate the dissimilarity of two feature vectors, and it is showed in the Equation (4). It is based on the shared part between two feature vectors. d represents the feature dimension, and its value is bounded in the range $[0,1]$.

$$Czekanowski\left(V_j, V_k\right) = 1 - \frac{2\sum_{d=1}^{n} \min_{j,k}\left(V_{jd}, V_{kd}\right)}{\sum_{d=1}^{n} \min_{j,k}\left(V_{jd} + V_{kd}\right)} \tag{4}$$

3.2 Feature Construction

Feature selection is designed to select a feature subset of D so as to eliminate redundant irrelevant features while reducing computation complexity. Both of the classification performance and the interpretability of model can be improved. The size of the training set is N, $\{(\mathbf{x_1}, \mathbf{y_1}), ..., (\mathbf{x_i}, \mathbf{y_i}), ..., (\mathbf{x_N}, \mathbf{y_N})\}$. For the i-th sample, $\mathbf{x}_i = [x_{i1}, x_{i2}, ..., x_{iD}]^T \in \mathbb{R}^D$ is the D-dimension eigenvector, and $\mathbf{y}_i = [y_{i1}, y_{i2}, ..., y_{iC}]^T \in \{1,0\}^C$ represents the C-dimension binary label vector, 1 represents the correlation, and 0 is independent. Two matrices are used to describe the training set:

$$X = [x_1, ..., x_i, ..., x_N]^T = [x^{(1)}, ..., x^{(k)}, ..., x^{(D)}] \tag{5}$$

$$Y = [y_1, ..., y_i, ..., y_N]^T = [y^{(1)}, ..., y^{(l)}, ..., y^{(C)}] \tag{6}$$

where $x^{(k)}$ represents the k-th feature vector, and $y^{(l)}$ represents the l binary label vector.

4 Abnormal Behaviour Division Mechanism Based on Multi-label Hyper-network

This paper focuses on the way to establish a credible mechanism between users and cloud. A cloud user behaviour authentication model based on multi-label hyper-network (MLHN) is proposed. It implements fine-grained division of user behaviour and improves the accuracy of abnormal detection. This method trains

users' normal behaviour database in a hyper-network, and it classifies the current user behaviour as an unknown instance in the hyper-network. The current user is identified as a normal user if the instance is classified as a label successfully. If the current instance does not find the label, the weight of the hyper-network is updated, the hyperedge is replaced, and the label is searched again. If the label is found, it is identified as a risk user, and on the contrary, it is identified as a malicious user. The flow diagram is shown in Figure 1.

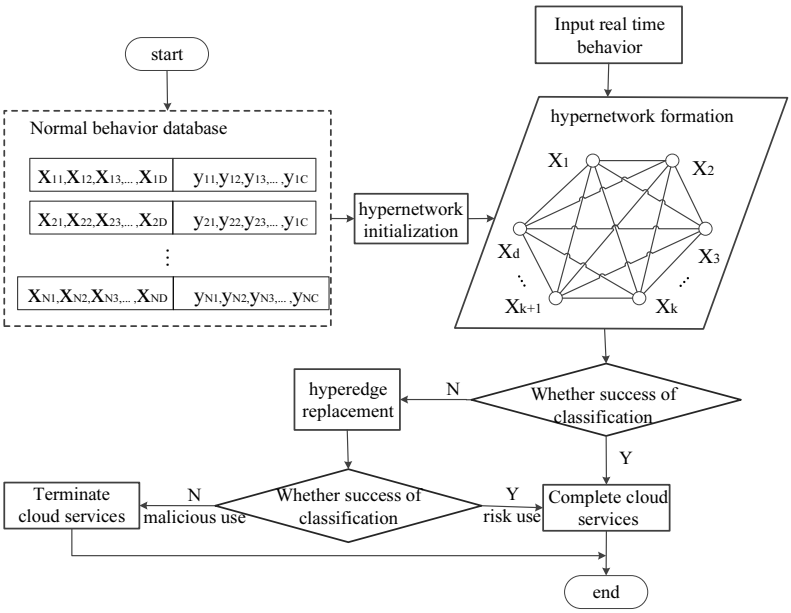


Figure 1. Flow diagram of cloud user behaviour authentication model

4.1 Multi-label Hyper-network

Hypernetwork is a weighted random hypergraph where high-order interactions among vertices are represented in hyperedges. Formally, a hypernetwork is defined by a triple $H=(V,E,W)$, where V denotes vertices, E denotes hyperedges, and W denotes hyperedge weights. In hypernetwork, a vertex corresponds to a feature or a data variable, a hyperedge represents an arbitrary relationship of two or more vertices. Each hyperedge of hypernetwork is associated with a weight to indicate the importance of the relationship among vertices. Therefore, a hypernetwork is regarded as a large collection of hyperedges that characterise high-order relationships among features. In hypernetwork, a learning task can be regarded as storing or memorising a given data set $D=\{\boldsymbol{x_i}\,|1\le i\le N\}$, where $\boldsymbol{x_i}$ is a data instance represented by a d -dimensional feature vector, so that the stored data can be retrieved later.

Multi-label hyper-network is also defined by a triple $H=(V,E,\boldsymbol{W})$, where V is a vertex set which corresponds to data features, E is a set of hyperedges, $\mathbf{W}=\left[\boldsymbol{w}_1,\boldsymbol{w}_2,\ldots,\boldsymbol{w}_{|E|}\right]^T$ is a real-valued matrix in which each row stands for the weight vector of a specific hyperedge. In multi-label hyper-network, each hyperedge e_i is composed of three parts, i.e., a vertex set $v_i\subseteq V$, a label vector $\boldsymbol{y}_i=[y_{i1},y_{i2},\ldots,y_{ic}]$, and a weight vector $\boldsymbol{w}_i=[w_{i1},w_{i2},\ldots,w_{ic}]$. v_i and $\boldsymbol{y}_i$ are generated from a training data instance, $\boldsymbol{w}_i$ are parameters learned from training data.

4.2 Training of Probabilistic Memory

Multi-label hyper-network is also regarded as a probabilistic memory where a multilabel training data set is stored. In supervised learning, $P(\boldsymbol{x},y_i\,|\boldsymbol{W})(1\le i\le m)$, each training data instance $\boldsymbol{x}$ is associated with the label y_i through the hyperedge and their weight vector w_i. Therefore, the hyper-network represents the joint probability distribution of the input instance x_i and the class label y_i, as shown in Equation (7).

$$P(\boldsymbol{x},y_i\,|\boldsymbol{W})=\frac{1}{Z(\boldsymbol{W})}\exp\left(-\varepsilon(\boldsymbol{x},y_i;\boldsymbol{W})\right) \qquad (7)$$

where $\varepsilon(\boldsymbol{x},y_i;\boldsymbol{W})$ is the energy function which can be expressed in many ways, such as log function and sigmoid function. $Z(\boldsymbol{W})$ is a normalising term, $\boldsymbol{W}$ is the weight matrix of multi-label hyper-network, and $y_i\in\{0,1\}$.

In this paper, we utilise a log function to describe the energy function of multi-label hyper-network as shown in Equation (8), where $|E|$ is the total number of hyperedges in multi-label hyper-network, w_{ji} is the

label weight of hyperedge e_j.

$$\varepsilon\left(\boldsymbol{x}, y_i \middle| \boldsymbol{W}\right) = -\ln\left(\sum_{j=1}^{|E|} w_{ji} I\left(\boldsymbol{x}, y_i; e_j\right)\right) \quad (8)$$

where $I\left(\boldsymbol{x}, y_1; e_j\right)$ is a matching function defined in the last equation, which is explained in Equation (9).

$$I\left(\boldsymbol{x}, y_i; e_j\right) = \begin{cases} 1 & \text{if distance}(x, e_j) \le \delta \text{ and } y_{ji} = y_i \\ 0 & \text{other} \end{cases} \quad (9)$$

For any label y_i, it has only two values, 0 and 1. I can be also expressed as Equation (10)

$$I\left(\boldsymbol{x}; e_j\right) = I\left(\boldsymbol{x}, y_i = 1; e_j\right) + I\left(\boldsymbol{x}, y_i = 0; e_j\right) \quad (10)$$

For the matching function, y_{ji} represents whether the label i is in the hyperedge e_j, distance$\left(x; e_j\right)$ represents the Euclidean distance between the instance x and the hyperedge e_j, δ is the threshold, and the calculation Equation (11) is as follows:

$$\delta = \frac{1}{|D|} \sum_{x \in D} \frac{1}{|G_x|} \sum_{x' \in G_x} \|x - x'\| \quad (11)$$

where G_x represents the nearest neighbour set of data instance x; D represents a training data set.

4.3 Fine-grained Division of User Abnormal Behaviour

For data classification using hypernetworks, given input instance $\boldsymbol{x}$, we compute the probability of each label that belongs to the data instance, as shown in Equation (12).

$$P\left(y_i = 1 \middle| \boldsymbol{x}\right) = \frac{P\left(\boldsymbol{x}, y_i = 1\right)}{P\left(\boldsymbol{x}\right)} = \frac{\sum_{j=1}^{|E|} w_{ji} I\left(\boldsymbol{x}, y_i = 1; e_j\right)}{\sum_{j=1}^{|E|} w_{ji} I\left(\boldsymbol{x}, y_i = 1; e_j\right) + \sum_{j=1}^{|E|} w_{ji} I\left(\boldsymbol{x}, y_i = 0; e_j\right)} \quad (12)$$

where, w_{ji} represents the i-th weight value of the weight vector of the hyperedge e_j. For a given input instance $\boldsymbol{x}$, its class label y_i^* is determined by calculating the conditional probability of each class as in Equation (13) and selecting the class label with the highest conditional probability as the output.

$$y_j^* = g\left(P\left(y_j = 1 \middle| \boldsymbol{x}\right), \sigma_j\right) = \begin{cases} 1 & \text{if } P\left(y_i = 1 \middle| \boldsymbol{x}\right) \ge \sigma_j \\ 0 & \text{other} \end{cases} \quad (13)$$

where σ_j is the threshold of label j and is calculated using Equation (14)

$$\sigma_j = \arg\max\left(\sum_{i=1}^{N} f\left(y_{ij}, y_j^*\right)\right) \quad 0 \le \sigma \le 1 \quad (14)$$

$$f\left(y_{ij}, y_j^*\right) = \begin{cases} 1 & \text{if } y_{ij} = y_j^* \\ 0 & \text{if } y_{ij} \ne y_j^* \end{cases} \quad (15)$$

where y_{ij} represents the true value of the label j of the i-th training example. y_i^* is the value of the label j predicted by the threshold σ. If we find the label of the current instance at a time, we determine that this behaviour is normal user behaviour. Otherwise, the weight is updated, the hyperedge is replaced, and the label prediction is carried out again.

In the multi-label classification, all labels overlap in the tag space. In order to reduce the uncertainty of the label prediction, we introduce the k-nearest neighbour method. In the learning and classification process, for each instance its k-nearest neighbours are identified firstly, and then only hyperedges that are generated from these neighbours are used to match this instance. (Note that the nearest neighbours set a boundary of hyperedges that are used for matching.) In this case, we consider that only the nearest neighbour related labels are considered candidates for the label prediction. In the way of predicting the label, the uncertainty of prediction will be greatly reduced.

4.4 Weight Update

Multi-label hyper-network represents a high-order relationship between feature subsets and multiple category labels, which is essentially a data probability model suitable for data set training. The weight update process begins with the initial multi-label hypernetwork. Training data is generated from the training data set. Given a training instance $\left(\boldsymbol{x}, \boldsymbol{y}\right)$, where $\boldsymbol{x} = \left[x_1, x_2, \ldots, x_D\right]$, $\boldsymbol{y} = \left[y_1, y_2, \ldots, y_C\right]$, a hyperedge is generated by randomly sampling a subset of vertices from the vertex set, taking $\boldsymbol{y}$ as the label vector, and associating an initial weight vector to this hyperedge. Suppose a hyperedge $e = \left(v, \boldsymbol{y}, \boldsymbol{w}\right)$ is generated, $v = \left\{x_1, x_2, \ldots, x_i, \ldots, x_N\right\}$ is the vertex set, $\boldsymbol{y} = \left[y_1, y_2, \ldots, y_C\right]$ is the label vector, $\boldsymbol{w} = \left[w_1, w_2, \ldots, w_C\right]$ is the weight vector, and $w_i = 1.0$ is the initial weight value. The learning task of the hyper-network is to update the weight of the hyperedge, which minimizes the classification error on the training data.

Due to the randomness of the hyperedge generation, it is necessary to replace hyperedge when the first classification is unsuccessful, so that the training data can be better adapted to the hyperedges. In the process of hyperedge substitution, we first calculate the fitness for each hyperedge e_i and then generate a new

hyperedge set from the same training instance that generated hyperedge e_i. Next the fitness of the new hyperedge set is computed. If the fitness of a new hyperedge is greater than the fitness of the hyperedge e_i, the hyperedge e_i will be replaced by a new hyperedge. The fitness Equation (16) is shows as:

$$fitness(e_i) = \frac{1}{|G|} \sum_{(x_i, y_i) \in G} \frac{1}{m} \sum_{j=1}^{m} \left| \left\{ j \middle| y_{ij} = y'_j \right\} \right| \quad (16)$$

In the equation, G is a set of training instances that match with the hyperedge e_i; m is the number of possible labels, y_{ij} is the value of the label j of instance (x_i, y_i), and y'_j is the value of the label j in the label vector of the hyperbranched e_i.

Algorithm 1 The cloud user behavior authentication model based on multi-label hyper-network	
Step 1:	The data in the user's normal behavior database are selected and constructed by the equations (1), (2), (3), and (4).
Step 2:	The original hyper-network model is formed by equations (7), (8), (9), (10), and (11).
Step 3:	Enter real-time behavior x in the initialized hyper-network.
Step 4:	The real time behavior is classified by equations (12), (13), (14), and (15). **If** the label is found through classification successfully **then** cloud user is identified as normal users, update the database; **else** use the equation (16) to replace the hyperedge, update the hyper-network.
Step 5:	Enter user behavior into the updated hypernetwork. Use equations (12), (13), (14), and (15) for label search again. **If** the label is found through classification successfully **then** cloud user is identified as risk users **else** it is identified as a malicious user and an alert is triggered.

The fitness of the hyperedges is calculated as the average between the labels of the hyperedge and training samples. Matching with the training samples, the higher the similarity is, the greater the fitness is. As it can be seen from the equation in 4.3, the higher the similarity between the label of the training instance and the label of hyperedge that matches this instance is, the higher the probability that the instance is correctly classified. After the replacement of the hyperedges, the equation in 4.3 is applied again to classify the instances. The users are identified as risk users if the classification is successful. If they have not been classified successfully, they will be identified as abnormal users. The aforementioned algorithm is described as Algorithm 1.

5 Experimental Results and Analysis

In this paper, the KDD CUP99 data set simulation environment is used to verify the algorithm, and it is proved that the model can describe user behaviour intuitively and accurately. It effectively validates the discrimination effect of algorithm on malicious users and risk users. In addition, the model is compared with the classic K-means based anomaly analysis algorithm and the exception analysis algorithm that is based on the nearest method in the detection rate, detection speed and accuracy rate. The comparison shows that the proposed model is efficient and accurate in the malicious user identification.

5.1 Simulation Tool

The hardware environment of this experiment is Intel Core i5-2400CPU, main frequency 3.10GHz, and memory 4GB. The operation system is win7, 64 bits. MATLAB (R2014b) is used as the programming tool, since MATLAB is a scientific computing language widely used in the fields of scientific modelling, data visualization, and interactive simulation, published by MathWorks. The proposed algorithm under test is run in the above experimental environment.

5.2 Experiment Setup

Because of the limited access to practical cloud platform where user's real data normally resides, KDD CUP99 simulation dataset is adopted as the experimental data. The KDD CUP99 dataset contains two collections, one as training set and the other as test set. The types of attack in the training centre can be divided into four categories: DDOS (Distributed Denial-of-Service attack), R2L (unauthorized access from remote host), U2R (unauthorized local super user privileges access), PROBE (port monitoring or scanning). The number of specific data has been introduced in detail later.

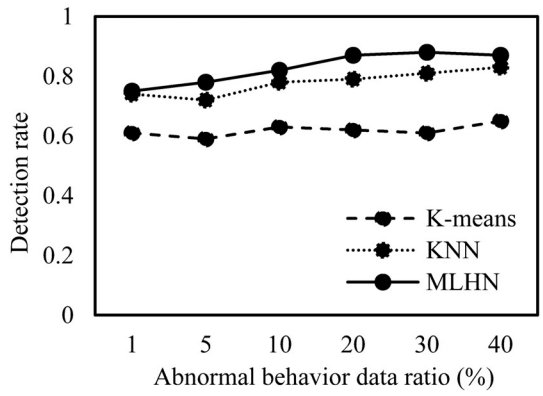
The experimental data set is very large. In order to facilitate the calculation, we choose only 10% of the training set and the test set to carry out the simulation test. The type and number of the attack are shown in Table 1.

Table 1. Attack type and number

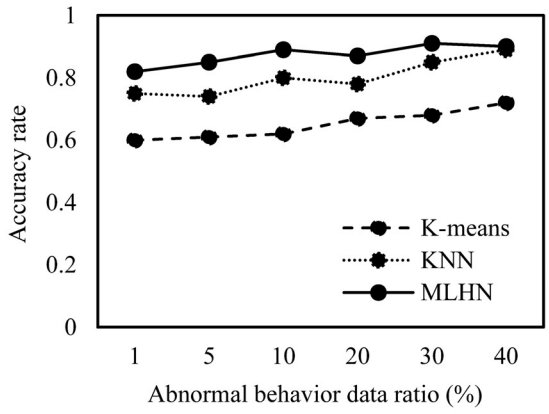
label	category	10% of the training set	10% of the text set
0	NORMAL	9727	6059
1	DDOS	7422	3629
2	PROBE	2152	271
3	R2L	99	358
4	U2R	5	22

The last bit (forty-second bits) of the experimental data set in this chapter records the type of attack. For convenience of computation, we only distinguish between the two states of attack and normal. When the last column (the forty-second bit of the data) is labelled as normal, the assignment is 1, while the other is the attack, and the assignment is -1.

First, we analyse the proportion of abnormal behaviours in data sets of different user behaviours. Then we choose different datasets to build training samples for multi-label training. Finally, the same verification data (random extraction from all user behaviour data) is used to compare the accuracy of the text method and the control method under the training samples of different balance degrees, and a suitable balance should be chosen. Figure 2 gives the comparison results of the accuracy index of training samples in the proportion of 1%, 5%, 10%, 20%, 30% and 40%, respectively. It can be seen from the figure that most of the control methods have the phenomenon that the accuracy increases gradually with the increase of the proportion of abnormal behaviour in the training samples.



(a) Comparison of detection rate



(b) Comparison of accuracy rate

Figure 2. Data balance analysis

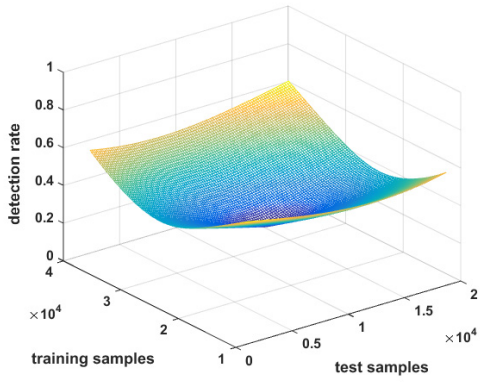
However, the algorithm proposed in this paper is relatively stable in overall performance, and it is more stable at 20%. Therefore, we choose the proportion of abnormal behaviour data to be around 20% in the next simulation experiment.

After determining the degree of data balance, we will verify the fine-grained division of users. Two groups of users are extracted from the dataset with abnormal behaviour ratio of 20% at arbitrary and unrepeated, which are recorded as TRAIN and TEST. TRAIN is used for training multi-label hyper-network, and TEST is used for algorithm verification. In order to verify the recognition ability of the behaviour identification model proposed for malicious users and risk users, we use KDD99 datasets to simulate the behaviour of two types of users respectively. Malicious users refer to forty-second bits of dataset, which are DDOS, R2L, U2R and PROBE data. Risk users will behave abnormally at some time, but normally in most cases. Therefore, we change the time-based network traffic statistics of the data, send a large number of HTTP requests to the target host at a certain time, and simulate the risk type users at a time. This behaviour is very similar to malicious users on the surface, but its attack time is very short.

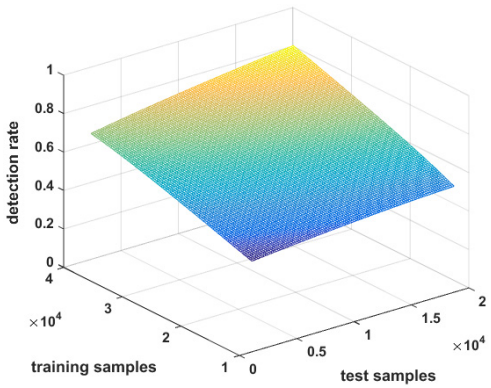
5.3 Simulation Results

This experiment uses the dataset TEST to test the cloud user behaviour identification model, and compares this model with the classic K-means based anomaly analysis algorithm and the abnormality analysis algorithm based on nearest neighbour method. 50,000 data were selected for model training (the proportion of abnormal behaviour data was selected to be around 20%). Then the test set was tested randomly. We simulate the risky user behaviour by sending HTTP requests with lots of random contents at random time. The content of requests is random pages. This user increases the number of access to sensitive pages, resulting in an exception of throughput and clicks, but the attack time is short and this user does not belong to a malicious user. (The forty-second bit flag is still Normal.)

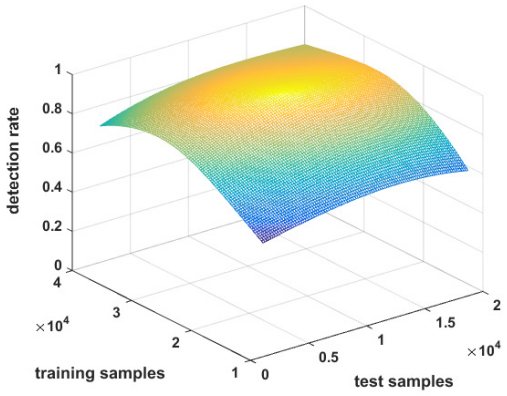
The experiment selects three key indexes, detection rate (DR), detection speed (DS) and accuracy rate (AR) to measure the detection performance of the model proposed in this paper. Among them, the detection rate refers to the proportion of the detected quantity of the attack user and the total number of the attack samples. The accuracy rate refers to the proportion of the detection amount of the malicious user and the total number of the abnormal samples (mainly on the number of the risk type users). The detection rate, detection speed and accuracy of the model proposed in this paper are tested by changing the proportion of malicious users and risk users. The algorithm is also assessed by be compared with the classic K-means-based anomaly analysis algorithm [28] and the abnormality analysis algorithm based on nearest neighbour method [10], as shown in Figure 3.



(a) K-means-based anomaly analysis algorithm



(b) Abnormality analysis algorithm based on nearest neighbour



(c) Abnormality analysis algorithm based on multi-label hyper-network

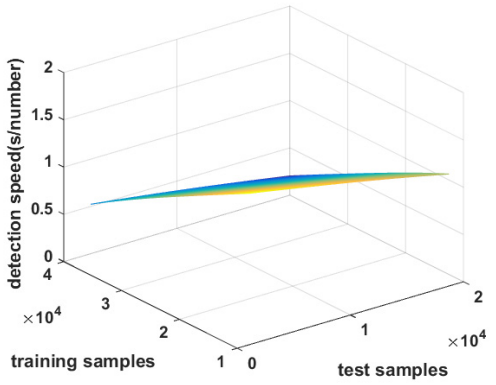
Figure 3. The comparison of detection rate

Figure 4 is the comparison of the detection speed (DS) of the three algorithms with different number of training samples and test samples. The X axis represents the number of test samples, the Y axis represents the number of training samples, and the Z axis represents the detection speed (S/number). The recognition speed of three user abnormal behaviour analysis algorithms is improved with the increase of test samples. In this paper, we use multi-label hyper-network to train user behaviour, and classify the current behaviour directly. The algorithm increases the speed of detection samples.

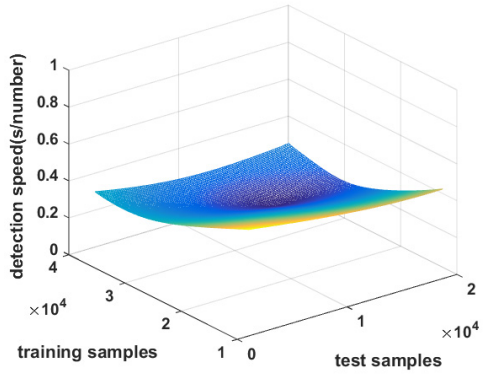
Figure 5 shows the comparison of three user abnormal behaviour analysis algorithms with different number of training samples and test samples. The X axis represents the number of test samples, the Y axis represents the number of training samples, and the Z axis represents the accuracy rate. With the increase of test samples, the accuracy rate has been improved. Owing to the influence of noise, the classic K-means based anomaly analysis algorithm is very unstable in the accuracy rate. The user behaviour analysis method based on nearest neighbour cannot identify DDOS attacks, so the accuracy rate of identifying abnormal behaviours will be affected. The algorithm proposed in this paper is not easy to be affected by noise and, so it can recognise DDOS attack. It also has certain recognition ability for the unknown attack type. Therefore, the results show outperformed stability and high accuracy of the proposed algorithm.

6 Conclusion

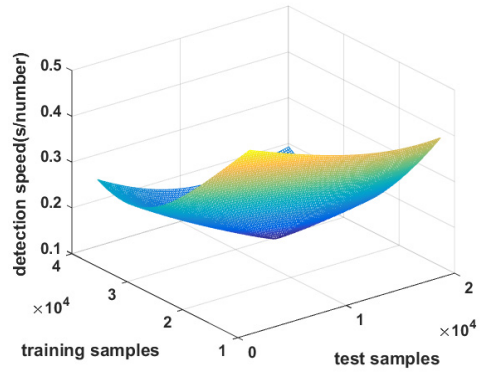
Through the analysis of user's behaviour data and the simulation of risk and malicious user behaviour, our research provides a scientific verification of the cloud user behaviour identification model based on multi-label hypernetwork. Through the comparison with other models, the experiment shows the advantages of our proposed model using different metrics. The proposed model in this paper can establish a good mutual trust relationship between users and mobile cloud environment, achieving fine-grained division of user behaviour and improving the accuracy of anomaly detection. However, the model is not effective enough to identify any unknown attack types accurately. There will be more research efforts being put into the future work to implement prediction and recognition of unknown attack types using metrics of detection rate and detection speed.



(a) K-means-based anomaly analysis algorithm

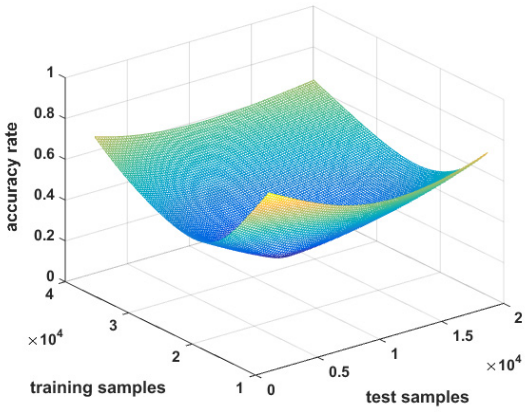


(b) abnormality analysis algorithm based on nearest neighbour

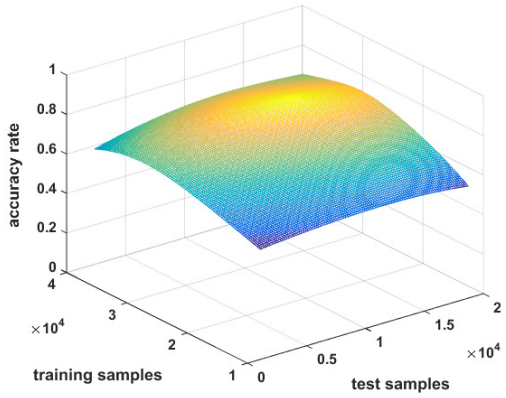


(c) abnormality analysis algorithm based on multi-label hyper-network

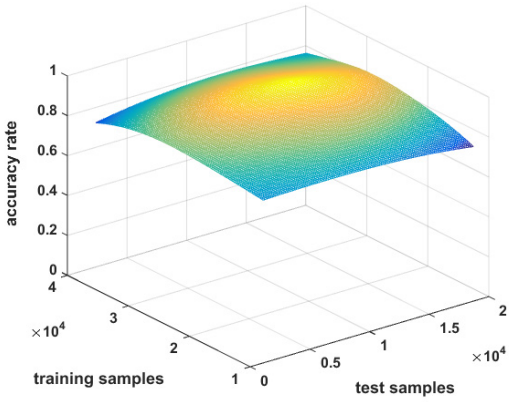
Figure 4. The comparison of detecting speed



(a) K-means-based anomaly analysis algorithm



(b) abnormality analysis algorithm based on nearest neighbour



(c) abnormality analysis algorithm based on multi-label hyper-network

Figure 5. The comparison of accuracy rate

Acknowledgement

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grants no. 61801171, in part by the China Postdoctoral Science Foundation under Grant no. 2018M630461, and in part by the Science Foundation of Ministry of Education of China under Grants No. 19YJC630174. Ruoshui Liu and Xin Wang contributed equally to this work.

References

[1] M. W. Li , Q. T. Wu , J. L. Zhu, R. J. Zheng, M. C. Zhang, A Computing Offloading Game for Mobile Devices and Edge Cloud Servers, *Wireless Communications and Mobile Computing*, Vol. 2018, Article ID 2179316, December, 2018.

[2] K. Peng, R. H. Lin, B.B. Huang, H. Zou, F. C. Yang, Link Importance Evaluation of Data Center Network Based on Maximum Flow, *Journal of Internet Technology*, Vol. 18, No. 1, pp. 23-31, January, 2017.

[3] I. A. Abbasi, A. S. Khan, S. Ali, Dynamic Multiple Junction Selection Based Routing Protocol for VANETs in City Environment, *Applied Sciences*, Vol.8, No.5, pp. 1-8, April, 2018.

[4] M. C. Zhang, M.Y. Yang, Q. T. Wu, R. J. Zheng, J. L. Zhu, Smart Perception and Autonomic Optimization: A Novel Bio-Inspired Hybrid Routing Protocol for MANETs, *Future Generation Computer Systems*, Vol. 81, pp. 505-513, April, 2018.

[5] W. Quan, Y. N. Liu, H. K. Zhang, Enhancing Crowd Collaborations for Software Defined Vehicular Networks, *IEEE Communications Magazine*, Vol. 55, No. 8, pp. 80-86, August, 2017.

[6] M. C. Zhang, P. Xie, J. L. Zhu, Q. T. Wu, R. J. Zheng, H. K. Zhang, NCPP-based Caching and NURbased resource Allocation for Information-centric Networking, *Journal of Ambient Intelligence & Humanized Computing*, No. 4-5, pp. 1-7, October, 2017.

[7] H. K. Zhang, W. Quan, H. C. Chao, Smart Identifier Network: A Collaborative Architecture for the Future Internet, *IEEE Network*, Vol. 30, No. 3, pp. 46-51, May-June, 2016.

[8] C. X. Zhou, Z. M. Cui, G. Y. Gao, On the Security of an Improved Identitybased Proxy Signature Scheme without Random Oracles, *Journal of Internet Technology*, Vol. 19, No.7, pp. 2057-2068, December, 2018.

[9] J. Du , P. Xie, J. L. Zhu, R. J. Zheng, Q. T. Wu, M. C. Zhang, Method for Detecting Abnormal Behaviour of Users Based on Selective Clustering Ensemble, *IET Networks*, Vol. 7, No. 2, pp. 85-908, March, 2018.

[10] J. N. Chen, Selective Bayes Classification Algorithm Research, Ph. D. Thesis, Beijing Jiaotong University, China, 2008.

[11] S. F. Ding, J. Zhang, X. K. Zhang, Y. X. An, Survey on Multi Class Twin Support Vector Machines, *Journal of Software* (in Chinese), Vol. 29, No. 1, pp. 89-108, 2018.

[12] H. X. Qiao, Research and Realization of the KNN Classification Algorithm Based on MapReduce, Ph. D. Thesis, Beijing Jiaotong University, China, 2012.

[13] A. K. Singh, C. C. Sekhar, A Two-Stage Conditional Random Field Model Based Framework for Multi-Label Classification, Pattern Recognition and Machine Intelligence, 2017.

[14] G. Madjarov, I. Dimitrovski, D. Gjorgjevikj, S. Dzeroski, Evaluation of Different Data-Derived Label Hierarchies in Multi-label Classification, New Frontiers in Mining Complex Patterns, *NFMCP'14 Proceedings of the 3rd International Conference on New Frontiers in Mining Complex Patterns*, Nancy, France, 2014, pp. 19-37.

[15] E. L. Mencía, F. Janssen, Learning Rules for Multi-label Classification: A Stacking and a Separate-and-conquer Approach, *Machine Learning*, Vol. 105, No. 1, pp. 1-50, April, 2016.

[16] S. Daengduang, P. Vateekul, Enhancing Accuracy of Multi-label Classification by Applying One-vs-one Support Vector Machine, *13th International Joint Conference on Computer Science and Software Engineering*, Khon Kaen, Thailand, 2016.

[17] G. Varando, C. Bielza, P. Larrañaga, Decision Functions for Chain Classifiers Based on Bayesian Networks for Multi-label Classification, *International Journal of Approximate Reasoning*, Vol. 68, pp. 164-178, January, 2016.

[18] R. Senge, J. J. D. Coz, E. Hüllermeier, On the Problem of Error Propagation in Classifier Chains for Multi-label Classification, Data Analysis, Machine Learning and Knowledge Discovery, *36th Annual Conference of the German Classification Society,* Hildesheim, Germany, 2012, pp. 163-170.

[19] G. Huang, Y. Yang, J. Bai, Selective Classifier Chains Based on Max-relevance and Min-redundancy for Multi-label Classification, *Iaeng International Journal of Computer Science*, Vol. 44, No. 3, pp. 327-336, July, 2017.

[20] Z. Zhang, H. Wang, L. Li, G. F. Liu, Core Network Based Multi-label Classification in Large-Scale Social Network Environments, *IEEE International Conference on Data Mining Workshop*, Atlantic, NJ, USA, 2016, pp. 940-947.

[21] K. H. Huang, H. T. Lin, Cost-sensitive Label Embedding for Multi-label Classification, *Machine Learning*, Vol. 106, No. 9-10, pp. 1725-1746, October, 2017.

[22] X. Kong, B. Cao, P. S. Yu, Multi-label Classification by Mining Label and Instance Correlations from Heterogeneous Information Networks, *19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Chicago, IL, USA, 2013, pp. 614-622.

[23] A. Tharwat, H. Mahdi, A. E. Hassanien, Plant Recommender System Based on Multi-label Classification, *International Conference on Advanced Intelligent Systems and Informatics*, Cairo, Egypt, 2016.

[24] Q. Tan, Y. Liu, X. Chen, G. X. Yu, Multi-Label Classification Based on Low Rank Representation for Image Annotation, *Remote Sensing*, Vol. 9, No. 2, pp. 109, January, 2017.

[25] A. Wosiak, K. Glinka, D. Zakrzewska, Multi-label

Classification Methods for Improving Comorbidities Identification, *Computers in Biology & Medicin*e, Vol. 100, No. 1, pp. 279-288, September, 2018.

[26] A. K. Singh, and C. C. Sekhar, A Two-Stage Conditional Random Field Model Based Framework for Multi-Label Classification, Pattern Recognition and Machine Intelligence, 69-76, 2017.

[27] K. Laghmari, C. Marsala, and M. Ramdani, A Distributed Recommender System Based on Graded Multi-label Classification, Networked Systems, 2017.

[28] K. Wang, Research and implementation of network anomaly detection based on clustering, Ph. D. Thesis, Beijing University of Posts and Telecommunications, China, 2016.

Biographies

Ruoshui Liu was born in Zhengzhou, China in Aug. 1983. He received his MEng degree with first class from University of York in 2007 and Ph.D. degree in Computer Science from University of Cambridge in 2012. His research interests include wireless sensor networks, Machine Learning and Cloud Computing.

Xin Wang was born in Harbin, China in April in 1980. She received Ph.D. degree in Management Science and Engineering at University of Posts and Telecommunications in China in 2017. She is now the assistant professor in Shanghai International Studies University. Her research interests include Neuroscience, artificial intelligence, Information System and Networks.

Juan Du was born in Luoyang, Henan Province, PRC in Mar. 1992. She received the master's degree in computer science and technology from the Henan University of Science and Technology, Luoyang, China, in 2018. Her research interests include cloud computing, anomaly detection.

Ping Xie was born in Hunan Province, PRC in Mar 1984. She received the Ph.D. degree in communication and information system from the Beijing University of Posts and Telecommunications, Beijing, China, in 2014. She works as a Lecture in Henan University of Science and Technology from Sept. 2014 to now. Her research interests include cognitive networks, physical layer security.