International Journal of Security and Its Applications
Vol.8, No.2 (2014), pp.157-166
http://dx.doi.org/10.14257/ijsia.2014.8.2.16

Suspicious Transaction Detection for Anti-Money Laundering

Xingrong Luo

Vocational and technical college in Enshi
Enshi, Hubei, China
es_lxr@126.com

Abstract

Money laundering activities in financial markets have been increasingly serious recent years. Although efforts on anti-money activities started at an early stage, the solutions seem to be restricted to a strategic level. Besides, even some research pioneers at employing data mining techniques to anti-money laundering, the situation in China is still difficult. To this end, in this paper, after presenting the systematic view of the data mining framework of anti-money laundering, we propose a classification based algorithm to effectively detect suspicious transactions. Specifically, we consider the financial transactions as a data stream, and try to construct a classifier based on a set of mined frequent rules. Our experiments on a simulated transaction dataset based on real world banking activities prove the efficiency of our proposed method.

Keywords: *Classification, Data mining, Anti-money laundering, suspicious transactions*

1. Introduction

As the increasing development of internet and database technologies, the data can be obtained has been more and more big [1, 2]. In order to understand the big data, data mining methodology is applied throughout various fields, such as marketing, customer relationship management and financial management. Specifically, as an emergent technology in financial area, efforts on data mining have been made on banker/customer relationship management, credit risk alert and market analysis on finance.

One of the toughest things is financial fraud, and worse still, money laundering. Money laundering is the convention of criminal incomes into assets that cannot be tracked to the underlying crime, where the process of concealing sources of money is referred to 'laundering' [3]. As the money laundering activities go increasingly wild, the financial growth and national security have been critically affected. Current strategies of anti-money laundering expect laws and regulations to be established to prevent and suppress money laundering activities. For example, possible measures of banks include validating customer identification validation before banking business, checking suspicious foreign exchange cash transactions, tracking large cash flows, and blacklisting accounts of suspected money laundering, *etc*. However, existing anti-money laundering methods reply on human intervention, and applying modern data mining techniques still remains at an initiating phase.

Detecting suspicious financial transactions is an essential precondition and key aspect of anti-money laundering [4]. Existing methods are based on the amount of transactions, and the identification implementation process is extremely restricted to the mechanism of unusual banking activities reporting [5]. Therefore, there are several limitations of traditional anti-money laundering efforts, such as narrow coverage of identification, long cycle of clue discovery, and extensive delay [6].

In order to solve above challenges, we propose a data mining [7] system for anti-money laundering, and focus on the suspicious transaction detection in this paper.

ISSN: 1738-9976 IJSIA

The overall workflow of the system is as follows, as shown in Figure 1. After storing the transactional data into data warehouse, some preprocessing work is performed for data cleaning and transformation. Then, related data is selected for data mining engine, where data mining algorithm is applied. After that, discovered knowledge is then abstracted into a knowledge base, which will be further used to visualization, recommender systems and other business application.

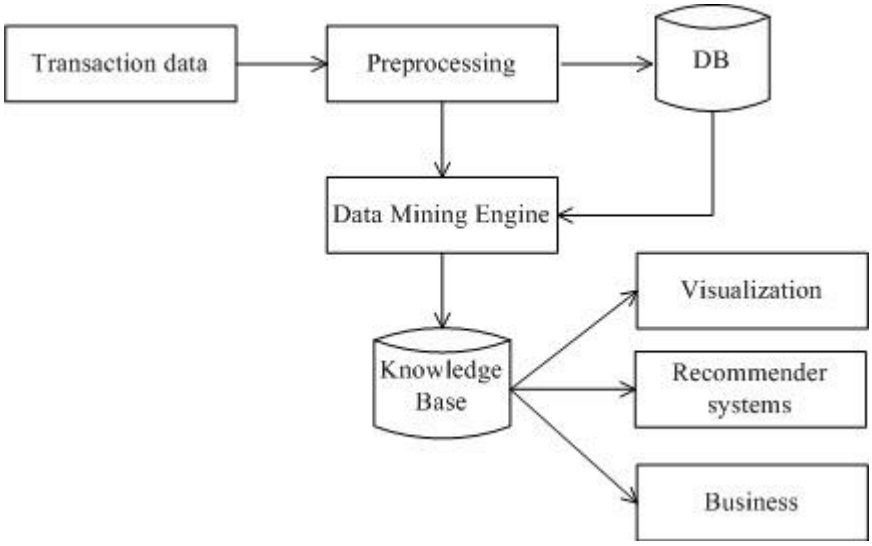

Figure 1. Framework of Data Mining System for Anti-money Laundering

Then, we focus on the dynamic detection mechanism of suspicious transactions. Typically, the abnormal relationships between transactional accounts are deemed as suspicious [5]. Within a stream of financial transaction data, discovering emerging patterns during various financial transactions, exhibited as interesting abnormity from chronic or regular behaviors, known as suspicious transaction patterns [5], is an efficient method. For example, Liu *et al.*, [8] proposed a method based on time series to detect suspicious transactions with the technique of scan statistics. Also, Keyan *et al.*, [9] tried to improve the accuracy of detecting suspicious transactions by using cross validation and grid search optimized support vector network model.

However, different from existing works, we regard the financial transactions as a stream data, and employ a dynamic mining method to detect suspicious patterns on stream transactions. Specifically, we propose a classification algorithm based on multiple class association rules on data streams, by constructing a FP-tree to improve the time and space efficiency, and then reducing frequent rules by using Hoeffding bound [10] over dynamic data streams.

To sum up, in this paper, our main contributions are as follows. First, we study on the problem of anti-money laundering and summarize the overall data mining framework as a solution, which can solve the challenges of traditional manual strategies of anti-money laundering policies. Second, we propose a classification based algorithm to dynamic detect suspicious transactions over the financial transactional data streams. Last, our experiments on a simulated banking transaction dataset improve the efficiency of our method.

The remain of this paper is organized as follows. Section 2 provides some related work. The overall framework is discussed in Section 3. Then our algorithm for detecting suspicious transactions is proposed in Section 4. Empirical experiments are conducted in Section 5. Finally, the paper is concluded in Section 6.

2. Related Work

The efforts on anti-money research started at an early stage. Senator *et al.*, [11] first formally proposed an artificial intelligence system named FinCEN (Financial Crimes Enforcement Network) to identify potential money laundering from reports of large cash transactions. The evaluation of suspicious transaction in FinCEN is implemented by a Bayes model. Petrus C van Duyne *et al.*, [12] pointed out that there is existing problems in the monitoring system of suspicious transactions and anti-money laundering strategies. Kingdon *et al.*, [13] developed a system to automatically identify unusual behavior in customers transactions.

However, none of above efforts can be efficiently applied to Chinese financial markets, due to the specific characteristics of Chinese financial transactions [14]. Therefore, in this paper, we propose an algorithm designed for Chinese banking transactions.

3. Overall Framework

In this section, we present our overall framework of anti-money laundering using data mining techniques.

Typically, as described in Figure 1, suppose we have a streaming financial transactions dataset. First, we perform preprocessing such as cleaning, reduction and sampling to select relevant sub-dataset for our suspicious transactions detection problem. Then, we store that data into a data warehouse. Later, feed data into a data mining engine module for suspicious transactions detection. Note that for incoming streaming data, dynamic data can be directly delivered to the data mining engine for real-time execution. After that, useful information is extracted and stored into a knowledge base for further applications, which might be used by decision makers for data visualization, recommender systems and other business applications with respect to anti-money laundering field.

Specifically, in this paper, we focus on the data mining engine module, and study on the problem of detecting suspicious transaction patterns. First of all, we will discuss the data preprocessing, which serves as the premise stage of mining.

3.1. Data Preprocessing

Typically, anti-money laundering activities involve multiple accounts during financial transactions. The data preprocessing include the following aspects.

3.1.1. Attributes Filtering: Only a subset of attributes is useful for anti-money laundering activities. For example, the name of accounts is usually useless, and can be removed to reduce the scale of dataset. Suppose the streaming financial transaction dataset can be represented as $DS = \{t_1, t_2, ..., t_n\}$. Each $t_i, i = 1,2,..., n$ is associated with a set of attributes A, and a target attribute y. Now we need to check whether attribute $x \in A$ is relevant to y.

An intuitive way is, sort all the values of A, and then get a corresponding distribution of y. Suppose x_i denotes the attribute x of the i-th record, and $0 < i < n$. Define $\sum_{2}^{n} \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}$ to describe the discreteness of the distribution [21]. If it exceeds some threshold, no relevance is exhibited as considered. Then, attribute x can be removed.

3.1.2. Feature Extraction based on Domain Knowledge: According to domain knowledge, some features such as capital flows and the frequency of access account are closely related to money laundering activities, and yet are not recorded in the original transaction dataset. However, we can calculate them using statistical method. The relations can be formulated as a linear regression model.

Notate the registered fund as f, and the summary amount of access account with a certain period as s. Given account i, f_i, s_i denote the registered fund and summary amount of i respectively. Suppose there exists a linear relationship between f and s, i.e., $f = \alpha + \beta s + \varepsilon$, where α, β are constants, and ε follows a normal distribution $N(0, \sigma^2)$. Parameters α, β can be inferred using the least square method [15] as follows:

$$\hat{\beta} = \frac{n\sum_{i=1}^{n} f_i s_i - \sum_{i=1}^{n} s_i \sum_{i=1}^{n} f_i}{n\sum_{i=1}^{n} s_i^2 - (\sum_{i=1}^{n} s_i)^2}, \hat{\alpha} = \bar{f} - \hat{\beta}\bar{s}. \quad (1)$$

3.1.3. Correlation matrix between trade accounts: Calculate the correlation coefficients between trade accounts among different industries, and store as a correlation matrix as features. For example, Table 1 gives the correlation matrix after preprocessing.

Table 1. Correlation Matrix between Trade Accounts

Import/export Trade	Steel	Mechanics	Food	Chemistry
Steel	0.5	0.9	0.1	0.1
Mechanics	0.7	0.5	0.6	0.7
Food	0.2	0.2	0.7	0.1
Chemistry	0.4	0.3	0.3	0.5

In next section, we will focus on the data mining engine module, and especially on the suspicious transaction detection function.

4. Proposed Algorithm

In this section, we propose a classification based algorithm to dynamic detect suspicious patterns for anti-money laundering. The objective is to find out the frequent associated accounts, which are deemed as suspicious.

4.1. Problem Formulation

Given a streaming financial transaction dataset $DS = \{t_1, t_2, ..., t_n\}$, and each $t_i, i = 1,2,..., n$ is associated with a set of attributes A. Define pattern p as a (A_i, v) pair, where A_i is the i-th attribute and v is the corresponding value. Suppose we have a set of patterns $P = \{p_1, p_2, ..., p_l\}$ and a tuple t. If for each $p \in P$, t satisfies $p = (A_i, v)$, we say that t matches P. Notate $P.count$ as the number of objects matched in P, and $P.sup = p.count / n$ as the support of P in DS.

Let c be the notation of class label. Define a class association rule $R: P \rightarrow c$, where $R.count$ is the number of objects in DS matching pattern P. Then, $R.sup = R.count / n$ is the support of rule R in DS, and $R.conf = R.count / P.count$ is the confidence of rule R in DS.

If $R.sup \geq min\_sup$, where min_sup is the minimum support, the corresponding pattern P is called frequent pattern or itemset, and R is called frequent association rule. Furthermore, if $R.conf \geq min\_conf$, where min_conf is the minimum confidence, R is called precise rule. For the convenience of description, we term rule R with pattern P of a length of k as $k-rule$. Besides, if R is frequent, R is termed as $k-freqrule$.

Now the objective is to find all the frequent and precise rules, and construct a classifier from above set of rules.

4.2. Algorithm Description

Generally, proposed algorithm in this paper is composed of two stages. First, discover all the frequent and precise rules from the training data. Second, based on the rules found in the former stage, model a classifier to learn the class labels and therefore distinguish the suspicious transaction patterns from the normal ones.

4.2.1. Discovering Frequent and Precise Rules: Before constructing a classifier upon association rules [22], first of all, we need to discover a proper set of rules. Our method is based on FP-tree algorithm [16].

Theorem 1. Given a data stream DS and the minimum support min_sup, suppose p as a frequent pattern. For any class label c, if rule $R: p \rightarrow c$ is not a frequent rule, then p should not be included in any frequent rules.

Base on theorem 1, we can prune those non-frequent items in the generation of frequent patterns when building FP-tree.

Since the streaming data is typically extremely large and cannot be stored in local storage. Therefore, we split the data stream into segments with certain length, called time window. Then, the rule mining algorithm is applied to the dataset with specific time window. After that, processed dataset can be abandoned or transferred to other places to release memory usage.

However, streaming data can only be scanned once, which means we should preserve not only frequent rules but all rules. But storing all rules encountered leads to a huge overhead. Therefore, we introduce Hoeffding bound [17] to estimate the support of rules. Given a variable r with value boundary R, suppose we have n independent observations, calculate the expectation of r as μ. Hoeffding bound indicates that the possibility of expect value of r being at least $\mu - \varepsilon$ is $1 - \delta$, where:

$$\varepsilon = \sqrt{\frac{R^2 \ln(1/\delta)}{2n}}. \quad (2)$$

We can observe that Hoeffding bound is independent on the distribution of samples. Suppose we need to estimate the support of some rule, and the value boundary $R = 1$. Given error factor ε and the probability of error δ, the length of data stream with support as $\sigma - \varepsilon$ and probability as $1 - \delta$ is calculated as follows:

$$n_l = \frac{R^2}{2\varepsilon^2}\ln(1/\delta) = \frac{1}{2\varepsilon^2}\ln(1/\delta) \tag{3}$$

Within time window of length n_l, only rules whose support is larger than $\sigma - \varepsilon$ should be preserved, and frequent rules with support larger than σ can be obtained at the probability of $1-\delta$. Accordingly, we have the following definitions.

Definition 1. Given a rule $R: p \rightarrow c$ within time window T, the support is $R.sup$, the minimum support is σ, and the maximum error of support is ε. If $R.sup \geq \sigma + \varepsilon$, then R is called *frequent rule*; if $\sigma - \varepsilon \leq R.sup < \sigma + \varepsilon$, then R is *potential frequent rule*; if $R.sup < \sigma - \varepsilon$, then R is *non-frequent rule.*

From Equation (3), we can observe that the length of time window n_l is negative related to ε^2. When ε is small, n_l will be very large. For example, if the value of ε is 0.001, and δ is 0.02, n_l will be 3912 thousands. That means lots of memory is needed for consuming such a big dataset, and the efficiency of algorithm will be affected. Therefore, ε should be initialized as a large value, and then decreases as the data stream comes in. In our settings, we initialize $\varepsilon = 0.95\sigma$. Later, after the processing within each time window, ε is updated as Equation (2).

The workflow of generating frequent rules is described in Algorithm 1. Note that the process of creating FP-tree is similar to [18], except that we further add a pruning strategy defined by theorem 1. Then we merge the rules mined by FP-tree P and our frequent rules set FR as lines 7-16.

Algorithm 1 generating frequent rules

Input: data stream DS, minimum support σ, error factor ϵ;
Output: frequent rules FR.

```
1: initialize σ = 0.01
2: initialize n = 0, FR = ∅, P = ∅
3: compute n_l = ln(1/σ)/2ε^2
4: for each time window in DS do
5:    create a FP-tree based on theorem 1
6:    derive rules from FP-tree and save to P
7:    for each rule R in FR but not in P do
8:       R.count++
9:    end for
10:   for each rule R in P do
11:      if R in FR then
12:         R.count++
13:      else
14:         insert R into FR
15:      end if
16:   end for
17:   n = n + n_l
18: end for
19: return FR
```

Figure 2. Algorithm of Generating Frequent Rules

All frequent rules can be mined by Algorithm 1. According to Hoeffding bound [17], any rule R has a support of $R.sup$, and $R.sup - \varepsilon$ with a probability of $1 - \delta$. Since n decreases as δ grows, the probability tends to be 1 if the support is $R.sup - \varepsilon$. That means, the support value of all rules satisfies $R.sup \geq \sigma - \varepsilon$. Interestingly, the rules generated from Algorithm 1 have a minimum support of $\sigma - \varepsilon$. Therefore, all rules are included in the results of Algorithm 1.

4.2.2. Constructing the classifier: Now we have mined all frequent rules, and the next step is to construct a classifier for further precise classification. Because of the dynamic variation of steaming data, we will not prune all unsatisfactory rules. The intuition is that even if a rule is not frequent currently, it might be frequent later when more data comes in.

Given a new transaction, the objective is to assign a class label to determine if the transaction is suspicious or not. If all the rules matching the new transaction have the same class label, then the transaction is assigned to that class directly. Otherwise, we get a group of class labels according to different rules applied to the new transaction.

Now we consider how to combine the effects of a group of class labels. For each rule $R: p \rightarrow c$, the rule with highest χ^2 is selected, where the upper bound of χ^2 is computed as follows [19]:

$$\max \chi^2 = (\min\{ sup(p), sup(c) \} - \frac{sup(p)sup(c)}{n})^2 ne , \quad (4)$$

where $sup(p), sup(c)$ denotes the number of objects with pattern p and label c respectively.

5. Experiments

In order to evaluate the efficiency of our algorithm, in this section, we conduct some experiments.

5.1. Settings and Data

The environment of our experiments is as follows. The PC has a Intel E5200 CPU with 1G memory, 250G hard disk memory, and the operation system is Windows 7.

We try to evaluate if our method can be used to detect suspicious transactions within a stream of financial transactions. Therefore, we simulate a large data stream of financial stream based a real world banking records, similar to existing study [20]. We have generated 100 millions records of transactions, and the fields of dataset are shown as Table 2. Note that the two sides of a transaction denote the in and out of financial stream, which can be represented as a directed edge between two nodes. Therefore, we notate the transactions between two specific accounts as (n_1, n_2), where n_1 is the number of transactions from Account_0 to Account_1, and n_2 is the number of transactions from Account_1 to Account_0.

Table 2. Fields of Banking Transaction Dataset

Field name	Description
ID	The identifier of a single transaction
Date	The date of transaction committed

Branch_ID_0	Branch of bank of the first account
Lower_branch_ID_0	Lower level of branch of the first account
Account_0	The first account of transaction
Branch_ID_1	Branch of bank of the second account
Lower_branch_ID_1	Lower level of branch of the second account
Account_1	The second account of transaction
Amount	The amount of transaction
Category	The sort of transaction, i.e., withdraw, deposit, and transfer

The parameters used in our experiment are set as follows. We set min_sup as 1%, min_conf as 50%, database coverage threshold as 4, error factor for confidence as 20%, and error factor for support ε as 0.01.

5.2. Experimental Results

Out of all 100 millions of transactions, we detect 317 accounts labeled as suspicious based on the rules mined as described in our algorithm. Furthermore, each suspicious account is associated with others in the form of suspicious transactions which are tied to two different accounts. Table 3 gives a partial results of our experiment.

Table 3. Partial Results of Suspicious Accounts and Transactions

Accounts	Associated accounts	Suspicious transactions
116054	(129161,564604,847321,...)	(0/8,0/12,11/0,...)
847321	(116054,165376,400528,..)	(0/11,0/5,0/7,...)
129161	(116054,165376,...)	(8/0,5/0,...)
564604	(116054,714340,...)	(12/0,0/2,...)
400528	(847321,...)	(7/0,...)
165376	(129161,847321,...)	(0/5,5/0,...)
...	...	...

For example, as shown in Table 3, account 116054 is suspicious and associated with 129161, 564604 and 847321. Besides, account 116054 is probably an export account and 129161 is probably an import account, given the fact that transactions involved 116054 are outflowing from 116054, while the transactions involved 129161 mostly have an incoming cash flow. From above observations, we can see that our results are consistent with the real world facts. Therefore, it is reasonable to claim that determining class labels by the rules generated from our algorithm to assert the suspicion of transaction is feasible.

Next, we evaluate the efficiency of our algorithm as well. Figure 3 gives the accuracy of our experiment. Note that the labels of transactions are done manually by domain knowledge. From the figure, we can see that as the number of transactions grows, our algorithm performs better. That means our method is scalable to large dataset.

6. Conclusion

In this paper, we propose a data mining system for detecting anti-money laundering actives, and focus on the discovering of suspicious transactions in financial transactions stream. Specifically, the proposed algorithm employs a classification method based on a set of frequent rules. Our experiments on a simulated transaction dataset based on real work banking activities show both feasibility and efficiency of our method.

Future work might include: (1) further improvement of the algorithm to support various data sources; and (2) adding more functions into our data mining system for anti-money laundering besides suspicious transaction detection.

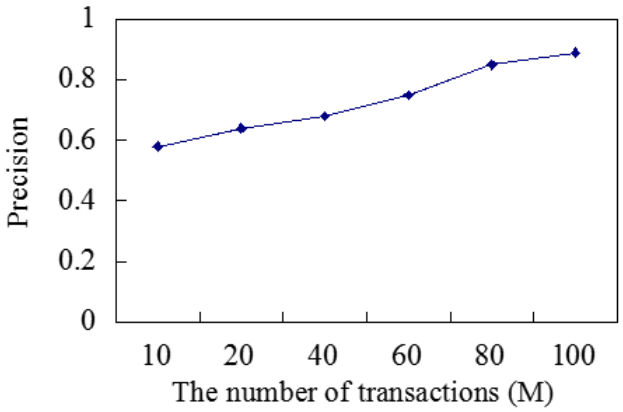

Figure 3. Precision of Proposed Algorithm on Various Size of Dataset

Acknowledgements

The authors would like to recognize the others who helped them and thank all our reviewers.

References

[1] S. LaValle, "Big data, analytics and the path from insights to value", MIT Sloan Management Review, vol. 52, no. 2, **(2011)**, pp. 21-31.
[2] P. Zikopoulos and C. Eaton, "Understanding big data: Analytics for enterprise class hadoop and streaming data", McGraw-Hill Osborne Media, **(2011)**.
[3] P. Reuter and E. M. Truman, "Chasing dirty money: The fight against money laundering", Peterson Institute, **(2004)**.
[4] R. Barone and D. Masciandaro, "Worldwide anti-money laundering regulation: estimating the costs and benefits", Global Business and Economics Review, vol. 10, no. 3, **(2008)**, pp. 243-264.
[5] R. Menon and S. Kuman, "Understanding the role of technology in anti-money laundering compliance", Infosys Technology Ltd, **(2005)**.
[6] B. Zagaris, "Problems applying traditional anti-money laundering procedures to non-financial transactions, parallel banking systems and Islamic financial systems", Journal of money laundering control, vol. 10, no. 2, **(2007)**, pp. 157-169.
[7] J. Han, M. Kamber and J. Pei, "Data mining: concepts and techniques", Morgan kaufmann, **(2006)**.
[8] X. Liu and P. Zhang, "A scan statistics based Suspicious transactions detection model for Anti-Money Laundering (AML) in financial institutions", Multimedia Communications (Mediacom), 2010 International Conference on. IEEE, **(2010)**.
[9] L. Keyan and Y. Tingting, "An Improved Support-Vector Network Model for Anti-Money Laundering. Management of e-Commerce and e-Government (ICMeCG)", 2011 Fifth International Conference on. IEEE, **(2011)**.
[10] O. Bousquet, S. Boucheron and G. Lugosi, "Introduction to statistical learning theory. Advanced Lectures on Machine Learning", Springer Berlin Heidelberg, **(2004)**, pp. 169-207.
[11] T. E. Senator, "The FinCEN Artificial Intelligence System: Identifying Potential Money Laundering from Reports of Large Cash Transactions", IAAI, **(1995)**.
[12] P. C. van Duyne and H. de Miranda, "The emperor's cloths of disclosure: Hot money and suspect

disclosures", Crime, Law and Social Change, vol. 31, no. 3, **(1999)**, pp. 245-271.
[13] J. Kingdon, "AI fights money laundering", Intelligent Systems, IEEE, vol. 19, no. 3, **(2004)**, pp. 87-89.
[14] J. Z. Xiao, H. Yang and C. W. Chow, "The determinants and characteristics of voluntary internet-based disclosures by listed Chinese companies", Journal of accounting and public policy, vol. 23, no. 3 **(2004)**, pp. 191-225.
[15] N. A. Weiss and C. A. Weiss, "Introductory statistics", Pearson Education, **(2012)**.
[16] J. Han, J. Pei and Y. Yin, "Mining frequent patterns without candidate generation", ACM SIGMOD Record, ACM, vol. 29, no. 2, **(2000)**.
[17] M. Medhat Gaber, A. Zaslavsky and S. Krishnaswamy, "Mining data streams: a review", ACM Sigmod Record, vol. 34, no. 2, **(2005)**, pp. 18-26.
[18] J. Han, "Mining frequent patterns without candidate generation: A frequent-pattern tree approach", Data mining and knowledge discovery, vol. 8, no. 1, **(2004)**, pp. 53-87.
[19] W. Li, J. Han and J. Pei, "CMAR: Accurate and efficient classification based on multiple class-association rules", Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on. IEEE, **(2001)**.
[20] J. Tang and J. Yin, "Developing an intelligent data discriminating system of anti-money laundering based on SVM"', Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference, IEEE, vol. 6, **(2005)**.
[21] G. H. John, R. Kohavi and K. Pfleger, "Irrelevant Features and the Subset Selection Problem", ICML, vol. 94, **(1994)**.
[22] R. Agrawal, T. Imieliński and A. Swami, "Mining association rules between sets of items in large databases", ACM SIGMOD Record, ACM, vol. 22, no. 2, **(1993)**.

Author

Xingrong Luo. Male, he was born in Enshi city, Hubei Province. Now, he is servicing in Vocational and technical college in Enshi as an associate Professor, and his research interests are computer software development and database technologies.

