Parallel Outlier Detection in Dial-back Fraud Calls

Bin Wu, Fandi Liao, Di Zhang
Beijing Key Lab of Intelligent Telecommunication Software and Multimedia,
Beijing University of Posts and Telecommunication, Beijing, China
wubin@bupt.edu.cn, liaofandi@126.com,zhangdi5004@126.com

Abstract—**With the intensified competition among telecommunications industry, we focused much on the quality of service. Illegal activities, especially dial-back fraud calls, may cause annoyance and inconvenience which will reduce user experience. The detection of dial-back fraud calls is an urgent issue that needs to be addressed. The rapid development of information technology which gives rise to the accumulated huge data will pose a greater challenge. However, traditional detecting methods to identify illegal activities cannot get acceptable accuracy. On the other hand, those methods become very inefficient or even unavailable when processing massive data. In this paper, we introduce a distributed outlier detection approach to locate illegal acts of the illegal users who have the characteristics as outliers. For a higher hit rate, we combine outlier detection with cluster coefficient. Besides, the method exploits parallel computation based on MapReduce in order to obtain vast time savings and improve the processing capability of the algorithm on large data. Extensive experimental results demonstrate the efficiently performances of proposed algorithm according to the evaluation criterions of speedup and scale up.**

Keywords-Illegal activities;Outlier detection;Cluster Coefficient; MapReduce;Parallelization;

I. INTRODUCTION

With the growing number of mobile users, a group of illegal activities via mobile communication emerges, one of which is dial-back fraud calls. Dial-back fraud calls is a phenomenon that the calls rang and hung up immediately, dialing back with advertising, fraud or deduct the high charges. It will disturb the normal life of users, and may cause the property loss, which will result in low user perceptions, even, customer defection. Fraud detection in telecommunication involves indentifying fraud as quickly as possible once it has occurred, which requires accurately and efficiently approaches to detect fraud. The regular practice of telecom operators is concluding from customer complaint or setting some certain thresholds to identify abnormal call-back. Analyzing the complaints from clients requires a lot of time and manpower. Furthermore, setting thresholds identification will not be so satisfactory on account of mass user data and may get low hit rate. Here, we extract two difficulties from dial-back fraud calls: handling the mass data and anomaly detecting precisely.

Parallel computing is an efficient way to process very large amount of data. MapReduce [1] [2] is a programming model, with which users can specify the algorithms in terms of map and reduce functions, and the underlying runtime systems automatically parallelize the computation across large-scale clusters of machines.

Outlier detection [3] is a data mining task whose goal is to isolate the observations which are considerably dissimilar from the remaining data. Outlier mining can be described as finding the top k objects that are considerably dissimilar, exceptional, or inconsistent with respect to the remaining data. It has wide applications in several domains such as fraud detection, customized marketing, medical analysis, and many others.

In this paper, we introduce an outlier approach to find outliers. We parallelize it to adapt to the very large amounts of data by MapReduce mechanism. Besides, in order to accurately identify the fraud calls outliers, we introduce the concept of clustering coefficient [4] which can reflects the tightness between records' neighbors. By adopting the outlier detection, users can select an important call to dial back and not dial back fraud calls, thereby avoiding time and economic loss caused by dial-back.

The remainder of the paper is organized as follows. In Section 2, we describe the related work and various existing approaches for outlier detection. We give a brief overview of LOF algorithm and cluster coefficient in Section 3. In Section 4, we display the procedure of identifying dial-back fraud calls user with parallelized methods. Next, in Section 5, we do the experiment and compare the performance and correctness of this parallelized algorithm with algorithm in Weka, what's more, validate the high accuracy rate of the entire fraud calls user detection process. Finally, we conclude this paper in Section 6.

II. RELATED WORKS: OUTLIER DETECTION

Recently parallel/distributed methods for outlier detection have attracted extensive academic research interests. Outlier detection can be mainly categorized into four approaches: the statistical approach, the distance-based approach, the density-based approach, and the deviation-based approach.

Considering the feature of object we detect, we prefer to introduce the density-based approach. The density-based approaches derived from a concept of local outlier. MM Breunig et al. [5] presented a specialized unit of measure, LOF (local outlier factor) for local outlier detection.

The LOF algorithm has many advantages over other outlier detection methods, for its results seem more meaningful and accurate. However, its computational complexity imposes a limitation.

To reduce the computation time of LOF, Seung Kim et al. [6] incorporates kd-tree indexing and an approximated k-nearest neighbors search algorithm in finding KNN(k nearest neighbor), which is an time-wasted procedure in LOF algorithm.

Agyemang et al. [7] developed a modified local outlier algorithm. To avoid computing the complex LOF, it put forward

Published by Atlantis Press, Paris, France.

a local sparsity coefficient that represents the outlierness of each data point.

Pokrajac et al. [8] proposed an incremental algorithm on LOF. It provides equivalent detection performance as the iterated static LOF algorithm (applied after insertion of each data record), which requires significantly less computational time.

Krishna et al. [9] improved the LOF, and it not only focuses on the density-based notion to discover local outliers but also reduces the number of passes to scan the complete database.

Based on the conception of neighbor-hood and local density, Yunxin Tao et al. [10] proposed a unified Density-Based Clustering and Outlier Detection algorithm (DBCOD in abbr.) for discovering clusters and detecting outliers in a multidimensional database. It solves clustering and outlier detection at the same time without losing the quality of clustering and outlier detection.

Although density-based outlier mining had been improved by aforementioned relevant literatures, the efficiency above increased can do few things when the amounts of data increase into mass. This paper combines the LOF algorithm with parallel computation, and introduces the clustering coefficient for high accuracy.

III. OVERVIEW

LOF is a sort of Outlier detection algorithm which is based on sample space's capacity. There is a parameter, k, representing the quantity of aggregation required to look up closed to each other.

In the Calculation of *k-distance(p)* which represents the most remote distance between *p* and its neighbor according to parameter k, processing *k-distance* query to each node p in a data set D is required. As usual, $N_k(p)$ is used to represent the *k-aggregation* close to the node *p*.

Computing the reachable distance between *p* and its *q*, in *k nearest neighbors*, by traversing the result of formula 1.

$$reach-dis_k(p,q)=\max\{dis(p,q),k-distance(q)\} \quad (1)$$

Among the formula, *dis(p,q)* represents the Euclidean distance from *p* to *q*

Computing the local reachability density (lrd) of each node *p*:

$$lrd_k(p)=\Big/\left(\frac{\sum_{q\in N_k(p)} reach-dis_k(p,q)}{|N_k(p)|}\right) \quad (2)$$

Computing the LOF of each node *p*:

$$LOF_k(p)=\frac{\sum_{q\in N_k(p)}\frac{lrd_k(q)}{lrd_k(p)}}{|N_k(p)|} \quad (3)$$

If a data record has a larger LOF value, a higher likelihood for it to be an outlier.

The degree to each node aggregate to each other is reflected by the clustering coefficient. In other words, it can indicate the degree nodes closing to each other in the graph.

Cluster coefficient is defined as:

$$CC_v=\frac{n}{C_k^2}=\frac{2n}{k(k-1)} \quad (4)$$

The n represents the number of edges between all k nearest neighbors of the node v.

Because of dial-back fraud calls' outlier character and a typical character that most of the users who are called by dial-back fraud calls do not know each other.

IV. IMPLEMENTATION

In this section, we present the main design for dial-back fraud calls detection based on MapReduce. The main task is the need of transforming the algorithms to adapt to the MapReduce programming mechanism. In MapReduce programming, the input dataset is stored on HDFS as a sequence file of <key, value> pairs, each of which represents a record in the dataset. The dataset is split and globally broadcast to all mappers. The reducers and combiners receive the <key, value> pairs from mappers' output. Theoretically, the pairs with the same key will be collected by the same reducer or combiner.

Considering the mechanism of MapRedcue, the procedure of LOF mainly split into 3 stages: (1)find k-nearest neighbors(KNN) of data records in dataset;(2)compute the lrd of each records;(3)compute the LOF of all the records.

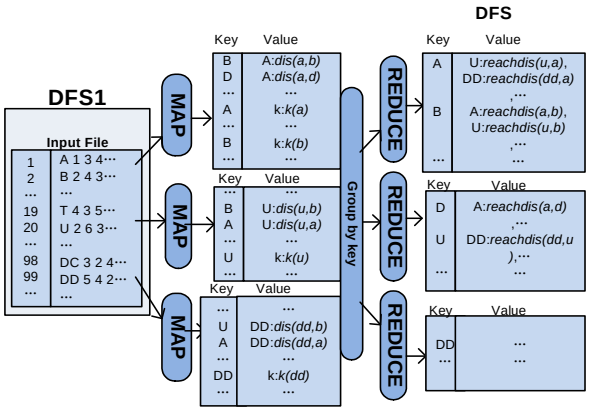

Figure 1. MapReduce data flow in computing reachability distance.

In the first job, as the Figure.1 shows, the map function responsible for computing the distance between each sample pair, and selects the top KNN of each sample, assumed p, via stack.

The reduce function performs the procedure of getting *reach-dis*$_k$*(p,q)* using Eq.1.

In order to reduce the time complexity in computing *lrd*, there are two kinds of <key, value> output. The One key is *p* , with value of *k-distance(p)*.The other kind is q that in p's KNN set In this way, what the reducers can receive are *k-distance(p)* and *dis(p,q)* where *q* contained in *p*'s KNN set. Consequently, the time complexity of computing lrd can fall into $O(n)$.

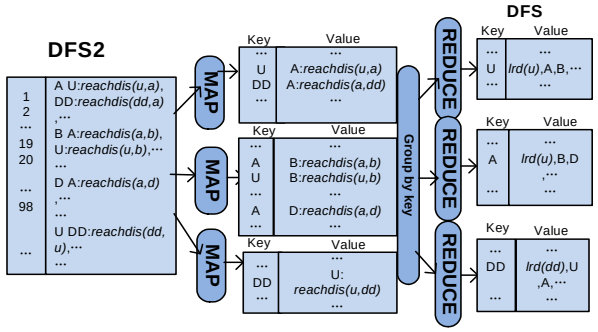

Figure 2. MapReduce data flow in computing *lrd*

Lrd is computed in the second job. Mappers output p'id and each *reach-dis$_k$(p,q)*where q is contained in $N_k(p)$. Reducers compute lrd using Eq.2 for every p in dataset.

LOF is computed in the last job. The map function forces all the record sending its lrd to all the nodes, which treated them as KNN. The reduce function computes LOF for the whole records in dataset using Eq.3. The data flow procedure of the last job is shown in Figure 3.

If a data record's density is similar to its surrounding neighbors', the LOF is in close in close proximity to 1.If a data record is located far from its neighbors or data cluster, On the contrary, it has an LOF value greater than 1. We select the nodes most likely to be outliers, which owned a larger LOF.

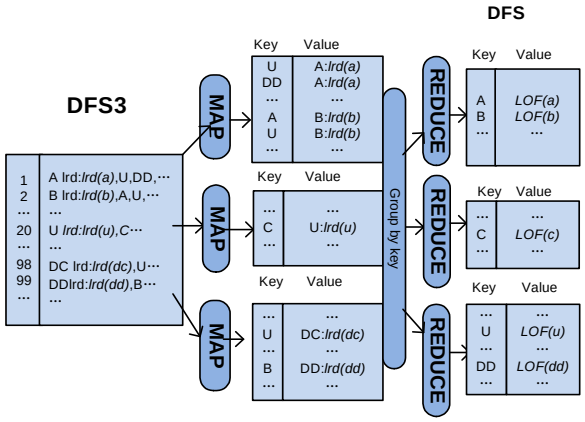

Figure 3. MapReduce data flow in computing LOF

We take the extracted files as one of the input of cluster coefficient algorithm. The other input should be the call records.

According to the definition of cluster coefficient, we split the algorithm into 3 stages, too. (1) Establish the adjacency matrix for each node and work out the degree as the result of calculation. (2) Find the number of triangles to the target point and any two neighbors. It should be emphasized that the neighbors here means that there is at least one call record between neighbors. (3) Compute the cluster coefficient for each node.

We construct 3 jobs to get the cluster coefficient. Mappers receive the call records and extract the from-call and to-call. The output pairs of mappers are <from-call, to-call> and <to-call, from-call>. Then the first reduce function construct the adjacent matrix. In the second job, besides of computing the degree for each node, the mappers break up the adjacent matrix and send out pairs such as <b c,a>(assuming b,c are both the neighbors of a) or <a c, T>(The 'T' is a mark means that a,c are neighbors). When reducers receive the output, and compute edges between neighbors for each node. What the third job do is just counts the cluster coefficient of each extracted node.

We can conclude from history calls record that the cluster coefficient of dial-back fraud calls users may be less. We select those users with lesser cluster coefficient value as the target outliers.

V. EXPERIMENTAL RESULTS

In this section, we evaluate the performance of our proposed algorithm with respect to accurate rate as well as efficiency by means of speedup and scaleup. [11].

We handled the experiments on a 10-node cluster and each node is configured as Table 1.

TABLE I. THE EXPERIMENT CONFIGURATION

HARD WARE	CPU	Intel(R) Xeon(R) CPU E5530 @ 2.40GHz 2*4 core
	RAM	16GB
	HardDisk	SATA 1T*4
	Connectivity	100 Ethernet LAN
SOFT WARE	Operating System	Linux redhat-6.2
	JDK	jdk1.6.0_20
	Hadoop	hadoop-0.20.2

We use the classified data set from UCI and increase their sizes as required for the purpose of the experiment. Besides, in order to verify the algorithm's correctness, we will take into consideration actual call record data set during three month, in which the users have been distinguished between normal and errant classes.

The original UCI data set, named as KEGG Metabolic Reaction Network Data Set, consists of 65554 terms with 29 attributes for each term, reaching a size of 11.2MB. The attributes are either integral or real numbers.

Firstly, we operate the LOF algorithm without improvement. With the increase of dataset, the algorithm's efficiency declined significantly. When the size of the dataset reaches 10000 terms, it takes 22'30'' to accomplish the work. And the memory will be overwhelmed with a 20000 dataset. In conclusion, we regard that the original LOF algorithm cannot handle with large datasets efficiently.

Then, we increase the number of node linearly and the dataset size by 10, 100, 500 times for each scale of clusters. Theoretically, the running time is expected to decrease as the

linearly increase of cluster scale. Figure4 illustrates that our parallelized algorithm shows a nice speedup.

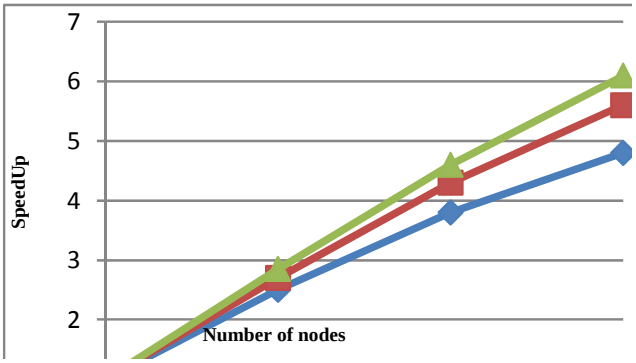

Figure 4. The SpeedUp

In order to evaluate the scaleup of the proposed algorithm, we increased the size of the dataset and the cluster scale by the same factor simultaneously. And the result, showed in Figure 5, is reasonable as predicted that the running are maintained relatively constant.

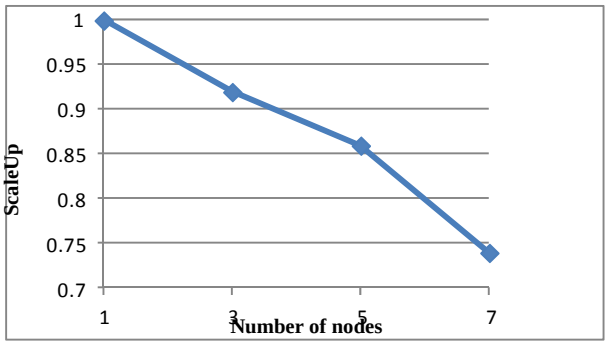

Figure 5. The ScaleUp

At last, we test the accuracy rate of the algorithm on actual datasets. It contains 10.4 million user records and its size is up to 1.6 GB. It takes nearly 15 hours to cover the entire process, including computing LOF, ranking and calculating Cluster Coefficient. The proposed algorithm figured out 84% of all the errant users in dataset by comparing with classification of the errant user and the normal ones, which is sensible and acceptable with respect to the accuracy rate. In sum, our algorithm does make sense in distinguishing most of the errant users.

VI. CONCLUSIONS

Fraud calls may cause annoyance and inconvenience which will reduce user experience. It is a direct result of user perception declining which would lead to customer and economic loss. In this paper, we introduce an outlier detection approach into illegal call-back user detection in consideration of this problem. For a higher hit rate, we combine outlier detection with cluster coefficient. Besides, the method exploits parallel computation based on MapReduce in order to obtain vast time savings and improve the processing capability of the algorithm on large data. The experimental results of extensive experiments on different datasets to evaluate the speedup and scaleup show how the proposed algorithm can process large datasets on commodity hardware cluster effectively. By adopting the outlier detection, users can select an important call to dial back and not dial back fraud calls, thereby avoiding time and economic loss caused by dial-back for fraud calls.

ACKNOWLEDGMENT

This work is supported by the National Key Basic Research Program(973 Program) of China(No.2013CB329603) and the National Natural Science Foundation of China(No.71231002).

REFERENCES

[1] Lammel.R.,"Google's MapReduce Programming Model-revisited",Science of Computer Programming,2008,70(1):1-30.

[2] Hadoop: Open source implementation of MapReduce.

[3] Han Jiawei,Micheline. K ,"Data Mining:concepts and techniques",2nd edition,San Froncisco,Morgan Kaufmann Publishers,2006

[4] M.E.J.Newman,"Scientific collaboration networks. I.Network construction and fundamental results" Phys.Rev.E,2001,64:016131.

[5] MM Breunig, HP Kriegel, RT Ng, J Sander," LOF: identifying density-based local outliers", In Proceedings of the ACM SIGMOD Conference, Dallas, TX, May 2000.

[6] S Kim, NW Cho, B Kang, SH Kang, "Fast outlier detection for very large log data", Expert Systems with Applications, 38 (2011) 9587–9596,2011.

[7] M Agyemang,CI Ezeife,"Lsc-mine: Algorithm for mining local outliers", Proceedings of the 15th Information Resources Management Association International Conference, pp. 5–8 ,2004.

[8] D. Pokrajac , A. Lazarevic and L. Latecki "Incremental local outlier detection for data streams", Proc. IEEE Symp. Comput. Intell. Data Mining, pp.504 2007.

[9] KG Sharma, A Ram, Y Singh," Efficient Density Based Outlier Handling Technique in Data Mining", Communications in Computer and Information Science, Volume 131, Part 4, 542-550, 2011

[10] Yunxin Tao, Dechang Pi, "Unifying Density-Based Clustering and Outlier Detection", The Second International Workshop on WKDD 2009, Moscow, pp. 644-647,2009.

[11] X. Xu, J. Jager, H.P. Kriegel, "A fast parallel clustering algorithm for large spatial databases", Data Mining Knowled. Disc., vol. 3, no. 3, pp.263 -290,1999.