ATLANTIS PRESS
Advances in Computer Science Research, volume 74
2nd International Conference on Computer Engineering, Information Science & Application Technology (ICCIA 2017)

Graph Structure Based Anomaly Behavior Detection

Kai Wang [a], Danwei Chen [b]
School of Computer, Nanjing University of Post and Telecommunications, Nanjing 210003, China;
[a]i.am.wack.1993@gmail.com, [b]chendw@njupt.edu.cn

Keywords: Anomaly Detection, Graph Mining, Unsupervised Learning, Social Graph.

Abstract. The analysis of malicious user behavior patterns in social networks has important implications for detecting malicious pages, fraudsters, and financial frauds.Traditional anomaly detection technology general based on classification algorithm using content feature and user behavior feature, but these type of methods are often with low efficiency, data acquisition difficulty and ignoring the network topology information.This paper puts forward a network graph structure based, unsupervised anomaly detection algorithm GBKD-Forest, we extracted three types of structure characteristics, within the Bagging method random sampling features to establish KD-Tree Forest, to isolate the abnormal samples.Evaluation through the experiment, the proposed algorithm in terms of accuracy and AUC is superior to other graph based anomaly detection algorithm and classical classification algorithm, at the same time, the time complexity of this algorithm has a linear relation with the number of nodes, low space complexity is suitable for large-scale network anomaly detection datasets.

1. Introduction

With the rapid development of Internet, people could contact each other in social network and shop online. However there are lots of risk behind the convenience of social medium—fake review, fake followers, phishing website, telecommunications fraud. The complex behavior of user makes it had to predict people's behavior and anomaly detection.

Existing anomaly behavior detection include content based method [1], behavior feature based method [2] and graph based method [3].Content based suspicious detection technology generally based on the user's personal information and the content of the message they published. Malicious users may publish spam ads, malicious links or illegal content but normal user won't. Behavior feature based detection focus on user's behavior, as for social network these feature include message sent time, number of tweets, comment and forward and online active time. Different from content based method, behavior based method classify the user according to the occurrence frequency of the content rather than the content itself. The advantage of this two kinds of method is their high accuracy of prediction, and their shortcoming is that the content is hard to collect, analysis and storage, the computational complexity is very high.

Graph-based anomaly detection methods have raised a generous concern abroad these years. Graph theory and machine learning method perform excellent in this area, outlier is an observation that differs so much from other observations as to arouse suspicion, and in a (static/dynamic) graph outlier is node, edge or substructure that differ from majority of other objects in the graph. The advantage of graph include: (1) Strong representation of data: such as who-follows-whom in Twitter, who-rates-what in Amazon and who-likes-what in Facebook these data can be absrtracted as directed/undirected, weighted/unweighted graph (2)Powerfull representation of relationship and reliance between objects: graph edge can represent friendship between users, dependency of related objects (3) Good rubust: different from content-based methods, graph-based methods is difficult for attacker to bypass.

Over the last few years, the security of social network has attracted the attention of researchers. Graph mining and user behavior analysis has aroused wide concern, graph anomaly is defined as: given a (plain/attributed, static/dynamic) graph database, find the graph objects (nodes/ edges/ substructure) that are rare and that differ significantly from the majority of the reference objects in

the graph [4]. In graph mining some method provide binary 0/1 classification of data points, i.e. outlier/normal while most method using outlier score the measure the level of outlierness of objects [8,9,10,11]. The goal of anomaly detection technology based on graph is to find nodes, edges or substructure that significantly differs from other reference, scholars in recent years in graph mining and anomaly detection has made a lot of research progress .

Breunig et al. [8] proposed a density based anomaly detection method LOF(Local Outlier Factor), based on fitting probability density function like k-Nearest Neighbor, The disadvantages of LOf is the result depends on the selection of parameter K and the computational complexity of this algorithm is not suitable for large networks.

Akoglu et al. [9] proposed a feature based anomaly detection method OddBall on weighted graph. By extracting 4 Egonet features simplifies the complexity but this method is only applicable to weighted graph and only consider the node's local characteristic.

Sun et al. [10] proposed a random walk based anomaly detection method within graph partition in bipartite graph. This method only use PageRank to define nodes' outlierness is not accurate enough and is easy for attackers to bypass.

Ding et al. [11] proposed a community-based intrusion detection algorithm, which consider bridging nodes or edges across communities are anomaly.But this approach is difficult to detect outlier in community and is not accurate enough .

The main contributions of this paper are as follow:

(1) Extract feature using graph based theory. We extract 3 types of feature: basic graph information, connection information and egonet information.

(2) We proposed a parameter-free unsupervised Anomaly detection algorithm GBKD-Forest (Graph Based K-Dimension Forest), by constructing KD Tree using graph based feature and within bagging method to improve the classification accuracy.

(3) Using real social network dataset and synthetic data to evaluate our algorithm, compare to existing method, our algorithm perform better on detecting fraudulent user and bots both in accuracy and efficiency.

The paper is organized as folows. The firs section is introduction and related work. In section 2, we discuss the three graph feature extraction model proposed in this paper. Section 3 introduce the the basic theory of our algorithm and explain it in detail. In section 4, we analyse our algorithm with experiment on large social network dataset, detecting anomaly nodes(users) and compare to existing method in accuracy and efficiency and finally section 5 for conclusion.

2. Graph Feature Extraction

For network graph model , given graph G = (V, E), V stands for node set, E represents edge set, the vertices number N = | V |, the number of edges M=| E |. This paper extract three types of feature.

2.1 Local Feature

The adjacency matrix of a directed graph $A_{n\times n}$, the in-degree of a node is $D_{in}(\mathrm{i}) = \sum_{j=1}^{n} a_{i,j}$, and out-degree is $D_{out}(i) = \sum_{j=1}^{n} a_{j,i}$, the sum of in-degree equals to the sum of out-degree in network. In social media, large in-degree means the user has a lot of fans while the degree that significantly greater than average may be obtained by buying. Out-degree represent the number of users the account follows, while zombie or bots trend to have similar out-degree.

2.2 Connection feature and Centrality Measure

This paper selects the nodes' PageRank, HITS values, and Centrality measurements.PageRank is a kind of node importance ranking algorithm based on random walk. For a given network graph adjacency matrix $M_{n\times n}$, set the initial transfer vector V_0 as uniform distribution, then $V_{i+1} = M\,V_i$, calculate until the iteration convergence gots V, and adding a damping coefficient alpha,then the steady state vector $\mathrm{V}' = (1-\alpha)\mathrm{MV} + \alpha V_0$. The PR value indicates the importance of the node, and the PR value of the anomaly is generally lower than the normal ones.

HITS algorithm [5] proposed the Hubness and Authoritativeness concept, for "Follower - Followee" network graph adjacency matrix, the first left singular vectors represent Hubness, and the

first right singular vectors represent Authoritativeness. Celebrities generally have higher Authoritativeness, while the fans that connect many celebrities have higher Hubness. Authoritativeness of abnormal accounts is lower than normal users, and the Hubness of fake followee is lower than normal users. According to the literature [6], we can easily distinguish normal users and suspicious users through these two groups of features.

In addition, this paper select 4 Centrality feature to measure the vertices' importance in graph. Degree Centrality [Linton] is defined as the number of links incident upon a node:

$$C_{Deg}(u) = \sum_{i=1}^{N} a_{i,j} \ (i \neq j) \tag{1}$$

Closeness Centrality [Bavelas] is defined as the average length of the shortest path between the node and other nodes in the graph, d(y,x) represent shortest path between x and y:

$$C_{close}(u) = \frac{N}{\sum_{i=1}^{N} d(y,x)} \tag{2}$$

Betweenness Centrality [Freeman] is defined as the count of a node go through the shortest path between other two nodes, σ_{st} represent shortest path between s and t， $\sigma_{st}(u)$ represent the number of path that go through node **u**:

$$C_{Bet}(u) = \sum_{u \neq s \neq t} \frac{\sigma_{st}(u)}{\sigma_{st}} \tag{3}$$

Eigenvector Centrality is defined as the average centrality of node's neighbors. N(v) represent the collection of neighbors:

$$x_v = \frac{1}{\lambda} \sum_{t \epsilon N(v)} x_t = \frac{1}{\lambda} \sum_{t \epsilon G} a_{v,t} x_t \tag{4}$$

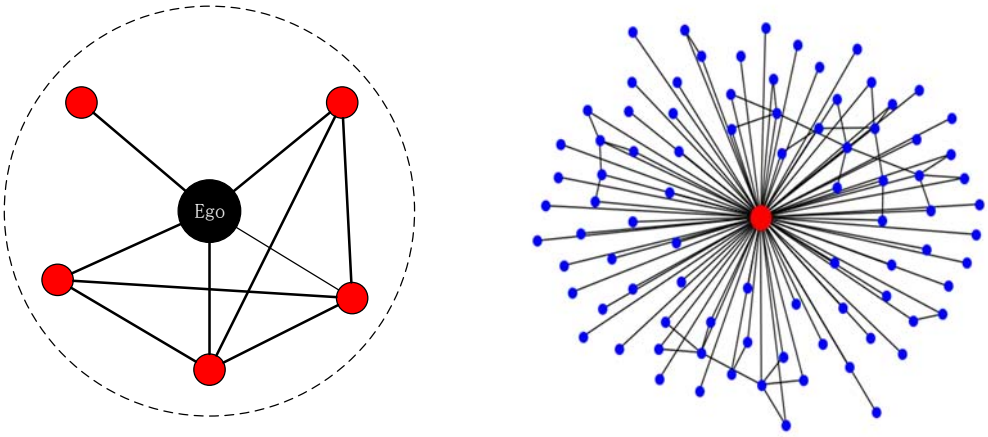

Fig. 1 Ego Network

2.3 Ego Feature

Ego Network consist of a focal node **u** and other nodes that directly connected to it, alone with edges between them, as shown in figure 1. Egonet feature is an importance local feature of a node and it community. The features of EgoNet we extract are shown in Table 1.

Table 1. Ego Network Feature

Symbol	Definition and Description
N_i	Number of nodes in Ego
E_i	Number of edges in Ego
λ_i	Principle eigenvalue
T_i	Number of Triangle
ND_i	Average Degree of neighbor
NE_i	Average Edges of neighbor

3. GBKD-Forest

Anomaly nodes are usually outliers that distributed sparse in the sample, their characteristics significantly different from normal ones in some dimensions, so easy to be separated by classification tree, random forests construct by multiple decision trees separate these abnormal points easily, outlier generally have lower average height [7].

3.1 KD-Tree

K-Dimension Tree is a space partitioning data structure that divides the characteristics of the K dimensional space and can quickly search them. As shown in Figure 2, 100 sample points of a two-dimensional Gaussian distribution requires only a height of 4 KD-Tree with to distinguish normal and abnormal values. The KD tree construction method used in this article is shown in Algorithm 1.

Algorithm 1: KD-Tree(root, X, K, h', h)

Input : root-Root of tree, X-Input data, K-Number of Dimension, h'-Current height, h-Limited height
Output : root of KD-Tree

1. **if** h'<h **and** $|X| > 1$ **then**
2. **for** i=1 to K **do**
3. d←Choose the dimensions with the greatest variance
4. d_j←select a split value in d-th dimension
5. $X_l \leftarrow \{X | X_d < d_j\}, X_r \leftarrow \{X | X_d \geq d_j\}$
6. return root{ left←KD-Tree(root->left,X_l,K,h'+1,h),
7. right←KD-Tree(root->right, X_r,K,h'+1,h),
8. splitAttr←d,
9. splitVal←d_j }
10. **else**
11. **return** root

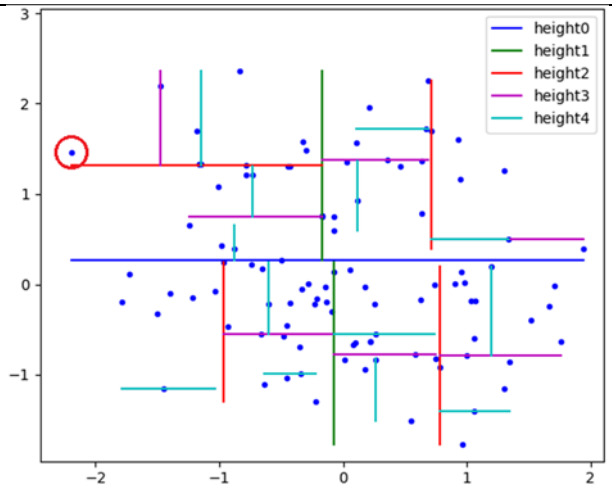

Fig. 2 KD-Tree Division (height=4)

3.2 KD-Forest

Ensemble Learning method can combine different classifiers to improve the generalization ability of the overall classifier.Bagging is a representative of parallel ensemble learning method, the first step is using Bootstrapping sampling to get the sample set $D_i(i = 1 \ldots n)$, and then training multiple classifier $H_i(i = 1 \ldots n)$,classification result is vote by multiple base classifier. In this way, the generalization ability of the base classifier can be effectively reduced, and reduce the errors caused by the random selection of the training data.

This paper proposes GBKD - Forest (Graph-Based K-Dimension Forest) anomaly detection algorithm, as an improvement of Isolation Forest [7] and random Forest. This algorithm is a fast anomaly detection algorithm based on Bagging method, builds random forest with KD-Tree, the method is shown in Algorithm 2.

KD-Tree forest take random sampling process twice, the first one is input sampling and the second one is dimension sampling. This process ensure the training will not easily get over fitting. Moreover

the construction of KD tree has near linear time complexity($O(n \log n)$), and the search time is $\log n$, also it only needs O(n) memory(n is the number of sampling).

Algorithm 2: KD-Forest(X, K, n, h, ψ)

Input : X-Input data , K-Number of dimension , n-Number of Tree ,h-Limited height , ψ-Sampling size
Output : KD-Forest
1. Set limited height h=$\log_2$ ψ,Forest={ϕ}
2. **for** i=1 to n **do**
3. X'←Bootstrap(X, ψ)
4. K ←Random select K dimension from all the Feature
5. Forest←Forest ∪ KD-Tree(root,X',K,0,h);
6. **end for**
7. **return** Forest

3.3 Anomaly Detection

Outliers are distributed sparse and away from dense group in the feature space, we can distinguish them with normal in decision tree. The fewer steps the samples take in a path, the more abnormal they are. The average height of abnormal and normal points generated by gaussian distribution is shown in Figure 3(maximum height is 11), average height of abnormal is lower than normal.The proposed algorithm in this paper is an unsupervised learning model without need to label the sample in advance, which is an improvement of Random Forest.

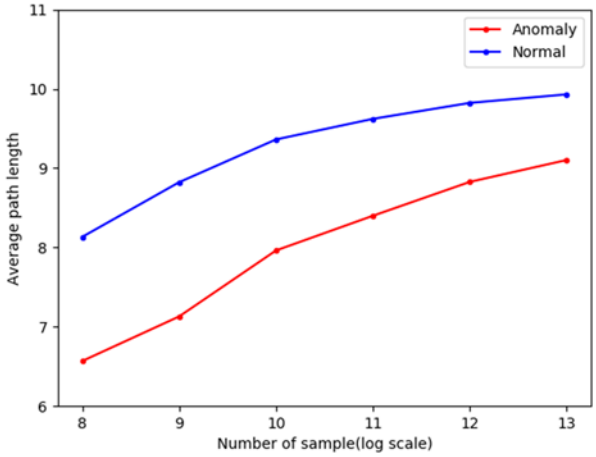

Fig. 3 Average height of abnormal and normal

KD-Tree is an extension of BST(Binary Search Tree) in high dimension space. Given a sample of n, let H_n be the height of a random BST, the average height $H = E(H_n) = \alpha \ln n + \beta \ln \ln n + O(1)$ where $\alpha = 4.311, \beta = 1.953$ [13], let $E(P_x)$ to be the average height of a sample in forest and E(P) be the average height of all samples. We define the Anomaly Score as :

$$S(x, h) = \frac{2}{1 + e^{\frac{E(P_x) - E(P)}{E(P_x)}}} \tag{5}$$

When $E(P_x) \to H$, $S < 0.5$, when $E(P_x) \to 0$, $S \to 1$ and when $E(P_x) \to E(P)$,S=0.5,as in Figure 4

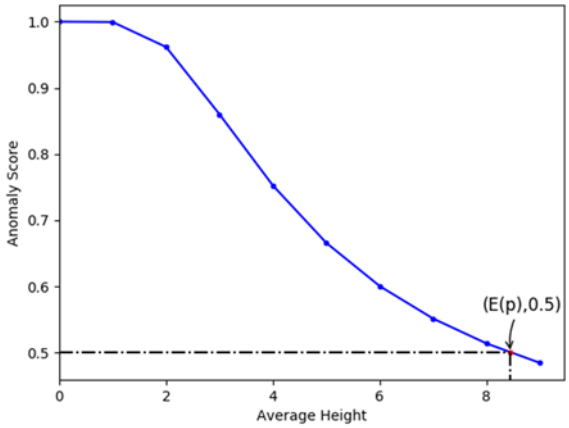

Fig. 4 Anomaly score with average height

4. Experiment Evaluation

4.1 Data Set and Evaluation

The data of network topology graph is large, and the cost of manual markup is expensive. Existing data sets do not contain labeled categories information. In this paper, synthetic data and real social network data are used, we injecting abnormal nodes and compare the proposed algorithm with existing graph-based anomaly detection algorithm. The graph structure information is shown in Table 2, and we use WS small-world Network generate synthetic data and inject anomaly nodes with edge.

Table 2 Data Set

Name	Type	Edge	Nodes	Anomaly	Percentage
Facebook	Undirected	96,140	4,039	200	4.95%
Wiki-Vote	Directed	111,662	7,115	400	5.62%
Synthetic	Directed	119,120	10,000	500	5.00%
Gnutella-P2P	Bipartite	74,698	22,687	1,000	4.41%
Email-Enron	Directed	427,643	36,692	2,000	5.45%

In this paper we use the following evaluation to evaluate the classification algorithm: Accuracy, Precision, F-Measure . As for the characteristics of anomaly detection is a binary classification problems with the distribution imbalanced of positive and negative samples, using Recall - Precision curve is highly associated with sample so we use AUC(Area Under Curve) to measure the performance.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{FP} + \text{TN}} \qquad \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{6}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \qquad \text{F1} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{7}$$

4.2 Result

At first we use synthetic data to compare our algorithm with LOF, the result is shown in Table 3. The average classification accuracy, AUC evaluation indexes of LOF is lower than GBKD-Forest, LOF is a distance based graph model, the time complexity is O (n^2) and differs significantly on the selection of parameter k distance (Figure 5), network with 5000 nodes needs 10 minutes computing, This kind of anomaly detection algorithm based on distance is not suitable for large network graph, so for large data set we did not do experiment using LOF.

Table 3. LOF and GBKD-Forest Result on Synthetic Data

Synthetic	LOF						GBKD-Forest
Nodes	1000	2000	3000	4000	5000	Average	10000
Precision	0.847	0.844	0.841	0.851	0.847	0.846	0.952
Accuracy	0.822	0.818	0.816	0.819	0.818	0.819	0.954
F1	0.762	0.749	0.741	0.704	0.784	0.748	0.96
AUC	0.712	0.704	0.705	0.702	0.696	0.704	0.940

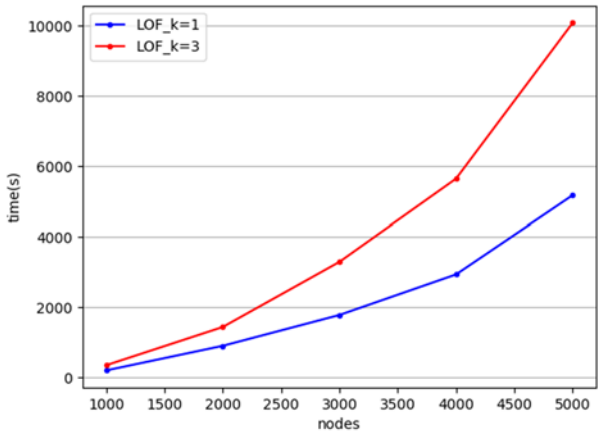

Fig. 4 LOF Computing Time

Anomaly detection of bipartite graph this paper selected the Wiki voting information, P2P network data and Email communication data, with random-walk based Exact NF algorithm to compare. The

classification performance of GBKD-Forest is superior to that of Exact NF as Table 4, and the performance of the Exact NF in different data sets is not stable, its AUC value fluctuates greatly.

Table 4. Exact NF and GBKD-Forest Result nn Bipartite Graph

	Exact NF			GBKD-Forest		
Nodes	7,115	22,687	36,692	7,115	22,687	36,692
Precision	0.612	0.725	0.767	0.964	0.968	0.973
Accuracy	0.663	0.869	0.892	0.960	0.951	0.964
AUC	0.674	0.914	0.809	0.962	0.967	0.968

Table 5 shows the AUC of LOF, Exact NF, SVM, LR and GBKD-Forest algorithm proposed in this paper, the proposed algorithm is superior to other algorithm in most of the data set, the SVM classification performance better in some data, but its classification performance is not stable and time complexity is high.

Table 5. AUC of different algorithm

Dataset	LOF	Exact NF	SVM	Random Forest	GBKD-Forest
Facebook	0.582	0.954	0.649	0.958	**0.972**
Wiki-Vote	0.512	0.674	0.500	0.961	**0.962**
Synthetic	0.704	0.946	**0.988**	0.946	0.940
Gnutella-P2P	NA	0.914	**0.978**	0.920	0.967
Email-Enron	NA	0.809	0.500	0.918	**0.968**

4.3 Time Complexity

As shown in table 6, the classification time of graph based KD-Forest algorithm is far lower than the LOF, and the time of SVM algorithm increases significantly with the increase of sample. Also GBKD-Forest is faster than Random Forest in large datasets because of low memory requirement .

Table 6. Classification Time

DataSet	Nodes	Feature Extract(s)	Classification Time(s)			
			GBKD-Forest	SVM	RF	LOF
Facebook	4,039	63	0.451	0.449	0.380	2940
Wiki-Vote	7,115	125	0.475	0.571	0.437	12361
Synthetic	10,000	278	0.510	0.576	0.469	17714
Gnutella-P2P	22,687	579	0.912	1.479	1.372	NA
Email-Enron	36,692	2,683	1.394	4.710	2.663	NA

As shown in Figure 5 in this paper, the graph structure information extraction time and classification time is nearly linear growth with the number of nodes in the network. Owing to eognet feature need recursive calculation, so the feature extraction time cost a long time. But classification time period is fast.

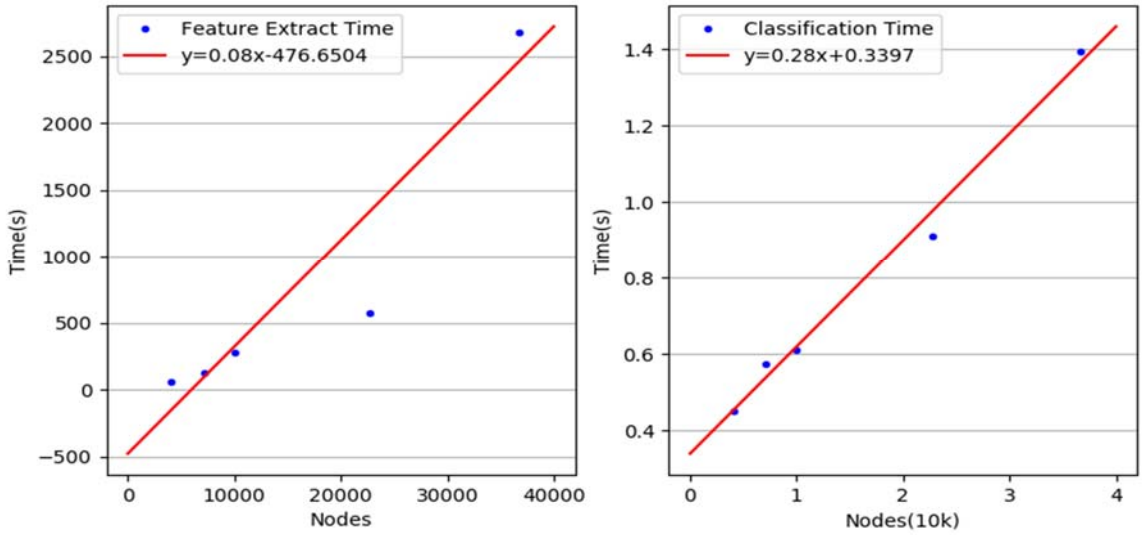

Fig. 5 Feature Extract and Classification Time

5. Conclusion

This paper proposed an unsupervised anomaly detection algorithm based on network topology structure information GBKD-Forest, using Ensemble learning Bagging method. Constructing multiple KD-Tree using different sample, randomly selected K dimension from all feature space and using feature with biggest variance as split feature until split completely or achieve limited height. For the nodes that are easy to partition is considered to be abnormal, which is reflected in the average tree height of the sample.

Comparing with the distance based algorithm LOF, random walk based Exact NF, SVM and Random Forest classification algorithm. GBKD-Forest has not only the highest accurate, but also has higher AUC than most of other algorithms. At the same time, the algorithm's classification time complexity is nearly linear with the number of nodes, and the memory requirement is very low.

References

[1]. Jindal N, Liu B. Analyzing and Detecting Review Spam [J]. 2015, PP(99):3367-3381.
[2]. Kriksciuniene D, Liutvinavicius M, Sakalauskas V, et al. Research of customer behavior anomalies in big financial data[C]// International Conference on Hybrid Intelligent Systems. IEEE, 2015:91-96.
[3]. Savage D, Zhang X, Yu X, et al. Anomaly detection in online social networks [J]. Social Networks, 2016, 39(1):62-70.
[4]. Akoglu L, Tong H, Koutra D. Graph based anomaly detection and description: a survey [J]. Data Mining and Knowledge Discovery, 2015, 29(3):626-688.
[5]. Kleinberg J M. Authoritative sources in a hyperlinked environment[C]// Acm-Siam Symposium on Discrete Algorithms. Society for Industrial and Applied Mathematics, 1998:668-677.
[6]. Meng Jiang, Peng Cui, Alex Beutel, Christos Faloutsos and Shiqiang Yang. “Catching Synchronized Behaviors in Large Networks: A Graph Mining Approach.” In ACM Transactions on Knowledge Discovery from Data (TKDD), 2015.
[7]. Liu F T, Kai M T, Zhou Z H. Isolation Forest [J]. 2008:413-422.
[8]. Breunig M M. LOF: identifying density-based local outliers[C]// ACM SIGMOD International Conference on Management of Data, May 16-18, 2000, Dallas, Texas, Usa. CiteSeer, 2000:93-104.
[9]. Akoglu L, Mcglohon M, Faloutsos C. OddBall: spotting anomalies in weighted graphs[C]// Advances in Knowledge Discovery and Data Mining, Pacific-Asia Conference, PAKDD 2010, Hyderabad, India, June 21-24, 2010. Proceedings. DBLP, 2010:410-421.
[10]. Sun J, Qu H, Chakrabarti D, et al. Neighborhood Formation and Anomaly Detection in Bipartite Graphs[C]// IEEE International Conference on Data Mining. IEEE Computer Society, 2005:418-425.
[11]. Ding Q, Katenka N, Barford P, et al. Intrusion as (anti)social communication: characterization and detection[C]// ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2012:886-894.
[12]. Schubert E, Zimek A, Kriegel HP (2014a) Generalized outlier detection with flexible kernel density estimates. In: Proceedings of the 14th SIAM International Conference on Data Mining (SDM), Philadelphia, pp 542–550. doi:10.1137/1.9781611973440.63
[13]. Reed, Bruce. "The height of a random binary search tree." Journal of the ACM (JACM) 50.3 (2003): 306-332.