International Journal of Advanced Research in

Computer Science Engineering and Information Technology

Volume: 3 Issue: 1 26-Jun-2014,ISSN_NO: 2321-3337

Improving identity crime detection using Scheduling for Fast Response Multi-pattern Matching over Streaming Events

Dr. G. Gunasekaran[1] , Mareeswari.V [2]

Principal, Meenakshi Engineering College, Chennai 600078, India. Research scholar, Department of Computer Science and Engineering, St. Peter's Institute of Higher

Education & Research, St. Peter's University, Avadi, Chennai 600054, India.

ABSTRACT *— Due to a rapid advancement in the electronic commerce technology, the use of credit cards has dramatically increased. As credit card becomes the most popular mode of payment for both online as well as regular purchase, cases of fraud associated with it are also rising. Credit card fraud is the specific crime in banking system. The credit card crime has been growing rapidly for the last few years. The process of making profit through credit card in the economy has been decreased about 8.2 crore annually in India. To avoid and predict the fraudulent activities on credit card application, in this paper a method of detecting the fraud over credit card on behalf of the cibil score. As datamining provides various ways to retrieve an appropriate data from the storage, here in proposed system an efficient way of matching the data provided by the applicants of credit card along with the cibil list to predict the fraudsters. The existing process of fraud detection has the drawbacks of effectiveness and scalability for multiple variants of data, the Scheduling for Fast Response multi-pattern matching algorithm used to match the large amount of attributes, In order to predict the fraudulent applicants with an appropriate time constraints. Together with the communal detection (CD) and spike detection (SD) algorithm that removes the redundant attributes and generates the credit score for cibil list or black list. The cibil score varies about 300 to 900, it has been recorded in the credit history and by considering its range the credits are provided to the lender or customer, and they are added to the white list.*

Index Terms—Data mining-based fraud detection, security, data stream mining, anomaly detection, Event-based (EBS) and Run-based (RBS) Scheduling.

1. INTRODUCTION

Generally, Datamining (sometimes called data or knowledge discovery) is the process of analyzing data from different perspectives and summarizing it into useful information. This information can be used to increase the revenue, cut costs and both. Datamining software is one

of the analytical tools that allow data from various categories to make a feature. Datamining in-turns is the process of finding patterns or correlations among the enormous data [10].

In recent years, the Datamining concerns with the fraud detection process since the problem here approached is the credit card fraud [13]. This system identifies the fraudsters and does not give chance in the credit card application. The credit card fraud becomes more prominent because there are large amount of data similar to other data. The fraudsters can easily make a fake account in order to get a credit card application. There may be two ways of data used by the fraudsters one refers to be plausible but identical data of other customer which is effortless to create but more difficult to apply successfully. The other is the real identity theft that is illegal use of innocent people's data [1].

The credit applications are paper-based forms or the online application form with request by the potential customers for credit card, mortgage loans, and personal loans.

In this case of credit card fraud, count of fraudster's increases that are highly experienced, organized and sophisticated [2]. Their visible patterns can be different to each other and constantly change. They are persistent, due to higher financial rewards the risk and effort involved are minimal. Based on the anecdotal observation of experienced credit card application investigation fraudsters can use software automation to manipulate particular values within an application and increase frequency of successful values [3].

The duplicate data of the fraudsters may refer to the applicants with common value. There are two types of duplicate data, one is exact duplicate which have all data similar to other data and another duplicate is the near duplicate (i.e.) approximately similar data with some alteration in the spellings. This system of credit card fraud detection argues that each successful credit application fraud pattern is represented by a sudden and sharp spike in duplicate within a short time [9].

The applicant who are new to the credit card application but have the facility of other credit service such as personal loan, mortgage loan but committed a crime are listed towards the blacklist or the cibil list which consists of the fraudsters data. Hence these kinds of persons are checked for the cibil score that varies 300 to 900 [10] that are generated by the spike detection (SD) algorithm are consider whether to provide further service or not.

This approach of credit card fraud detection uses another database of Whitelist which stores the innocent applicants of credit card. Whitelisting uses real social relationships on a set of attributes [4].

1.1 Main challenges in fraud detection system

Generally fraud is the unauthorized activity taking place in various applications and the process of identifying the unauthorized person is said to be the fraud detection.
The detection system needs "Defence in Depth" with multiple, sequential and independent layers

of defence [5] to cover different types of attacks.

The two main challenges for the Datamining based layer of defence are adaptivity and the use of quality data. These challenges are need to be addressed in order to reduce false positives [3].

Adaptivity

Adaptivity denotes to the morphing of fraud behaviour. In credit application domain, changing legal behaviour is exhibited by communal relationships (such as rising/falling number of siblings) and can be caused by external events. That is the legal behaviour is quiet difficult to differentiate from fraud behaviour.

Quality data

Quality data are highly desirable for Datamining. The quality of data can be increased by removing the errors in data. The detection system has to find the duplicates and ignore the redundant attributes.

2. EXISTING SYSTEM

The main objective of existing research was to achieve resilience by adding two new, real times, data mining-based layers to complement the two existing non-data mining layers discussed in the section. These new layers improved detection of fraudulent applications because the detection system can detect more types of attacks, better account for changing legal behaviour, and remove the redundant attributes.

These new layers were not human resource intensive. They represent patterns in a score where the higher the score for an application, the higher the suspicion of fraud (or anomaly). In this way, only the highest scores require human intervention. These two new layers, communal and spike detection do not use external databases, but only the credit application database per se. And crucially, these two layers are unsupervised algorithms which are not completely dependent on known frauds but use them only for evaluation.

The main contribution of this paper is the demonstration of resilience, with adaptively and quality data in real-time data mining-based detection algorithms. The first new layer was Communal Detection (CD): the whitelist-oriented approach on a fixed set of attributes. To complement and strengthen CD, the second new layer was Spike Detection (SD): the attribute-oriented approach on a variable-size set of attributes.

The second contribution was the significant extension of knowledge in credit application fraud detection because publications in this area were rare. In addition, this research uses the key ideas from other related domains to design the credit application fraud detection algorithms.

Finally, the last contribution was the recommendation of credit application fraud detection as one of the many solutions to identity crime. Being at the first stage of the credit life cycle, credit application fraud detection also prevents some credit transactional fraud.

The communal and spike detection alone can handle the prediction process of fraud. Here there is a drawback of scalability, efficiency of matching, long time constraints, imbalanced classes of data etc [3].

There exist a fusion approach which uses the Dempster – Shafer theory, Bayesian learning and rule based filtering to predict the credit card fraud that results shows good results but it leads to the generation of too many false predictions [6].

3. PROPOSED SYSTEM

The main objective of this project is to detect the credit card fraud detection in the very initial stage of credit card application. This system uses the efficient approach for prediction of general frauds and crime activities. In order to achieve the identification of fraud this paper proposes two layers to complement the existing system are the Communal Detection (CD) and Spike Detection (SD) along **with two static scheduling algorithms: Event-based (EBS) and Run-based (RBS) Scheduling,** then come up with a hybrid method called Fast Response Time Scheduling (FRTS) to dynamically manage the scheduling in order to further reduce the average response time.[8]

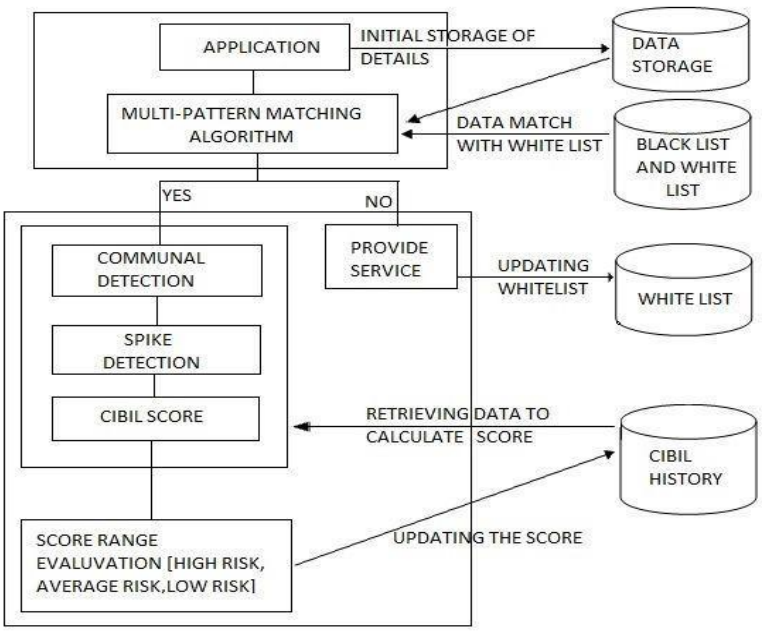

Figure 1

The processes involved in this paper have the contribution of Cibil score that are assigned to the applicants. Higher the score lower the risk in providing the application. Initially the score is assigned to be 900 which may be varied by the transaction and other crime history of customer (Fig.1). The Cibil score is about 300 to 900, lower the score results more risk of providing the credit card [12].

The main contribution of this project is to achieve the challenges of adaptively, and quality data. And the existing drawbacks of effectiveness, scalability, high response time, efficiency, imbalance of data, false predictions etc. Being the first stage of credit life cycle the fraudulent applicant is detected and hence the further transaction crimes will be prevented.

4. METHODOLOGY

This section is divided into three, to systematically explain the CD algorithm, SD algorithm and Multi pattern matching algorithm with clear discussion about their purposes.

4.1 Communal Detection

This section motivates the need for CD and its adaptive approach. Suppose there were two credit card applications that provided the same postal address, home phone number, and date of birth, but one stated the applicant's name to be John Smith, and the other stated the applicant's name to be Joan Smith. These applications could be interpreted in three ways:

1. Either it is a fraudster attempting to obtain multiple credit cards using near duplicated data.

2. Possibly there are twins living in the same house who both are applying for a credit card.

3. Or it can be the same person applying twice, and there is a typographical error of one character in the first name.

With the CD layer, any two similar applications could be easily interpreted as (1) because this paper's detection methods use the similarity of the current application to all prior applications (not just known frauds) as the suspicion score. However, for this particular scenario, CD would also recognize these two applications as either (2) or (3) by lowering the suspicion score due to the higher possibility that they are legitimate.

To account for legal behavior and data errors, CD is the whitelist-oriented approach on a fixed set of attributes. The whitelist, a list of communal and self-relationships between applications, is crucial because it reduces the scores of these legal behaviors and false positives. Communal relationships are near duplicates which reflect the social relationships from tight familial bonds to casual acquaintances: family members, housemates, colleagues, neighbors, or friends [3]. The family member relationship can be further broken down into more detailed relationships such as husbandwife, parent-child, brother-sister, male-female cousin (or both

male, or both female), as well as uncle-niece (or uncle nephew, auntie- niece, auntie-nephew). Self-relationships highlight the same applicant as a result of legitimate behavior (for simplicity, self-relationships are regarded as communal relationships). Broadly speaking, the whitelist is constructed by ranking link-types between applicants by volume. The larger the volume for a link-type, the higher the probability of a communal relationship. On when and how the whitelist is constructed, it is in the CD algorithm [3].

This section explains the need for CD, its approach and working in real time. The CD algorithm is mostly based on the Cibil score on a fixed set of attributes. The CD can be used to compare the current application with the prior one based on the similarity in the attributes. In general, CD is a Whilelist based approach and crucial to reduce the score in terms of communal and self-relationships. The communal relationship reflects the family bonds and legitimate behavior of the same applicant. The Whilelist is constructed by link-types and its volume.

The CD algorithm works in real time by exact or similar matches between categorical data, giving scores. It consists of nine inputs, three outputs and six steps. The overall stream consists of minidiscrete and microdiscrete streams of data from current and previous applicants. The algorithm reconstructs the Whilelist for a period of time and reset the parameter values.

Overview of Communal Detection Algorithm

Inputs
Vi -current application
W -number of Vj (previous application)
Rx, link-type – link-type in current Whilelist
$T_{similarity}$ - String similarity threshold
$T_{attribute}$ - attribute threshold
Ɗ - Duplicate filter
A - exponential smoothing factor
T_{input} - input size threshold
SoA - State of Alert

Outputs
S (Vi) - Cibil score
Same or new parameter value
New Whilelist

CD Algorithm

1. Multi-attribute link- match current application value (Vi) against W number of previous application value (Vj) to determine if a single attribute exceeds $T_{similarity}$ and create multi attribute links, if near duplicates exceeds $T_{attribute}$ or an exact duplicates time difference exceeds Ɒ.
2. Single-link score- calculate the single link score (attribute weight) by matching multi-link attribute with Rx, link-type. It focuses on single link between two applications.
3. Single-link average previous score- calculate average previous score from linked previous application for inclusion into current application score.
4. Multiple-link score- calculate Cibil score based on every link and average previous application score.
5. Parameter value change- determines same or new parameter value through SoA. In first case, when input size is high and output Cibil score is low, the SoA values to low. In second case, the SoA value is high when the condition is opposite to first case.
6. Whilelist change- the link types are sorted in the decreasing order by number of links and higher ranked link type are given low link type weight. The new Whilelist is founded at the end of gx.

4.2 Spike Detection

This method is contrast to the CD method. The need of SD is to improve the resilience and adaptivity. The SD Cibil score is calculated on redundant attributes that are continually filtered. Only selected attributes in the form not-too-parse and not-too-dense attributes are used. SD strengthens CD by providing attributes weight, reflecting the degree of importance of the attribute. This method trades off effectiveness for efficiency to improve computation speed. The SD is attribute-oriented approach on a variable set of attributes and based on blacklist approach.

Overview of Spike Detection Algorithm Design

Inputs
Vi-current application
W-number of Vj (previous application)
t- current step
$T_{similarity}$ - String similarity
threshold Ɵ - time difference filter
α - exponential smoothing factor

Outputs
S (Vi)- Cibil score
wk- attribute weight

SD Algorithm

1. Single-step scaled counts- matches' current value (Vi) against W number of moving window Vj to determine if a single value exceeds $T_{similarity}$ and time difference exceeds Ɵ. The first

case is used for cross matching between current and previous value and time. The second case is non-match as the values are not similar.

2. Single Value spike detection- calculates current values score based on weighted average using α of t.
3. Multiple value score- calculate S(Vi) for every current application score using all values score and attribute weights.
4. SD attributes selection- determines wk for SD at end of gx, highlighting the probe-reduction of selected attributes.
5. CD attribute weights change determines wk for CD at end of gx.

4.3 Multi-Pattern Matching Algorithm

This method consists of two basic method of scheduling approaches, **Event-Based Scheduling (EBS) and Run-Based Scheduling (RBS).**

1) **Event-based Scheduling EBS**(*S, R,O*).

All the runtime patterns test the same event *ei* for possible state transitions before processing the next event *ei*+1. We call this scheduling approach *Event-based Scheduling*. The runtime patterns' processing order for *ei* is *Oi*. For example, given the event stream $S = \{e1, e2\}$, and runtime pattern set $R = \{r1, r2\}$ with the order $O = \{o1, o2\}$ where $o1 = o2 = \{1, 2\}$. If we use the <{runtime pattern}, event> pair sequence to demonstrate the scheduling process, the event processing order should be $< \{r1, r2\}, e1 >, < \{r1, r2\}, e2 >$. In this scheduling approach, each event is tested by multiple runtime patterns in each round. Therefore, we also call it "*n* : 1" method.

2) **Run-based Processing RBS**(*S, R,O, L*).

Each runtime pattern is being processed until it is matched or is rejected, then the next runtime pattern can start to be processed. The events in the buffer may be revisited many times. This approach is called *Run-based Processing*. The order of processing runtime patterns is *O*. To avoid long waiting time for the runtime patterns which do not have matchings, a longest execution time *l j* for *r j* is defined by user or by system. For example, given the event stream $S = \{e1, e2\}$, and runtime patterns $R = \{r1, r2\}$. The order $O = \{1, 2\}$. The upper bound of execution time $L = \{2$ *events*, 2 *events*$\}$, then, the event processing order should be $< r1, e1 >, < r1, e2 >, < r2, e1 >, < r2, e2 >$. In Run-based scheduling, each runtime pattern tests multiple events in each round. Thus, it is also called "1 : *m*" method.

Volume: 3 Issue: 1 26-Jun-2014,ISSN_NO: 2321-3337

Pseudo code of Fast Response Time Scheduling Algorithm

```
1: for each query q in the query set do
2: create a root run and put it into the pending list;
3: end for
4: while ! the end of event stream do
5: check if the input event buffer has been updated with new events since last check;
6: if the input event buffer has been updated then
7: sort the pending list by the priority of runtime pattern in descending order;
8: merge the pending list into the running list with order preservation;
9: end if
10: if if the running list is not empty then
11: select the first K runs in the running list;
12: for each run r in the K runs do
13: execute r until it reaches pending or stopped state;
```

14: put the generated child runs into the children list;
15: **if** *r* stops **then**
16: destroy *r*;
17: **else**
18: move *r* from the running list to the pending list;
19: **end if**
20: **end for**
21: sort the children list by the priority of runtime pattern in descending order;
22: merge the children list into the running list with order preservation;
23: **end if**
24: **end while**

5. CONCLUSION

The main target that focused in this project is to safeguard the credit application in the initial stage of credit life cycle. The implementation of Multi pattern matching algorithm in order to compare the attributes makes the identification process reliable with less time complexity. Here the two main challenges of adaptivity and quality of data have been achieved with balanced data load. This project has been proposed with the efficiency in scalability by updating the evaluation of data.

REFERENCES

[1]. R.Bolton and D.Hand,"Unsupervised Profiling Methods for Fraud Detection," Statistical Science, vol. 17, no. 3, pp. 235-255, 2001.

[2]. G.Gordon, D.Rebovich, K.Gordon, "Identity Fraud Trends and Patterns: Building a Data-Based Foundation for Proactive Enforcement," Center for Identity Management and Information Protection, Utica College, 2007.

[3]. Clifton Phua, Kate Smith-Miles, Vincent Cheng-Sion Lee and Ross Gayler, "Resilient Identity Crime Detection," IEEE Transactions On Knowledge And Data Engineering, Vol. 24, No. 3, March 2012.

[4]. J. Neville, O. Simsek, D. Jensen, J. Komoroske, K. Palmer, and H.Goldberg, "Using Relational Knowledge Discovery to Prevent Securities Fraud," Proc. 11th ACM SIGKDD Int'l Conf. Knowledge Discovery in Data Mining (KDD '05), 2005, doi: 10.1145/1081870.1081922.

[5]. B. Schneier,, Schneier on Security. Wiley, 2008.

[6]. Suvasini Panigrahi, Amlan Kundu, "Credit card fraud detection: A fusion approach using

Dempster–Shafer theory and Bayesian learning" Int'l Journal on Information fusion, Elsevier vol 10, issue 4, 2009.

[7]. A. Bifet and R. Kirkby Massive Online Analysis, Technical Manual, Univ. of Waikato, 2009.

[8]. Ying Yan, Jin Zhang, Ming-Chien Shan, "Scheduling for Fast Response Multi-pattern Matching over Streaming Events," ICDE Conference, 978-1-4244-5446-4/10 , 2010 IEEE.

[9]. T.Oscherwtiz, "Sythetic Identity Fraud: Unseen Identity Challenge," Bank Security News, vol.3, p.7, 2005.

[10]. R.Caruana and A.Niculescu-Mizil, "Data Mining in Metric Space: An Empirical Analysis of Supervised Learning Performance Criteria," Proc. 10th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD '04), 2004, doi: 10.1145/1014052.1014063.

[11]. E.Ramkumar and P. Kavitha, "Online Credit Card Application and Identity Crime Detection," Int'l Journal of Engg. Research and Technology (IJERT) vol. 2 Issue 2, Feb-2013.

[12]. Hideaki Tanaka, Shigeyuki Tsukao, Daiki Yamashita, TakahideNiimura, Ryuichi Yokoyama, "Multiple Criteria Assessment of Substation Conditions by Pair-Wise Comparison of Analytic Hierarchy Process," IEEE Transactions On Power Delivery, Vol. 25, No. 4, October 2010.

[13]. IDAnalytics, "ID Score-Risk: Gain Greater Visibility into Individual Identity Risk," Unpublished, 2008.

[14]. B. Head, "Biometrics Gets in the Picture," Information Age, pp. 10-11, Aug.-Sept. 2006.

[15]. J. Jonas, "Non-Obvious Relationship Awareness (NORA)," Proc. Identity Mashup, 2006.

[16]. B. Schneier, Beyond Fear: Thinking Sensibly about Security in an Uncertain World. Copernicus, 2003.

[17]. W. Wong, "Data Mining for Early Disease Outbreak Detection," PhD thesis, Carnegie Mellon Univ., 2004.

[18]. S. Romanosky, R. Sharp, and A. Acquisti, "Data Breaches and Identity Theft: When Is Mandatory Disclosure Optimal?," Proc. Ninth Workshop Economics of Information Security (WEIS), 2010.